Robust 3D Hand Pose Estimation From Single Depth Images Using Multi-View CNNs

Liuhao Ge, Hui Liang, *Member, IEEE*, Junsong Yuan, *Senior Member, IEEE*, and Daniel Thalmann

***Abstract*— Articulated hand pose estimation is one of core technologies in human–computer interaction. Despite the recent progress, most existing methods still cannot achieve satisfactory performance, partly due to the difficulty of the embedded high-dimensional nonlinear regression problem. Most existing data-driven methods directly regress 3D hand pose from 2D depth image, which cannot fully utilize the depth information. In this paper, we propose a novel multi-view convolutional neural network (CNN)-based approach for 3D hand pose estimation. To better exploit 3D information in the depth image, we project the point cloud generated from the query depth image onto multiple views of two projection settings and integrate them for more robust estimation. Multi-view CNNs are trained to learn the mapping from projected images to heat-maps, which reflect probability distributions of joints on each view. These multi-view heat-maps are then fused to estimate the optimal 3D hand pose with learned pose priors, and the unreliable information in multi-view heat-maps is suppressed using a view selection method. Experimental results show that the proposed method is superior to the state-of-the-art methods on two challenging data sets. Furthermore, a cross-data set experiment also validates that our proposed approach has good generalization ability.**

***Index Terms*— 3D hand pose estimation, convolutional neural networks, multi-view CNNs.**

I. Introduction

ARTICULATED hand pose estimation is one of core technologies in vision-based human-computer interaction, especially in virtual, augmented and mixed reality applications. With the emergence of commercial depth cameras in recent years, many research works have focused on 3D hand pose estimation from depth images [2]–[7]. However, it is still challenging to estimate 3D hand pose from depth images robustly and accurately in real-time, because of large hand pose variations, severe self-occlusions and high-dimensional motion of hand joints.

Methods for 3D hand pose estimation can be divided into data-driven and model-driven methods. Different with model-driven methods, which require complex model calibration and are sensitive to initialization, data-driven methods map input depth images to 3D hand joint locations using discriminative models, such as the isometric self-organizing map (ISO-SOM) model [8], random forests [3], [9]–[14], the joint matrix factorization and completion (JMFC) model [15] and convolutional neural networks (CNNs) [4], [5], [16]–[21], which are trained on large annotated hand pose datasets [4], [10], [12].

We concentrate on the CNN-based approach in this paper. One of primary issues of the CNN-based approach for accurate 3D pose estimation is how to effectively utilize the depth image. If the neural network directly maps the 2D depth image to 3D joint locations, it will suffer from high learning complexity and low generalization ability, since the mapping is highly nonlinear [22]–[24]. To tackle this issue, Tompson *et al.* [4] proposed mapping input depth image to a set of heat-maps representing the 2D probability distributions of hand joints on the image plane and recovering 3D locations using the depth information with model fitting. Nonetheless, this approach cannot effectively exploit the 3D information in the depth image, since the estimated heat-maps only provide 2D information of hand joints projected on the image plane.

In this paper, we propose a novel multi-view CNN-based 3D hand pose estimation method that can effectively utilize depth information to accurately infer 3D hand joint locations without model fitting, as depicted in Fig. 1. In detail, human hand is first segmented from the input depth image; the point cloud generated from the hand depth image is projected onto multiple projection planes; the projected image is then fed into its corresponding network to regress a set of heat-maps encoding the 2D probability distributions of hand joints on the corresponding projection plane. The 3D probability distribution of hand joint can be inferred from the combination of heat-maps on multiple views. By formulating multi-view fusion as the maximum a posteriori estimation with pre-learned hand pose constraints, we are able to obtain the optimal 3D hand pose and mitigate the ambiguity of estimations.

Compared with the single view CNN-based method [4], our proposed multi-view CNN-based method has the following advantages:

- The single view CNN-based method [4] directly takes the depth value at the inferred 2D hand joint position as the

Manuscript received November 18, 2017; revised March 21, 2018; accepted April 25, 2018. Date of publication May 10, 2018; date of current version June 1, 2018. This paper was presented at the IEEE Conference CVPR, 2016 [1]. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xudong Jiang. *(Corresponding author: Junsong Yuan.)*

L. Ge is with the Institute for Media Innovation, Nanyang Technological University, Singapore 639798 (e-mail: ge0001ao@e.ntu.edu.sg).

H. Liang is with Amazon, Seattle, WA 98121 USA (e-mail: hulia@amazon.com).

J. Yuan is with the Department of Computer Science and Engineering, The State University of New York at Buffalo, Buffalo, NY 14260 USA (e-mail: jsyuan@buffalo.edu).

D. Thalmann is with the School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland (e-mail: daniel.thalmann@epfl.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2834824

1057-7149 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

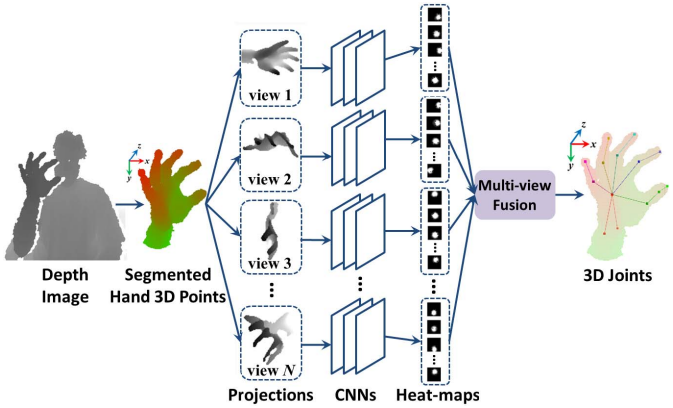


Fig. 1. Overview of our proposed multi-view CNN-based approach for 3D hand pose estimation. We generate heat-maps for multiple views by projecting 3D points onto a number of projection planes. Multi-view CNNs are trained in parallel to map projected images on multiple views to their corresponding heat-maps, which are then fused together to estimate 3D hand joint locations.

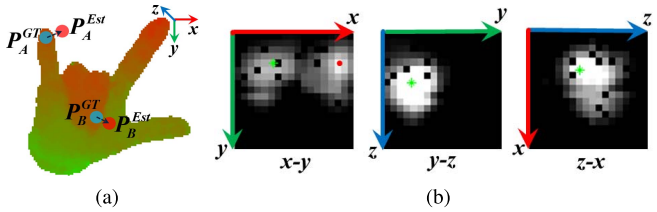


Fig. 2. Two problems of single view CNN-based method. (a) A case of hand pose estimation in single view. Blue points are true locations, and red points are estimated locations. Small 2D estimation error on the image plane may cause large depth estimation error. The tip of little finger is misestimated on the background and the tip of middle finger is misestimated on the palm. (b) A case of ambiguous estimation. We project the depth image onto *x*-*y*, *y*-*z* and *z*-*x* planes of a Cartesian coordinate system. Green dot indicates the true joint location in 2D projection plane, while red dot indicates the wrong joint location. Although the heat-map on *x*-*y* view contains two hotspots that are difficult to determine, from the heat-map on *z*-*x* view, it is obvious that the *x* value should be small with high confidence. Therefore, the left hotspot in the heat-map on *x*-*y* view should be true.

hand joint depth. As presented in Fig. 2a, the error of depth estimation may be large, even if the inferred joint position just deviates a little from the true joint position on the image plane. On the contrary, our proposed method, generating and fusing heat-maps on multiple views, can estimate 3D locations more robustly.

- In the case of ambiguous estimation, as presented in the *x*-*y* view of Fig. 2b, the single view CNN cannot determine the true joint position among multiple hotspots in the heat-map. When adopting our proposed multi-view CNNs, the ambiguity can be eliminated by using heat-maps on other views, such as the case in Fig. 2b.
- Different with the single view CNN-based method [4] that requires a pre-defined hand model to optimize estimated hand joint locations, our proposed method implicitly imposes hand pose constraints learned from training samples in the optimization problem, instead of manually defining hand model parameters.

This paper is an extension of our conference paper [1]. The new contributions of this paper are summarized as follows:

- We have proposed a new multi-view representation which projects the 3D point cloud onto six views of the oriented bounding box (OBB) and the axis-aligned bounding box (AABB) to better leverage 3D information in the depth image. However, in our conference paper [1], we only projected the 3D point cloud onto OBB's three views. Experimental results have shown that the combination can further boost the estimation performance.
- We have designed two new networks by applying architectures of the residual network (ResNet) and the fully convolutional network (FCN) for estimating more accurate heat-maps. Experimental results have shown that our proposed networks can achieved better performance than the network adopted in our conference paper [1].
- We have proposed a 3D data augmentation method for training CNNs of AABB's three views which are not rotation invariant, to make the multi-view CNNs more robust to various global hand orientations.
- We have proposed a view selection method to suppress unreliable information in heat-maps for multi-view fusion. In addition, the multi-view fusion approach proposed in this paper is more general since it does not restrict the number of projection views. But the fusion method in our conference paper [1] is only applicable to three orthogonal projection views.
- We have conducted more extensive self-comparison experiments and have compared with more existing methods on both MSRA dataset [10] and one additional dataset (NYU dataset [4]). We have also evaluated our method in real scenarios using the SoftKinetic's DepthSense camera.

The rest of this paper is organized as follows. Section II reviews related work on hand pose estimation, especially the CNN-based methods. Section III introduces the methods of multi-view representation and multi-view CNNs learning. The multi-view fusion algorithm is presented in Section IV. Section V presents extensive experimental results and Section VI concludes this paper.

II. Related Work

A. Hand Pose Estimation

3D hand pose estimation has been extensively studied over many years. The most common 3D hand pose estimation techniques can be classified into model-driven approaches and data-driven approaches [25]. Model-driven methods usually find the optimal hand pose parameters via fitting a deformable 3D hand model to input image observations. Such methods have demonstrated to be quite effective, especially with the depth cameras [26]–[33]. However, there are some shortcomings for model-driven methods. For example, they usually need to explicitly define the anatomical size and motion constraints of the hand for different subjects. In addition, due to the high dimensional parameter space of the hand pose, they can be sensitive to initialization for the iterative model-fitting procedure which will converge to a local optimal pose.

In contrast, the data-driven methods do not require the explicit specification of the hand model and motion

constraints. On the contrary, such information is implicitly encoded in the trained discriminative model. Therefore, many recent methods are built upon such a scheme [3]–[5], [9]–[19], [34], [35]. Among them, the random forest and its variants have proved to be reasonably accurate and fast. In [3], the authors propose to use the random forest to directly estimate hand joint angles from depth images.

A similar method is presented in [11], which further adopts transfer learning to make up for the inconsistence between synthetic and real-world data. As the estimation from random forest can be ambiguous for complex hand poses, pre-learned hand pose priors are sometimes utilized to better fuse independently predicted hand joint distributions [36], [37]. In [10], the cascaded pose regression algorithm [38] is adapted to the problem of hand pose estimation by regressing hand joints hierarchically, in order to preserve the hand pose constraints during the process of pose regression.

Some other approaches combine the model-based fitting and the data-driven methods in order to take advantages of both methods [39]–[41]. In [39], a multi-layered discriminative model, which is trained on synthetic training data, is first applied to infer candidate hand poses. Then, the optimal hand pose is estimated by a model-fitting stage based on particle swarm optimization. In [40] and [41], a random decision forest is trained to detect hand parts. Then, a model-based generative tracking method is applied to estimate hand pose by combining hand part labels and Gaussian mixture representation of depth data into an objective function.

