Stable Hand Pose Estimation under Tremor via Graph Neural Network

Zhiying Leng[1*] Jiaying Chen[1†] Hubert P. H. Shum[2‡] Frederick W. B. Li[2§] Xiaohui Liang[1¶]

1 State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, China
2 Department of Computer Science, Durham University, U.K.

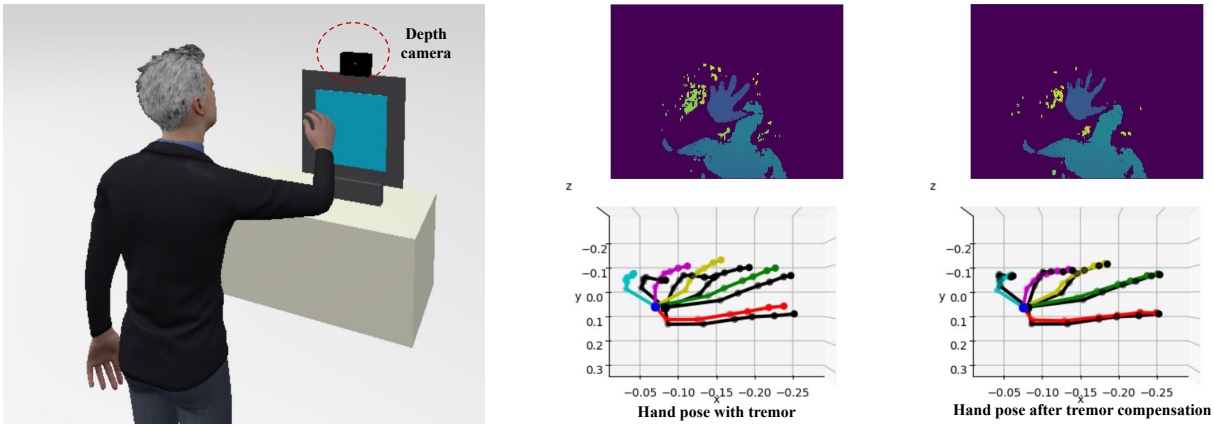

Figure 1: The aim of our method is to estimate the stable 3D hand pose under tremor. Left: the environment setting for collecting test data. Right: input consecutive frames and hand poses of neighbor frames before and after tremor compensation.

Abstract

Hand pose estimation, which predicts the spatial location of hand joints, is a fundamental task in VR/AR applications. Although existing methods can recover hand pose competently, the tremor issue occurring in hand motion has not been completely solved. Tremor is an involuntary motion accompanied by a desired gesture or hand motion, leading to hand pose that deviates from user's intentions. Considering the characteristic of tremor motion, we present a novel Graph Neural Network for stable 3D hand pose estimation. The input is depth images. The constraint adjacency matrix is devised in Graph Neural Network for dynamically adjusting the topology of a hand graph during message passing and aggregation. Firstly, since there are rich potential constraints among hand joints, we utilize the constraint adjacency matrix to mine the suitable topology, modeling spatial-temporal constraints of joints and outputting the precise tremor hand pose as the pre-estimation result. Then, for obtaining a stable hand pose, we provide a tremor compensation module based on the constraint adjacency matrix, which exploits the constraint between control points and tremor hand pose. Concretely, the control points represented the voluntary motion are employed as constraints to edit the tremor hand pose. Our extensive quantitative and qualitative experiments show that the proposed method has achieved decent performance for 3D tremor hand pose estimation.

Index Terms: Computing methodologies—Artificial intelligence—Computer vision—Computer vision problems; Human-centered computing—Human computer interaction—Interaction techniques—Gestural input

*joint first author, e-mail: zhiyingleng@buaa.edu.cn
†joint first author, e-mail: chenjiaying@buaa.edu.cn
‡e-mail: hubert.shum@durham.ac.uk
§e-mail: frederick.li@durham.ac.uk
¶corresponding author, e-mail: liang_xiaohui@buaa.edu.cn

1 Introduction

Human-computer interaction (HCI) plays a key role in VR/AR applications. Gesture interaction is the most intuitive and natural way in HCI. During the interaction, users may feel fatigued or nervous, especially after a long duration. The user's hand may involuntarily shake in this case. This is called physiological tremor. In addition, hand shaking may also be caused by some diseases [29], such as Parkinson's. This is called pathological tremor. These tremor motions are relatively slow and local, shaking around a desired pose. The tremor motions are different from the fast hand motion [26], which typically distorts the depth image captured the hand pose due to motion blur. However, the tremor motions affect the robustness of human-computer interaction. Taking the selection gesture as an example, tremor motions make the user's hand deviate from the target object and shake around the target object. In this work, we study the task of estimating a stable hand pose to cope with the tremor issue in VR/AR applications.

Existing state-of-the-art methods have achieved considerable performance in hand pose estimation. Because a hand is articulated by bones, many researchers have attempted to apply hand skeleton constraints to improve the accuracy. These methods can be divided into two categories. The first one is called the structured-based method [17, 21–24, 31, 43]. These methods implicitly embed physical constraints into Convolutional Neural Network (CNN) or loss function. The second one is called the multi-branch based method [4, 7, 12]. These methods utilize the hand anatomy knowledge to explicitly split hand pose estimation into multiple sub-tasks. The latter improves the discriminative ability of networks by the pre-defined constraint among joints. However, these methods cannot fully utilize potential constraints among joints, the performance of which is limited by hand-crafted constraints. For example, some researchers only considered the constraint relationship between the palm and fingers, without defining the constraint relationship among other fingers [7].

Another difficulty is that existing methods cannot handle the

tremor problem. A tremor hand pose consists of both involuntary tremor motion and voluntary motion. The tremor component can be eliminated by tremor compensation methods based on Fourier Linear Combiner (FLC) and its variants [1, 10, 25, 28, 32, 36]. Nevertheless, these filters are sensitive to parameter adjustments, making them difficult to be generalized to tremor data. Moreover, the tremor data of these methods is collected with an acceleration sensor [27]. In some VR/AR applications, there is no acceleration sensor, tremor data can only be captured through camera as images. These methods are inapplicable in these cases, since the frequencies of image data and sensor data are inconsistent. Furthermore, existing tremor compensation algorithms only focus on the tremor of a hand joint. A direct deployment of these algorithms for compensating the tremor of all hand joints will be computationally experience and will lead to the loss of physiological consistency among the joints.

Specifically, we observe that there are rich structural relationships among hand joints. The Graph Neural Network (GNN) [42] studied by many researchers recently can fully model the structural constraint, through defining and updating a hand graph. This method captures more structure information than CNN-based methods. Cai et al. [2] proposed a GNN-based method to perform 3D pose estimation. The local-to-global network based on a fixed graph structure is designed to learn multi-scale features. However, the fixed graph structure potentially limits the ability of GNN to exploit the relationship among joints.

