3D CNN HAND POSE ESTIMATION WITH END-TO-END HIERARCHICAL MODEL AND PHYSICAL CONSTRAINTS FROM DEPTH IMAGES

Z.Z. Xu[*], *W.J. Zhang*[†]

Abstract: Previous studies are mainly focused on the works that depth image is treated as flat image, and then depth data tends to be mapped as gray values during the convolution processing and features extraction. To address this issue, an approach of 3D CNN hand pose estimation with end-to-end hierarchical model and physical constraints is proposed. After reconstruction of 3D space structure of hand from depth image, 3D model is converted into voxel grid for further hand pose estimation by 3D CNN. The 3D CNN method makes improvements by embedding end-to-end hierarchical model and constraints algorithm into the networks, resulting to train at fast convergence rate and avoid unrealistic hand pose. According to the experimental results, it reaches 87.98 % of mean accuracy and 8.82 mm of mean absolute error (MAE) for all 21 joints within 24 ms at the inference time, which consistently outperforms several well-known gesture recognition algorithms.

Key words: *hand pose estimation, 3D CNN, depth image, hierarchical model, physical constraints*

Received: October 1, 2020 **DOI:** 10.14311/NNW.2023.33.003
Revised and accepted: February 28, 2023

1. Introduction

In order to display and simulate the details of hand action, hand pose estimation algorithm is required to locate all 21 joints in the hand model with high efficiency and precision from a depth image, which is essential for the development of human-computer interaction (HCI) technical solution. Since the mapping between depth image and articulated hand gesture is highly nonlinear and difficult, it remained a challenging task to build an efficient, reliable and functional gesture recognition and control system.

Previous studies are mainly focused on the works that depth image is treated as optical flat image, and then depth data tends to be mapped as gray values during

[*] Zhengze Xu; Department of Communication, East China Normal University, Shanghai 200241, China

[†] Wenjun Zhang – Corresponding author; College of Information Technology, Shanghai Jian-Qiao University, Shanghai 201306, China, E-mail: 18096@gench.edu.cn

the processing and features extraction. Labeled pixels are merged to generate joints location by clustering algorithm such as mean shift, after pixels are classified into hand parts by random decision forests (RDF) [1]. Identification of joints by direct regression from Hough forests [2] has also been tried to obtain the optimized final results.

Thanks to the popularity of convolutional neural networks (CNN), substantial progress towards highly precise hand pose estimation has been made. Many algorithms have been proposed in the literature expanded from single depth map to depth sequence. DeepJoint [3] detects hand joints by Gaussian heat-map as middle representative, which down-scaling to 18×18 by max-pooling reduction is an obstacle for high precision. Moreover, z-coordinate of joint position is deduced by depth value after identification in 2D map is especially difficult for the occluded joint to obtain. DeepPrior [4] embeds a bottleneck structure into the fully connected layer to force the model to learn "pose prior" in lower dimensional space, and then refinement with overlapping regions (ORRef) is applied to fine-tuning the final outputs. DeepPrior++ [5] boost the performance of DeepPrior network by data augment, refined hand localization and residual network block.

Gesture recognition based on spatio-temporal data from continuous depth sequence has also made a new direction in the recent research. 3D CNN is introduced for VIVA challenge dataset to [6] combine information from multiple adjacent frames. Convolutional recurrent neural network module (CRNN) [7] which is the integration of convolutional neural network (CNN) and recurrent neural network (RNN) shows that it makes great improvements of the accuracy in the hand pose estimation experiment. Pavlo et al. use a unified recurrent three-dimensional convolutional neural network (R3DCNN) architecture [8] to segment and recognize hand action with connectionist temporal classification (CTC) as loss function.

Although depth image is displayed in 2D formation, pixels captured by depth sensor are able to provide 3D spatial structure of captured objects in the scene, which is essentially different from the traditional optical image. Many researchers come to realize that images captured by the same hand gesture are co-related in multi-view projection, thus it is possible to utilize or approximate the hand's appearance from a different viewpoint by learning its latent representation. Ge et al. [9] first convert image into 3D point cloud, then project them onto three orthogonal planes and each one is fed into a different CNN to regress for 2D heat-maps. Lastly, they are fused to output final 3D coordinates of hand pose estimation. Poier et al. implement the observations by a CNN with encoder-decoder architecture [10], which consistently surpasses the counterparts with learned latent representations.

More and more recent researches are focused on algorithms directly working in 3D space which are beyond the concept that depth image is just the projection of hands on planar space. 3D volumetric representations are generated by truncated signed distance function (TSDF) after 3D reconstruction of hand depth image into 3D point cloud [11]. Then they are fed into the 3D CNN to learn the mapping between 3D volumes and 3D joint locations. V2V-PoseNet [12] first converts 3D point cloud to 3D voxel grid, and then processed it with 3D CNN model to produce a 3D probability density map for each joint. At inference time, detection of each joint on 3D heat-maps expanded from 2D heat-maps similar in DeepJoint [3] boosts the performance significantly. Compared from 3D volumetric representations produced by TSDF and voxel grid, Ge et al. [13] inputs 3D point cloud into 3D CNN directly with oriented bounding box (OBB) to transform the original hand into canonical shape which is robust to variations in global orientation.

