

# Context-Aware Deep Spatio-Temporal Network for Hand Pose Estimation from Depth Images

Yiming Wu, Wei Ji, Xi Li\*, Gang Wang, Jianwei Yin and Fei Wu

**Abstract**—As a fundamental and challenging problem in computer vision, hand pose estimation aims to estimate the hand joint locations from depth images. Typically, the problem is modeled as learning a mapping function from images to hand joint coordinates in a data-driven manner. In this paper, we propose Context-Aware Deep Spatio-Temporal Network (CADSTN), a novel method to jointly model the spatio-temporal properties for hand pose estimation. Our proposed network is able to learn the representations of the spatial information and the temporal structure from the image sequences. Moreover, by adopting adaptive fusion method, the model is capable of dynamically weighting different predictions to lay emphasis on sufficient context. Our method is examined on two common benchmarks, the experimental results demonstrate that our proposed approach achieves the best or the second-best performance with state-of-the-art methods and runs in 60fps.

**Index Terms**—Hand Pose Estimation, Context-Aware Deep Spatio-Temporal Network, Adaptive Fusion.

## I. INTRODUCTION

HAND pose estimation is a fundamental and challenging problem in computer vision, and it has a wide range of vision applications such as human-computer interface (HCI) [1], [2] and augmentation reality (AR) [3]. So far, there have been a number of methods [4]–[17] proposed on this topic, and significant progress has been made with the emergence of deep learning and low-cost depth sensors. The performance of previous RGB image based methods is limited to the clustered background, while the depth sensor is able to generate depth images in low illumination conditions, and it is simple to do hand detection and hand segmentation. Nevertheless, it is still difficult to accurately estimate 3D hand pose in the practical scenarios due to data noises, high freedom of degree for hand motion, and self-occlusions between fingers.

The typical pipeline for estimating hand joints from depth images could be separated into two stages: 1) extracting robust features from depth images; 2) regressing hand pose based on the extracted features. Current approaches concentrate on improving the different algorithms in each of the stages. Specifically, in [4], the features extracted by multi-view CNNs are utilized for regression, then the same author in [6] replaces

Y. Wu, W. Ji, J. Yin, F. Wu are with College of Computer Science, Zhejiang University, Hangzhou 310027, China (e-mail: ymw, jiwei@zju.edu.cn zjuyjw, wufei@cs.zju.edu.cn).

X. Li\*(corresponding author) is with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China, and also with the Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Hangzhou 310027, China (e-mail: xilizju@zju.edu.cn).

Gang Wang is with AI Labs of Alibaba Group, Hangzhou 310027, China (email: gangwang6@gmail.com).

the 2D CNN by 3D CNN to fully exploit 3D spatial information. Besides, hierarchical hand joint regression [10] and iterative refinement [18], [19] for hand pose are deployed in several approaches.

In this paper, we propose a novel method named as CADSTN (Context-Aware Deep Spatio-Temporal Network) to jointly model the spatio-temporal context for 3D hand pose estimation. We adopt sliced 3D volumetric representation, which keeps the structure of the hand to explicitly model the depth-aware spatial context. Moreover, motion coherence between the successive frames is exploited to enforce the smoothness of the predictions of the images in sequence. The model is able to learn the representations of the spatial information and the temporal structure in the image sequences. Moreover, the model is capable of combining the spatio-temporal properties for final prediction. As shown in Figure 1, the architecture is composed of three parts: first, we sufficiently capture the spatial information via *Spatial Network*; second, the dependency between frames is modeled via a set of LSTM nodes in *Temporal Network*; third, to exploit spatial and temporal context simultaneously, the predictions from the above two networks are utilized for the final prediction via *Fusion Network*. The main contributions of our work are summarized as the following two folds:

- First, we propose a unified spatio-temporal context modeling approach with group input and group output, which benefits from the inter-image correspondence structures between consecutive frames. The proposed approach extracts the feature representation for both unary image and successive frames, which generally leads to the performance improvement of hand pose estimation.
- Second, we present an end-to-end neural network model to jointly model the temporal dependency relationships among the multiple images while preserving spatial information for an individual image in a totally data-driven manner. Moreover, an adaptive fusion method is enabled to make the network dynamically adapt to the various situations.

We evaluate our proposed method on two public datasets: NYU [20] and ICVL [8]. With detailed analysis of the effects of different components of our method, and the experimental results demonstrate that most results of our method are best comparing with state-of-the-art methods. Our method runs in 60fps on a single GPU that satisfies the practical requirements<sup>1</sup>.

