Linear Attention, Graph and Compact Interhand Reconstruction

Anonymous CVPR submission

Paper ID 33

Abstract

In the metaverse, hand reconstruction is becoming more and more compelling. With the development of virtual reality and mixed reality applications, the demand for reconstruction and interaction of human hands with efficient transformer has greatly increased. However, current research still cannot guarantee high efficiency, so there is still a lot of room for improvement. In this paper, we propose a lightweight hand reconstruction method (LAGCHand) based on linear and compact modules to reconstruct interactive hands from a single RGB image. In order to reduce computing consumption and improve real-time performance, we put forward three modules. The first is Linear Attention Encoder module to extract hand features from coarse to fine, the second is Compact Single Hand heatmap Linear module to get global features and left and right hand features, the third is Linear Graph module, Two-handed attention-enhancing features were obtained using linear GCN and linear attention mechanisms. Our model has good performance on the InterHand2.6M benchmark, with MPJPE 21.89 and Flops is reduced to 0.81G, which greatly improves the computing efficiency.

1. Introduction

3D hand reconstruction and hand pose estimation play an important role in human-computer interaction and human behavior modeling. With the introduction of InterHand2.6M [18], a more efficient solution has been proposed.By building different network architectures to recreate the interacting hands, For example, the 2.5D heat map [6, 11, 18] is used to estimate the position of hand joint, and the dense image feature map [14, 23] is used to obtain hand features for reconstruction and the image convolutional neural network is used Method of collation [15]. Although these methods have gradually improved the reconstruction effect and achieved good results, they are still difficult to be applied to the mobile terminal and mixed reality due to their large consumption of computing and high demand for hardware.

In this paper, Linear Attention, Graph and Compact Interhand Reconstruction (LAGCHand) is proposed. Firstly, Linear Attention Enocder was used to extract coarse to fine features from input monocular RGB two-handed images. In order to reduce computing consumption, we use a new lightweight visual transformer to extract features from interactive hand information. The lightweight transformer combines the convolutional neural network (CNNs) and Vision Transformer (ViTs) to fuse the features of the local representation block and the global representation block together. In the end, the input features are fused to make full use of the following features of the interactive hand. It can still perform well while keeping the computation low. Next, we proposed the Compact Single Hms Linear module to process the features extracted in the previous step, so as to obtain the two-hand separate features and global features for subsequent use. Finally, we put forward Linear Graph module. Through linear GCN and linear attention mechanism, it not only reduces computing consumption but also obtains hand features with enhanced attention. In summary, our contributions are summarized as follows:

1. We propose a Linear Attention, Graph and Compact Interhand Reconstruction method–LAGCHand(as shown in Figure 2), and demonstrate the effectiveness of Linear Attention for the task of hand reconstruction.
2. We propose Compact Single Hms Linear module to process the features and get the two-hand separate features and global features.
3. We propose a Linear GCN module to make full use of global and local information through linear graph layer and efficient feature fusion mechanism..

2. Related works

Hand pose estimation. Hand pose estimation can be divided into two ways based on depth map and RGB map [5]. RGB-based hand pose estimation [17] can be divided into two paradigms: model-based methods [2, 3, 13, 17, 24] and model-free methods. The model-based approaches collect