B. CNN-Based Articulated Pose Estimation

Recently, convolutional neural networks have shown to be powerful in articulated pose estimation. In [22], CNNs are tuned to regress for the 2D human poses by directly minimizing the pose estimation error on the training data. Results have shown to outperform traditional methods largely. However, it takes more than twenty days to train the network and the dataset only contains several thousand images. Considering the relatively small size of the dataset used in [22], it may be difficult to use this method on larger datasets such as [4], [10], and [12], which consist of more than 70K images. It is reported in [23] and [42] that such direct mapping with CNNs from image features to continuous 2D/3D locations is of high nonlinearity and complexity as well as low generalization ability, which renders it difficult to train CNNs in such a manner.

An alternative way of CNN-based articulated pose estimation is to predict the heat-maps of joint positions instead of the articulated pose parameters. The intensity of a pixel on the heat-map indicates the likelihood for a joint occurring there. The network is trained to minimize the difference between the estimated heat-maps and the ground truth heat-maps. In this way, the network can be trained efficiently and this method has achieved state-of-the-art performance in body pose estimation [23], [43]. Similarly, such a framework has also been applied in 3D hand pose estimation [4]. However, the heat-map only provides 2D information of the hand joint and the depth information is not fully utilized. To address this issue, a model-based verification stage is adopted to estimate the 3D hand pose based on the estimated heat-maps and the input depth image [4]. Such heat-map based approaches are promising as heat-maps can reflect the probability distribution of hand joints on the projection plane.

TABLE I
NOTATIONS

K	the number of hand joints
$\boldsymbol{\phi}_k$	the k-th hand joint location
N	the number of projection planes
I_n	the projected image on the n-th view
T	the number of training samples
X_t	the depth image of the t-th training sample
$I_{t,n}$	the projected image n-th view of the t-th training sample
$G_{k,n}$	the ground truth heat-map for the k-th joint on the n-th view
$\boldsymbol{w}_n$	the network weights for the n-th view
$\boldsymbol{\phi}_{k,n}$	the projection of $\boldsymbol{\phi}_k$ on the n-th view
$d_{k,n}$	the signed distance from $\boldsymbol{\phi}_k$ to the n-th projection plane
$\boldsymbol{\mu}_k$	the weighted mean vector for 3D Guassian of the k-th joint
$\boldsymbol{\Sigma}_k$	the weighted covariance matrix for 3D Guassian of the k-th joint
M	the number of principal components used in PCA on training data
$\boldsymbol{e}_m$	the m-th principal components for training data
$\boldsymbol{e}_{m,k}$	the k-th 3D sub-vector of $\boldsymbol{e}_m$ corresponding to the k-th joint
α_m	the coefficient of the principal component $\boldsymbol{e}_m$
$\boldsymbol{u}_k$	the 3D empirical mean vector of the k-th joint for training data

Inspired by the above methods, we propose to generate heat-maps on multiple views and fuse them together to estimate the probability distribution of hand joints in 3D space. Multi-view CNNs have shown superior performance in 3D object recognition and retrieval [44], [45], as well as human action recognition [46]. Different from existing methods using multi-view CNNs to extract compact descriptors of 3D shapes for classification tasks [44]–[46], our proposed method trains multi-view CNNs to generate multi-view heat-maps and estimates 3D hand joint locations through a fusion stage from single depth images in real-time, which is a regression task in 3D space.

III. Multi-View Representation and Learning

Our method estimates 3D hand joint locations from the single depth image. Specifically, the input of this task is a depth image containing a human hand and the outputs are K 3D hand joint locations which represent the hand pose. Let the K objective hand joint locations be $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_k\}_{k=1}^K \in \boldsymbol{\Lambda}$, here $\boldsymbol{\Lambda}$ is the $3 \times K$ dimensional hand joint space. We summarize the notations of important variables in Tab. I.

In this section, we describe our proposed method of multi-view representation and multi-view learning. We first generate the multi-view hand representation by projecting 3D hand points onto multiple projection planes. For each view, we train a CNN model which maps the projected image to a set of heat-maps representing probability distributions of hand joint locations on the projected image.

A. Multi-View Representation

The objective for multi-view representation is to generate a set of projected images $\{I_n\}_{n=1}^N$ on multiple views from the depth image of the segmented hand. As illustrated in Fig. 1, the segmented hand in the depth image I_D is first converted

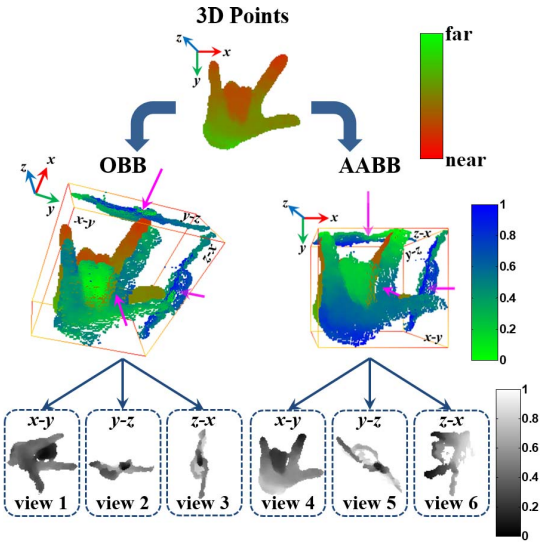


Fig. 3. Illustration of multi-view representation. 3D hand points obtained from the segmented hand depth image are projected onto *x-y*, *y-z* and *z-x* planes of the OBB projection coordinate system and the AABB projection coordinate system, respectively. Color bars of 3D points and projected images are shown in this figure.

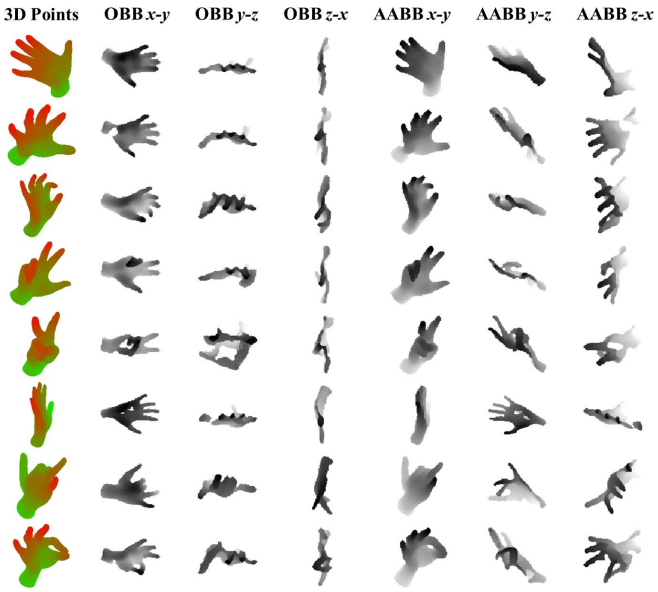


Fig. 4. Examples of projected images on six views which are *x-y*, *y-z*, *z-x* views in the OBB projection coordinate system and *x-y*, *y-z*, *z-x* views in the AABB projection coordinate system. The 3D points visualization in this figure is the same as the corresponding depth image.

to a set of 3D points in the camera coordinate system by using the depth camera's intrinsic parameters. To generate the multi-view representation, we project these 3D points onto N projection planes. In order to fully utilize the depth information in the 2.5D depth image, the projected images of one frame should provide complementary information as much as possible. Thus, orthogonal planes, on which the projected images can reflect different features of hand shapes from independent views, are good candidates for projection planes.

1) Projection With Oriented Bounding Box: We first project 3D points onto three orthogonal side planes of the oriented bounding box (OBB), as shown in Fig. 3 (left). The OBB is generated by performing principal component analysis (PCA) on the set of 3D points, which is a tight fit around these 3D points in local space [47]. The origin of the OBB projection coordinate system is set at the center of the bounding box, and its x, y, z axes are respectively aligned with the 1st, 2nd and 3rd principal components. Since OBB is rotation invariant, projected images on OBB's side planes are also rotation invariant. Thus, this projection method using OBB is robust to variation in global hand orientations.

2) Projection With Axis-Aligned Bounding Box: The axis-aligned bounding box (AABB) is another commonly used bounding volume. To provide more robust hand pose estimation, we also project the 3D points onto three orthogonal side planes of AABB, as shown in Fig. 3 (right). The AABB is the minimum bounding box of which the edges are parallel with the axes of a fixed coordinate system [47]. Here, the fixed coordinate system is the camera coordinate system. The origin of the AABB projection coordinate system is set at the center of the bounding box, and its x, y, z axes are respectively aligned with the x, y, z axes of the camera coordinate system. We have found experimentally that the projected images generated using AABB are complementary to those generated using OBB, and their integration can further boost the estimation performance.

For generating the projected image on one view, the distances from segmented hand 3D points to the projection plane are first normalized between 0 and 1 (with nearest points set to 0, farthest points in the segmented hand point cloud set to 1). The values of pixels which are not belong to the hand region are set as 1. Then, 3D points are orthographically projected onto the quantized projection plane, and corresponding normalized distances are stored as pixel values of the projected image. If multiple 3D points are projected onto the same pixel region, the smallest normalized distance will be stored as the pixel value. As the projected images may be noisy and may have missing pixels due to self-occlusion in depth images [48], we perform median filtering and morphological operations to further smooth the projected images.

Fig. 4 presents some examples of projected images on the six views. As can be seen, projected images on different views can reflect different features of hand shapes. The projected images on OBB's three views are rotation invariant and can reflect front, top and lateral shapes of the hand. The projected images on the AABB's *x-y* view are almost the same as input depth images, because the *x-y* plane in AABB projection coordinate system is aligned with the projection plane of the depth image. In addition, even in the situation where the hand rotation is large, the front hand shape can still be recovered in the projected images, *e.g.* the sixth row's OBB *x-y* view and AABB *y-z* view in Fig. 4. Although these six views are not mutually orthogonal and there may be some redundancy between OBB's three views and AABB's three views, we combine all these six views to make more robust hand pose estimation, and experiments in Section V will

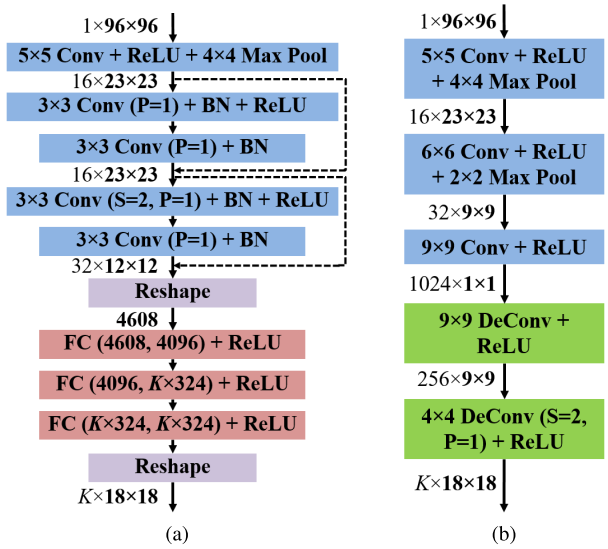


Fig. 5. (a) Residual Network. (b) Fully Convolutional Network (FCN). We experiment with these two network architectures separately in our experiments. Both networks take a 96×96 projected image as input and generate K heat-maps with the size of 18×18 pixels. The dotted arrows in (a) denote residual connections. 'BN' is short for batch normalization, 'S' means stride, 'P' means padding. For convolutional and deconvolutional layers, the default stride is 1, and the default padding is 0. For max pooling layers, the default stride is the same as kernel size, and the default padding is 0. For simplicity, we do not present stride and padding when using the default values.

show that the combination method can make more accurate estimation.