With 2D CNN substituted by 3D CNN, methods commonly outperform in the precision of the hand pose estimation in the following reasons. 1) 2D CNN are not suitable for the prediction of 3D coordinates due to the lack of spatial information. 2) The mapping between 3D coordinates and a 2D image is highly nonlinear, which is difficult for training. 3) Position in z-coordinate deduced by depth value hurts the performance of occluded joints.

An approach of 3D CNN hand pose estimation with end-to-end hierarchical model and physical constraints is proposed. Reconstruction of hand into 3D point cloud makes recognition model focused on the 3D spatial structure, where depth information is fully utilized. Afterward, 3D model is converted into voxel grid for further gesture recognition and pose estimation by regression-based 3D CNN. Although direct regression of the 3D coordinates is less accurate than heat-map detection, it achieves better performance after adding hierarchical and physical constraint algorithms.

The contributions in this thesis can be mainly summarized as follows.

Firstly, hierarchical cascaded model is normally deployed in previous 2D CNN method, but generally it is hard to apply in the 3D CNN due to huge scale of its framework or long-latency caused by multi-stage operation. Conversions from sparse point cloud to dense voxel grid and shared convolutional layers into the networks are both help to reduce the dimension of input data in large scale. End-to-end hierarchically structured CNN architecture is an alternative to multi-stage method in order that shorten the latency of the estimation is essential to HCI solution. Different hierarchical architectures have also been tested in the experiment to find the best case scenario.

Secondly, explicit spatial constraints by embedded into the loss function in order to rectify the estimated pose which does not properly fit with appearance, and they perform especially when occlusion or self-occlusion occurs. We presume joints on the same finger are supposed to be collinear or coplanar. Thus we formulate the global loss function along with these constraints applied to hand joints during the training process in the case of violation of this condition. To our knowledge, this is the first work to integrate 3D CNN with end-to-end hierarchical model and physical constraints for hand pose estimation method.

2. Method

A. 3D voxel grid To begin with, after hand is segmented from depth image, reconstruction of 3D point cloud from hand region is shown in Eq. 1, where I_{w} is width of the image, I_{h} is height of the image and f is defined as focal length of the camera. They are 320, 240 and 241.42 respectively in Intel® RealSense$^{\mathrm{TM}}$ sensor.

$$\begin{aligned} x &= \left(u - \frac{I_{\mathrm{w}}}{2}\right) \times d_{(u,v)} \times \frac{1}{f} \\ y &= \left(\frac{I_{\mathrm{h}}}{2} - v\right) \times d_{(u,v)} \times \frac{1}{f} \\ z &= d_{(u,v)} \end{aligned} \tag{1}$$

Directly processing 3D point cloud without proper reduction in 3D CNN is high consumption on memory resources and less efficiency in computation. In the experiment, we adopt GPU as hardware device during the training and inference, whose memory size is 6–12 GB in common. To avoid memory exhaustion due to large storage in tensor by sparse raw point cloud, voxel is introduced from Voxnet [14] and VoxelNet [15] aimed to reduce the number of inputs and increase computational efficiency. Voxel which is a discretized binary variable represents a basic unit of graphic information in 3D space compare to pixel in 2D plane. Voxelized grid is much faster than TSDF [16] or D-TSDF [11] algorithm on CPU to shorten the latency in the pre-process stage of hand recognition.

Each cloud point is mapped to discrete voxel coordinates by occupancy grid models as shown in Eq. 2. $V_{(i,j,k)}$ is defined as 1 if the voxel is occupied by any depth point and 0 otherwise. Volumetric representation of hand has a fixed voxel size in our model. The resolution is chosen by $80 \times 80 \times 80$ in order to avoid either loss of detail information when the grid size is too small or enlargement of computational cost when the grid size is too big.

$$V_{(i,j,k)} = \begin{cases} 0, & Points_{(x,y,z)} = 0 \\ 1, & Points_{(x,y,z)} \geq 1 \end{cases} \tag{2}$$

The pipeline of 3D volumetric representation prepared for the next stage of 3D CNN processing from depth datasets is visualized in Fig. 1.