In this paper, we propose a novel method based on Graph Neural Network for stable hand pose estimation under tremor. The novel Graph Neural Network is named as CAM-GNN. We define a learnable constraint adjacency matrix called CAM, which is a weighted adjacency matrix representing the graph topology to characterize the dynamic graph structure. The input is depth images. Firstly, we design a pre-estimation module to produce the accurate hand pose under tremor. CAM dynamically adjusts the topology of a hand graph by the joint-by-joint message passing mechanism, establishing the spatial-temporal constraint. The constraint is more rich than any manual and fixed design. Secondly, we devise a tremor compensation method that utilizes control points to eliminate the tremor component of all joints. The tremor compensation method is inspired by the characteristic that the tremor is often accompanied by a target object. In this module, we first adopt the WaveNet [35] to extract control points, which represent non-tremor fingertips. Then, we introduce a motion editing method based on control points, which is an extension of CAM-GNN to stabilize the tremor hand pose towards control points..

To verify the performance of our method, we employ TIM-Tremor [27] and NYU [34] datasets to form suitable datasets for tremor hand pose estimation. A set of experiments have been conducted through the datasets to demonstrate that our proposed method effectively eliminates the tremor component. We also conducted a significant number of experiments on the NYU and MSRA15 [32] datasets, demonstrating that the proposed CAM-GNN module can be used as a transplantable module to boost existing methods in enhancing the accuracy of hand pose estimation. The main contributions of this work are summarized as follows:

- We propose a novel GNN-based method to estimate the stable hand pose under tremor in VR/AR applications.
- By introducing a constraint adjacency matrix in GNN module, the spatial-temporal constraint is dynamically learned and the error estimation of existing methods is corrected.
- According to the characteristic of tremor motion, we design a novel tremor compensation method to eliminate the tremor component, thereby extending the hand pose estimation algorithm to tackle the tremor problem in VR/AR applications.

2 Related Work

This work is closely related to following topics: 3D hand pose estimation, tremor compensation and Graph Neural Network. In this section, we review related works on these topics.

2.1 3D Hand Pose Estimation

As the performance of computing equipment and Convolutional Neural Network has been improved, there are many CNN-based methods developed for 3D hand pose estimation. These methodologies can be categorized into detection-based and regression-based methods [6]. The former generates a heat map for each joint [8, 37]. The location of each joint is obtained by an argmax operation. In contrast, the latter directly predict the location of each joint [5, 9, 22]. Besides, Spurr et al. [30] applied the VAE framework to learn a cross-modal latent space, estimating 3D hand poses from RGB or depth images. However, these methods just learn the mapping between images and 3D pose space, without modeling and utilizing the hand structure. Hence, some researchers introduce the structural constraint into CNN. These methods can be divided into two categories, structured-based methods and multi-branch based methods.

Structure-based Methods. These methods embed physical structural constraints into the CNN model [21, 22] or loss function [17]. Zhou et al. [43] proposed a model-based approach that adopted a forward kinematics based layer to ensure the geometric validity of estimations. Madadi et al. [17] included the structural constraint in the loss function. Sun et al. [31] used phalanges instead of joints for representing pose and defined a loss function that encoded long-range interaction between phalanges. For fitting a 3D pose, Oberweger et al. [22, 24] trained a feedback loop to correct mistakes. Oberweger et al. [21, 22] learned a prior model and integrated it into the network by introducing a bottleneck in the last convolution layer. These methods establish limited structural constraints based on the kinematic skeleton.

Multi-branch Based Methods. These methods mine discriminative cues with multi-output branches. These methods fuse features of different joints according to a tree-like structure of hand, forming the constraint among joints. Guo et al. [12] employed multi-output branches to predict coordinates of each finger. On the basis of Guo's work [12], Chen et al. [4] fused features of different joints according to the hand topology. Concretely, joint features belonging to the same finger were integrated into the first layer. Features of all fingers were fused in following layers to predict the accurate hand pose. Other methods divided hand joints by joint types [11] instead of fingers. Du et al. [7] decomposed the hand pose estimation task into the palm pose estimation and the finger pose estimation. They adopted a two-branch cross-connection structure to share the beneficial complementary information between sub-tasks. However, the performance of these methods is limited, because the structural relationship is hand-crafted.

2.2 Tremor Compensation

The purpose of the tremor compensation is to eliminate the tremor component and to retain the voluntary motion. This technology plays an important role in the fields of medical diagnosis and surgical robot control. The tremor compensation eliminates the tremor component by estimating tremor motion. Some researchers used the Fourier Linear Combiner (FLC) to model the tremor signal. Such a method requires model parameters to be adjusted in order to deal with the variations of tremor frequency and magnitude [1, 10, 25, 28, 32, 36]. Veluvolu [36] used the band limited multiple Fourier Linear Combinator to track the frequency and magnitude. Sun [32] proposed a correction method based on the enhanced band-limited multiple Fourier Linear Combiner. These methods based on FLC have poor generalization, because of complex parameter settings.

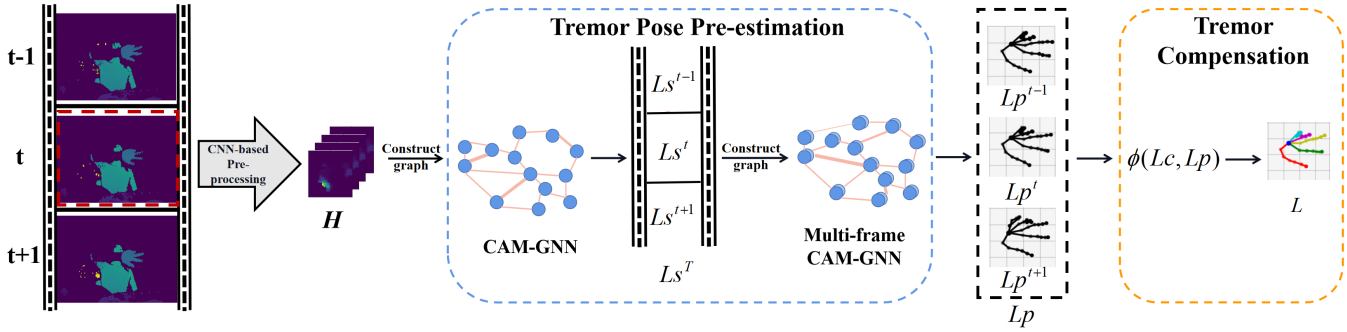

Figure 2: The overview of our proposed method for hand pose estimation under tremor. For the t-th frame of consecutive frames, the CNN-based processing module estimates the rough result H. Then, the tremor pose pre-estimation module outputs the more accurate tremor hand pose Lp. Finally, the tremor compensation module eliminates the tremor component to obtain the stabilized hand pose L. Lc represents control points.