<sup>1</sup>A demo video is available online at <https://goo.gl/tbnWse>

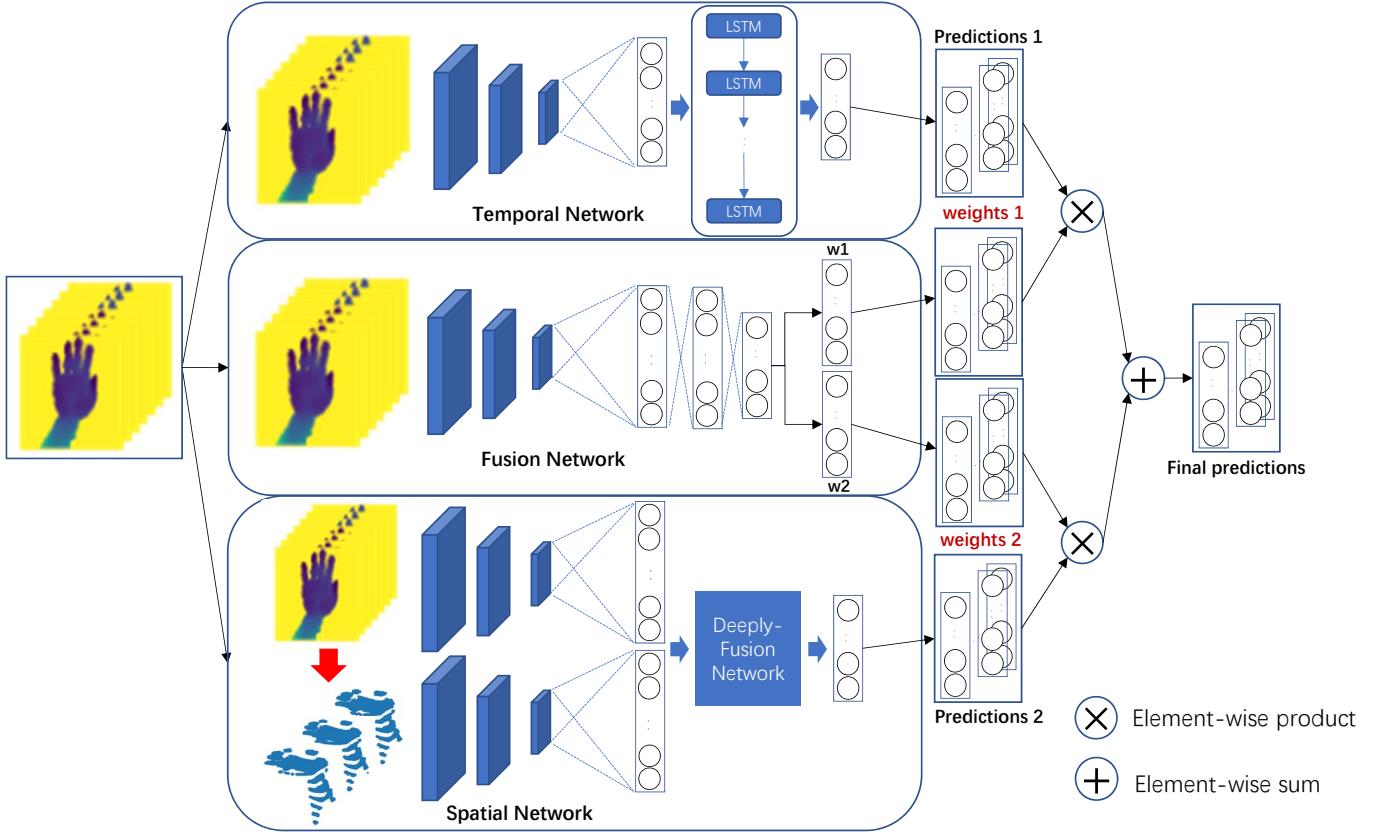


Fig. 1. The overview of total network. The network is composed of three parts: *Temporal Network*, *Spatial Network* and *Fusion Network*. (top) *Temporal Network* extracts the features considering the temporal coherence of input images via LSTM, and outputs a sequence of joint locations. (bottom) *Spatial Network* employs the Deeply-Fusion framework to hierarchically fuse the features from a single depth image and corresponding sliced 3D volumetric representations, then gives out prediction. (center) *Fusion Network* fuses the predictions from two networks, it learns the weights for each prediction and the final prediction is calculated as summation of weighted predictions.

The rest of the paper is organized as follows. In Section II, we review some works for 3D hand pose estimation. In Section III, we introduce our proposed method. In Section IV, we present the experimental results of the comparison with state-of-the-art methods and ablation study. Finally, the paper is concluded in Section V.

## II. RELATED WORK

### A. Feature Representation Learning

Most state-of-the-art performance hand pose estimation methods based on deep learning rely on visual representation learned from data. Given a depth image, different methods try to learn the sufficient context and get the robust feature. For example, in [18], the original segmented hand image is downsampled by several factors, and each scaled image is followed with a neural network to extract multi-scale feature. Although increasing the number of scale factor may improve the performance, the computational complexity will also be increased. In [21], Guo et al. propose a regional feature ensemble neural network and extend it into 3D human pose estimation in [22], Chen et al. [23] propose a cascaded framework denoted as Pose-REN to improve the performance of region ensemble network. In [24], various combinations of voxel and pixel representations are experimented, and finally

use a voxel-to-voxel prediction framework to estimate per-voxel likelihood.

Furthermore, recovering the fully 3D information catches researcher's attention. Multi-View CNN [4] is introduced for hand pose estimation, which exploits depth cues to recover 3D information of hand joints. In [5], 3D CNN and truncated signed distance function (TSDF) [25] are adopted to learn both the 3D feature and the 3D global context. Ge et al. [6] take advantage of the above two methods, and propose a multi-view 3D CNN to regress the hand joint locations. In addition, data augmentation is usually considered for extracting robust feature. A synthesizer CNN is used to generate synthesized images to predict an estimate of the 3D pose by using a feedback loop in [19]. Wan et al. [26] use two generative models, variational autoencoder (VAE) [27] and generative adversarial network (GAN) [28], to learn the manifold of the hand pose, with an alignment operation, pose space is projected into a corresponding depth map space via a shared latent space. Our method is different from the aforementioned architectures in the following ways: 1) we deploy the sliced 3D volumetric representation to recover 3D spatial information. 2) we extract the temporal property from an image sequence to enhance the consistency between images.

### B. Spatio-Temporal Modeling

Towards the tasks based on the video, spatio-temporal modeling is widely used (e.g. hand tracking, human pose tracking and action recognition).

In the matters of hand tracking problems, Franziska et al. [29] propose a kinematic pose tracking energy to estimate the joint angles and to overcome the challenge occlusion cases. In [30], Oberweger et al. propose a semi-automated method to annotate the hand pose in an image sequence by exploiting spatial, temporal and appearance constraints.

In the context of human pose tracking, Fragkiadaki et al. [31] propose an Encoder-Recurrent-Decoder (ERD) module to predict heat maps of frames in an image sequence. More recently, Song et al. [32] introduce a spatio-temporal inference layer to perform message passing on general loopy spatio-temporal graphs.

In terms of action recognition problem, Molchanov et al. [33] employ RNN and deep 3D CNN on a whole video sequence for spatio-temporal feature extraction. Our approach is closely related to [34] who propose a two-stream architecture to encode the spatial and temporal information separately. Our approach is based on the depth image, which leads us to learn spatio-temporal feature via LSTM instead of optical flow.

### C. Regression Strategies

Regression strategy has the close relationship with the performance of methods, and it can be roughly classified into three camps: 1) 2D heat-map regression and 3D recovery; 2) directly 3D hand joint coordinates regression; 3) cascaded or hierarchical regression. In [4], [20], 2D heat-maps, which represent the 2D location for hand joints, are regressed based on the extracted feature, then the 3D positions are determined via recovery. Directly 3D hand joint coordinates regression predicts the hand locations in a single forward-pass, which exhibits superior performances comparing to the previous 2D heat-map regression [5]–[7], [21], [26], [35].

Cascaded and hierarchical regression has shown good performance in hand pose estimation. Oberweger et al. [18] use the iterative refinement stage to increase the accuracy, and in [19], they use the generated synthesized images to correct the initial estimation by using a feedback loop. Tang et al. [36], [37] introduce hierarchical regression strategies to predict tree-like topology of hand. Similarly, the strategy to first estimate hand palm and sequentially estimate the remaining joints is adopted in [8], [9]. Recently, Ye et al. [10] combine the attention model with both cascaded and hierarchical estimation, and propose an end-to-end learning framework.

## III. HAND POSE ESTIMATION

Figure 1 illustrates the main building blocks of our method. Given an image sequence, two predictions are estimated by *Temporal Network* and *Spatial Network*, then the predictions are fused by the output from *Fusion Network*.

### A. Overview

We denote a single depth image as  $I$ , and the corresponding  $K$  hand joints are represented as a vector  $\mathbf{J} = \{\mathbf{j}_k\}_{k=1}^K \in \mathbb{R}^{3K}$ , where  $\mathbf{j}_k$  is the 3D position of  $k$ -th joint in the depth image  $I$ .

Currently, most discriminative approaches concentrate on modeling a mapping function  $F$  from the input image to hand pose. We define a cost function as follows:

$$C(F, \mathbf{J}) = \frac{1}{N} \sum_{i=1}^N \|F(I_i) - \mathbf{J}_i\|_2 \quad (1)$$

where  $\|\cdot\|_2$  calculates the  $l_2$  norm,  $N$  is the number of images. Then the expected mapping function  $\hat{F}$  is expressed as  $\hat{F} = \operatorname{argmin}_{F \in \mathcal{F}} C(F, \mathbf{J})$ , where  $\mathcal{F}$  is the hypothesis space.

To obtain a robust mapping function, most approaches concentrate on the hand structure to capture features for estimation. In this paper, we capture the spatial and temporal information simultaneously to learn features for prediction. As shown in Figure 1, the architecture is separated into three parts. The first part is denoted as *Spatial Network*, which is applied on a single frame to learn a mapping function  $F_{spa}$ . Inspired by the texture based volume rendering used in the medical image [38], we slice the depth image into several layers, which makes the different joints scatter in the different layer. We feed the depth image and corresponding sliced 3D volumetric representation into the network, depth and spatial information are extracted hierarchically for hand pose estimation.