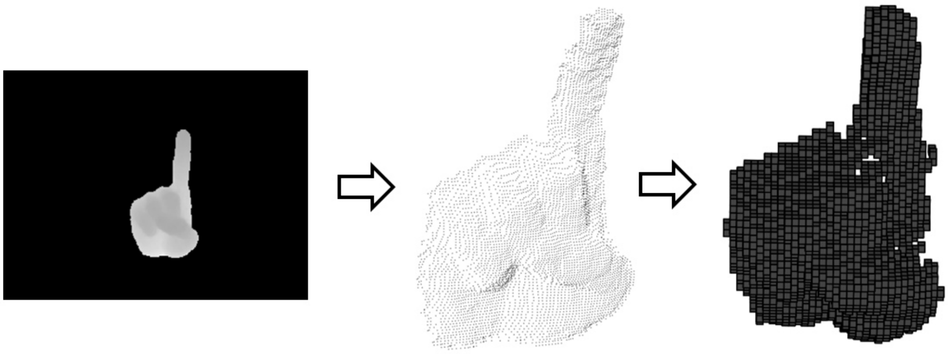


Fig. 1 *Reconstruction of 3D point cloud from hand region in depth map, then be converted into* $80 \times 80 \times 80$ *size of 3D voxel grids by occupancy models.*

B. Hierarchical method Our hand model has 21 joints as the same number and position as the hand labeled in MSRA15 datasets [17], including TIP, DIP (distal interphalangeal point), PIP (proximal interphalangeal point), MCP (metacarpophalangeal point) on each finger and palm keypoint. Different from other fingers, MCP on thumb is much close to the palm.

It is confirmed in [18] that single direct regression model has difficulty in the prediction of the whole 21 joints on hand model, since it may has insufficient capacity to learn the complex variations of hand. Therefore, deployment of a hierarchically structured model which has multiple branches for global features and local features is highly recommended to achieve better performance for 3D articulated hands.

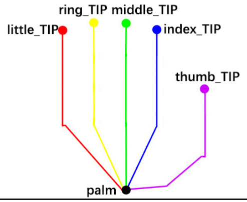
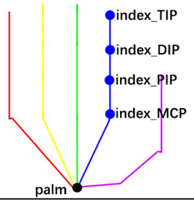
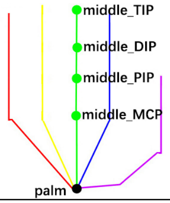
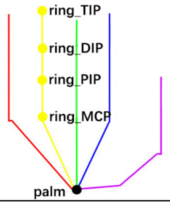
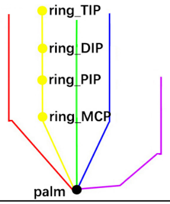
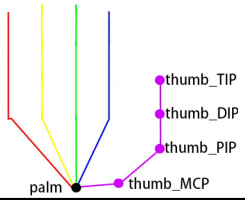
Hierarchical method divides the gesture recognition problem into sub-tasks such as hand joints by finger in order that global hand pose can be separated into a serial of local poses. Solving such complex problems in multi-stage by cascaded method is previously used in RDF and CNN, which reduces the nonlinearity of global hand pose. Cascaded method from coarse to fine is heavily time consuming although it works better than single-stage method during the inference since it iteratively updates results from previous stages. Training the network in an end-to-end way and fusing the outputs of each branch into the final results once for all can significantly improve the real-time performance of pose estimation algorithm.

Thus in our model hand pose estimation from holistic regression is replaced by end-to-end hierarchically structured approach, which is divided into one global hand pose branch and five local hand pose branches. Global branch composed of fingertips and palm joint is focused on the recognition of basic hand gesture, while five local branches each one is concentrated on the estimation of each finger. Overlapped joints such as six key joints will be predicted in multiple values since each branch outputs once during the inference. Obtaining the final result by computing average value of them will further improve the accuracy of the hierarchical model.

Detail solution shown in Tab. I. reveals fast convergence during the training in the experiment. Our solution defining six key joints as representative of basic hand gesture achieves better performance than previous method which utilizes the whole 21 joints as global hand pose.

C. Physical constraints Structure constraints between correlated joints and fingers have been introduced in many works of literature [19, 20] to estimate pose of articulated hands. It has been proved in [21] that unrealistic hand pose are avoided by the application of explicit modeling of physical constraints and spatial relation. Previous studies are mainly focused on the traditional approach when physical constraints are applied in pose verification and error recovery after formal hand recognition. By adding penalties into loss function when physical constraints of hand are violated during the learning stage, such constraints can be integrated into the holistic estimation model by end-to-end training.

There are two types of constraints in our spatial relation model, collinear or coplanar from key joints on each finger. According to the spatial relation of these joints, loss function can be computed in three different ways, and they are described as follows.

Branch name	Global or local hand pose	Joints in the branch	Illustrate of joints' location	Number of joints
Palm & tips	Global hand pose	palm index_TIP middle_TIP ring_TIP little_TIP thumb_TIP	 ring_TIP middle_TIP little_TIP index_TIP thumb_TIP palm	6
Palm & index	Local hand pose	palm index_MCP index_PIP index_DIP index_TIP		5
Palm & middle	Local hand pose	palm middle_MCP middle_PIP middle_DIP middle_TIP	 middle_TIP middle_DIP middle_PIP middle_MCP palm	5
Palm & ring	Local hand pose	palm ring_MCP ring_PIP ring_DIP ring_TIP		5
Palm & little	Local hand pose	palm little_MCP little_PIP little_DIP little_TIP		5
Palm & thumb	Local hand pose	palm thumb_MCP thumb_PIP thumb_DIP thumb_TIP		5