B. Multi-View CNNs Architecture

Multi-view CNNs aim to learn the relations between the projected images and the heat-maps reflecting hand joint locations on each view. Since we project 3D points onto multiple views, for each view, we construct a convolutional neural network having the same network architecture and the same architectural hyperparameters. In our conference paper [1], we employ the multi-resolution network architecture following the work in [4]. However, with the progress of neural networks, this network architecture is not optimal. In this paper, we propose to employ the residual network architecture (ResNet) [49] and the fully convolutional network architecture (FCN) [50] for inferring more accurate heat-maps from projected images, as shown in Fig. 5.

The input projected images are first resized and padded to 96×96 pixels and filtered by local contrast normalization (LCN) [51] to normalize the contrast in the image. The outputs of the network are K heat-maps with 18×18 pixels, of which the intensity indicates the confidence of a joint locating on the 2D position of a specific view.

We denote a training sample as $(X_t, \mathbf{\Phi}_t)$, where X_t is the depth image in the training dataset, $\mathbf{\Phi}_t$ is the corresponding joint locations in the camera coordinate system, $t = 1, \ldots, T$, T is the number of training samples. The depth image X_t is converted to the multi-view representation $\{I_{t,n}\}_{n=1}^{N}$ as inputs to multi-view CNNs. The ground truth heat-map $G_{k,n}(\mathbf{\Phi}_t)$ for the k-th joint on the n-th view is generated by applying 2D Gaussian centered at the 3D ground truth joint location's

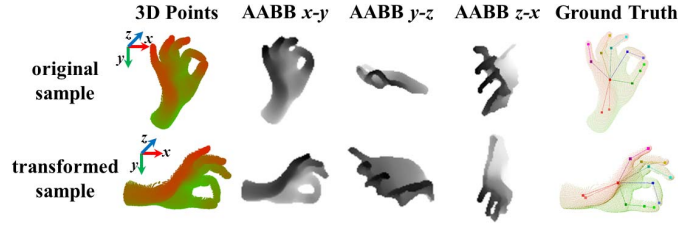


Fig. 6. An example of 3D data augmentation. The original training sample is shown in the first line, including 3D point cloud, projected images on AABB's three views and ground truth joint locations. The transformed training sample is shown in the second line. In this example, the 3D points are rotated by rotation angles $\theta_x = 10°$, $\theta_y = -20°$ and $\theta_z = 60°$.

2D projection point on the projection plane. The standard deviation of Gaussian is set to $\sigma = 1.8$ with an output heat-map size of 18×18, and the Gaussian is normalized to have a sum of 1. In the training stage, we minimize the following objective function:

$$\boldsymbol{w}_n^* = \arg\min_{\boldsymbol{w}_n} \sum_{t=1}^{T}\sum_{k=1}^{K} \left\| G_{k,n}(\mathbf{\Phi}_t) - \mathcal{H}_k\left(I_{t,n}, \boldsymbol{w}_n\right) \right\|_F^2, \quad (1)$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\boldsymbol{w}_n$ denotes network weights for the n-th view, $\mathcal{H}_k$ represents the k-th hand joint's heat-map output from the CNN regressor.

C. 3D Data Augmentation

Since AABB is sensitive to rotation, we propose to augment the training data by varying global hand orientations. Different with 2D image data augmentation, our proposed 3D data augmentation directly rotates the point cloud of hand in 3D space and then projects the 3D points onto AABB's three side planes.

The 3D point cloud of hand is rotated around x, y, z axes of the camera coordinate system with rotation angles θ_x, θ_y and θ_z, respectively. The 3D point $\boldsymbol{p}$ is transformed into $\boldsymbol{p}'$ after 3D rotation:

$$\boldsymbol{p}' = \mathcal{R}_x(\theta_x) \cdot \mathcal{R}_y(\theta_y) \cdot \mathcal{R}_z(\theta_z) \cdot \boldsymbol{p}, \quad (2)$$

where $\mathcal{R}_x$, $\mathcal{R}_y$ and $\mathcal{R}_z$ are 3×3 rotation matrices around x, y, z axes, respectively. Fig. 6 shows an example of 3D data augmentation. As can be seen, the projected images are not simply rotated in 2D image space. 3D rotation is performed on both hand point cloud and corresponding ground truth joint locations. Projected images on AABB's three views are generated from the transformed point cloud. Note that since OBB is rotation invariant, there is no need to augment training data on OBB's three views.

In this work, a transformed training set for AABB's three views is generated by randomly rotating original training samples. The rotation angles θ_x and θ_y are chosen uniformly at random from the interval $[-30°, 30°]$. The rotation angle θ_z is chosen uniformly at random from the interval $[-90°, 90°]$. During the training stage, both the original training set and the transformed training set are used for training CNNs of AABB's three views, in order to make them robust to variations in global hand orientations.

IV. The Multi-View Fusion Algorithm

In this section, we describe our proposed multi-view fusion method. The objective for multi-view fusion is to estimate the 3D hand joint locations $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_k\}_{k=1}^{K}$ from all the N views' heat-maps. We first formulate the multi-view fusion problem as the maximum a posteriori (MAP) estimation, and propose view selection method for suppressing unreliable view's estimation. We then derive the analytical solution to the optimization problem and obtain final estimation of 3D hand joint locations. Note that compared with the earlier version of this paper [1], the multi-view fusion method proposed in this section is more general without restricting the number of projection views and more robust to disagreement among multi-view heat-maps.

A. Problem Formulation

We estimate the hand joint locations $\boldsymbol{\Phi}$ by applying the MAP estimator on the basis of projections $I_1, I_2, \ldots, I_N$, which can be viewed as the observations of the 3D hand pose. Given the query hand depth image I_D, we assume that the N projections $I_1, I_2, \ldots, I_N$ are independent, conditioned on the joint locations $\boldsymbol{\Phi}$ [52]. Although this is not a strict assumption when projection planes are not mutually orthogonal, we will show experimentally that our proposed fusion method based on this assumption is quite effective compared with state-of-the-art methods in Section V. We also assume that the K hand joint locations are independent conditioned on each view's projection. Under these two assumptions and the assumption of equal a priori probability $P(\boldsymbol{\Phi})$, the posterior probability of joint locations can be formulated as the product of the individual estimations from all the N views. The problem to find the optimal hand joint locations $\boldsymbol{\Phi}^*$ is thus formulated as follows:

$$\begin{aligned}\boldsymbol{\Phi}^* &= \arg\max_{\boldsymbol{\Phi}} P(\boldsymbol{\Phi} | I_1, I_2, \ldots, I_N) \\ &= \arg\max_{\boldsymbol{\Phi}} \prod_{k=1}^{K} \prod_{n=1}^{N} P(\boldsymbol{\phi}_k | I_n) \\ &\quad s.t.\ \boldsymbol{\Phi} \in \boldsymbol{\Omega},\end{aligned} \tag{3}$$

where $\boldsymbol{\Phi}$ is constrained to a low dimensional subspace $\boldsymbol{\Omega} \subseteq \boldsymbol{\Lambda}$ in order to resolve ambiguous joint estimations.

The posterior probability of the k-th hand joint location on the n-th view $P(\boldsymbol{\phi}_k | I_n)$ $(k = 1, 2, \ldots, K; n = 1, 2, \ldots, N)$ can be estimated from heat-maps generated by multi-view CNNs. Since the intensity on a heat-map indicates the confidence of a joint locating in the 2D position on the projection plane, we can get the corresponding probability distribution $P(\boldsymbol{\phi}_{k,n} | I_n)$ from the n-th view's k-th heat-map, where $\boldsymbol{\phi}_{k,n}$ is the k-th 3D hand joint location $\boldsymbol{\phi}_k$'s 2D projection point on the n-th view's projection plane.

We denote the signed distance from 3D joint location $\boldsymbol{\phi}_k$ to the n-th view's projection plane as $d_{k,n}$. Thus, $\boldsymbol{\phi}_k$ can be decomposed into $\boldsymbol{\phi}_{k,n}$ and $d_{k,n}$ which are independent of each other. Assuming that, conditioned on the n-th view's projection I_n, the distribution of the signed distance variable $d_{k,n}$ is uniform, we have:

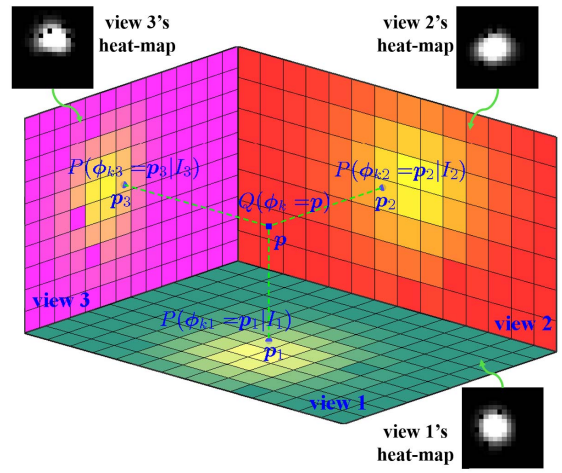


Fig. 7. Illusion of multi-view fusion. For illustration purpose, we only present a case having three views for fusion in this figure. The probability distribution $P(\boldsymbol{\phi}_{k,n} | I_n)$ on the n-th view is obtained from its corresponding heat-map ($n = 1, 2, \ldots, N$; in this figure, $N = 3$). For a 3D sampling point $\boldsymbol{p}$, it is projected onto all the N views to get its 2D projection points $\boldsymbol{p}_n$ and their corresponding heat-map intensities $P(\boldsymbol{\phi}_{k,n} = \boldsymbol{p}_n | I_n)$. Then, $Q(\boldsymbol{\phi}_k = \boldsymbol{p}) = \prod_{n=1}^{N} P(\boldsymbol{\phi}_{k,n} = \boldsymbol{p}_n | I_n)$.

$$\begin{aligned}P(\boldsymbol{\phi}_k | I_n) &= P(\boldsymbol{\phi}_{k,n}, d_{k,n} | I_n) \\ &= P(\boldsymbol{\phi}_{k,n} | I_n) P(d_{k,n} | I_n) \\ &\propto P(\boldsymbol{\phi}_{k,n} | I_n).\end{aligned} \tag{4}$$

Thus, the optimization problem in Eq. 3 can be transformed into:

$$\begin{aligned}\boldsymbol{\Phi}^* &= \arg\max_{\boldsymbol{\Phi}} \prod_{k=1}^{K} \prod_{n=1}^{N} P(\boldsymbol{\phi}_{k,n} | I_n) \\ &= \arg\max_{\boldsymbol{\Phi}} \prod_{k=1}^{K} Q(\boldsymbol{\phi}_k),\end{aligned} \tag{5}$$

where $Q(\boldsymbol{\phi}_k) = \prod_{n=1}^{N} P(\boldsymbol{\phi}_{k,n} | I_n)$ for the k-th hand joint.