The second part is denoted as *Temporal Network*. This network concentrates on the time coherence property in image sequences, and learns the mapping  $F_{temp}$  from an image sequence to a set of hand poses. By using LSTM layer, this network takes the features from previous frames into account when estimating the pose in the current frame.

The third part is named as *Fusion Network*. For the sake of predicting pose via the combination of information from the individual and consecutive frames, we fuse the aforementioned two networks as an integration by adaptive fusion method, the final prediction is calculated as the summation of the different networks' outputs. By means of employing this approach, spatial information from the individual frame and temporal property between frames are thought to have in mind as a unified scheme.

### B. Proposed Method

In this section, we present details of our method for hand pose estimation.

#### Spatial Network

In an individual frame, the spatial information is critical for hand pose estimation. By adopting 3D sliced volumetric representation and Deeply-Fusion network [39], we extract features to learn a robust mapping function.

In the original depth image  $I$ , the value of the pixel encodes the depth information. We first crop the hand region out of the image, this step is same as [18]. As shown in Figure 2, the cropped depth image is resized to  $M \times M$ , and depth  $D(x, y)$  is represented as a function of  $(x, y)$  coordinates. Previous methods represent the 3D hand in different ways.

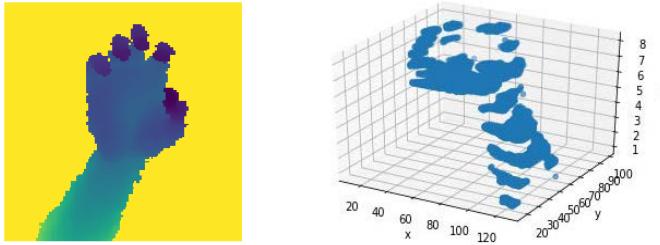


Fig. 2. Depth image and sliced 3D volumetric representation.

In [40], the depth image is recovered as a binary voxel grid for training. The hand surface and occluded space [41] are filled with 1, the free space is filled with 0. In [5], [6], TSDF and projective Directional TSDF (D-TSDF) are adopted to encode more information in the 3D volumetric representation. However, the size of 3D volume used in these methods is commonly set as  $32 \times 32$ . Different from previous methods, we employ a solution that the hand surface is sliced into  $L$  pieces. Then our sliced 3D volumetric representation is an  $M \times M \times L$  voxel grid, where  $L$  represents the number of sliced layers in  $z$ -axes. For an exact cropped depth image, the free space and occluded space are omitted, then we denote  $D_{min}$  and  $D_{max}$  as the depth of closest and farthest surface point, then we divide  $[D_{min}, D_{max}]$  equally into  $L$  pieces. So the binary sliced 3D volumetric representation  $V[x, y, l]$  is filled with 1 if the depth value  $D(x, y)$  is in the range of  $[D_{min} + \frac{D_{max}-D_{min}}{L}l, D_{min} + \frac{D_{max}-D_{min}}{L}(l+1)]$ , where  $l = 0, 1, \dots, L-1$ . In Figure 2, from the top layer to the bottom layer, the fingertips and palm center scatter in the different layers.

Sliced 3D volumetric representation keeps the structure of the hand. To obtain the hand structure and depth information simultaneously, we input depth image and sliced 3D volumetric representation at the same time. As shown in Figure 3, the left part extracts features from depth image, and the right part extracts features from sliced 3D volumetric representation. We denote the features from max-pooling layers as  $\phi_{depth}$  and  $\phi_{3D}$ , then the features are hierarchically fused by Deeply Fusion Network [39] as follows:

$$\begin{aligned} \phi_0 &= \phi_{depth} \oplus \phi_{3D} \\ \phi_m &= H_m^{depth}(\phi_{m-1}) \oplus H_m^{3D}(\phi_{m-1}) \\ m &= 1, 2. \end{aligned} \quad (2)$$

where  $\oplus$  is element-wise mean for features,  $H$  is the transformation for features learned by a fully connected layer,  $l$  is the index for layer.

Owing to the adoption of several fully connected layers, the network tends to overfitting, so we use an auxiliary loss as the regularization as well as dropout. In Figure 3, every fully connected layer is followed by a dropout layer, the nodes are dropout randomly with 30% probability. For the purpose of obtaining more representative capability for each input, the auxiliary paths are added in training stage. As shown in the green boxes, the layers in the auxiliary paths are same as the main network, the layers connected by the blue dotted line share parameters. And in the training stage, the total losses are

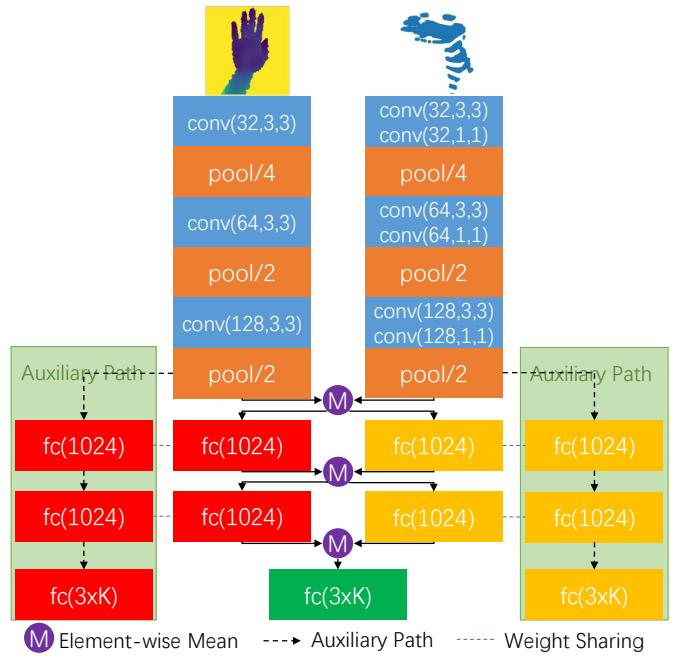


Fig. 3. *Spatial Network*. Depth image and sliced 3D volumetric representation are input into *Spatial Network*, the different features are fused together hierarchically. For the purpose of obtaining more representative capability for each input, the auxiliary paths are added in training stage, and the layers in the auxiliary paths are same as the main network. In the training stage, the total losses are consist of three losses, i.e. three regression loss in auxiliary paths and main network. When testing, the auxiliary pathes are removed. For convolution layer, the number in the parenthesis is the kernel size. For pooling layer, the number is stride. For fully connected layer, the number is the length of output feature.

consist of three losses, i.e. three regression losses in auxiliary paths and main network. When testing, the auxiliary paths are removed.

## Temporal Network

As we know, the hand poses between successive frames are closely related (e.g. when grabbing, the joints on fingers are most likely to be closer). The temporal property in the between frames are also critical for estimating hand joints. So we extend the mapping function from single frame to a sequence of images. The sequential prediction problem could be reformulated as follows:

$$(\mathbf{J}_1, \dots, \mathbf{J}_T) = F_{temp}(I_1, \dots, I_T) \quad (3)$$

where  $T$  is the number of images in a sequence.