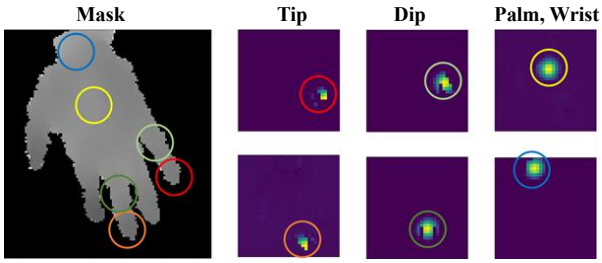

Figure 3: Shape-aware heat maps for several joints. Tip: the fingertip, Dip: the digital pulp, Palm Wrist: the joints of palm and wrist. Given a hand mask, shape-aware heat maps of fingertips are related to the hand shape. In contrast, the heat map of palm is still a Gaussian heat map, since the palm joint is in the hand region.

2.3 Graph Neural Network

Graph Neural Network is a connectionist model that captures the dependence of graph via message passing and aggregation from its neighbors with arbitrary depth [41, 42]. Due to the powerful ability to learn features of structured data, GNN has been widely used in recognition [13], classification [14], Natural Language Processing [19] and other fields. For example in object detection, Liu et al. [16] presented the Structure Inference Network to exploit the scene information and the relationship among objects.

Unlike Convolutional Neural Network, the data processed by GNN is a graph. According to the way of processing graph, GNN can be divided into two categories, spatial-based and spectral-based methods. The latter transforms the graph from the spatial domain to the frequency domain through Fourier transformation. Cai et al. [2] adopted the spectral-based method to learn the mapping of 2D hand pose to 3D hand pose. Simultaneously, they proposed a temporal graph representation for hand to exploit the temporal relationship among frames. A local-to-global network architecture is designed to capture multi-scale features based on the fixed graph structure. These methods for hand pose estimation exploit limited structure information based on the fixed graph structure. Besides, Wang et al. [39] devised a Dynamic Graph CNN (DGCNN) for point cloud learning, by dynamically selecting K-nearest points as graph nodes. DGCNN changes graph nodes, not the graph topology.

3 Methodology

The aim of this work is to estimate stable 3D hand pose under tremor, which is represented by a vector of the size of $3 \times N \times F$, where 3 is the coordinate dimension of joints, N is the number of joints and F is the number of frames, given consecutive depth images I as the input. Our method includes three modules: 1) preprocessing, 2) tremor hand pose pre-estimation and 3) tremor compensation. The framework of our method is shown in Figure 2. The preprocessing step roughly estimates the hand pose with tremor. The pre-estimation module computes a more accurate tremor hand pose. Finally, the tremor compensation module is employed to obtain the stable 3D hand pose L. In this section, we describe our method based on the processing of an input depth image I^t captured at frame t.

3.1 Preprocessing Module

The preprocessing module, as the backbone network, estimates a rough hand pose from a depth image. We apply an existing CNN-based method as the backbone network, DenseReg [37]. For an input depth image I^t, the preprocessing module outputs a heat map H for joints, $H = \{h_i | h_i \in R^{H \times W \times C}, i = 1 : N\}$, where C is the channel number of heat maps. The value of C is 9 referred to DenseReg. The peak of heat maps represents the location of joints.

In this work, we propose a shape-aware heat map to represent the ground-truth position of each joint, which is generally represented as a Gaussian heat map. The strategy is inspired by Li's work [15] that gradually reducing the kernel size of Gaussian can refine the localization accuracy. In other words, the Gaussian distribution outside the object region is invalid. Our shape-aware map is obtained by the intersection between a Gaussian heat map and a hand mask, as shown in Figure 3. For hand pose estimation, the distribution in the hand region can effectively supervise the training of the network under L2 loss. As listed in Table 1, the shape-aware heat map is a simple but practical strategy.

3.2 Tremor Pose Pre-estimation

Before eliminating the tremor component, the tremor pose pre-estimation module is proposed to obtain an accurate hand pose. The preprocessing module generates a rough estimation, since CNN-based methods only capture the structural constraint by hand designed. There is rich structure information among hand joints, which is limitedly captured by CNN-based methods.

We design a CAM-GNN module to refine the rough estimation by establishing the spatial-temporal constraint. This is motivated by the fact that GNN can learn the rich structural constraint by propagating and aggregating messages among neighbors. The input is the rough estimation result H. Firstly, the single-frame CAM-GNN refines H to yield the refined result Ls^t, $Ls^t = \{ls_i^t | ls_i^t = (x_i, y_i, z_i), i = 1 : N\}$. The multi-frame CAM-GNN further optimizes Ls^t to produce a more accurate result Lp^t, $Lp^t = \{lp_i^t | lp_i^t = (x_i, y_i, z_i), i = 1 : N\}$.

3.2.1 Constraint Adjacency Matrix

Recapping Graph Neural Network. Typically, the Graph Neural Network handles the graph-structured data. Given a graph $G =$

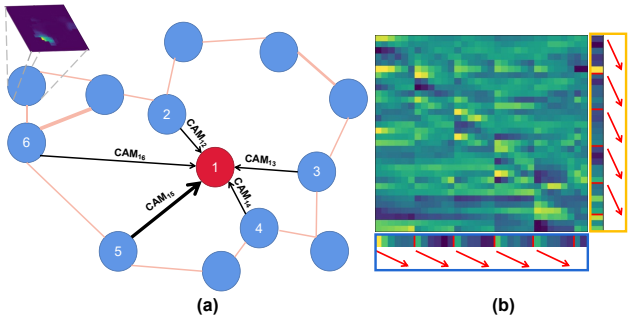

Figure 4: (a): The aggregation of node features. A hand graph (the graph is simplified for illustration purpose.) The thickness of each edge indicates the value of CAM_{ij}. For the red node, FA accumulates features of neighbor nodes multiplied by CAM_{ij}. (b): The visualization of learned $CAM \in R^{32\times32}$ on NYU dataset. The value of 32 is defined based on the total number of joints, which comprise 30 finger joints (6 joints of each finger) as well as one palm joint and one wrist joint. Both the rows and the columns of CAM maintain the same order of different hand joints, including the little finger tip, the digital pulp of the little finger, the palm, etc. Each CAM entry shows the correlation between a joint pair, where brighter color indicates that the two joints are more correlative. Values of the blue box and the yellow box are the cumulative sum of row and column, respectively.

(V,E), where nodes are represented as $V = \{v_i | i = 1:N\}$ and edges are defined as $E = \{\langle v_i, v_j \rangle | i, j = 1:N\}$, the graph is updated by a node-to-node message passing mechanism. For the node v_k, features of neighbor nodes Ω are first aggregated by an aggregate function FA defined as follows:

$$f_k^{\Omega} = FA(f_k^{in}) = \sum_{j=1}^{m} (f_j^{in}) \tag{1}$$

where f_j^{in} is the feature of the neighbor node v_j and m is the number of neighbor nodes. Then, the feature of v_k and neighbor information f_k^{Ω} are concatenated to update the new state of v_k as follows:

$$f_k^{out} = h(f_k^{\Omega}, f_k^{in}) \tag{2}$$

where the output function h is neural networks, such as fully connected networks or convolutional neural networks.