Eq. 5 indicates that we can get the optimal hand joint locations by maximizing the product of $Q(\boldsymbol{\phi}_k)$ for all the joints. The distribution of $Q(\boldsymbol{\phi}_k)$ can be estimated from sampled values of $Q(\boldsymbol{\phi}_k = \boldsymbol{p})$, where $\boldsymbol{p}$ denotes the 3D sampling point. In this work, when using OBB/AABB's three views, the 3D points are uniformly sampled in the corresponding bounding box; when using the combination of OBB and AABB's six views, the 3D points are uniformly sampled in the intersection of OBB and AABB. As shown in Fig. 7, the 3D sampling point $\boldsymbol{p}$ is projected onto all the N views to get its 2D projection points $\boldsymbol{p}_n$ and their corresponding heat-map intensities $P(\boldsymbol{\phi}_{k,n} = \boldsymbol{p}_n | I_n)$ $(n = 1, 2, \ldots, N)$. Then, the value of $Q(\boldsymbol{\phi}_k = \boldsymbol{p})$ for a 3D point $\boldsymbol{p}$ can be computed by multiplying these intensities.

B. View Selection

In order to handle the disagreement among multiple views' estimations, if the confidence values of some views are evidently smaller than those of all the other views, these

values should be suppressed. For each 3D sampling point $\boldsymbol{p}$, we define the set of views whose intensities are below a threshold as $\mathbf{U}_{\boldsymbol{p}} = \left\{ n' \middle| P\left(\phi_{k,n'} = \boldsymbol{p}_{n'} \middle| I_{n'}\right) < \tau_{k,n'} \right\}$, where $\tau_{k,n'}$ is the threshold determined by the Otsu's thresholding method on the corresponding heat-map. If there is only a small portion of views whose intensities are below the threshold, namely $\left|\mathbf{U}_{\boldsymbol{p}}\right| \leq \varepsilon N$, it means that confidence values provided by these views may be unreliable and the value of $Q\left(\boldsymbol{\phi}_k = \boldsymbol{p}\right)$ will be computed from all the other views' intensities:

$$Q\left(\phi_k = \boldsymbol{p}\right) = \left(\prod_{n=1, n \notin \mathbf{U}_{\boldsymbol{p}}}^{N} P\left(\phi_{k,n} = \boldsymbol{p}_n \middle| I_n\right) \right)^{\frac{N}{N - \left|\mathbf{U}_{\boldsymbol{p}}\right|}}$$

when

$$\left|\mathbf{U}_{\boldsymbol{p}}\right| \leq \varepsilon N, \tag{6}$$

where the power $N \big/ \left(N - \left|\mathbf{U}_{\boldsymbol{p}}\right|\right)$ is used for normalization.

If more views' intensities are below the threshold, namely $\left|\mathbf{U}_{\boldsymbol{p}}\right| > \varepsilon N$, we no longer suppress the small intensities and the value of $Q\left(\boldsymbol{\phi}_k = \boldsymbol{p}\right)$ will be assigned to zero. In our implementation, the coefficient ε is set to $\varepsilon = 1/5$. Thus, when we use OBB/AABB's three views, namely $N = 3$, no views' intensities will be suppressed; when we use the combination of OBB and AABB's six views, namely $N = 6$, at most one view's intensity will be suppressed. By applying the method of view selection, information in heat-maps can be more effectively utilized, thus the estimation will be more robust and accurate.

C. Solution to the Optimization Problem

For simplicity of the optimization problem, the distribution of $Q\left(\boldsymbol{\phi}_k\right)$ is approximated as a 3D Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $\boldsymbol{\mu}_k$ is the weighted mean vector, $\boldsymbol{\Sigma}_k$ is the weighted covariance matrix. These parameters of the Gaussian distribution can be estimated from the sampled values of $Q\left(\boldsymbol{\phi}_k = \boldsymbol{p}\right)$:

$$\boldsymbol{\mu}_k = \sum_{\boldsymbol{p}} w_{\boldsymbol{p}}^{(k)} \boldsymbol{p}, \quad \boldsymbol{\Sigma}_k = \sum_{\boldsymbol{p}} w_{\boldsymbol{p}}^{(k)} \left(\boldsymbol{p} - \boldsymbol{\mu}_k\right)\left(\boldsymbol{p} - \boldsymbol{\mu}_k\right)^T, \tag{7}$$

where $w_{\boldsymbol{p}}^{(k)} = Q\left(\boldsymbol{\phi}_k = \boldsymbol{p}\right) \Big/ \sum_{\boldsymbol{p}'} Q\left(\boldsymbol{\phi}_k = \boldsymbol{p}'\right)$ is the weight of sampling point $\boldsymbol{p}$ for joint k.

Based on above assumptions and derivations, the optimization problem in Eq. 5 can be approximated as follows:

$$\begin{aligned} \boldsymbol{\Phi}^* &= \arg\max_{\boldsymbol{\Phi}} \sum_{k=1}^{K} \log Q\left(\boldsymbol{\phi}_k\right) \\ &= \arg\max_{\boldsymbol{\Phi}} \sum_{k=1}^{K} \log \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &= \arg\min_{\boldsymbol{\Phi}} \sum_{k=1}^{K} \left(\boldsymbol{\phi}_k - \boldsymbol{\mu}_k\right)^T \boldsymbol{\Sigma}_k^{-1} \left(\boldsymbol{\phi}_k - \boldsymbol{\mu}_k\right) \\ &\quad s.t.\ \boldsymbol{\Phi} = \sum_{m=1}^{M} \alpha_m \boldsymbol{e}_m + \boldsymbol{u}, \end{aligned} \tag{8}$$

where $\boldsymbol{\Phi}$ is constrained to take the linear form. In order to learn the low dimensional subspace $\boldsymbol{\Omega}$ of hand configuration constrains, PCA is performed on joint locations in the training dataset during the training stage [37]. $\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_M]$ are the principal components, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_M]^T$ are the coefficients of the principal components, $\boldsymbol{u}$ is the empirical mean vector, and $M \ll 3 \times K$.

As proved in Appendix VI, given the linear constraints of $\boldsymbol{\Phi}$, the optimal coefficient vector $\boldsymbol{\alpha}^* = \left[\alpha_1^*, \alpha_2^*, \cdots, \alpha_M^*\right]^T$ is:

$$\boldsymbol{\alpha}^* = \mathbf{A}^{-1} \boldsymbol{b}, \tag{9}$$

where $\mathbf{A}$ is an $M \times M$ symmetric matrix, $\boldsymbol{b}$ is an M-dimensional column vector:

$$\mathbf{A}_{ij} = \sum_k \boldsymbol{e}_{j,k}^T \boldsymbol{\Sigma}_k^{-1} \boldsymbol{e}_{i,k}, \quad \boldsymbol{b}_i = \sum_k \left(\boldsymbol{\mu}_k - \boldsymbol{u}_k\right)^T \boldsymbol{\Sigma}_k^{-1} \boldsymbol{e}_{i,k},$$

$\boldsymbol{e}_i = \left[\boldsymbol{e}_{i,1}^T, \boldsymbol{e}_{i,2}^T, \cdots, \boldsymbol{e}_{i,K}^T\right]^T$; $\boldsymbol{u} = \left[\boldsymbol{u}_1^T, \boldsymbol{u}_2^T, \cdots, \boldsymbol{u}_K^T\right]^T$; $i,\ j = 1,\ 2, \cdots,\ M$.

The optimal joint locations $\boldsymbol{\Phi}^*$ are reconstructed by back-projecting the optimal coefficients $\boldsymbol{\alpha}^*$ in the subspace $\boldsymbol{\Omega}$ to the original joint space $\boldsymbol{\Lambda}$:

$$\boldsymbol{\Phi}^* = \sum_{m=1}^{M} \alpha_m^* \boldsymbol{e}_m + \boldsymbol{u}. \tag{10}$$

To sum up, the proposed method for multi-view fusion consists of two main steps. The first step is to estimate the parameters of Gaussian distribution $Q\left(\boldsymbol{\phi}_k\right)$ for each joint using all the N views' heat-maps. Unreliable intensities of some views will be suppressed when calculating the sampled value of $Q\left(\boldsymbol{\phi}_k = \boldsymbol{p}\right)$. The second step is to calculate the optimal coefficients $\boldsymbol{\alpha}^*$ and reconstruct the optimal joint locations $\boldsymbol{\Phi}^*$. Since an analytical solution has been derived, the 3D joint locations can be inferred efficiently using the proposed multi-view fusion algorithm.

V. Experiments

A. Dataset

We conduct self-comparisons and comparisons with state-of-the-art methods on two datasets released in [4] and [10].

The MSRA Hand Pose Dataset released in [10] (called *MSRA2015* dataset for short) contains nine subjects' right hands, each subject contains 17 gestures and each gesture contains about 500 frames, which are captured using the Intel's Creative Interactive Gesture Camera. In the following experiments, we use eight subjects as the training set containing about 84,000 frames for training multi-view CNNs and the remaining one as the testing set containing about 10,500 frames. This experiment is repeated nine times for all subjects. For each depth image, the ground truth contains 21 3D hand joint locations. As shown in Fig. 8a, the 21 hand joints are the wrist center, five metacarpophalangeal joints, five proximal interphalangeal joints, five distal interphalangeal joints and five finger tips, respectively.

The NYU Hand Pose Dataset released in [4] (called *NYU* dataset for short) contains 72,757 training frames and 8,252 testing frames, which are captured by the PrimeSense™ 3D sensor. For each frame, the ground truth contains 36 3D hand joint locations. However, our evaluation is performed on a subset of 14 hand joints, following previous work [4], [5], [17]. As shown in Fig. 8b, the 14 objective hand joints are five finger tips, five proximal interphalangeal joints, the distal interphalangeal joint of the thumb, two wrist

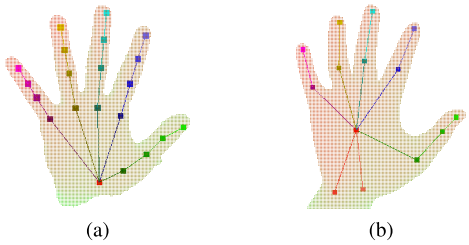


Fig. 8. Illustration of objective hand joints. (a) Objective hand joints in MSRA2015 dataset. The number of hand joints is $K = 21$. (b) Objective hand joints in NYU dataset. The number of hand joints is $K = 14$.

joints and the palm center, respectively. Since the NYU dataset provides original depth images containing human body and background, the hand should be segmented from the original depth image. Similar to [4], random decision forest (RDF) [53] is applied for hand segmentation. Note that in our conference paper [1], we did not perform any experiments on this dataset.

In addition, in order to evaluate the generalization ability of our method, we conduct a cross-dataset experiment by training the multi-view CNNs on MSRA2015 dataset and testing on the MSRA Hand Tracking Dataset released in [31] (called *MSRA2014* dataset for short). The definition of objective hand joints in MSRA2014 dataset is the same as that in MSRA2015 dataset. But the hand poses, hand sizes and hand shapes are different in these two datasets.

B. Evaluation Metrics

We employ three metrics to evaluate the estimation performance. The first metric is the mean error distance for each hand joint across all test frames, which is a standard evaluation metric. The second metric is the proportion of good test frames in the entire test frames. A test frame is regarded as good only when all estimated joint locations are within a maximum allowed distance from the ground truth, namely the error threshold. This worst case accuracy proposed in [54] is a very strict criterion. The third metric is the proportion of joints within an error threshold among all test joints [39].