Currently, due to the feedback loop structure in the recurrent neural network (RNN), it possesses powerful "long-term dependencies" modeling capability, and is widely used in computer vision tasks. In general, RNN is difficult to train due to the vanishing gradient and error blow-up problems [42]. In this paper, we adapt LSTM [43], [44], a variant of RNN which is expressive and easy to train, to model the temporal context of successive frames. As shown in Figure 4, an image sequence is taken as input and the network gives out the estimated poses. Without loss of generality, we think about the  $t$ -th depth image  $I_t$ , the image is fed into the convolution neural network and

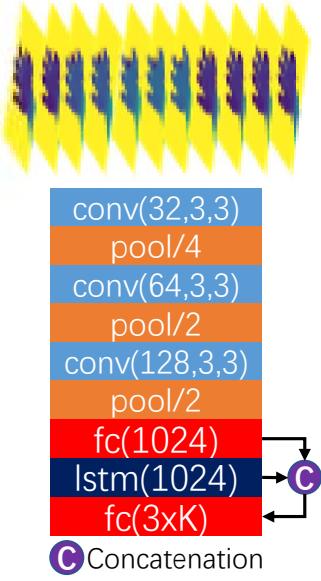


Fig. 4. *Temporal Network*. It learns a mapping function  $F_{temp}$  to process an image sequence, a set of images input and LSTM layer captures the temporal context in these images. For LSTM layer, the number in the parenthesis is the dimension of the hidden state feature.

the feature  $\phi_t$  extracted from the first fully connected layer is denoted as follows:

$$\phi_t = H_{\theta_c}(I_t). \quad (4)$$

Then we feed the features  $\phi_t$  into LSTM layer, and get the hidden state  $h_t$  as the new features.

$$\begin{aligned} i_t &:= \sigma(W_{hi} * h_{t-1} + W_{xi} * \phi_t + b_i) \\ f_t &:= \sigma(W_{hf} * h_{t-1} + W_{xf} * \phi_t + b_f) \\ o_t &:= \sigma(W_{ho} * h_{t-1} + W_{xo} * \phi_t + b_o) \\ c_t &:= (f_t \odot c_{t-1}) + (i_t \odot \tanh(W_{hc} * h_{t-1} + W_{xc} * \phi_t + b_c)) \\ h_t &:= o_t \odot \tanh(c_t) \end{aligned} \quad (5)$$

where  $\sigma(\cdot)$  is sigmoid function,  $\tanh(\cdot)$  is tanh function,  $\odot$  is element-wise product, and matrices  $W_{h*}, W_{x*}, b_*$  ( $*$  is in  $i, f, o, c$ ) are parameters for the gates in LSTM layer. The features  $h_t$  output from LSTM layer and the original features  $\phi_t$  are concatenated, and then are fed into the last fully connected layer to regress the hand pose coordinates.

## Fusion Network

The aforementioned *Temporal Network* and *Spatial Network* estimate the joints by placing emphasis on capturing spatial and temporal information, we name the predictions from these two networks as  $J_{temp}$  and  $J_{spa}$  respectively. Due to the importance of spatial and temporal information, we jointly model the spatio-temporal properties, which adaptively integrates different predictions for the final estimation. As shown in Figure 5, *Fusion Network* uses an activation function *sigmoid* after the last fully connected layer, then output  $w_1$  and  $w_2$ . *Fusion Network* fuses two predictors, and gives out the final prediction as follows:

$$J_{out} = w_1 \odot J_{temp} + w_2 \odot J_{spa} \quad (6)$$

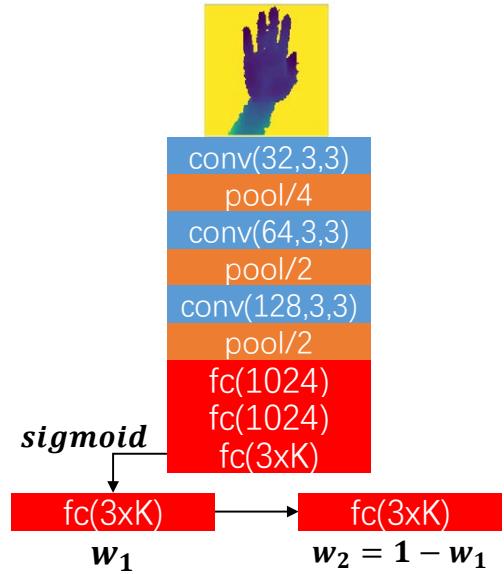


Fig. 5. *Fusion Network*, the network outputs weights for two predictions from *Spatial Network* and *Temporal Network*.  $w_1$  and  $w_2$  are learned as the confidence of two predictions.

where  $w_1 + w_2 = \mathbf{1}$ , and  $\mathbf{1} \in \mathbb{R}^{3K}$  is a vector that all elements are one. The weights  $w_1$  and  $w_2$  are learned as the confidence of two predictions, and the final prediction  $J_{out}$  is estimated as the weighted summation of each prediction.

Because *Temporal Network* considers the temporal information and *Spatial Network* extracts the spatial information, then the network infers the hand joint locations depend on spatial and temporal features.

Finally, we summarize our proposed method in Algorithm 1.

---

### Algorithm 1: Summary for Context-Aware Deep Spatio-Temporal Network

---

**Input:** Training data  $D = \{I_i, J_i\}_{i=1}^N$ , testing images  $D' = \{I_j\}_{j=1}^{N'}$ , number of LSTM cells in LSTM layer  $T$   
**Output:** The predicted hand poses  $\{J_f\}_{j=1}^{N'}$   
*/\* Training stage \*/*  
1 *Spatial Network* is trained with  $D$ , and learn the mapping function  $F_{spa}$  for an individual frame to minimize Equation 1;  
2 Train *Temporal Network* to learn the mapping function  $F_{temp}$  in Equation 3 for a set of images  $F_{temp} : \{I_1, \dots, I_T\} \rightarrow \{J_1, \dots, J_T\}$ ;  
3 Fix the parameters in the above two networks, then train the parameters in *Fusion Network* to learn parameters to output confidential weights  $w_1$  and  $w_2$  in Equation 6;  
*/\* Testing stage \*/*  
4 Forward  $T$  testing images into trained network, get the predictions  $J_{spa}$  and  $J_{temp}$  from *Spatial Network* and *Temporal Network*;  
5 Calculate the prediction via Equation 6.

---

## IV. EXPERIMENTS

In this section, we evaluate our method on two public datasets for comparison with the state-of-the-art methods. In addition, we do ablation experiments and analyse the performance of each component.