Single-frame CAM-GNN. For tremor hand pose estimation, we define a hand graph G_{hand} with rough estimation results H, where the number of nodes is set based on the number of hand joints and the feature of each node is a heat map of a joint. The graph topology is vital for hand pose estimation, which represents the constraint relationship among joints.

A straightforward and brute force way is to define a complete graph to model the structural relationship among nodes. For hand pose estimation, it means that each joint is related to other joints and the contribution of each joint is equal when aggregating neighborhood information. Experiments show that a network trained based on this graph topology does not coverage. The reason is that the aggregated information is redundant. Besides, such a topology setting also resembles the structural and multi-branch methods, offering only a fixed skeleton relationship.

Hence, we propose a learnable constraint adjacency matrix (CAM) to characterize the graph topology dynamically. This allows covering the rich potential constraint among joints, as oppose to a manually defined topology that has limited coverage. CAM is defined as $CAM = (CAM_{ij}) \in R^{N\times N}$, where N is the number of joints. Each value of CAM is initialized to a random number between -1 and 1, forming an uncertain graph. During training, the

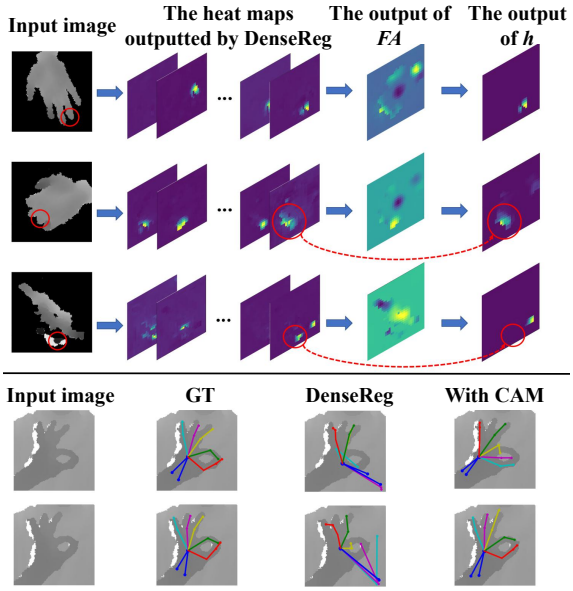

Figure 5: Some qualitative results obtained by our CAM-GNN module. Top: Our CAM-GNN module corrects the result of ring fingertip estimated by DenseReg. The first row shows the case that DenseReg estimates well. In the following two rows, the CAM-GNN module corrects the incorrectly predicted heat maps by DenseReg. Bottom: The failed case (first row) and the success case (second row) about correcting errors of DenseReg.

graph topology is dynamically adjusted along with other network parameters under a gradient-based optimizer, such as Adam. Edges beneficial to the loss function are strengthened, and vice versa. Finally, the trained CAM defines the optimal topology for the hand graph. With CAM, the aggregation function FA is updated by:

$$f_k^{\Omega} = FA(CAM, H) = \sum_{j=1}^{N} (CAM_{kj} * h_j) \tag{3}$$

The aggregation of node features is shown in Figure 4(a). The function h is implemented by the non-local module [38] and a convolution layer. This CAM-GNN module is trained under the supervision of the cross entropy loss. Finally, after an argmax operation on output heat maps, the pre-estimation Ls^t is obtained, which is a coordinate vector with the size of $3 \times N$. Ls^t is the refinement of the preprocessing result under the spatial constraint formed by the CAM-GNN module.

Figure 5 shows some results of the heat maps generated by FA and h. The heat map from FA shows the strength of correlation among different hand joints, where regions with brighter colors indicate that joints located nearby have stronger correlation, and vice versa. As depicted from the second and the third rows of Figure 5, the heat map from h shows our network can successfully correct the confusing results obtained from DenseReg [37] through CAM-GNN.

3.2.2 Multi-frame CAM-GNN

While performing CAM-GNN on a single frame image can improve the accuracy of CNN-based methods, by inspecting the failed case as shown in Figure 5, we found that it cannot handle hole or noise artifact very well. Since gestures are continuous, we propose a novel tactic to address this problem, which extends CAM-GNN for multi-frame to exploit the multi-frame spatial-temporal constraint. The extra temporal information utilizes the continuity of gestures to effectively compensate short-term artifacts.

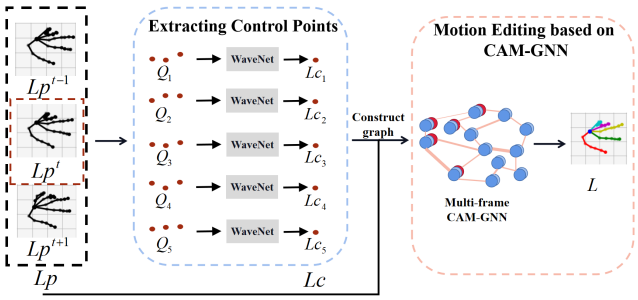

Figure 6: The pipeline of the tremor compensation module. The hand pose with tremor Lp is the result of pre-estimation module. The graph is a simplified graph for exhibition. Fingertips of successive frames are fed into WaveNet to extract control points Lc in parallel. The concatenation of control points and fingertip coordinates is the input feature of fingertip node. L is the result of eliminating the tremor of Lp based on Lc.

Here, we define a graph $G_{multi-frame}$ for multi-frame CAM-GNN. The number of nodes is also the number of hand joints. The feature of each node f_k^{in}, $f_k^{in} = [ls_k^1, ..., ls_k^t]$, is the concatenation of multi-frame coordinate vectors outputted by single-frame CAM-GNN. The multi-frame CAM-GNN module also includes an aggregate function FA and an output function h. The function FA is structurally similar to Equation 3, which is shown as follows:

$$f_k^{tmp1} = FA(CAM, Ls^T) = \sum_{j=1}^{N} (CAM_{kj} * f_j^{in}) \tag{4}$$

We set T as 8, which is mentioned in the experiment section. The function h is implemented with fully connected layers. With the supervision of L2 loss, the topology is also dynamically adjusted to the optimal by CAM. The more accurate result Lp^t is yielded under spatial-temporal constraints formed by multi-frame CAM-GNN.

3.3 Tremor Compensation Module

A tremor hand pose Lp^t consists of both tremor component α and voluntary motion L, which is defined as follows:

$$Lp^t = g(L, \alpha) \tag{5}$$

where g is a combination function. The tremor compensation module is proposed to eliminate the tremor component of Lp^t. Applying an existing tremor compensation method to estimate the stable state of each joint in parallel, the estimated hand pose will not meet the physiological constraint, because of cumulative error and other factors. Hence, we propose a tremor compensation method based on control points. As illustrated in Figure 6, this method is constituted by two components, extracting control points and motion editing based on CAM-GNN.

3.3.1 Extracting Control Points

The Definition of Control Points. The tremor motion is around a target object or a desired pose. We extract some points to restrain tremor hand poses, which is named as control points Lc. Generally, fingertips are end controllers of 3D manipulations in VR applications, which represent the user intent. As shown in Figure 4(b), values in the blue box indicate that fingertips have stronger weights than other joints. Values in the yellow box reveal that other joints have weaker weights than fingertips in message propagating. Hence, we extract voluntary motions of fingertips as control points.