C. Implementation Details

Our proposed multi-view CNNs are implemented within the Torch7 [55] framework. When training the multi-resolution network in [1], we use the stochastic gradient descent algorithm with learning rate of 0.2, batch size of 64, momentum of 0.9 and weight decay of 0.0005. When training ResNet and FCN, we use the RMSprop algorithm with learning rate of 0.001, batch size of 64, epsilon of 0.01. Training is stopped after 50 epochs to prevent overfitting. All experiments are conducted on a computer with two Intel Core i7-5930K processors, 64GB of RAM and two Nvidia Tesla K80 GPUs each having 12GB of GPU memory.

D. Self-Comparison

1) Single View CNN Versus Multi-View CNNs: To evaluate the superiority of the multi-view method over the single view method, we implement the single view CNN-based approach. In our implementation, only projected images on the x-y plane of OBB projection coordinate system are fed into the CNN, which is similar to the method in [4]. We can only estimate x and y coordinates of hand joints from output heat-maps using the 2D Gaussian fitting method as that in [4]. The z coordinate can be inferred using the pixel value of the projected image. If the estimated 2D position of a hand joint is on the background of the projected image, the z coordinate will be set as zero in the OBB projection coordinate system rather than the maximum depth value, which can lower the estimation error on z direction. As presented in Fig. 9 (left and middle), the multi-view regression approach significantly outperforms the single view regression approach.

2) Evaluating Fusion Methods: To evaluate the effectiveness of our proposed optimal fusion method, we implement the average fusion method as baseline, which can be regarded as a simplified alternative of the proposed optimal fusion method. This method first estimates 2D projection points of the 3D hand joint location on x-y, y-z and z-x planes of the OBB projection coordinate system by fitting Gaussian models on corresponding heat-maps. The 3D hand joint location is determined by averaging x, y and z coordinates, which are obtained from three views' 2D projection points, in the OBB projection coordinate system. As can be seen in Fig. 9 (left and middle), the mean error over all joints of the optimal fusion method is 13.1mm on MSRA2015 dataset, while that of the average fusion method is 15.8mm; for the worst case accuracy, the optimal fusion method performs better than the average fusion method when the error threshold is smaller than 50mm, but is a little bit worse than the average fusion method when the error threshold is larger than 50mm. The reason may be that the average fusion method takes the average among ambiguous estimations, which may not deviate from the right estimation very large. The optimal fusion method is able to choose the optimal one among ambiguous estimations. But once this method chooses the wrong estimation, the estimation error will be very large. However, high accuracy at small error threshold should be more favorable. The optimal fusion method is overall better than the average fusion method and we adopt the optimal fusion method in the following experiments.

An experimental example of the ambiguous estimation is presented in Fig. 10, where the tip of index finger is very likely to be confused with the tip of little finger. As shown in this figure, the single view regression method only exploits the heat-map of OBB's x-y view containing two hotspots, thus estimates the location of index fingertip with large error distance. However, the multi-view optimal fusion method fuses heat-maps of three views and estimates the 3D location with high accuracy. The estimation of the multi-view average fusion method is between those of the above two methods, as this method underutilizes the information of heat-maps.

Additionally, we evaluate the impact of different numbers of principal components M used for the constraint in Eq. 8 on the worst case accuracy under different error thresholds. As can be seen in Fig. 9 (right), it is suitable to use 35 principal components for constraint considering the estimation accuracy. We set $M = 35$ for all the other experiments.

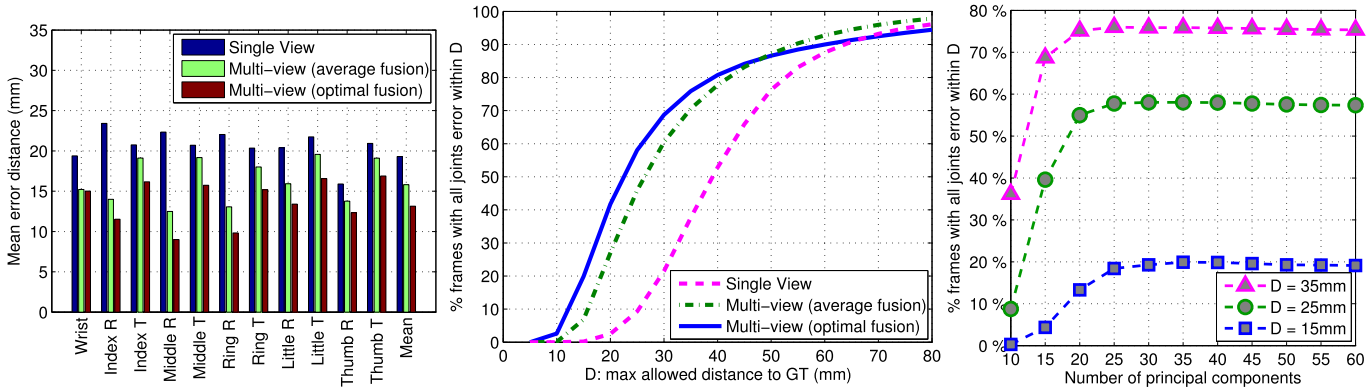


Fig. 9. Self-comparison of different methods on MSRA2015 dataset [10] using OBB's three views. **Left**: the mean error distance for each joint across all the test frames (R: root, T: tip). **Middle**: the proportion of good test frames in the entire test frames over different error thresholds. **Right**: the influence of different numbers of principal components used in hand pose constraints on the estimation accuracy.

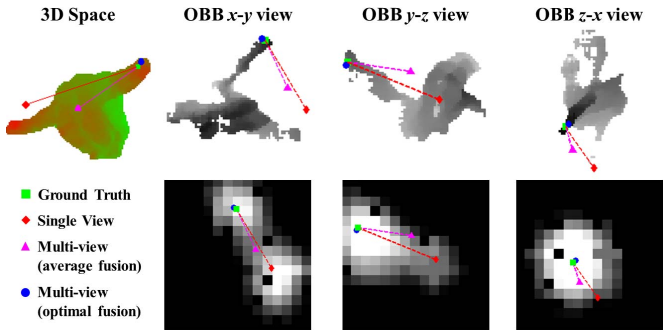


Fig. 10. A case of ambiguous estimation. **Top-left**: 3D point cloud with ground truth and estimated 3D locations of the tip of index finger. **Top-right**: projected images on OBB's three views. **Bottom-right**: heat-maps of OBB's three views. The ground truth and estimated 3D locations of the tip of index finger are projected onto three views and their heat-maps for comparison. Lines indicate offsets between estimated joint locations and the ground truth.

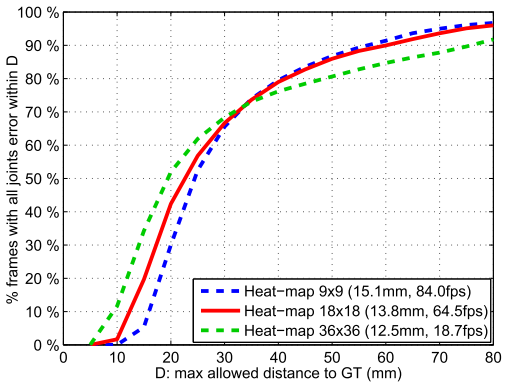


Fig. 11. Self-comparison of different heat-map resolutions on NYU [4] hand pose dataset. The mean error distances over all joints and the average frame rates are shown in the legends.

3) Impact of Heat-Map Resolution: We evaluate the impact of the heat-map resolution on accuracy and real-time performance. In this experiment, we apply the FCN architectures with view selection and data augmentation for generating heat-maps with different resolutions, *i.e.*, 9×9, 18×18 and 36×36. To make fair comparison, we design the FCNs with comparable numbers of network parameters for different heat-map resolutions. As shown in Fig. 11, when the error threshold is smaller than 30mm, the method with 36×36 heat-maps performs best, and the method with 9×9 heat-maps performs worst. However, when the error threshold is larger than 30mm, the method with 36×36 heat-maps performs worst. Moreover, the method with 36×36 heat-maps is time-consuming, which runs at 18.7fps and cannot achieve real-time performance. Balancing between the runtime and the estimation accuracy, we choose 18×18 as the heat-map resolution.

4) Effectiveness of Combination OBB With AABB: We experiment with the hand pose estimation method using OBB/AABB's three views as well as the method using both OBB and AABB's six views. As shown in Fig. 12, compared with the method using OBB's three views, the method using AABB's three views performs almost the same on MSRA2015 dataset, and performs worse on NYU dataset. But the method using both OBB and AABB's six views outperforms the first two methods over most error thresholds even without using the view selection method. These results demonstrate that the projected images of OBB and AABB's six views are complementary to each other, and the fusion of six views' heat-maps can produce more accurate and robust results.

5) Effectiveness of View Selection: When fusing heat-maps from multiple views, our method selects views with reliable estimations and suppresses unreliable estimations. As shown in Fig. 12, when using the combination of OBB and AABB's six views, the view selection method will further improve the estimation accuracy, which indicates that it is effective to deal with the disagreement among multiple views' estimations by using our proposed view selection method.

6) Effectiveness of 3D Data Augmentation: In Section III-C, we augment the training data for AABB's three views by randomly rotating the 3D point cloud. As shown in Fig. 12, when applying 3D data augmentation, the estimation accuracy is slightly better than the method without using data augmentation. If the variation in hand global orientations is larger

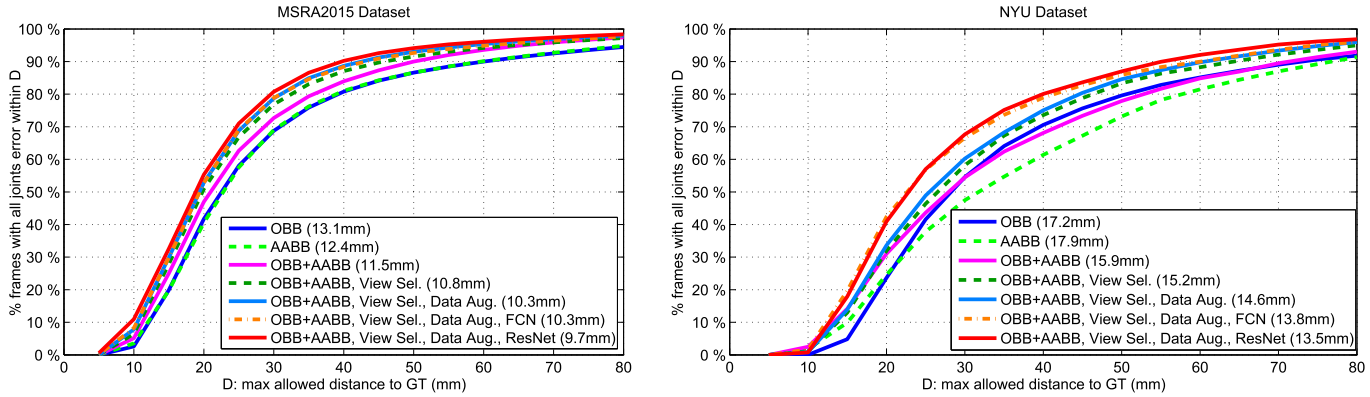


Fig. 12. Evaluation of combining OBB and AABB's six views, view selection method, 3D data augmentation and different network architectures on MSRA2015 dataset [10] and NYU dataset [4]. The mean error distances over all joints of different methods are shown in the legend titles.