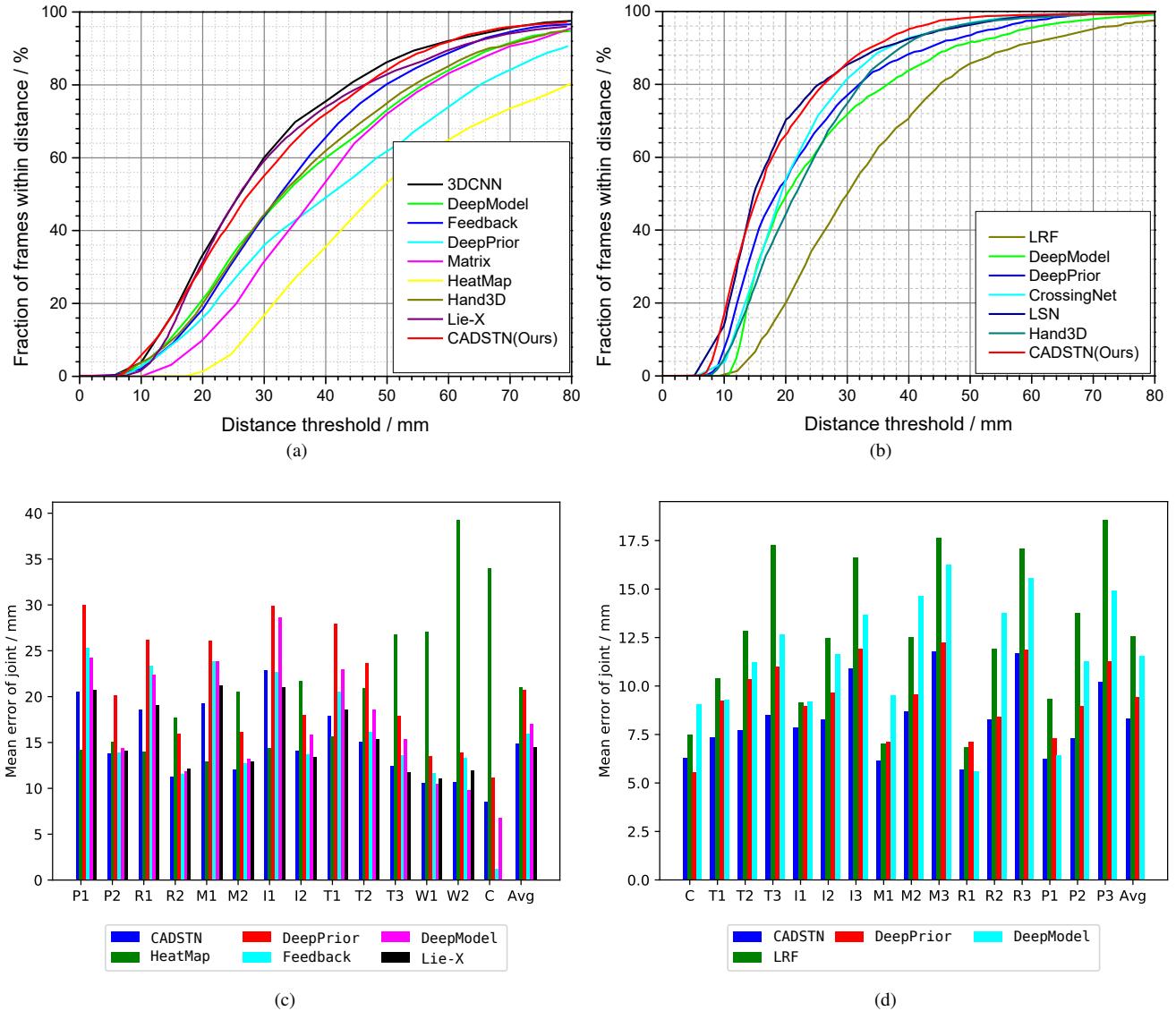


Fig. 6. Comparison with state-of-the-art methods on NYU dataset and ICVL dataset. Figure(a) and Figure(b): the proportion of good frames over different thresholds on NYU dataset and ICVL dataset. Figure(c) and Figure(d): per-joint error distance on NYU dataset and ICVL dataset. The palm and fingers are indexed as C: palm, T: thumb, I: index, M: middle, R: ring, P: pinky, W: wrist.

### A. Experiment Setting

1) **Datasets:** We evaluate our method on NYU [20] and ICVL [36], details about the datasets are summarized in Table I, where **Train** is the number of train images, **Test** is the number of test images and the number in bracket is the number of sequences; **Resolution** is the resolution of images; **Annotation** is the number of annotated joints. NYU dataset is challenging because of its wider pose variation and noisy image as well as limited annotation accuracy. On ICVL dataset, the scale of image is small and the discrepancies between training and testing is large, these make the estimation difficult.

2) **Evaluation Metrics:** The evaluation follows the standard metrics proposed in [18], including accuracy, pre-joint error distances and average error distance. As mentioned above, we denote  $j_{kn}$  as the predicted  $k$ -th joint location in the  $n$ -th frame,  $j_{kn}^*$  is the corresponding ground truth. The total number of

TABLE I  
DATASET FOR EXPERIMENTS.

Dataset	Train	Test	Resolution	Annotation
NYU	72k (1)	8k(2)	640 × 480	36
ICVL	22k (10)	1.6k(2)	320 × 240	16

frames is denoted as  $N$ , and  $K$  is the number of hand joints.

Per-joint error distance calculates the average Euclidean distance between the predicted joint location and the ground truth in 3D space.

$$err_k = \frac{\sum_{n=1}^N (\|j_{kn} - j_{kn}^*\|)}{N} \quad (7)$$

Average error distance computes the mean distance for all joints.

$$ave = \frac{\sum_{k=1}^K err_k}{K} \quad (8)$$

Accuracy is the fraction of frames that all predicted joints from the ground truth in a frame below the given distance threshold  $\delta$ .

$$acc_\delta = \frac{\mathbb{1}(\max_k(\|j_{kn} - \hat{j}_{kn}^*\|) \leq \delta)}{N} \quad (9)$$

where  $\mathbb{1}(\cdot)$  is the indicator function.

3) *Implementation Details*: We implement the training and testing with Caffe [45] framework. The pre-process step follows [18], then the cropped image is resized to  $128 \times 128$  with the depth value normalized to  $[-1, 1]$ , and for sliced 3D volumetric representation,  $L$  is set to 8 in the experiments. Moreover, we augment the dataset from the available dataset.

**Data augmentation** On NYU dataset, we do augmentation by random rotation and flipping. On one side, we randomly rotate the images by an angle selected from  $90^\circ - 90^\circ$  and  $180^\circ$ . On the other side, we flip the image vertically or horizontally. For each image in the dataset, we just generate one image by the above-mentioned augmentation method. So the size of the dataset is twice as many as the original. On ICVL dataset, in-plane rotation has been applied and there are 330k training samples in total, so the augmentation is not used on ICVL.

**Training configuration** Theoretically, training the three networks jointly is considerable, but it takes a longer training time to optimize the parameters of the three networks. Besides, we have tried to fine-tune the spatial and temporal network, but there is no much difference to freeze the Spatial Network and Temporal Network due to the limited size of data set. Practically, we strike a balance between training complexity and efficiency, so we adopt the simpler method as follows. We train *Temporal Network* and *Spatial Network* first, then train *Fusion Network* by fixing the above two networks. The training strategy is same for *Spatial Network* and *Temporal Network*, we optimize the parameters by using back-propagation and apply Adam [46] algorithm, the batch size is set as 128, the learning rate starts with 1e-3 and decays every 20k iterations. *Spatial Network* and *Temporal Network* are trained from scratch with 60k iterations. *Fusion Network* is trained by 40k iterations with the above two networks fixed. Our training takes place on machines equipped with a 12GB Titan X GPU.

## B. Experimental Results

1) *Comparison with State-of-the-art*: We compare our method with several state-of-the-art hand pose estimation methods [6], [7], [18]–[20], [26], [35], [36]. What worth mentioning is that some methods provide predicted joints but some not, we calculate the metrics of some methods [18]–[20], [35], [36] available online and estimate the others from the figures in their papers.

**NYU Dataset** contains 72k training images from one subject and 8k testing images from two subjects. There are totally 36 annotated joints, and we only evaluate a subset of 14 joints as used in other papers for a fair comparison. The annotation for hand is illustrated in Figure 7.

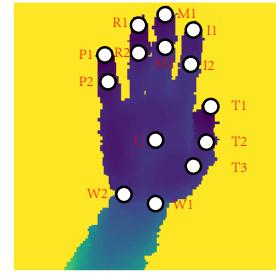


Fig. 7. Hand annotation used on NYU dataset.

We compare our method with several state-of-the-art approaches on NYU dataset, including 3DCNN [6], DeepModel [35], Feedback [19], DeepPrior [18], Matrix [7], HeatMap [20], Lie-X [47].