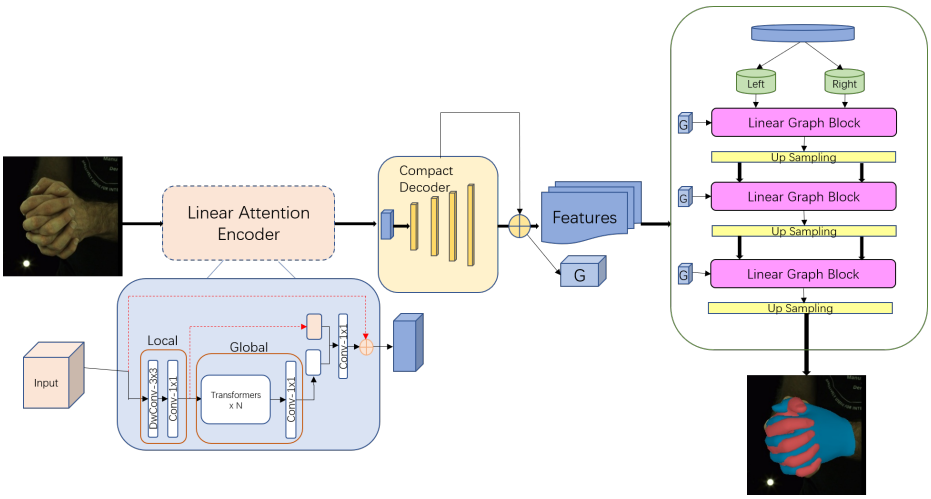

Figure 1. Overview our network architecture.Given an RGB image as input

synthetic data for pre-training and then use MANO [21] to capture the hand shape to generate hand pose. For example, Kulon et al. [13] used an image encoder and a mesh convolution decoder trained by direct 3D hand mesh reconstruction loss to process the data. Model-free methods, on the other hand, estimate the pose by generating hand joint information through neural networks learning hand joint coordinates [8, 16, 22, 25] or hand joint heat maps [1, 4, 10, 19].For example,Aboukhadra et al. [1] used a keypoint RCNN to extract 2D pose, feature maps, heat maps, and boxes in bounding monocular RGB images. This 2D information is then modeled as two plots and passed to two branches of the reconstruction stage.Instead, we use a model-based approach to help generate hand postures using the MANO model.

Transformers in Hand Reconstruction. As for hand pose and mesh, Huang et al. [9] proposed a transformer-based network which estimates 3D hand pose from 3D hand point cloud. Hampali, et al. [7] proposed HandsFormer, which combines the recognition capability of transformers with the accuracy of heat map-based methods, using attention mechanisms to sort out the correct configuration of joints and output the 3D pose of both hands. And it can be extended to estimate the 3D pose of objects operated by one or both hands.liu et al. [17] perform explicit contextual inference between hand and object representations beyond the limited 3D annotations in a single image, using spatio-temporal consistency in large-scale hand-object videos as a constraint for generating pseudolabels in semi-supervised learning. And the most relevant work to us is the HandOccNet proposed by Park et al. [20] which makes full use of information from the occluded region as an aid to enhance image features proposing feature injection and self-enhancement modules to inject hand information into the occluded region. Whereas they only considered hand information injection, our dual-fit module achieves better performance by utilizing both information from the occluded region and hand information.

3. Method

All network modules are shown in Figure 1. Our system mainly includes three parts: Linear Attention Encoder, Compact Single Hms Linear module and Linear Graph module. For a given Single RGB graph, it is first sent to Linear Attention Encoder for processing encoding and feature extraction, and then encoded features are sent to Compact Single Hms Linear module for feature processing and decoding. This structure produces an intermediate global eigenvector F and several bound eigenmaps $\left\{G_t \in \mathbb{R}^{C_t \times H_t \times W_t}, t = 0, 1, 2\right\}$, Where t represents the t th feature level corresponding to the t th Linear Graph Block. $H_t \times W_t$ is the resolution of feature mapping, and C_t is the feature channel. Finally, the global feature vector F was divided into the vertices of the left hand and the right hand, which were input into the Linear Graph module together with the feature mapping of the binding of each layer, and the hand reconstruction information was finally obtained. Three of these modules are discussed in sections 3.1, 3.2, and 3.3.

3.1. Linear Attention Encoder

Linear Attention Encoder module takes an image as the first input $\left\{X \in \mathbb{R}^{H_0 \times W_0 \times 3}\right\}$ enters feature fusion module after feature processing through local feature extraction module and global feature extraction module and finally outputs $\left\{Out \in \mathbb{R}^{H_1 \times W_1 \times 3}\right\}$. The specific feature processing in the module is calculated in the following ways:

$$Out = M(X) = Concat\left[T(x), X\right], \qquad (1)$$

$$T(x) = Concat\left(TC(x), CNN_{Depth-wise}(X)\right), \quad (2)$$

$$TC(x) = Trans\left(CNN_{Depth-wise}(X)\right), \quad (3)$$

Here, $M(X)$ represents the complete Linear Attention Encoder function, $T(x)$ represents the output of local and global features after feature fusion, $CNN_{Depth-wise}$ represents the depth wise convolution, $TC(x)$ represents the depth wise convolution, The $Trans$ function represents the characteristics of the transformer operation. We use convolution of 1×1 in local feature module $CNN_{Depth-wise}$, global feature module and finally fusion module.