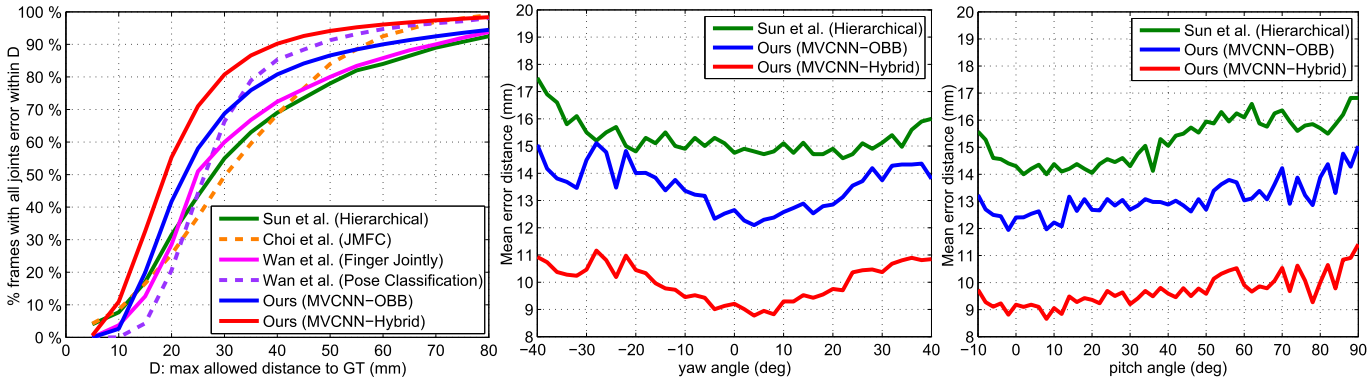


Fig. 13. Comparison with state-of-the-art approaches on MSRA2015 dataset [10]. **Left**: the proportion of good test frames in the entire test frames over different error thresholds. **Middle & right**: the mean error distance over different yaw and pitch angles of the viewpoint. Some curves are cropped from corresponding figures reported in [10], [15], and [34].

for the test frames, the improvement achieved by 3D data augmentation will be more evident.

7) Comparison of Network Architectures: In all of the above experiments, we employ the multi-resolution network in [1] to estimate the heat-maps. Here, we experiment with the ResNet and FCN. As shown in Fig. 12, both the ResNet and FCN can improve the estimation accuracy, especially on the NYU dataset, which shows that both networks have good generalization ability. Moreover, the FCN has fewer parameters (24 million on both datasets) than the multi-resolution network (99 million on the MSRA dataset and 56 million on the NYU dataset) and ResNet (93 million on the MSRA dataset and 58 million on the NYU dataset). Thus, the model size of FCN is smaller. But the estimation accuracy of ResNet is slightly better than that of FCN. In following comparisons with state-of-the-art methods, we present results of the ResNet due to its better estimation accuracy.

The last four self-comparison experiments are newly added in this paper compared with our conference paper [1].

E. Comparison With State-of-the-Art

In this section, we compare our methods with state-of-the-art methods. We denote the method proposed in our conference paper [1] using OBB's three views and multi-resolution network as *MVCNN-OBB*, and our method using OBB and AABB's six views with the ResNet architecture, view selection and data augmentation as *MVCNN-Hybrid*.

1) Comparison on MSRA2015 Dataset: On MSRA2015 dataset, we compare our multi-view CNN-based method with three state-of-the-art methods: the random forest based hierarchical regression method [10], the JMFC based collaborative filtering method [15] and the local surface normals based method with finger jointly regression and pose classification [34]. Note that the first two methods have been validated to be superior to methods in [3], [12], and [27]. Thus, we indirectly compare with these methods. However, in our conference paper [1], we only compared with the first method.

As shown in Fig. 13 (left), our MVCNN-Hybrid method achieves the best performance when the error threshold is larger than 5mm, and our MVCNN-OBB method achieves the second best performance when the error threshold is between 15mm and 30mm. When the error threshold is 5mm, the good frame proportions of our methods are slightly inferior to those of the hierarchical regression method [10] and the JMFC based method [15]. This may be induced by the relatively low resolution of heat-maps (18×18) adopted in our methods.

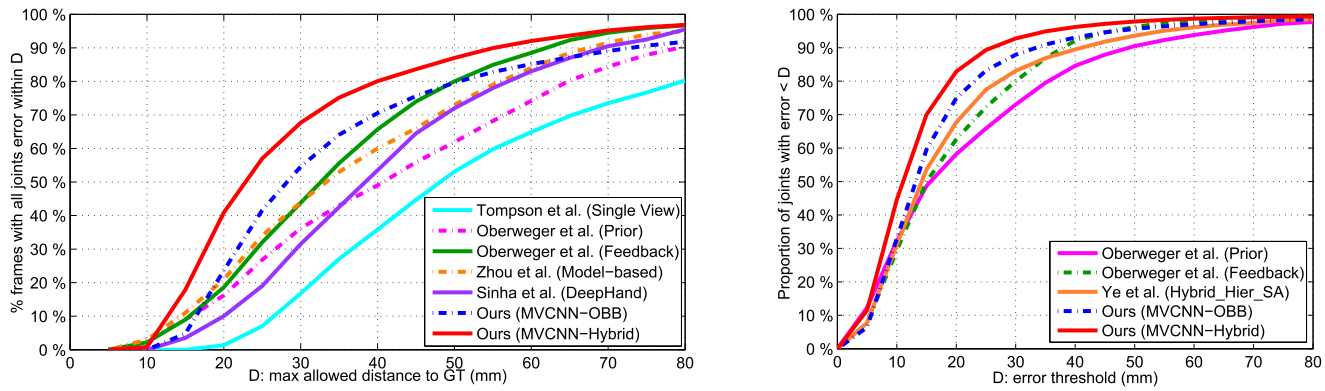


Fig. 14. Comparison with state-of-the-art approaches on NYU dataset [4]. **Left**: the proportion of good test frames in the entire test frames over different error thresholds. **Right**: the proportion of joints within different error thresholds. Some curves cropped from corresponding figures reported in [5] and [16]–[19].

We evaluate the mean error distances over different yaw and pitch angles of these methods. As presented in Fig. 13 (middle and right), the mean error distances of our MVCNN-OBB method and MVCNN-Hybrid method are about 2mm and 5mm smaller than those of the hierarchical regression method [10] over all the yaw and pitch angles, respectively. Moreover, our methods are more robust to the pitch angle variation with smaller stand deviations (0.64mm for MVCNN-OBB and 0.58mm for MVCNN-Hybrid) than the hierarchical regression method (0.79mm) [10].

2) Comparison on NYU Dataset: On NYU dataset, we first compare the worst case accuracies of our proposed multi-view CNN-based methods with five state-of-the-art methods. The first method is the single view CNN-based heat-map regression method proposed in [4] which directly adopts the depth image as the input of the network (denoted as *Single View*). For comparison, the estimated 2D joint location in the depth image is converted to 3D location with the information of its corresponding depth value. The second method is the CNN-based direct hand pose estimation with a prior proposed in [17] (denoted as *Prior*). The third method is the CNN-based hand pose estimation using a feedback loop proposed in [5] (denoted as *Feedback*). The fourth method is the CNN-based hand model parameters regression proposed in [18] (denoted as *Model-based*). The last method is the deep feature based matrix completion method proposed in [16] (denoted as *DeepHand*). As shown in Fig. 14 (left), our MVCNN-Hybrid method significantly outperforms these five state-of-the-art methods when the error threshold is larger than 10mm, and our MVCNN-OBB method achieves the second best performance when the error threshold is between 20mm and 45mm.

In order to make a fair comparison with the spatial attention network based hierarchical hybrid method proposed in [19] (denoted as *Hybrid_Hier_SA*), we evaluate the proportion of joints within different error thresholds on the subset containing 11 hand joints following the experimental setting in [19] (removing palm joints except the root joint of thumb). As shown in Fig. 14 (right), our MVCNN-Hybrid method outperforms the methods in [5], [17], and [19] over all the error thresholds, and our MVCNN-OBB method achieves the second best performance when the error threshold is larger than 10mm.

TABLE II

AVERAGE ESTIMATION ERRORS (IN *mm*) OF 6 SUBJECTS FOR 7 METHODS EVALUATED ON MSRA2014 DATASET [31]

Subject	1	2	3	4	5	6	Avg
FORTH	35.4	19.8	27.3	26.3	16.6	46.2	28.6
PSO	29.3	14.8	40.2	17.3	16.2	24.3	23.6
ICP	29.9	20.7	30.8	23.9	18.5	32.8	26.1
ICP-PSO	10.1	24.1	13.0	12.8	11.9	20.0	15.3
ICP-PSO*	8.6	7.4	9.8	10.4	7.8	11.7	9.2
MVCNN-OBB	30.1	19.7	24.3	19.9	21.8	20.7	22.8
MVCNN-Hybrid	21.8	16.3	16.7	16.3	14.7	16.1	17.0

F. Cross-Dataset Experiment

We conduct a cross-dataset experiment to further validate the generalization ability of our proposed multi-view CNN-based hand pose estimation method. In this experiment, we aim at adapting the multi-view CNNs trained on the source MSRA2015 dataset [10] to the target MSRA2014 dataset [31].

We train the multi-view CNNs on all the nine subjects in MSRA2015 dataset [10]. The fully trained multi-view CNNs are evaluated on all the six subjects in MSRA2014 dataset [31]. Following the evaluation criterion in [31], we only evaluate the average estimation errors for the wrist center and five fingertips. As shown in Table II, we compare our MVCNN-OBB method and MVCNN-Hybrid method with model-based tracking approaches presented in [31], which are FORTH [27], PSO [31], ICP [56], ICP-PSO [31] and ICP-PSO* (ICP-PSO with finger based initialization) [31]. In our conference paper [1], we did not evaluate our MVCNN-Hybrid method in this experiment.

As reported by [31], these model-based tracking methods require a carefully calibrated hand model for each subject's hand, and these methods rely on the temporal information for tracking. Particularly, these methods utilize ground truth information for the first frame initialization. However, our methods do not utilize these information, thus are more flexible for different subjects and are robust to estimation

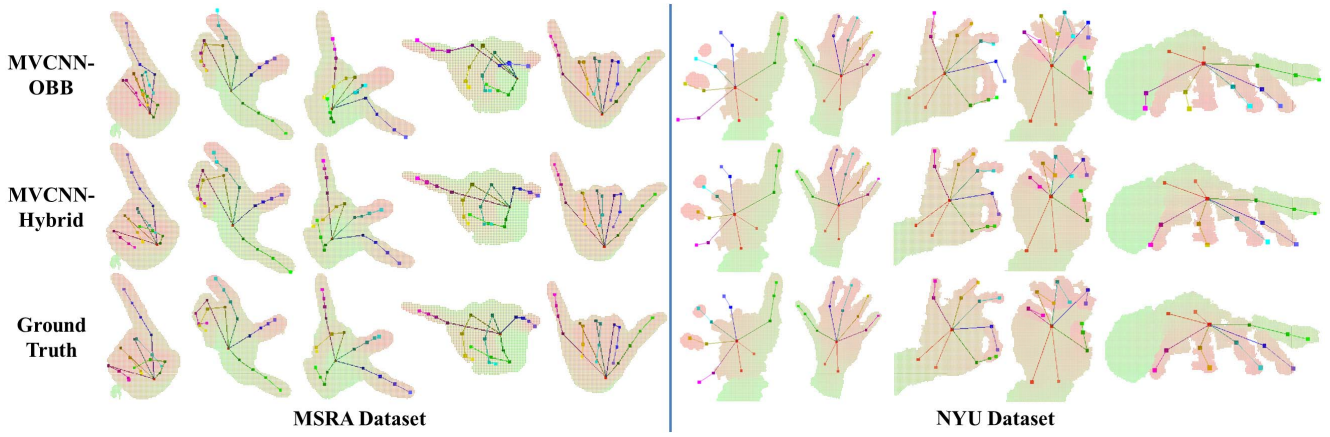


Fig. 15. Qualitative results for MSRA2015 dataset [10] and NYU dataset [4] of two methods: the MVCNN-OBB method (in the first line) and the MVCNN-Hybrid method (in the second line). The ground truth hand joint locations are shown in the last line. We show hand joint locations and bones with the point cloud. Different hand joints and bones are visualized using different colors. This figure is best viewed in color.