Extracting Control Points with WaveNet. We adopt WaveNet [35] to extract control points Lc. WaveNet is a deep neural network for generating raw audio waveforms. Analogously, we exploit WaveNet to generate fingertip coordinates without tremor in parallel. In other words, WaveNet generates a control point for one fingertip at a time. Given coordinates of fingertip i from successive frames, $Q_i = [lp_i^1, lp_i^2, ..., lp_i^T]$, WaveNet builds a conditional probability model to predict a control point Lc_i. The probability is factorised as a product of conditional probabilities as follows:

$$p(Lc_i) = \prod_{t=1}^{T} p(lp_i^t | lp_i^1, lp_i^2, ..., lp_i^{t-1}) \tag{6}$$

3.3.2 Motion Editing Based on CAM-GNN

For obtaining L in Equation 5, we introduce compensation ϕ for α to make Lp^t consistent with control points Lc, as follows:

$$L = \phi(Lc, Lp^t) \tag{7}$$

The tremor motion is a temporal movement, shaking around a desired pose. Hence, we novelly import the motion editing module based on multi-frame CAM-GNN as ϕ, which employs tremor hand poses of successive frames and control points to generate the non-tremor hand pose. The multi-frame CAM-GNN module models the constraint between tremor hand poses and control points.

The graph fed into multi-frame CAM-GNN is defined as G_{edit}. The number of nodes in G_{edit} is the number of joints. For each node corresponding to a fingertip, its feature is the concatenation of pre-estimation results of T frames and a corresponding control point. The feature of other nodes is pre-estimation results of T frames. The formula of a node feature is shown as follows:

$$f_k^{in} = \begin{cases} [Lp_k^t, Lc_k], t = 1:T & v_k \Leftrightarrow \text{fingertip} \\ [Lp_k^t], t = 1:T & otherwise \end{cases} \tag{8}$$

We set T as 8, which is validated in the experiment section.

In this module, the function FA is the same as Equation 4. Since node features are still coordinates, the function h is also implemented by fully connected layers. The loss function is also L2 loss. During propagating message and updating feature, CAM dynamically adjusts the graph topology to the optimal through CAM, forming the constraint between tremor hand poses and control points. The motion editing based on CAM-GNN achieves that the stabilized pose L complies with the desired pose constraint and the spatial constraint.

4 Experiments

4.1 Implementation Details

We employ DenseReg [37] as the preprocessing module to generate a rough estimation. We form our tremor pose pre-estimation module by extending DenseReg with our single-frame CAM-GNN module and multi-frame CAM-GNN module. The whole module is jointly trained with DenseReg, where the training parameters are adopted from DenseReg.

For the tremor compensation module, WaveNet is applied to extract control points. WaveNet is trained separately. The training parameters refer to WaveNet [35]. After training WaveNet, we freeze the preprocessing module, tremor pose pre-estimation and WaveNet to train the motion editing module. When training the motion editing module, the parameters is the same as the pre-estimation module.

4.2 Evaluation of Tremor Pose Pre-estimation

4.2.1 Datasets and Metrics

Datasets. We evaluate our method for 3D hand pose estimation on two publicly hand pose datasets, NYU dataset [34] and MSRA15 [32] dataset. The NYU dataset is proposed by Tompson et al. [34]. It provides 8252 test images and 72757 training images captured by a depth camera. The ground-truth label for a hand is 3D coordinates of 36 joints. The MSRA15 dataset consists of 76375 RGBD images.

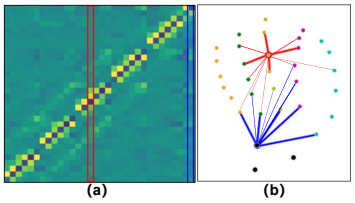

Figure 7: (a) The visualization of learned $CAM \in R^{32\times32}$ for a layer of GNN on NYU dataset. Here, CAM has the same setting as that in Figure 4. Brighter color indicates that two joints are more correlative. (b) The skeleton corresponding to the learned CAM.

Table 1: The result of ablation experiments on NYU dataset.

Methods	The mean error of all joints (mm)
DenseReg	10.241
DenseReg+GNN(Shape-aware heat map)	9.58
DenseReg+CAM-GNN(Gaussian map)	9.73
DenseReg+CAM-GNN(Shape-aware heat map)	9.498

In this dataset, there are 21 joints labeled with 3D coordinates. We adopt these two datasets because they provide adequate labeled depth images for hand pose estimation.

Metrics. We evaluate our method on three metrics. One of metrics is the per-joint mean error, which is averaged on all images. The second metric is the mean error of all joints. The error is the Euclidean distance between estimated joints and ground-truth. The third metric is the error of the phalange length, which is used to evaluate the physiologically constraint.

4.2.2 Ablation Studies

Ablation Study for CAM. As depicted in Figure 5, our CAM-GNN corrects the error of DenseReg. The result of ablation experiments on NYU dataset is listed in Table 1. It indicates that the CAM-GNN module greatly improves the performance of DenseReg. Mean errors of the most joints are reduced under the CAM-GNN module as shown in Figure 8 (a). The error of the phalange length is also significantly reduced from 4.127mm to 3.577mm with the CAM-GNN module, indicating that the constraint formed by CAM makes the estimated hand pose conform to the physiologically constraint very well.

We compared the performance of GNN on the manual fixed topology with the learned topology formed by CAM. In the experiment, we set the skeleton topology as the fixed topology. As manifested in Table 1, the mean error of the fixed graph on GNN is 9.58mm, which exceeded our method, 9.498mm. The learned CAM is visualized in Figure 7(a). Values in red box indicate the relationship between the digital pulp and other joints. Values in blue boxes indicate the relationship between the palm and other joints. The brighter locations in CAM exactly correspond to joints directly connected by the skeleton, as shown in Figure 7(b). Moreover, joints that are not directly connected by the skeleton are also related to each other, as shown in Figures 7(a) and 7(b). The result demonstrates that the learned CAM can capture rich potential constraints among joints.

Table 2: The performance of our CAM-GNN module on detection-based and regression-based methods. The metric is the mean error of all joints.