Tab. I *Hierarchical method by branching strategy in detail.*

1) Collinear relation from any three different joints belonging to the same finger can be checked by Eq. 3 where thr_1 is set to 0.02, since the linear distance between two points is the shortest.

$$\|P_1P_2\| + \|P_2P_3\| \leq \|P_1P_3\| \times (1 + thr_1) \tag{3}$$

If given joints P_1, P_2, P_3 are considered to be collinear from ground truth joints labeled in depth image datasets, then we compute vector $v = \overrightarrow{P_1P_3}$ as the direction of this line and also its unit vector $e = \frac{\overrightarrow{P_1P_3}}{\|\overrightarrow{P_1P_3}\|}$. Vector v' and its unit vector e' are calculated in the same way by the predicted location of these three joints from the networks. Two lines with respect to ground truth and inferred joints should be in same direction when the recognition model is able to output accurate joints' position in 3D space. If they meet this condition, the result of dot product $e \cdot e' = \|e\| \, \|e'\| \cos\theta$ is supposed to be close to 1. So the loss for collinear joints is defined in Eq. 4 where thr_2 is set to 0.95 in the experiment.

$$C_{\text{collinear}} = thr_2 - \min(thr_2, e \cdot e') \tag{4}$$

2) If neither ground truth joints nor inferred joints are considered as collinear by Eq. 3, we try to check coplanar spatial relation for all subsets of three joints in a finger. Under this situation, direction of line is replaced by the plane normal vector of these three joints through cross product $N = \overrightarrow{P_1P_2} \times \overrightarrow{P_2P_3}$ and its unit vector N_e by ground truth joints or $N_e{}'$ by inferred joints. Plane normal vector from ground truth joints must be parallel to the inferred one if dot product of $N_e \cdot N_e{}'$ is close to 1, which shows the estimation model can acquire quite precise outputs. Therefore, the loss functions for coplanar joints are computed in the similar way as shown in Eq. 5.

$$C_{\text{coplanar}} = thr_2 - \min(thr_2, N_e \cdot N_e{}') \tag{5}$$

3) Apparently, loss value should be a large number in the situation that ground truth joints are not collinear, while inferred joints checked by Eq. 3 are considered to be collinear. Theoretically cross product of $\overrightarrow{P_1P_2}$ and $\overrightarrow{P_2P_3}$ should be zero if P_1, P_2, P_3 are collinear, but it usually is a very small value in practical application. The influence of this small miscalculation will be certainly introduced into the constraints method when computing the unit vector of plane normal vector due to its length is amplified to 1. To avoid this issue, the result of $N_e \cdot N_e{}'$ is set to zero directly under the circumstance in practice, then the final result of loss function is thr_2 by Eq. 5.

D. 3D CNN architecture 3D residual block extended from 2D residual block in resnet [22] is deployed in 3D CNN architecture as basic convolution processing unit. To further enhance the performance, hybrid dilated convolution framework [23] (HDC) is introduced for preventing gridding effect and increasing receptive field. Each 3D residual block is comprised of $r = 1$ and $r = 2$ dilated convolution layer as shown in Fig. 2, which attains accurate prediction while reducing the computational cost.

Our 3D CNN architecture by hierarchically structured model as shown in Fig. 3, first utilizes original 3D CNN layer in the beginning stage with kernel size at 7 to significantly decrease the cost and memory resources in computation. Next it shares features in two earlier network layers by using same 3D residual blocks. In the end, main branch is divided into six sub-branches of convolution blocks and fully connected (FC) layers to regress global and local pose jointly.

Each branch learns more specific features for basic hand gesture or each finger by optimizing network parameters in order to minimize the loss function for global and local hand pose between predicted values and ground truth values of hand joints.

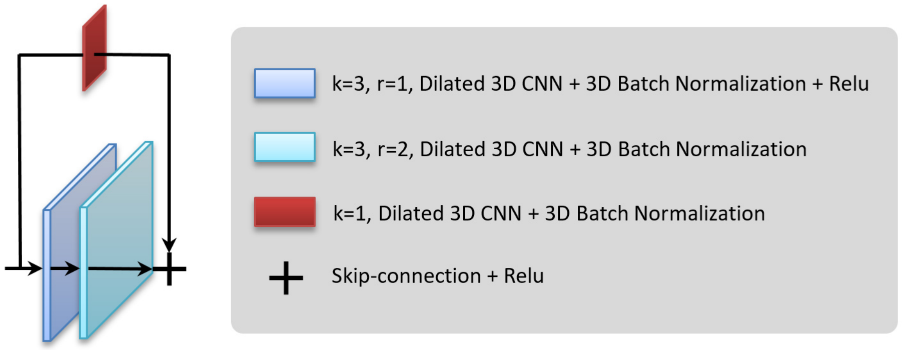

k=3, r=1, Dilated 3D CNN + 3D Batch Normalization + Relu
k=3, r=2, Dilated 3D CNN + 3D Batch Normalization
k=1, Dilated 3D CNN + 3D Batch Normalization
Skip-connection + Relu