3.2. Compact Single Hand Heatmap Linear

The feature $\left\{Out \in \mathbb{R}^{H_1 \times W_1 \times 3}\right\}$output in Linear Attention Encoder module is input into Single Hand Heatmap Linear module. Final output processed feature F_{maps} and global feature G for subsequent modules $\left\{G \in \mathbb{R}^{H_2 \times W_2 \times C_2}\right\}$:

$$F_{maps}, G = Hms_{single}\left[Out\right] \quad (4)$$

3×3 convolutional neural network is used for feature processing in Hms_{single} network.

3.3. Linear Graph

L_f, R_f respectively represent the left and right hand to be processed features, which are obtained by processing features in Single Hand Heatmap Linear Module. Then, they are sent to Linear Graph Module for processing and finally the output hand vertex features HF_L, HF_R are obtained. The specific process is as follows:

$$L_f^{'}, R_f^{'} = Di - Concat\left(Linear_{GCN}\left[L_f R_f\right]\right) \quad (5)$$

$$Linear_{Graph}\left[L_f R_f\right] = Linear\left[Concat\left(L_f, R_f\right)\right] \quad (6)$$

$Linear_{Graph}$ represents the Linear graph module, and the $Linear$ function represents the linear feature extraction operation. The $L_f^{'}, R_f^{'}$ processed through the linear graph module is sent to the subsequent attention module.

$$H_L = Attention\left[Concat\left(L_f^{'}, F_{maps}\right)\right] \quad (7)$$

$$H_R = Attention\left[Concat\left(R_f^{'}, F_{maps}\right)\right] \quad (8)$$

Here, $Attention$represents the vertex feature H_L, H_R of attention enhancement by stitching the left and right hand features processed by linear convolution in the last step with the image feature F_{maps} through multi-head self-attention mechanism.

$$HF_L = HF^{R \longrightarrow L} = \beta H_L + (1 - \beta) H_R \quad (9)$$

$$HF_R = HF^{L \longrightarrow R} = \alpha H_R + (1 - \alpha) H_L \quad (10)$$

Finally, by injecting left hand features into right hand and right hand features into left hand through linear attention module, the final enhanced left and right hand feature vertex features HF_L, HF_R, where β and α are self-learning parameters are obtained.

3.4. Loss Function

In order to compare the proposed network with the most advanced model, we use the same loss vertex loss, regression joint loss and network smoothing loss as IntagHand [15].

Vertex Loss. Use the three-dimensional coordinates of L1 loss monitor hand point and the two-dimensional projection of the vertex with MSE loss monitor:

$$L_v = \sum_{i=1}^{N} \left|V_{h,i} - V_{h,i}^{GT}\right| 1 + \left|\prod\left(Vh, i\right) - \prod\left(V_{h,i}^{GT}\right)\right|_2^2, \quad (11)$$

Where $V_{h,i}$ is the i-th vertex, $h = L, R$ represents left or right hand, and $\prod$ is a two-dimensional projection operation. Vertex losses are applied to each submesh.

Regressed Joint Loss.Regression of hand joints from predicted hand vertices by multiplying by the predefined joint regression matrix $\jmath$.

$$\begin{aligned} L_J = & \sum_{i=1}^{V} \left\|\jmath V_{h,i} - \jmath V_{h,i}^{GT}\right\|_1 \\ & + \sum_{i=1}^{V} \left\|\prod\left(\jmath V_{h,i}\right) - \prod\left(\jmath V_{h,i}^{GT}\right)\right\|_2^2, \end{aligned} \quad (12)$$

Mesh Smooth Loss. The smooth losses term is used to monitor the geometric smoothness. By using two different smooth losses, the normal consistency between the predicted value and the ground truth grid is regularized at first:

$$L_n = \sum_{f=1}^{F} \sum_{e=1}^{3} \left|e_{f,i,h} \cdot n_{f,h}^{GT}\right|_1, \quad (13)$$