Methods	NYU	MSRA15
DenseReg	10.241	7.234
DenseReg+(CAM-GNN)	9.498	6.523
CPM	19.85	8.54
CPM+(CAM-GNN)	17.79	7.0
DeepPrior++	20.75	8.67
DeepPrior+++(CAM-GNN)	18.467	7.185

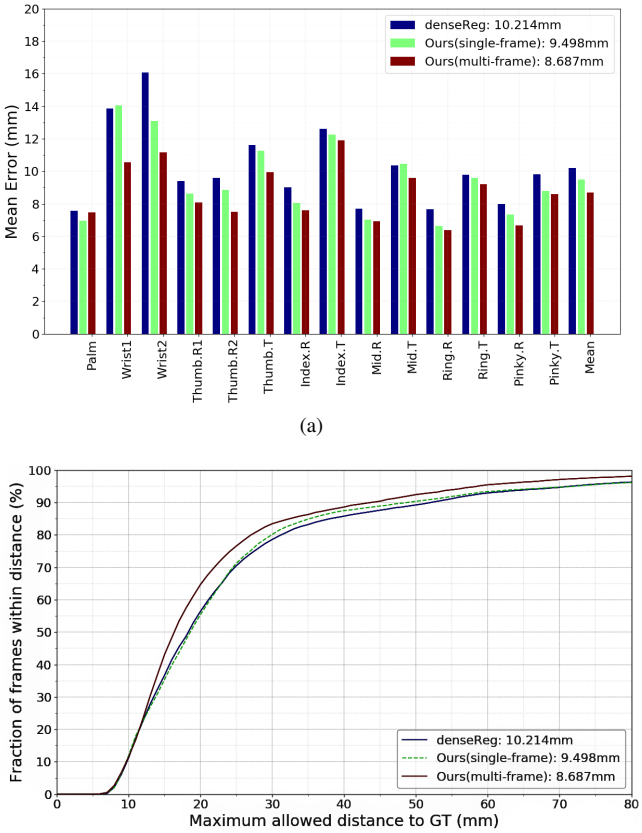

Figure 8: The ablation study for the multi-frame CAM-GNN module on the NYU dataset. (a): The mean error of per joint. (b): The fraction of frames within a certain distance. Mean errors of all joints on methods are shown in the legend.

DenseReg is a detection-based method. There are another methods based on regression, such as CPM [40], DeepPrior++ [21]. We also carried out experiments on regression-based methods to verify the effectiveness of CAM-GNN. The result is shown in Table 2. On NYU dataset, the CAM-GNN module reduces the error of the regression-based method by 2.04mm and 2.283mm. The maximum reduced error is 1.54mm on MSRA dataset. The result indicates that the CAM-GNN module can be used as an independent post-processing module to improve the accuracy of existing methods.

Ablation Study for the Multi-frame CAM-GNN Module. The result in Figure 8(a) shows that the multi-frame CAM-GNN module further reduces the mean error of most joints. The mean error is reduced from 9.498mm to 8.687mm. Moreover, as in Figure 8(b), given a maximum allowed error distance from the ground truth, the fraction of frames that have all predicted joints within the threshold is significantly increased on the basis of the single frame CAM-GNN. The improvement is significant at the 20mm threshold. Supancic et al. [33] stated that the error distance between the manual label and the ground truth is about 20mm. The improvements of our multi-frame CAM-GNN module demonstrate that our method effectively reduces errors by strengthening the spatial-temporal constraint.

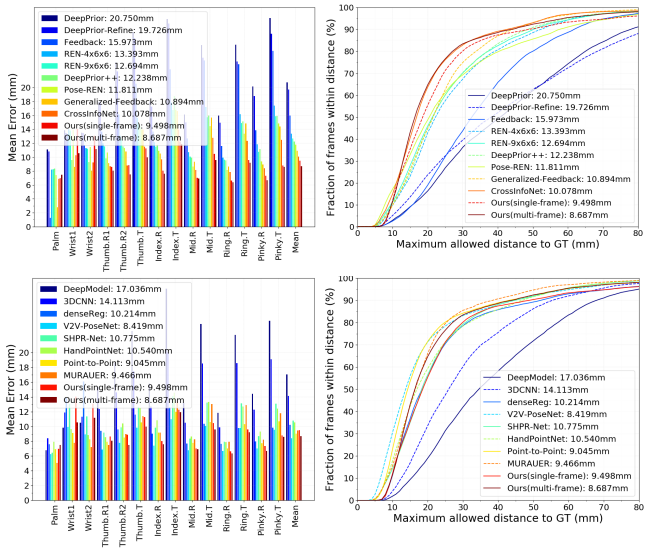

Figure 9: The performance of different methods on the NYU validation set. Left: The mean error of per joint. Right: The fraction of frames within a certain distance. Mean errors of all joints on methods are shown in the legend.

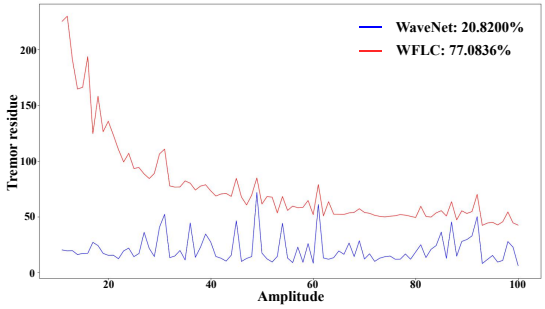

Figure 10: The tremor residue line chart of WaveNet and WFLC on the motion trajectories of different amplitudes. Amplitude refers to the amplitude of the tremor.

4.2.3 Comparison with the State-of-the-art

We compare our method on NYU dataset with state-of-the-art methods, which are committed to explore joint constraints. These methods are divided into two categories, structural methods and multi-branch methods. Structural methods include Feedback [23], Generalized-Feedback [24], DeepPrior [22] and DeepPrior++ [21]. Multi-branch methods include REN [11], Pose-REN [4], and CrossInfoNet [7]. In Figure 9 (top), the experiment result demonstrates that our method achieves the best performance. The proposed method achieves an error of 8.687mm on the validation set, which is the least than others.

In addition, we compare our method with other existing methods [3], not limited to methods that explore joint constraints. As depicted in Figure 9 (bottom), the result of our method is only inferior to V2V-PoseNet [20]. The reason is that V2V-PoseNet achieved the performance by stacking 10 times. The result of V2V-PoseNet baseline is 9.22mm. In the work of Jameel et al. [18], the mean error is 8.72mm. It indicates that our method is decent for 3D hand pose estimation.

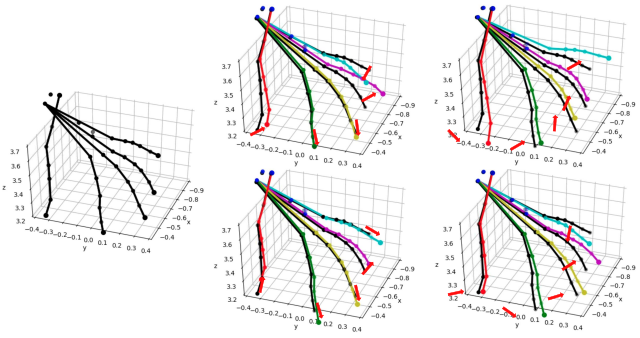

Figure 11: The intuitive visualization of motion editing based on the different control points. The black hand pose is the pose before motion editing. The red arrows represent different offsets. The color hand pose is the result of motion editing.

4.3 The Evaluation of Tremor Compensation

For evaluating the tremor compensation module, we firstly evaluate these two submodules, extracting control points and motion editing based on CAM-GNN, and then we verify the feasibility.