Fig. 2 *3D residual block by HDC.*

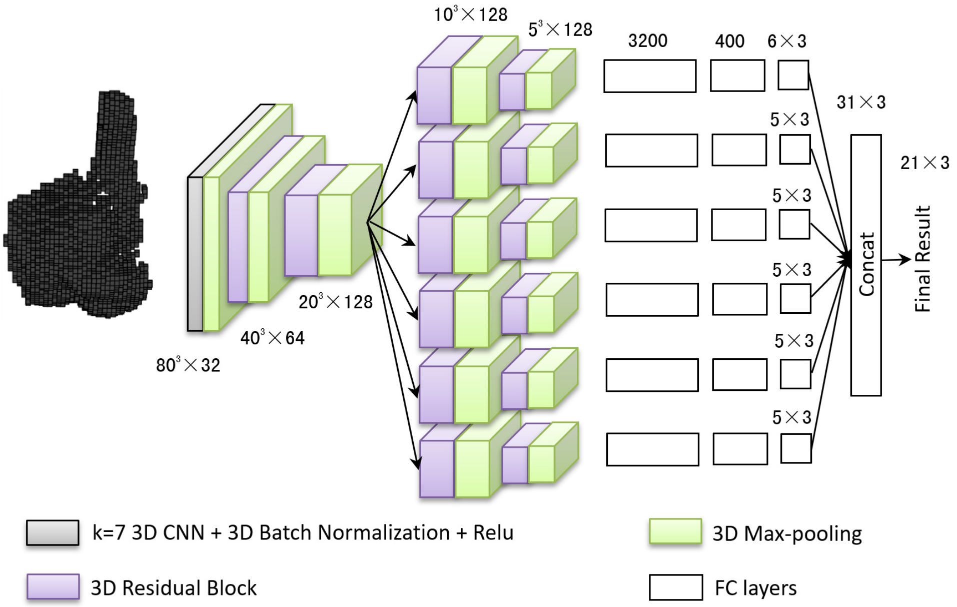


Fig. 3 *3D CNN architecture by hierarchically structured model.*

Loss function is defined in Eq. 6 where α is set to 15 to enlarge the penalty values of physical constraints. Each component of all cost is explained in the following three equations.

$$
\begin{aligned}
C &= C_{\text{global}} + C_{\text{fingers}} + \alpha \times C_{\text{constraints}} \\
C_{\text{global}} &= \frac{1}{6}\sum_{i=1}^{6} \left\| P_i\left(x,y,z\right) - \tilde{P}_i\left(x,y,z\right) \right\|^2 \\
C_{\text{fingers}} &= \frac{1}{5}\sum_{f=1}^{5}\sum_{j=1}^{5} \left\| P_{f,j}\left(x,y,z\right) - \tilde{P}_{f,j}\left(x,y,z\right) \right\|^2 \\
C_{\text{constraints}} &= \sum_{m=1}^{M} C_{\text{collinear}} + \sum_{n=1}^{N} C_{\text{coplanar}}
\end{aligned}
\tag{6}
$$

3. Experiment

A. Experiment setup MSRA15 [17] hand depth datasets labeled 21 accurate ground truth joints' position in 3D space satisfy the requirements according to our hand model, which captured 76375 frames of hand depth images with resolution at 320 × 240 from 9 subjects. Because training 3D CNN model is by far the most time consuming part, a 6-core CPU and a NVIDIA RTX 2070 GPU are employed to keep the training time practical. We execute the experiments under Linux and GPU-based PyTorch 1.5.0 resulting in a huge speed boost.

The network is trained with mini-batch gradient descent and Adam optimization algorithm, and decay learning rate strategy is also introduced. At the end of experiments, fine training 3D CNN with end-to-end hierarchical model and constraints algorithm to 20 epochs converges our estimation network.

B. Experiment results Mean 3D distance error to each joint and success rate of all joints on different error thresholds are applied as two evaluation metrics in the experiment. For better comparison, other pose estimation methods are also tested in the experiment, including 3D heat-map detection, 3D heat-map detection by a hierarchical model, direct holistic regression as baseline1, baseline2 and baseline3 respectively. To our knowledge, any end-to-end hierarchical model for 3D heat-map detection can be only executed by considerable enlargement of complexity of the model, thus making inference in real-time infeasible. Physical constraints by adding penalties into loss function can only be applied in direct regression model, since the outputs of 3D heat-map are only middle representatives to final result of joints' location.

Firstly, we compute Euclidean distance between predicted coordinates and ground truth coordinates for per hand joint in 3D space. The sequence numbers in Fig. 4 from 1 to 21 corresponding to each joint are arranged in order as follows: palm, index_MCP, index_PIP, index_DIP, index_TIP, middle_MCP, middle_PIP, middle_DIP, middle_TIP, ring_MCP, ring_PIP, ring_DIP, ring_TIP, little_MCP, little_PIP, little_DIP, little_TIP, thumb_MCP, thumb_PIP, thumb_DIP, thumb_TIP.