Where $f$is the face exponent of the hand grid, $e_{f,i}\,(i = 1, 2, 3)$ is the three sides of the face f, and n_f^{GT} is the normal vector of this face calculated from the ground real grid. Secondly, minimize the $L1$ distance of each edge length between the predicted mesh and the ground truth mesh:

$$L_e = \sum_{e=1}^{E} \left|e_{i,h} \cdot e_{i,h}^{GT}\right|_1. \quad (14)$$

4. Experiment

4.1. Experimental Settings

Implementation Details. Our network is implemented using PyTorch. We use the MobileVitV3-XXS as backbone

to encode the image feature.We used the Adam optimizer [12] to train our model on four NVIDIA RTX 3090 GPUs, each with a small batch size set to 64. The entire training takes 100 epochs and lasts 1 days. The learning rate of the 50-th epoch decays from the initial 1×10^{-4} to 1×10^{-5}.

Evaluation Metrics. To evaluate the pose and shape of the reconstructed hand, we compared the mean position error per joint (MPJPE) with the mean position error per vertex (MPVPE) in millimeters. In a fair comparison with other state-of-the-art models, we scaled the metacarpal mid-length of each hand to 9.5cm for training and re-scaled it to ground truth bone length for evaluation based on IntagHand [15]. This is done after the root joints of each hand are aligned.

4.2. Datasets

InterHand2.6M Dataset [18]. All the networks in this paper are trained on the InterHand2.6M [18] dataset. This dataset provides two-handed interaction data and human and machine annotations. In the experiment, we specifically used 366K training samples and 261K test samples in InterHand2.6M. During pre-processing, we clipped the hand part area according to the 2D projection of the hand vertex, and adjusted its size to 256 × 256 resolution [15].

4.3. Quantitative and Qualitative Results

We first compare the LCHand network with the State-Echo-Art hand reconstruction method and more recent hand reconstruction methods as shown in Table 1. The Inter-Hand [18] model can directly return the three-dimensional model of two hands through the regression network. The Interacting-two-hand [23] model reconstructed 3D hands by predicting the position and shape parameters of two MANO models. Finally, the IntagHand [15] model uses the improved GCN network to predict the position and pose of the interactive hand for regression reconstruction. For a fair comparison, we also run their published source code on the same subset of Inter-Hand2.6M [18]. The comparison results are shown in Table 1. Based on the results in Table 1, we not only show MPJPE and MPVPE, but also show the effect of our model on the reduction of computational delay FLOPS. It can be seen that our method greatly reduces FLOPS while maintaining a certain MPJPE and MPVPE reconstruction effect. While the most advanced models have 10G or higher FLOPS.

4.4. Ablation study

We gradually increase the three modules designed by us through the ablation experiment, and the specific experimental effects can be intuitively perceived from Table 2. By increasing modules one by one, we can see that the Flops consumed in extracting image features under the condition of baseline using MobileVitv3-xxs is about 2G. However,

Method	MPJPE	MPVPE	FLOPS
Inter-Hand	16.0	-	19.49
Interacting-two-hand	13.48	13.95	28.98
IntagHand	8.79	9.03	8.42
Ours	21.89	22.26	0.81

Table 1. Comparison with the state-of-the-art models on performance and flops. The MPJPE and MPVPE are calculated in millimeters. The flops are calculated in GFlops.

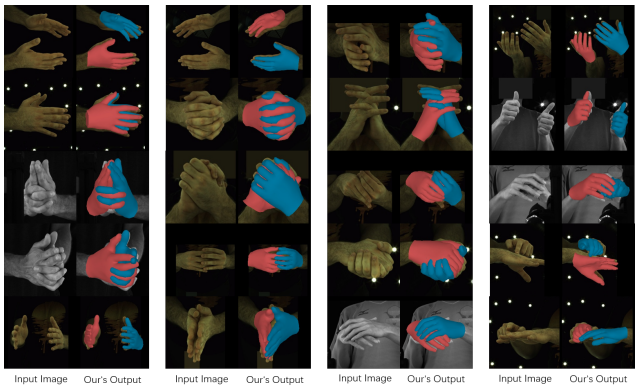

Figure 2. Qualitative results of our method on InterHand2.6M.

the module we designed increased constantly, and finally Flops dropped to about 0.8G, and MPJPE and MPVPEBI remained almost the same, which can be seen that our module design was successful.