The accuracy and the pre-joint error distances are shown in Figure 6, our proposed method has a comparable performance with state-of-the-art methods. In Figure 6a, our method outperforms the majority of methods. For example, the proportion of good frames is about 10% higher than Feedback when distance threshold is 30mm. 3DCNN adopts the augmentation and projective D-TSDF method, and trains a 3D convolution neural network which maps the sliced 3D volumetric representation to 3D hand pose, our method could not as accurate as 3D CNN from 15-60mm due to the sufficient 3D spatial information captured by the 3D CNN. Lie-X infers the optimal hand pose in the manifold via a Lie group based method, it stands out over the range of 20-46mm, but our method overtakes at other threshold. In Figure 6c, the per-joint error distance for five methods and our method are illustrated.

Table II reports the average error distance for different methods. Generally, results show that our method has a comparable performance with competing methods. Our method outperforms [18]–[20], [35] by a large margin, and is comparing to [47].

To have an intuitive sense, we present some examples as depicted in Figure 8. Our method could get an acceptable prediction even in the extreme circumstance.

TABLE II  
MEAN ERROR DISTANCE OF DIFFERENT METHODS ON NYU DATASET.

Methods	Average joint error (mm)
HeatMap [20]	21.02
DeepPrior [18]	19.73
Feedback [19]	15.97
DeepModel [35]	16.90
Lie-X [47]	<b>14.51</b>
CADSTN	14.83

**ICVL Dataset** contains 22k training images (totally 330k by augmentation), and is separated into ten sequences. And testing dataset includes two sequences, there are 1.6k images in total.

On ICVL dataset, we compare our proposed method against five approaches: CrossingNet [26], LRF [36], DeepPrior [18], DeepModel [35] and LSN [9]. The quantitative results are shown in Figure 6b. Our method is better than LRF, Deep-

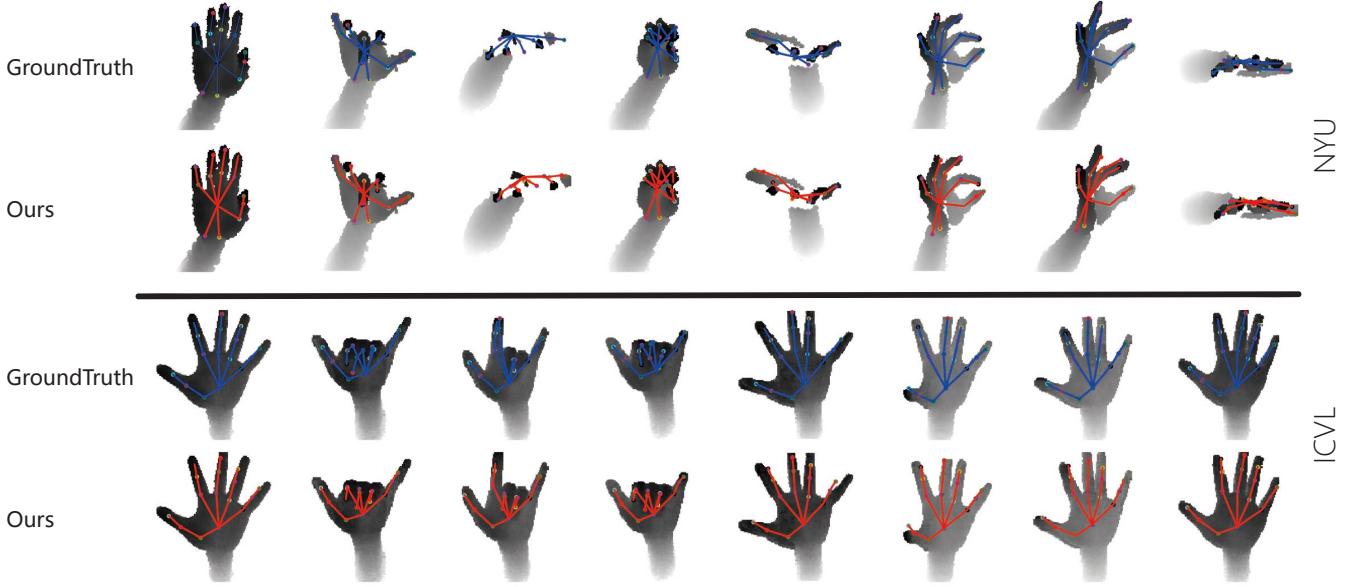


Fig. 8. Qualitative results for NYU and ICVL dataset. Best viewed in color.

Prior<sup>2</sup>, DeepModel and CrossingNet, and is roughly same as LSN. CrossingNet employs GAN and VAE for data augmentation, and our method surpasses it when the threshold is bigger than 9mm. Compared to the hierarchical regression framework LSN, our method is not as accurate from 14-30mm but outperforms when the threshold is bigger than 30mm. Furthermore, Figure 6d reveals the same case that fingertips estimation is generally worse than the palm of the hand. As summarized in Table III, the mean error distance of our method is the lowest in four methods, which obtains a 1.4mm error decrease than DeepPrior and is 0.16mm smaller than LSN.

TABLE III  
MEAN ERROR DISTANCE OF DIFFERENT METHODS ON ICVL DATASET.

Methods	Average joint error (mm)
LRF [36]	12.56
DeepPrior [18]	9.42
DeepModel [35]	11.55
CrossingNet [26]	10.22
LSN [9]	8.20
<b>CADSTN</b>	<b>8.04</b>

2) *Ablation Study*: For the sake of analyzing the effects of different parts in our method, we perform an extensive ablation experiments on NYU dataset.

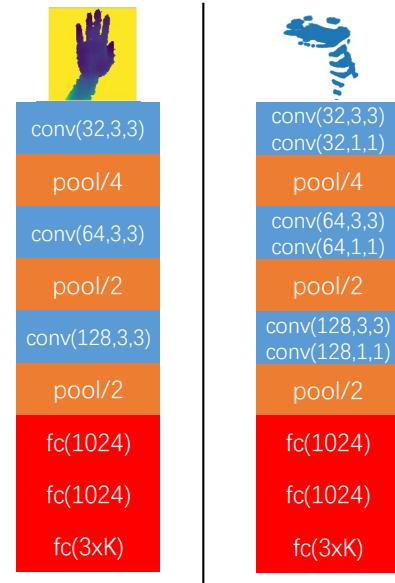
**Ablation Study For Feature Representation** Except for *Spatial Network* and *Temporal Network*, we train two networks named *Baseline Regression Network* and *Sliced 3D Input Network*. As shown in Figure 9, these two networks are parts of *Spatial Network*.

We show the results in Figure 10 and Table IV. The experimental results reveal that *Spatial Network* and *Temporal Network* improve the performance by exploiting the spatial and temporal context respectively, and our final model performs best by utilizing spatio-temporal context simultaneously.

<sup>2</sup>DeepPrior [18] only provides test results on the first sequence on ICVL, the accuracy curve is plotted based on the provided result.

TABLE IV  
ABLATION STUDY ON NYU DATASET.

Methods	Average joint error (mm)
Sliced 3D Input	16.56
Baseline Regression	15.95
Temporal	15.47
Spatial	15.03
<b>CADSTN</b>	<b>14.83</b>

Fig. 9. *Baseline Regression Network* and *Sliced 3D Input Network*, these two networks are parts of *Spatial Network*.