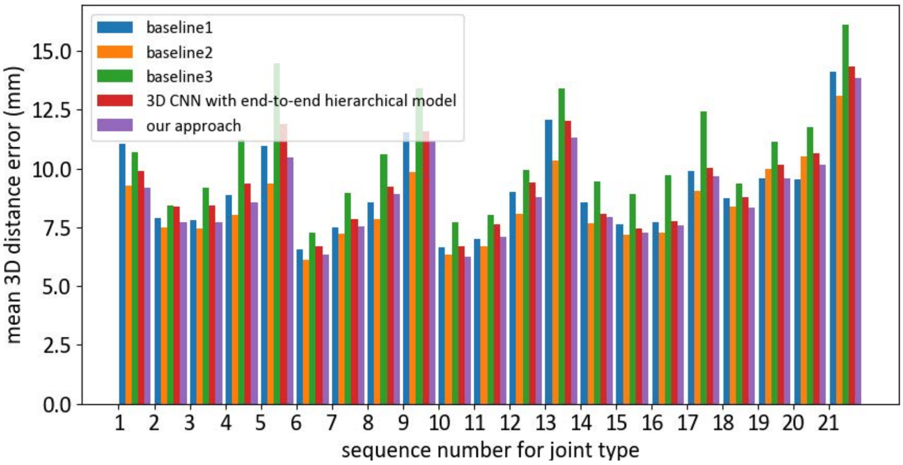


Fig. 4 *Mean 3D distance error of per joint on MSRA15.*

Secondly, success rate of all joints on different error thresholds is recorded to draw the curve in Fig. 5. Since distance between adjacent joints is commonly larger than 15 mm, error threshold set to 15 mm is reasonable. The success rate within the threshold is up to 87.98 % in our method.

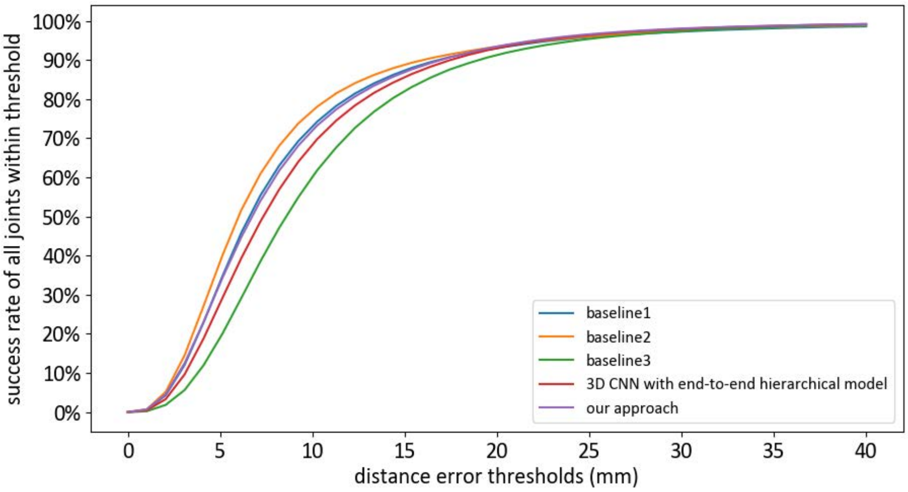


Fig. 5 *Success rate of all joints for different thresholds.*

Moreover, we compare state-of-the-art approaches between 2D CNN and 3D CNN with respect to ours for average 3D error of all joints in Tab. II. Comparison between the proposed method and existing 3D detection methods is also shown in Tab. III. Both of them show that our 3D CNN approach outperforms others except for baseline2.

Approaches of 2D CNN	Average 3D error [mm]
Sun et al. [18]	15.2
Ge et al. [9]	13.2
DeepPrior++ [5]	9.5

Approaches of 3D CNN	Average 3D error [mm]
Ge et al. [11]	9.6
Our approach	**8.82**

Tab. II *Average 3D error on MSRA15 between 2D CNN and 3D CNN.*

Approaches	Average 3D error
3D heat-map detection (baseline1)	9.12 mm
3D heat-map detection by a hierarchical model (baseline2)	8.43 mm
direct holistic regression (baseline3)	10.58 mm
3D CNN with end-to-end hierarchical model	9.35 mm
3D CNN with end-to-end hierarchical model and physical constraints (our approach)	**8.82 mm**

Tab. III *Average 3D error of 3D detection methods on MSRA15.*

In addition, duration time is also recorded in Tab. IV. Considerable enlargement of complexity of the hierarchically structured model increases the intensity and duration both in training and testing. The application of physical constraints with penalties into loss function which is only performed during the training has no effects on the duration time for testing. Most of methods can execute hand pose estimation in real time except for baseline2 which is not fast enough to fulfill the task for the depth image sequences at 25 frames per second.