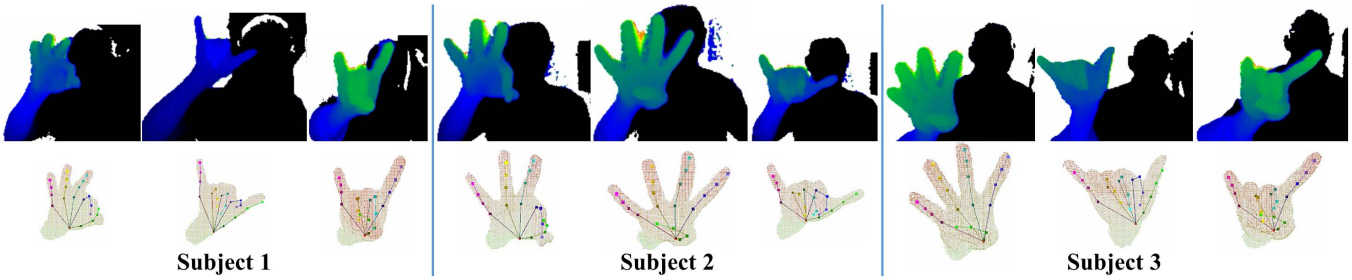


Fig. 16. Qualitative results for testing in real scenarios on different subjects with different hand sizes and hand poses. For each subject, the first line are the depth images captured by the SoftKinetic's DepthSense camera; the second line is the segmented hand depth images and the estimated hand joint locations. Different hand joints and bones are visualized using different colors. This figure is best viewed in color.

failure. Despite these unfavorable conditions, both of our MVCNN-OBB method and MVCNN-Hybrid method still outperform FORTH, PSO and ICP methods, as presented in Table II, which demonstrates that our methods have good generalization ability. It is not surprising that our methods are inferior to ICP-PSO and ICP-PSO*, since ICP-PSO and ICP-PSO* are tracking methods that require a carefully calibrated hand model and ground truth for first frame initialization, while our methods do not require any hand model and do not utilize any temporal information for tracking or ground truth for initialization. Furthermore, we conduct this experiment on cross-dataset that is more challenging. It is worth noting that the average estimation error of our MVCNN-Hybrid method is only 1.7mm worse than that of the ICP-PSO method.

G. Qualitative Results

Fig. 15 shows qualitative results of the MVCNN-OBB method and the MVCNN-Hybrid method on several challenging test frames in MSRA2015 dataset [10] and NYU dataset [4]. As can be seen, the estimation accuracy of the MVCNN-Hybrid method is generally better than that of the MVCNN-OBB method. It is worth noting that even though some depth images in NYU dataset are very noisy and some frames are incomplete, *e.g.*, the 6th and 10th columns in Fig. 15, our proposed multi-view CNN-based methods are still able to make accurate estimations of 3D hand joint locations from these depth images.

We additionally evaluate our pre-trained multi-view CNN models in real scenarios, which is a newly added experiment compared with our conference paper [1]. We train the multi-view CNN models on all the 9 subjects of the MSRA2015 dataset in [10] and apply these models to perform real-time hand pose estimation. As shown in Fig. 15 (the 1st line), the depth images are captured by the SoftKinetic's DepthSense camera which is different with the Intel's Creative Interactive Gesture Camera used in MSRA2015 dataset [10]. We experiment on three subjects, who are not included in MSRA2015 dataset [10], with different hand sizes and hand poses. The qualitative results of hand pose estimation using our MVCNN-Hybrid method are shown in Fig. 16 (the 2nd line). As can be seen, our proposed multi-view CNN-based hand pose estimation method is tolerant for different hand poses and hand sizes, and it is even tolerant for different types of depth cameras of which the noise distribution may be different. This experiment further demonstrates the good generalization ability of our proposed multi-view CNN-based method.

H. Runtime

The runtime of our MVCNN-OBB method is 12.2ms in average, including 1.8ms for depth image preprocessing

and multi-view projection, 6.8ms for multi-resolution networks forward propagation and 3.6ms for multi-view fusion. Therefore, the MVCNN-OBB method runs in real-time at about 82fps. The runtime of our MVCNN-Hybrid method is 16.8ms in average, including 3.2ms for depth image preprocessing and multi-view projection, 9.3ms for ResNets forward propagation and 4.3ms for multi-view fusion. Therefore, the MVCNN-Hybrid method runs in real-time at about 60fps. Note that tasks of multi-view projection and multi-view fusion are executed on CPU with parallelism, and the task of CNNs forward propagation is executed on GPU with parallelism. Although the MVCNN-Hybrid method is slower than the MVCNN-OBB method, the MVCNN-Hybrid method can achieve more accurate and robust results than the MVCNN-OBB method and it can still run fast in real-time.

VI. Conclusion

In this paper, we have presented a multi-view CNN-based approach for robust 3D hand pose estimation. The 3D point cloud generated from the depth image is projected onto both OBB and AABB's six views. The multi-view CNNs are trained to map projected images to heat-maps representing the probability distributions of hand joints on projected images. Two network architectures are proposed using ResNet and FCN which have good generalization ability. 3D data augmentation is performed on training CNNs of AABB's three views to make them more robust to various hand orientations. Heat-maps from different views are fused to make the optimal estimation of 3D hand joint locations. Furthermore, we have proposed a view selection method to suppress unreliable information in multi-view heat-maps. Experimental results are presented to show the superior performance and good generalization ability of our proposed methods.

In the future work, our multi-view CNN-based method can be extended to 3D human pose estimation and tracking from depth images. We are also looking forward to extending the multi-view CNN-based method to multi-object tracking [57]–[61] with multiple cameras for visual surveillance.

Appendix
Derivation of the Optimal Solution

The optimization problem is formulated as follows:

$$\boldsymbol{\Phi}^* = \arg\min_{\boldsymbol{\Phi}} \sum_k (\boldsymbol{\phi}_k - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\phi}_k - \boldsymbol{\mu}_k)$$
$$s.t.\ \boldsymbol{\Phi} = \sum\nolimits_{m=1}^{M} \alpha_m \boldsymbol{e}_m + \boldsymbol{u} = \boldsymbol{E}\boldsymbol{\alpha} + \boldsymbol{u}$$

Let $R(\boldsymbol{\Phi}) = \sum_k (\boldsymbol{\phi}_k - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\phi}_k - \boldsymbol{\mu}_k)$, which is in the quadratic form of the variable $\boldsymbol{\alpha}$, the optimal solution of $\boldsymbol{\alpha}$ can be obtained by setting the derivative of R with respect to $\boldsymbol{\alpha}$ to zero.

$$\frac{\partial R(\boldsymbol{\Phi})}{\partial \boldsymbol{\alpha}} = \frac{\partial R(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}} \cdot \frac{\partial \boldsymbol{\Phi}(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}}$$
$$\because\ \frac{\partial R(\boldsymbol{\Phi})}{\partial \boldsymbol{\phi}_k} = \frac{\partial}{\partial \boldsymbol{\phi}_k}\left[(\boldsymbol{\phi}_k - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\phi}_k - \boldsymbol{\mu}_k)\right] = 2(\boldsymbol{\phi}_k - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}$$
$$\therefore\ \frac{\partial R(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}} = 2\begin{bmatrix} \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\phi}_1 - \boldsymbol{\mu}_1) \\ \vdots \\ \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\phi}_k - \boldsymbol{\mu}_k) \\ \vdots \\ \boldsymbol{\Sigma}_K^{-1}(\boldsymbol{\phi}_K - \boldsymbol{\mu}_K) \end{bmatrix}^T$$
$$\because\ \frac{\partial \boldsymbol{\Phi}(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = \frac{\partial}{\partial \boldsymbol{\alpha}}(\boldsymbol{E}\boldsymbol{\alpha} + \boldsymbol{u}) = \boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_M]$$
$$\therefore\ \frac{\partial R(\boldsymbol{\Phi})}{\partial \boldsymbol{\alpha}} = 2\begin{bmatrix} \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\phi}_1 - \boldsymbol{\mu}_1) \\ \vdots \\ \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\phi}_k - \boldsymbol{\mu}_k) \\ \vdots \\ \boldsymbol{\Sigma}_K^{-1}(\boldsymbol{\phi}_K - \boldsymbol{\mu}_K) \end{bmatrix}^T \cdot [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_M] = \boldsymbol{0}$$

Thus, we can get M linear equations for M unknown variables $\alpha_1, \alpha_2, \cdots, \alpha_M$:

$$\sum_{k=1}^{K}\left[\left(\sum_{j=1}^{M} \alpha_j \boldsymbol{e}_{j,k}^T + \boldsymbol{u}_k^T - \boldsymbol{\mu}_k^T\right)\boldsymbol{\Sigma}_k^{-1}\boldsymbol{e}_{i,k}\right] = 0, \quad i = 1,\ 2, \cdots,\ M$$
$$\therefore\ \sum_{j=1}^{M}\left[\left(\sum_{k=1}^{K} \boldsymbol{e}_{j,k}^T \boldsymbol{\Sigma}_k^{-1}\boldsymbol{e}_{i,k}\right)\alpha_j\right] = \sum_{k=1}^{K}(\boldsymbol{\mu}_k - \boldsymbol{u}_k)^T \boldsymbol{\Sigma}_k^{-1}\boldsymbol{e}_{i,k}$$

Let $\mathbf{A}\boldsymbol{\alpha} = \boldsymbol{b}$, then:

$$\mathbf{A}_{ij} = \sum_{k=1}^{K} \boldsymbol{e}_{j,k}^T \boldsymbol{\Sigma}_k^{-1}\boldsymbol{e}_{i,k},\ \ \boldsymbol{b}_i = \sum_{k=1}^{K}(\boldsymbol{\mu}_k - \boldsymbol{u}_k)^T \boldsymbol{\Sigma}_k^{-1}\boldsymbol{e}_{i,k}$$
$$i,\ j = 1,\ 2, \cdots,\ M$$

At last, we can get the solution for the optimization problem: $\boldsymbol{\alpha}^* = \mathbf{A}^{-1}\boldsymbol{b}$.