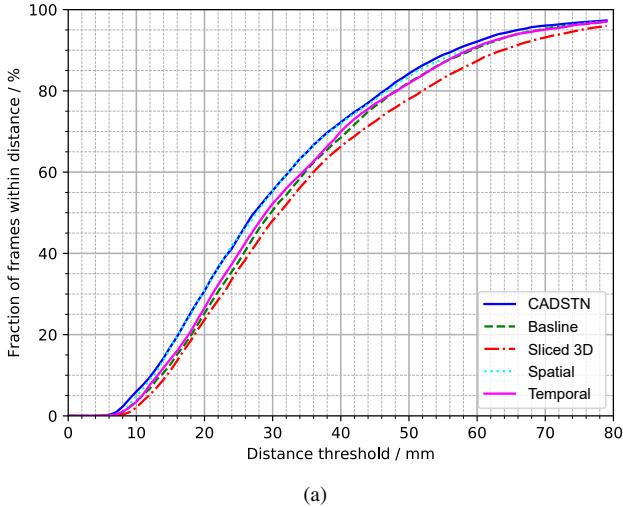
**Baseline Regression Network** is a simple network composed of convolution, max-pooling, and fully connected layers. The network regresses the labels directly, and the average error

TABLE V  
SIMPLE COMBINATION FOR THE RESULTS FROM SPATIAL NETWORK AND TEMPORAL NETWORK ON NYU DATASET.

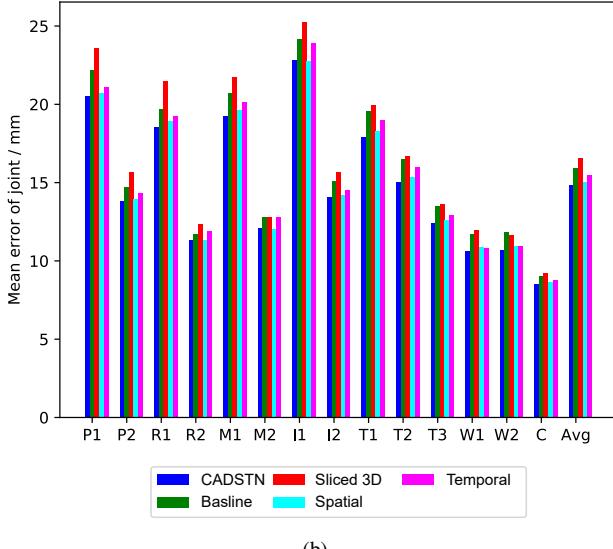
w	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	ours
Average joint error (mm)	14.98	14.94	14.89	14.88	14.89	14.90	14.92	15.06	15.24	<b>14.83</b>

distance is shown in Table IV. The results reveal that *Baseline Regression Network* already surpasses some methods.

**Sliced 3D Input Network** regresses the hand joints with the sliced 3D volumetric representation. Due to the absence of depth information, it is difficult to estimate the depth for hand joint.



(a)



(b)

Fig. 10. Performance of different part in our proposed network. Figure(a): proportion of good frames over different threshold for different part in our proposed network (**Sliced 3D Input** is the sliced 3D volumetric representation input branch in *Spatial Network* Figure(b): per-joint error distance for different parts. The palm and fingers are indexed as C: palm, T: thumb, I: index, M: middle, R: ring, P: pinky, W: wrist.

**Spatial Network** hierarchically fuses the features from depth image input and sliced 3D volumetric representation. We could find that **Sliced 3D** is the worst in several competitors. In our opinion, the bad performance is because the depth in-

formation is sacrificed in sliced 3D volumetric representation. Via the Deeply-Fusion network, *Spatial Network* borrow the spatial information from **Sliced 3D** branch, and it achieves a better result than *Baseline Regression Network* and *Sliced 3D Input Network*.

**Temporal Network** replaces the second fully connected layer with LSTM layer and concatenates the hidden output with the input features. We find that *Temporal Network* slightly improves the performance due to the temporal information from experimental results, and the  $T$  value for LSTM is set as 16 in training.

**CADSTN** is our proposed method, which integrates *Spatial Network* and *Temporal Network* by *Fusion Network*. *Fusion Network* fuses the predictions from two networks and yields the final prediction. The three networks are connected with the implicitly adaptation weights, and influence each other in the optimization process. Because of the temporal coherence and spatial information concerned, our method performs best in ablation experiments, and Figure 10b reveals that the joint error distance for every joint is the lowest. Table IV reports that our final method decreases the average joint error by 1.12mm.

**Ablation Study For Regression** To demonstrate the advantage of our proposed Fusion Network, we compare Fusion Network with simple combination for the results from Spatial Network and Temporal Network on NYU dataset. And the experimental results are shown in Table V. The experimental result demonstrates our proposed Fusion Network performs better than simply combination method.

## V. CONCLUSION

In this paper, we propose a novel method for 3D hand pose estimation, termed **CADSTN**, which models the spatio-temporal context with three networks. The modeling of spatial context, temporal property and fusion are separately done by three parts. The proposed *Spatial Network* extracts depth and spatial information hierarchically. Further on making use of the temporal coherence between frames, *Temporal Network* gives out a sequence of joints by feeding into a depth image sequence. Then we fuse the predictions from the above two networks via *Fusion Network*. We evaluate our method on two publicly available benchmarks and the experimental results demonstrate that our method achieves the best or the second-best result with state-of-the-art approaches and can run in real-time on two datasets.

## ACKNOWLEDGMENT

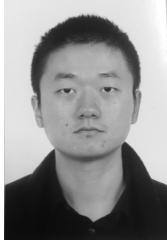
This work was supported in part by NSFC under Grant U1509206 and Grant 61472353, in part by the Key Program of Zhejiang Province under Grant 2015C01027, in part by the National Basic Research Program of China under Grant 2015CB352302, and in part by the Alibaba-Zhejiang University Joint Institute of Frontier Technologies.