4.3.1 Evaluation of Extracting Control Points

Dataset and Metric. Since the existing tremor motion dataset TIM-Tremor [27] is collected with an accelerometer sensor (ACL300, 1000Hz) attached to wrists of 55 patients, the frequency is inconsistent with the frequency of image acquisition and the number of joint is also inconsistent with hand pose estimation. To align with the frame rate of NYU dataset, we firstly downsample TIM-Tremor dataset by equidistant sampling. The downsampled tremor signal is converted to discrete trajectory coordinates by double integral. After coordinate conversion, the tremor coordinates is added to fingertip coordinates of NYU dataset, forming a tremor fingertip dataset to train WaveNet. The non-tremor fingertip coordinates in original NYU dataset are the ground truth. The division of the training set and the test set is the same as NYU dataset. For evaluating the performance of extracting control points, we employ the tremor residue as the metric, which is the proportion of residual tremor to tremor motion.

Comparison with Other Methods. We compare our method with other methods based on Fourier Linear Combinator (FLC). Here, we choose Weighted Fourier Linear Combiner (WFLC) [1] for comparison. As depicted in Figure 10, there is a considerable gap between WaveNet and WFLC at different amplitudes. Especially at low amplitudes, the tremor residue of WFLC is greater than 1, since the output of WFLC has a time delay. It illustrates that our method is more accurate than FLC-based methods.

4.3.2 The Evaluation of Motion Editing Based on CAM-GNN

Dataset and Metric. For training and evaluating the motion editing module, we generate a tremor hand pose dataset, analogous to Section 4.3.1. The fingertip in NYU dataset is selected as control points. The tremor hand pose is formed by superposing the downsampled TIM-Tremor on NYU dataset. Finally, the data set is constituted by replacing the tremor fingertip coordinates with control points. The ground truth is the non-tremor hand pose in NYU dataset. The division of the training set and the test set is also the same as NYU dataset. The metric is also the tremor residue.

Quantitative Experiments on Motion Editing. In order to figure out how many frames can achieve the best performance, we conduct quantitative experiments on the multi-frame CAM-GNN module. Experiments indicate that the tremor residue decreases steadily with the increase of input frames, and gradually becomes stable when the number of frames reaches 8. It demonstrates that

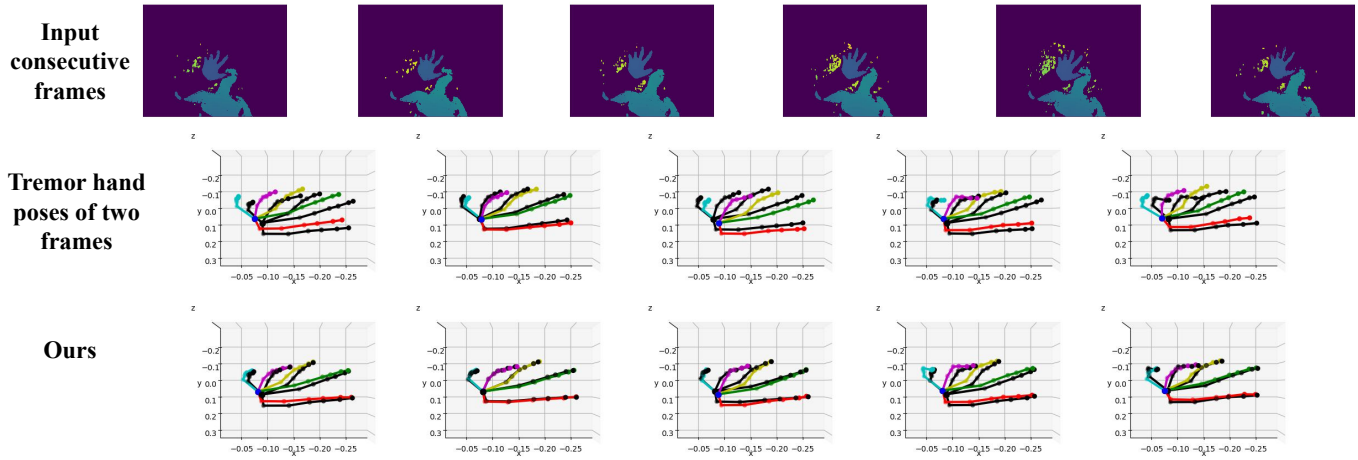

Figure 12: The result of tremor compensation on the static hand with tremor. The black and color hand pose are respectively represented as results of neighbor frames.

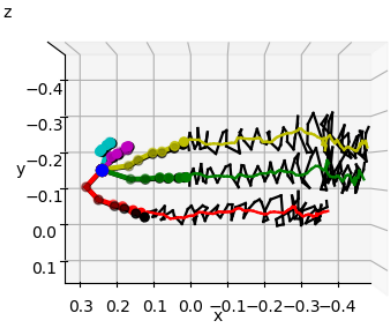

Figure 13: The result of tremor compensation on the special hand motion with tremor. The black and color hand pose are respectively represented as the hand pose before and after tremor compensation. The black and color curves show the trajectory of their fingertips.

applying multi-frame CAM-GNN to motion editing can form the temporal constraint between the desired hand pose and the tremor hand pose within specific frames, where the desired hand pose is the intent of tremor hand pose.

Qualitative Experiments on Motion Editing. We carried out qualitative experiments to evaluate motion editing module on NYU dataset. The aim of motion editing is to make tremor fingertips consistent to control points, simultaneously other joints consistent to anatomical and desired pose constraints. In the experiment, hand poses of consecutive frames from NYU dataset are fed into the module. Control points are fingertip coordinates of the last frame added random offset. The result of motion editing based on the different control points is presented in Figure 11. It demonstrates that the motion editing based on control points yields the reasonable hand pose, which conforms to the constraint of anatomy and control points.

4.3.3 Qualitative Experiments on Tremor Compensation.

The tremor compensation module is an independent module to eliminate the tremor component. For verifying the availability of the module, we collected two sets of tremor hand motion by a depth camera (Intel RealSense SR300), the static hand with tremor and the specific hand motion with tremor. The setting of the environment for collecting test data is shown in the left part of Figure 1. A user who stood in front of the camera performed the specific gesture motion. We take drawing a line as the specific hand motion, as the trajectory of which intuitively demonstrates the effect of tremor compensation.

For the static hand with tremor, the ideal estimation is stable and relatively static. Hence, we compare hand poses of neighbor frames before and after compensation. Results of randomly selected 6 consecutive frames are shown in Figure 12. Before tremor compensation, there are significant variations between hand poses of neighbor frames. After tremor compensation, hand poses of neighbor frames are more closer to each other and these hand poses are more stable. It demonstrates that our tremor compensation method is valid when the tremor occurs on the static hand.

For the moving hand with tremor, the ideal motion trajectory is smooth and stable. Therefore, we select the motion trajectory of fingertips as the presentation of results. As shown in Figure 13, the hand poses are the results of the last frame. Curves are fingertip trajectories of all preceding frames. In the scenario of drawing a line, the tremor leads to conspicuous fluctuation on the fingertip trajectories. We find that the fingertip trajectories are more smooth under the tremor compensation module. It indicates that our method works well on the moving hand with tremor.