Method	MPJPE	MPVPE	FLOPS
LAencoder	20.88	20.59	2.091
LAencoder+ Compact	21.31	21.77	1.023
LAencoder+ Compact+ LGraph (ours)	21.89	22.26	0.81

Table 2. Ablation study of module choice on InterHand2.6M.

5. Conclusion

We propose LAGCHand, a lightweight linear compact method for reconstructing interactive hands from monocular RGB images. In this paper, we first introduce the feature extraction module of lightweight transformer based on linear Attention, which is used to extract hand features in pictures.Next, we propose Compact Single Hand Heatmap Linear module, which uses a compact convolutional network to obtain two-handed features and global features. Finally, we propose a linear GCN module to make full use of both left and right hand global and local information to reconstruct the hand module. This is our initial work, and we are going to reduce the FLOPs to 0.4G and improve the accuracy of hand reconstruction.

References

[1] Ahmed Aboukhadra, Jameel Malik, Ahmed Elhayek, Nadia Robertini, and Didier Stricker. Thor-net: End-to-end graformer-based realistic two hands and object reconstruction with self-supervision, 10 2022. 2

[2] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1067–1076, 2019. 1

[3] Adnane Boukhayma, Rodrigo de Bem, and Philip H.S. Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[4] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 678–694, Cham, 2018. Springer International Publishing. 2

[5] Bardia Doosti. Hand pose estimation: A survey, 03 2019. 1

[6] Zicong Fan, Adrian Spurr, Muhammed Kocabas, Siyu Tang, Michael J. Black, and Otmar Hilliges. Learning to disambiguate strongly interacting hands via probabilistic per-pixel part segmentation, 2021. 1

[7] S. Hampali, S. D. Sarkar, M. Rad, and V. Lepetit. Handsformer: Keypoint transformer for monocular 3d pose estimation ofhands and object in interaction, 2021. 2

[8] Yana Hasson, Gül Varol, Cordelia Schmid, and Ivan Laptev. Towards unconstrained joint hand-object reconstruction from rgb videos. In *2021 International Conference on 3D Vision (3DV)*, pages 659–668, 2021. 2

[9] Lin Huang, Jianchao Tan, Ji Liu, and Junsong Yuan. Hand-transformer: Non-autoregressive structured modeling for 3d hand pose estimation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 17–33, Cham, 2020. Springer International Publishing. 2

[10] Umar Iqbal, Pavlo Molchanov, Thomas Breuel, Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5d heatmap regression. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 125–143, Cham, 2018. Springer International Publishing. 2

[11] Dong Uk Kim, Kwang In Kim, and Seungryul Baek. End-to-end detection and pose estimation of two interacting hands. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11169–11178, 2021. 1

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 4

[13] Dominik Kulon, Riza Alp Güler, Iasonas Kokkinos, Michael M. Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4989–4999, 2020. 1, 2

[14] Mengcheng Li, Liang An, Hongwen Zhang, Lianpeng Wu, Feng Chen, Tao Yu, and Yebin Liu. Interacting attention graph for single image two-hand reconstruction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2751–2760, 2022. 1

[15] Mengcheng Li, Liang An, Hongwen Zhang, Lianpeng Wu, Feng Chen, Tao Yu, and Yebin Liu. Interacting attention graph for single image two-hand reconstruction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2751–2760, 2022. 1, 3, 4

[16] Moran Li, Yuan Gao, and Nong Sang. Exploiting learnable joint groups for hand pose estimation, 12 2020. 2

[17] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14682–14692, 2021. 1, 2

[18] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX*, page 548–564, Berlin, Heidelberg, 2020. Springer-Verlag. 1, 4

[19] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 49–59, 2018. 2

[20] JoonKyu Park, Yeonguk Oh, Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Handoccnet: Occlusion-robust 3d hand mesh estimation network. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1486–1495, 2022. 2

[21] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Trans. Graph.*, 36(6), nov 2017. 2

[22] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 89–98, 2018. 2

[23] Baowen Zhang, Yangang Wang, Xiaoming Deng, Yinda Zhang, Ping Tan, Cuixia Ma, and Hongan Wang. Interacting two-hand 3d pose and shape reconstruction from single color image. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11334–11343, 2021. 1, 4

[24] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2354–2364, 2019. 1

[25] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4913–4921, 2017. 2