## REFERENCES

- [1] Y. Sato, M. Saito, and H. Koike, "Real-time input of 3d pose and gestures of a user's hand and its applications for hci," in *Virtual Reality, 2001. Proceedings. IEEE*. IEEE, 2001, pp. 79–86. I
- [2] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Computer Vision and Image Understanding*, vol. 108, no. 1, pp. 52–73, 2007. I
- [3] E. Barsoum, "Articulated hand pose estimation review," *arXiv preprint arXiv:1604.06195*, 2016. I
- [4] 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 *CVPR*, 2016. I, II-A, II-C
- [5] X. Deng, S. Yang, Y. Zhang, P. Tan, L. Chang, and H. Wang, "Hand3d: Hand pose estimation using 3d neural network," *arXiv preprint arXiv:1704.02224*, 2017. I, II-A, II-C, III-B
- [6] 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 *CVPR*, 2017. I, II-A, II-C, III-B, IV-B1, IV-B1
- [7] A. Sinha, C. Choi, and K. Ramani, "DeepHand: Robust hand pose estimation by completing a matrix imputed with deep features," in *CVPR*, 2016. I, II-C, IV-B1, IV-B1
- [8] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," in *CVPR*, 2015. I, II-C
- [9] C. Wan, A. Yao, and L. Van Gool, "Direction matters: hand pose estimation from local surface normals," in *ECCV*, 2016. I, II-C, IV-B1, III
- [10] Q. Ye, S. Yuan, and T.-K. Kim, "Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation," in *ECCV*, 2016, pp. 346–361. I, II-C
- [11] P. Li, H. Ling, X. Li, and C. Liao, "3d hand pose estimation using randomized decision forest with segmentation index points," in *ICCV*, 2015. I
- [12] Y. Zhang, C. Xu, and L. Cheng, "Learning to search on manifolds for 3d pose estimation of articulated objects," *arXiv preprint arXiv:1612.00596*, 2016. I
- [13] R. Y. Wang and J. Popović, "Real-time hand-tracking with a color glove," in *ACM transactions on graphics (TOG)*, vol. 28, no. 3. ACM, 2009, p. 63. I
- [14] J. Romero, H. Kjellström, and D. Kragic, "Monocular real-time 3d articulated hand pose estimation," in *Humanoid Robots*. IEEE, 2009, pp. 87–92. I
- [15] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single rgb images," in *ICCV*, Oct 2017. I
- [16] P. Panteleris and A. Argyros, "Back to rgb: 3d tracking of hands and hand-object interactions based on short-baseline stereo," in *ICCVW*, 2017. I
- [17] M. Oberweger and V. Lepetit, "DeepPrior++: Improving fast and accurate 3d hand pose estimation," in *ICCV workshop*, vol. 840, 2017, p. 2. I
- [18] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands deep in deep learning for hand pose estimation," in *CVWW*, 2015. I, II-A, II-C, III-B, IV-A2, IV-A3, IV-B1, IV-B1, II, IV-B1, III, 2
- [19] Oberweger, Markus and Wohlhart, Paul and Lepetit, Vincent, "Training a feedback loop for hand pose estimation," in *ICCV*, 2015. I, II-A, II-C, IV-B1, IV-B1, II
- [20] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Transactions on Graphics (ToG)*, vol. 33, no. 5, 2014. I, II-C, IV-A1, IV-B1, IV-B1, II
- [21] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, and H. Yang, "Region ensemble network: Improving convolutional network for hand pose estimation," *ICIP*, 2017. II-A, II-C
- [22] G. Wang, X. Chen, H. Guo, and C. Zhang, "Region ensemble network: Towards good practices for deep 3d hand pose estimation," *Journal of Visual Communication and Image Representation*, 2018. II-A
- [23] X. Chen, G. Wang, H. Guo, and C. Zhang, "Pose guided structured region ensemble network for cascaded hand pose estimation," *arXiv preprint arXiv:1708.03416*, 2017. II-A
- [24] G. Moon, J. Chang, and K. M. Lee, "V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. II-A
- [25] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *ISMAR*. IEEE, 2011, pp. 127–136. II-A
- [26] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation," in *CVPR*, 2017. II-A, II-C, IV-B1, IV-B1, III
- [27] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014. II-A
- [28] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014. II-A
- [29] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt, "Real-time hand tracking under occlusion from an ego-centric rgb-d sensor," in *Proceedings of International Conference on Computer Vision (ICCV)*, vol. 10, 2017. II-B
- [30] M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit, "Efficiently creating 3d training data for fine hand pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4957–4965. II-B
- [31] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *ICCV*, 2015, pp. 4346–4354. II-B
- [32] J. Song, L. Wang, L. Van Gool, and O. Hilliges, "Thin-slicing network: A deep structured model for pose estimation in videos," in *CVPR*, July 2017. II-B
- [33] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network," in *CVPR*, 2016, pp. 4207–4215. II-B
- [34] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, 2014, pp. 568–576. II-B
- [35] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei, "Model-based deep hand pose estimation," *IJCAI*, 2016. II-C, IV-B1, IV-B1, II, IV-B1, III
- [36] D. Tang, H. Jin Chang, A. Tejani, and T.-K. Kim, "Latent regression forest: Structured estimation of 3d articulated hand posture," in *CVPR*, 2014. II-C, IV-A1, IV-B1, IV-B1, III
- [37] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton, "Opening the black box: Hierarchical sampling optimization for estimating human hand pose," in *ICCV*, 2015. II-C
- [38] M. Hopf and T. Ertl, "Accelerating 3d convolution using graphics hardware," in *Visualization*, 1999. III-A
- [39] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *CVPR*, 2017. III-B, III-B
- [40] J. S. Supancic, III, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan, "Depth-based hand pose estimation: Data, methods, and challenges," in *ICCV*, 2015. III-B
- [41] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *CVPR*, 2015. III-B
- [42] K. Kawakami, "Supervised sequence labelling with recurrent neural networks," Ph.D. dissertation, PhD thesis. Ph. D. thesis, Technical University of Munich, 2008. III-B
- [43] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. III-B
- [44] W. Zaremba and I. Sutskever, "Learning to execute," *arXiv preprint arXiv:1410.4615*, 2014. III-B
- [45] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014. IV-A3
- [46] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015. IV-A3
- [47] C. Xu, L. N. Govindarajan, Y. Zhang, and L. Cheng, "Lie-x: Depth image based articulated object pose estimation, tracking, and action recognition on lie groups," *International Journal of Computer Vision*, 2017. IV-B1, II



**Yiming Wu** is now a PhD student in College of Computer Science and Technology at ZheJiang University, Hang Zhou, China. His mentor is Professor Li Xi. Prior to that, he received a bachelor's degree in engineering from Beijing Jiaotong University. His current research direction is computer vision and machine learning.



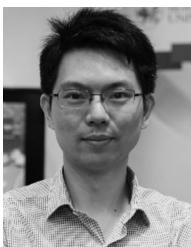
**Wei Ji** is currently a third-year PhD student in College of Computer Science at Zhejiang University, Hangzhou, China. His advisors are Prof. Xi Li and Prof. Yueling Zhuang. Earlier, he received his bachelor's degree in Computer Science and Technology from Nanjing University of Science and Technology in 2015. His current research interests are primarily in computer vision and machine learning, object recognition and detection.



**Fei Wu** received the B.S. degree from Lanzhou University, Lanzhou, Gansu, China, the M.S. degree from Macao University, Taipa, Macau, and the Ph.D. degree from Zhejiang University, Hangzhou, China. He is currently a Full Professor with the College of Computer Science and Technology, Zhejiang University. He was a Visiting Scholar with Prof. B. Yu's Group, University of California, Berkeley, from 2009 to 2010. His current research interests include multimedia retrieval, sparse representation, and machine learning.



**Xi Li** is currently a full professor at the Zhejiang University, China. Prior to that, he was a senior researcher at the University of Adelaide, Australia. From 2009 to 2010, he worked as a postdoctoral researcher at CNRS Telecom ParisTech, France. In 2009, he got the doctoral degree from National Laboratory of Pattern Recognition, Chinese Academy of Sciences, Beijing, China. His research interests include visual tracking, motion analysis, face recognition, web data mining, image and video retrieval.



**Gang Wang** received the BEng degree in electrical engineering from the Harbin Institute of Technology and the PhD degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign. He is an associate professor in the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore. He had a joint appointment at the Advanced Digital Science Center (operated by UIUC) as a research scientist from 2010 to 2014. His research interests include deep learning, scene parsing, object recognition, and action analysis. He is selected as a MIT Technology Review innovator under 35 for Southeast Asia, Australia, New Zealand, and Taiwan. He is also a recipient of Harriett & Robert Perry Fellowship, CSAI award, best paper awards from PREMIA (Pattern Recognition and Machine Intelligence Association) and top 10 percent paper awards from MMSP. He is an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence and an area chair of ICCV 2017. He is a member of the IEEE.



**Jianwei Yin** received the PhD degree in computer science from Zhejiang University, in 2001. He is currently a professor in the College of Computer Science, Zhejiang University, China. He was a visiting scholar at the Georgia Institute of Technology, Georgia, in 2008. His research interests include service computing, cloud computing, and information integration. Currently, he is the associate editor of the IEEE Transactions on Service Computing. He is a member of the IEEE.