	Baselines	Training	Testing
1	baseline1	52 hours	30 ms
2	baseline2	190 hours	108 ms
3	baseline3	10 hours	10.8 ms
4	3D CNN with end-to-end hierarchical model	30 hours	22.3 ms
5	3D CNN with end-to-end hierarchical model and physical constraints (our approach)	40 hours	22.3 ms

Tab. IV *Duration time for training and testing.*

C. Discussion Hand pose estimation methods have been evaluated as shown in Fig. 4, Fig. 5 and Tab. II, Tab. III. Although baseline2 attains highest accuracy in our experiments, it is not suitable for practical deployment in real application due to its long-latency in the inference. Our approach achieves better performance in high-accuracy than other benchmarks within real-time.

3D CNN achieves higher precision than 2D CNN method for it makes full utilize of spatial information provided by depth values. Our approach makes great improvements when hierarchically structured model is applied to fuse the outputs of each branch into the final results. Physical constraints algorithm is only performed in direct regression model which is able to avoid unrealistic hand pose as shown in Fig. 6.

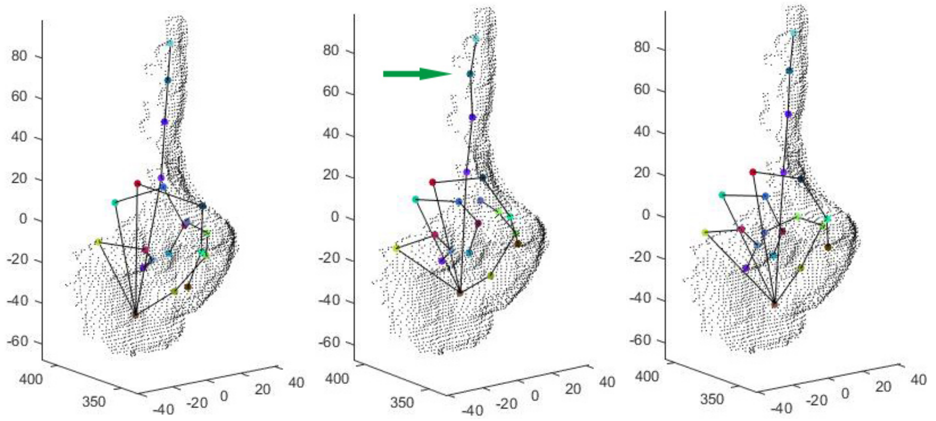


Fig. 6 *Hand skeleton model by ground truth (left), by 3D heat-map detection without physical constraints (middle), direct regression with physical constraints (right).*

As shown in Fig. 7, our 3D CNN approach is able to acquire accurate hand shape in various typical poses and orientations from depth image.

4. Conclusions and future works

In this paper, we have proposed an optimized 3D CNN approach for hand pose estimation to create hand skeleton model by all 21 joints in 3D space, which is essential to display and simulate the details of hand shape and action. It is observed that our 3D CNN approach with end-to-end hierarchical model and physical constraints achieves higher performance than most of existing baselines. Within short processing time, our approach makes hand pose estimation performed in real-time (≤ 24 ms on RTX 2070 GPU).

In the future, our approach is capable of application in the LiDAR-based hand pose estimation, which has the advantage for its longer effective sensing distance. Depth image datasets with precisely labeled joints are first utilized to pre-train the gesture recognition model due to their ease of obtainment, then we plan to fine-tuning the model by small amounts of training 3D points data captured from LiDAR, and expect similar conclusions could be made.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