This research is supported by the BeingTogether Centre, a collaboration between NTU Singapore and UNC at Chapel Hill. The BeingTogether Centre is supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative. This work is also supported in part by Singapore Ministry of Education Academic Research Fund Tier 2 MOE2015-T2-2-114, a grant from Microsoft Research Asia, and start-up grants from University at Buffalo.

References

[1] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3593–3601.

[2] Y. Wu and T. S. Huang, "Hand modeling, analysis and recognition," *IEEE Signal Process. Mag.*, vol. 18, no. 3, pp. 51–60, May 2001.

[3] C. Xu and L. Cheng, "Efficient hand pose estimation from a single depth image," in *Proc. Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3456–3462.

[4] J. Tompson, M. Stein, Y. LeCun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Trans. Graph.*, vol. 33, no. 5, 2014, Art. no. 169.

[5] M. Oberweger, P. Wohlhart, and V. Lepetit, "Training a feedback loop for hand pose estimation," in *Proc. Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3316–3324.

[6] M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit, "Efficiently creating 3D training data for fine hand pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4957–4965.

[7] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using Kinect sensor," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1110–1120, Aug. 2013.

[8] H. Guan, R. S. Feris, and M. Turk, "The isometric self-organizing map for 3D hand pose estimation," in *Proc. 7th Int. Conf. Autom. Face Gesture Recognit.*, Apr. 2006, pp. 263–268.

[9] C. Keskin, F. Kıraç, Y. Kara, and L. Akarun, "Hand pose estimation and hand shape classification using multi-layered randomized decision forests," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 852–863.

[10] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 824–832.

[11] D. Tang, T.-H. Yu, and T.-K. Kim, "Real-time articulated hand pose estimation using semi-supervised transductive regression forests," in *Proc. Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3224–3231.

[12] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim, "Latent regression forest: Structured estimation of 3D articulated hand posture," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3786–3793.

[13] 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 *Proc. Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3325–3333.

[14] P. Li, H. Ling, X. Li, and C. Liao, "3D hand pose estimation using randomized decision forest with segmentation index points," in *Proc. Int. Conf. Comput. Vis.*, Dec. 2015, pp. 819–827.

[15] C. Choi, A. Sinha, J. H. Choi, S. Jang, and K. Ramani, "A collaborative filtering approach to real-time hand pose estimation," in *Proc. Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2336–2344.

[16] A. Sinha, C. Choi, and K. Ramani, "DeepHand: Robust hand pose estimation by completing a matrix imputed with deep features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4150–4158.

[17] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands deep in deep learning for hand pose estimation," in *Proc. Comput. Vis. Winter Workshop*, 2015, pp. 21–30.

[18] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei, "Model-based deep hand pose estimation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 2421–2427.

[19] Q. Ye, S. Yuan, and T.-K. Kim, "Spatial attention deep net with partial PSO for hierarchical hybrid hand pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 346–361.

[20] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5679–5688.

[21] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Real-time 3D hand pose estimation with 3D convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, Apr. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8338122/

[22] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1653–1660.

[23] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 648–656.

[24] T. Pfister, J. Charles, and A. Zisserman, "Flowing convnets for human pose estimation in videos," in *Proc. Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1913–1921.

[25] J. S. Supancic, III, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan, "Depth-based hand pose estimation: data, methods, and challenges," in *Proc. Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1868–1876.

[26] A. Tagliasacchi, M. Schröeder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly, "Robust articulated-ICP for real-time hand tracking," *Comput. Graph. Forum*, vol. 34, no. 5, pp. 101–114, 2015.

[27] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Efficient model-based 3D tracking of hand articulations using Kinect," in *Proc. 22nd Brit. Mach. Vis. Conf.*, 2011, pp. 101.1–101.11.

[28] M. de La Gorce, D. J. Fleet, and N. Paragios, "Model-based 3D hand pose estimation from monocular video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1793–1805, Sep. 2011.

[29] L. Ballan, A. Taneja, J. Gall, L. V. Gool, and M. Pollefeys, "Motion capture of hands in action using discriminative salient points," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 640–653.

[30] J. Chen, S. Nie, and Q. Ji, "Data-free prior model for upper body pose estimation and tracking," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4627–4639, Dec. 2013.

[31] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, "Realtime and robust hand tracking from depth," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1106–1113.

[32] M. Ding and G. Fan, "Articulated and generalized Gaussian kernel correlation for human pose estimation," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 776–789, Feb. 2016.

[33] M. Ye, Y. Shen, C. Du, Z. Pan, and R. Yang, "Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1517–1532, Aug. 2016.

[34] C. Wan, A. Yao, and L. Van Gool, "Direction matters: Hand pose estimation from local surface normals," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 1–16.

[35] Y. Liu, P. Lasang, M. Siegel, and Q. Sun, "Geodesic invariant feature: A local descriptor in depth," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 236–248, Jan. 2015.

[36] F. Kirac, Y. E. Kara, and L. Akarun, "Hierarchically constrained 3D hand pose estimation using regression forests from single frame depth data," *Pattern Recognit. Lett.*, vol. 50, pp. 91–100, Dec. 2014.

[37] H. Liang, J. Yuan, and D. Thalmann, "Resolving ambiguous hand pose predictions by exploiting part correlations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 7, pp. 1125–1139, Jul. 2015.

[38] P. Dollár, P. Welinder, and P. Perona, "Cascaded pose regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1078–1085.

[39] T. Sharp *et al.*, "Accurate, robust, and flexible real-time hand tracking," in *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst.*, 2015, pp. 3633–3642.

[40] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt, "Fast and robust hand tracking using detection-guided optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3213–3221.

[41] S. Sridhar, F. Mueller, M. Zollhöfer, D. Casas, A. Oulasvirta, and C. Theobalt, "Real-time joint tracking of a hand manipulating an object from RGB-D input," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 294–310.

[42] A. Jain, J. Tompson, M. Andriluka, G. W. Taylor, and C. Bregler. (2014). "Learning human pose estimation features with convolutional networks." [Online]. Available: https://arxiv.org/abs/1312.7302

[43] J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *Proc. Neural Inf. Process. Syst.*, 2014, pp. 1799–1807.

[44] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. Int. Conf. Comput. Vis.*, Dec. 2015, pp. 945–953.

[45] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5648–5656.

[46] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang, and P. O. Ogunbona, "Action recognition from depth maps using deep convolutional neural networks," *IEEE Trans. Human Mach. Syst.*, vol. 46, no. 4, pp. 498–509, Aug. 2016.

[47] J. M. Van Verth and L. M. Bishop, *Essential Mathematics for Games and Interactive Applications: A Programmer's Guide*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann, 2008.

[48] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshop*, Jun. 2010, pp. 9–14.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[50] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[51] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 2146–2153.

[52] J. Xu, J. Yuan, and Y. Wu, "Multimodal partial estimates fusion," in *Proc. Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 2177–2184.

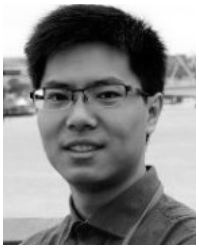
[53] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1297–1304.

[54] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon, "The Vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 103–110.

[55] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A MATLAB-like environment for machine learning," in *Proc. Neural Inf. Process. Syst. Workshop, BigLearn*, 2011, pp. 1–6.

[56] S. Pellegrini, K. Schindler, and D. Nardi, "A generalization of the ICP algorithm for articulated bodies," in *Proc. Brit. Mach. Vis. Conf.*, 2008, pp. 1–10.
[57] Z. Tu *et al.*, "Fusing disparate object signatures for salient object detection in video," *Pattern Recognit.*, vol. 72, pp. 285–299, Dec. 2017.
[58] X. Wang *et al.*, "Greedy batch-based minimum-cost flows for tracking multiple objects," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4765–4776, Oct. 2017.
[59] A. Maksai, X. Wang, F. Fleuret, and P. Fua, "Non-Markovian globally consistent multi-object tracking," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2563–2573.
[60] X. Wang, E. Türetken, F. Fleuret, and P. Fua, "Tracking interacting objects using intertwined flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2312–2326, Nov. 2016.
[61] A. Maksai, X. Wang, and P. Fua, "What players do with the ball: A physically constrained interaction modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 972–981.

Liuhao Ge received the B.Eng. degree in detection guidance and control technology from the Nanjing University of Aeronautics and Astronautics in 2011 and the M.Eng. degree in control theory and engineering from Southeast University in 2014. He is currently pursuing the Ph.D. degree with the Institute for Media Innovation, Interdisciplinary Graduate School, Nanyang Technological University, Singapore. His research interests mainly include computer vision and machine learning.



Hui Liang received the B.Eng. degree in electronics and information engineering and the M.Eng. degree in communication and information system from the Huazhong University of Science and Technology in 2008 and 2011, respectively, and the Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 2016. He was a research Scientist with the Institute of High Performance Computing, A*STAR, Singapore, a Research Associate with the Institute for Media Innovation, and a Research Fellow with the Rapid-Rich Object Search Laboratory, NTU. He is currently a Research Scientist with Amazon, USA. His research interests mainly include computer vision, machine learning, and human–computer interaction.

Junsong Yuan received the B.Eng. degree from the Special Class for the Gifted Young of Huazhong University of Science and Technology, China, in 2002, the M.Eng. degree from the National University of Singapore, and the Ph.D. degree from Northwestern University. He was an Associate Professor with the School of Electrical and Electronics Engineering, Nanyang Technological University (NTU), Singapore. He is currently an Associate Professor with the Computer Science and Engineering Department, University at Buffalo, The State University of New York, USA. He has authored over 200 papers in computer vision, pattern recognition, and multimedia. He is a fellow of the International Association of Pattern Recognition. He received the 2016 Best Paper Award from IEEE TRANSACTIONS ON MULTIMEDIA, the Nanyang Assistant Professorship from NTU, and the Outstanding EECS Ph.D. Thesis Award from Northwestern University. He is the program co-chair of ICME 2018 and VCIP 2015 and the Area Chair of the ACM MM 2018, ACCV 2014 and 2018, ICPR 2016 and 2018, CVPR 2017, and ICIP 2017 and 2018. He served as a guest editor of the *International Journal of Computer Vision*. He is currently a senior area editor of the *Journal of Visual Communication and Image Representation*, an associate editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

Daniel Thalmann received the Ph.D. degree in computer science from the University of Geneva in 1977 and an Honorary Doctorate from Paul-Sabatier University, Toulouse, France, in 2003. He has been the Founder of the Virtual Reality Laboratory, École Polytechnique Fédérale de Lausanne (EPFL), and a Visiting Professor with the Institute for Media Innovation, Nanyang Technological University, Singapore. He is currently an Honorary Professor with EPFL, Switzerland, and the Director of Research Development, MIRALab Sarl. He is a pioneer in research on virtual humans. He has authored over 600 papers in graphics, animation, and virtual reality. His current research interests include real-time virtual humans in virtual reality, crowd simulation, and 3D interaction. He is a member of the editorial board of 12 other journals. He received the Eurographics Distinguished Career Award in 2010, the 2012 Canadian Human Computer Communications Society Achievement Award, and the CGI 2015 Career Achievement. He was the program chair and a co-chair of several conferences, including the IEEE-VR, ACM-VRST, and ACM-VRCAI. He is a co-editor-in-chief of the *Journal of Computer Animation and Virtual Worlds*.