5 Conclusion

The tremor on a hand is a potential obstacle when performing 3D manipulations on virtual objects. In this work, we proposed a neoteric method based on Graph Neural Network for 3D hand pose under tremor. We first generate an accurate 3D hand pose with tremor. Concretely, we invent a shape-aware heat map in the preprocessing module to improve the estimation accuracy of CNN-based methods. Since CNN-based methods limitedly capture potential constraints among joints, we design a CAM-GNN pre-estimation module for single frame and multi-frame to learn the spatial-temporal constraint, which further improves accuracy of CNN-based methods as an independent module. Then we devise a tremor compensation method to eliminate tremor components, which adopts multi-frame CAM-GNN to edit the tremor hand pose based on control points. For training the tremor compensation module, we build tremor hand pose datasets based on the NYU and TIM-Tremor dataset. Finally, experiments demonstrate that our CAM-GNN method improves the performance of existing CNN-based methods as a transplantable module. Moreover, the tremor compensation method is a novel method to effectively eliminate the tremor, which does not need to wear additional equipments on the hand.

REFERENCES

[1] K. Adhikari, S. Tatinati, W. T. Ang, K. Veluvolu, and K. Nazarpour. A quaternion weighted fourier linear combiner for modeling physiological tremor. *IEEE transactions on bio-medical engineering*, pp. 2336–2346, 2016.

[2] Y. Cai, L. Ge, J. Liu, J. Cai, and N. M. Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[3] X. Chen. Awesome hand pose estimation. https://github.com/xinghaochen/awesome-hand-pose-estimation.

[4] X. Chen, G. Wang, H. Guo, and C. Zhang. Pose guided structured region ensemble network for cascaded hand pose estimation. *Neurocomputing*, 395:138–149, 2020.

[5] X. Deng, S. Yang, Y. Zhang, P. Tan, L. Chang, and H. Wang. Hand3d: Hand pose estimation using 3d neural network. *CoRR*, abs/1704.02224, 2017.

[6] B. Doosti. Hand pose estimation: A survey. *CoRR*, abs/1903.01013, 2019.

[7] K. Du, X. Lin, Y. Sun, and X. Ma. Crossinfonet: Multi-task information sharing based hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9896–9905, 2019.

[8] L. Ge, H. Liang, J. Yuan, and D. Thalmann. Robust 3d hand pose estimation in single depth images: From single-view CNN to multi-view cnns. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 3593–3601, 2016.

[9] L. Ge, H. Liang, J. Yuan, and D. Thalmann. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 5679–5688, 2017.

[10] J. G. González, J. G. Gonz'Alez, J. G. Gonz'Alez, J. C. Cavanaugh, and D. Ph. A new approach to suppressing abnormal tremor through signal equalization. *Proc.resna Annu.conf*, pp. 707–709, 1995.

[11] H. Guo, G. Wang, X. Chen, and C. Zhang. Towards good practices for deep 3d hand pose estimation. *arXiv preprint arXiv:1707.07248*, 2017.

[12] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, and H. Yang. Region ensemble network: Improving convolutional network for hand pose estimation. In *IEEE International Conference on Image Processing*, pp. 4512–4516, 2017.

[13] M. Kampffmeyer, Y. Chen, X. Liang, H. Wang, Y. Zhang, and E. P. Xing. Rethinking knowledge graph propagation for zero-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 11487–11496, 2019.

[14] C. W. Lee, W. Fang, C. K. Yeh, and Y. C. F. Wang. Multi-label zero-shot learning with structured knowledge graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[15] W. Li, Z. Wang, B. Yin, Q. Peng, Y. Du, T. Xiao, G. Yu, H. Lu, Y. Wei, and J. Sun. Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*, 2019.

[16] Y. Liu, R. Wang, S. Shan, and X. Chen. Structure inference net: Object detection using scene-level context and instance-level relationships. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[17] M. Madadi, S. Escalera, X. Baró, and J. Gonzalez. End-to-end global to local cnn learning for hand pose recovery in depth data. *arXiv preprint arXiv:1705.09606*, 2017.

[18] J. Malik, I. Abdelaziz, A. Elhayek, S. Shimada, S. A. Ali, V. Golyanik, C. Theobalt, and D. Stricker. Handvoxnet: Deep voxel-based network for 3d hand shape and pose estimation from a single depth map. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7111–7120, 2020.

[19] M. Miwa and M. Bansal. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

[20] G. Moon, J. Yong Chang, and K. Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5079–5088, 2018.

[21] M. Oberweger and V. Lepetit. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 585–594, 2017.

[22] M. Oberweger, P. Wohlhart, and V. Lepetit. Hands deep in deep learning for hand pose estimation. *arXiv preprint arXiv:1502.06807*, 2015.

[23] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a feedback loop for hand pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pp. 3316–3324, 2015.

[24] M. Oberweger, P. Wohlhart, and V. Lepetit. Generalized feedback loop for joint hand-object pose estimation. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[25] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 1989.

[26] G. Park, A. Argyros, J. Lee, and W. Woo. 3d hand tracking in the presence of excessive motion blur. *IEEE Transactions on Visualization and Computer Graphics*, 26(5):1891–1901, 2020.

[27] S. L. Pintea, J. Zheng, X. Li, P. J. M. Bank, J. J. van Hilten, and J. C. van Gemert. Hand-tremor frequency estimation in videos. In *ECCV Workshops (6)*, vol. 11134, pp. 213–228, 2018.

[28] P. O. Riley and M. J. Rosen. Evaluating manual control devices for those with tremor disability. *Journal of Rehabilitation Research & Development*, 24(2):99, 1987.

[29] J. Shahed and J. Jankovic. Exploring the relationship between essential tremor and parkinson's disease. *Parkinsonism Relat Disord*, 13(2):67–76, 2007.

[30] A. Spurr, J. Song, S. Park, and O. Hilliges. Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 89–98, 2018.

[31] X. Sun, J. Shang, S. Liang, and Y. Wei. Compositional human pose regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2602–2611, 2017.

[32] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun. Cascaded hand pose regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[33] J. S. Supančič, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan. Depth-based hand pose estimation: methods, data, and challenges. *International Journal of Computer Vision*, 126(11):1180–1198, 2018.

[34] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33, August 2014.

[35] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. In *The 9th ISCA Speech Synthesis Workshop*, p. 125, 2016.

[36] K. C. Veluvolu and W. T. Ang. Estimation and filtering of physiological tremor for real-time compensation in surgical robotics applications. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 6(3):334–342, 2010.

[37] C. Wan, T. Probst, L. V. Gool, and A. Yao. Dense 3d regression for hand pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[38] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, 2018.

[39] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

[40] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4724–4732, 2016.

[41] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[42] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.

[43] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei. Model-based deep hand pose estimation. *arXiv preprint arXiv:1606.06854*, 2016.