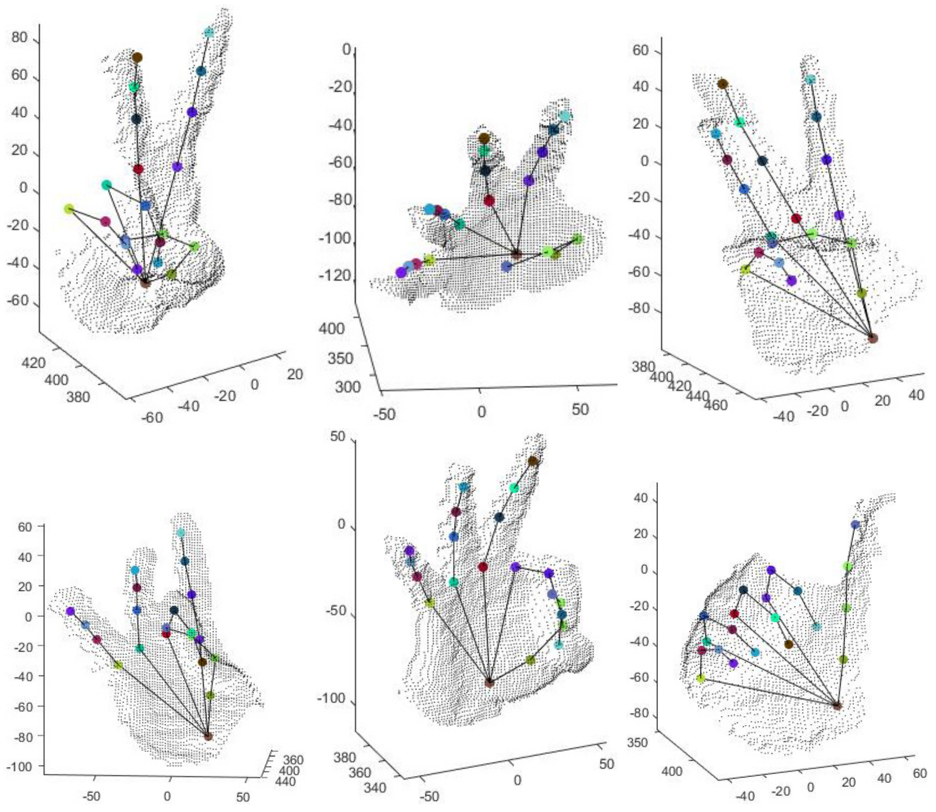


Fig. 7 *Examples of our experiment results.*

Acknowledgement

The research did not receive specific funding, but was performed as part of the employment of the authors, their employer is East China Normal University and Shanghai JianQiao University.

References

[1] KESKIN C., KIRAC F., KARA Y.E., AKARUN L. Real time hand pose estimation using depth sensors. In: *ICCVW*, 2011, pp. 1228–1234.

[2] XU C., CHENG L. Efficient Hand Pose Estimation from a Single Depth Image. In: *ICCV*, 2013, pp. 3456–3462.

[3] TOMPSON J., STEIN M., LECUN Y., PERLIN K. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 2014, 33(5).

[4] OBERWEGER M., WOHLHART P., LEPETIT V. Hands Deep in Deep Learning for Hand Pose Estimation. In: *CVWW*, 2015.

[5] OBERWEGER M., LEPETIT V. DeepPrior++: Improving Fast and Accurate 3D Hand Pose Estimation. In: *ICCVW*, 2017, 1, pp. 585–594.

[6] MOLCHANOV P., GUPTA S., KIM K., KAUTZ J. Hand Gesture Recognition with 3D Convolutional Neural Networks. In: *CVPR*, 2015.

[7] HU Z., HU Y., LIU J., WU B., HAN D., KURFESS T. A CRNN module for hand pose estimation. *Neurocomputing*, 2019, 333, pp. 157–168.

[8] MOLCHANOV P., YANG X., GUPTA S., KIM K., TYREE S., KAUTZ J. Online Detection and Classification of Dynamic Hand Gestures with Recurrent 3D CNN. In: *CVPR*, 2016, pp. 1063–6919.

[9] GE L., LIANG H., YUAN J., THALMANN D. Robust 3D Hand Pose Estimation in Single Depth Images: from Single-View CNN to Multi-View CNNs. In: *CVPR*, 2016, pp. 3593–3601.

[10] POIER G., SCHINAGL D., BISCHOF H. Learning pose specific representations by predicting different views. In: *CVPR*, 2018, pp. 60–69.

[11] GE L., LIANG H., YUAN J. 3D Convolutional Neural Networks for Efficient and Robust Hand Pose Estimation from Single Depth Image. In: *CVPR*, 2017, pp. 5679–5688.

[12] GYEONGSIK M., YONG C.J., MU L.K. V2V-PoseNet: Voxel-to-Voxel Prediction Network for Accurate 3D Hand and Human Pose Estimation from a Single Depth Map. In: *CVPR*, 2018, pp. 5079–5088.

[13] GE L., CAI Y., WENG J., YUAN J. Hand PointNet: 3D Hand Pose Estimation using Point Sets. In: *CVPR*, 2018, pp. 8417–8426.

[14] MATURANA D., SCHERER S. Voxnet: A 3d convolutional neural network for real-time object recognition. In: *IROS*, 2015, pp. 922–928.

[15] ZHOU Y., TUZEL O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In: *CVPR*, 2018, pp. 4490-9.

[16] DENG X., YANG S., ZHANG Y., TAN P., CHANG L., WANG H. Hand3D: Hand Pose Estimation using 3D Neural Network. *Computer Vision and Pattern Recognition*, 2017.

[17] SUN X. MSRA Hand Tracking Dataset. https://jimmysuen.github.io/, 2015.

[18] SUN X., WEI Y., LIANG S., TANG X., SUN J. Cascaded hand pose regression. In: *CVPR*, 2015, pp. 824–832.

[19] LIN J., WU Y., HUANG T.S. Modeling the constraints of human hand motion. In: *Workshop on Human Motion*, 2000, pp. 121–126.

[20] WU Y., HUANG T.S. Hand modeling, analysis and recognition. *IEEE Signal Processing Magazine*, 2001, 18(3), pp. 51–60.

[21] YUAN S., HERNANDO G.G., STENGER B., MOON G. CHANG J.Y., LEE K.M., MOLCHANOV P., KAUTZ J., HONARI S. GE L., YUAN J. Depth-Based 3D Hand Pose Estimation From Current Achievements to Future Goals. In: *CVPR*, 2018.

[22] HE K.M., ZHANG X.Y., REN S.Q., SUN J. Deep Residual Learning for Image Recognition. In: *CVPR*, 2016, pp. 770–778.

[23] WANG P., CHEN P.F. Understanding Convolution for Semantic Segmentation. In *WACV*, 2018, pp. 1451–1460.