Multimedia Tools and Applications (2022) 81:41661–41676
https://doi.org/10.1007/s11042-021-11020-w

1181: MULTIMEDIA-BASED HEALTHCARE SYSTEMS USING COMPUTATIONAL INTELLIGENCE

An enhanced self-attention and A2J approach for 3D hand pose estimation

Mei-Ying Ng[1] · Chin-Boon Chng[2] · Wai-Kin Koh[1] · Chee-Kong Chui[2] · Matthew Chin-Heng Chua[1]

Received: 5 October 2020 / Revised: 17 February 2021 / Accepted: 5 May 2021 /
Published online: 15 June 2021

Abstract

Three dimensional (3D) hand pose estimation is the task of estimating the 3D location of hand keypoints. In recent years, this task has received much research attention due to its diverse applications in human-computer interaction and virtual reality. To the best of our knowledge, there has been limited studies that model self-attention in 3D hand pose estimation despite its use in various computer vision tasks. Hence, we propose augmenting convolution with self-attention to capture long-range dependencies in a depth image. In addition, motivated by a recent work which uses anchor points set on a depth image, we extend anchor points to the depth dimension to regress 3D hand joint locations. Validation experiments using the proposed approaches are performed on various hand pose datasets, and we obtain performances that are comparable to other state-of-the-art methods. The results demonstrate the potential of these approaches in a hand-based recognition system.

Keywords Attention · Convolution · Hand pose estimation · Neural network

✉ Chin-Boon Chng
mpeccbo@nus.edu.sg

Mei-Ying Ng
e0402087@u.nus.edu

Wai-Kin Koh
waikin_koh@nus.edu.sg

Chee-Kong Chui
mpecck@nus.edu.sg

Matthew Chin-Heng Chua
isscchm@nus.edu.sg

[1] Institute of Systems Science, National University of Singapore, Singapore, Singapore

[2] Department of Mechanical Engineering, Faculty of Engineering, National University of Singapore, Singapore, Singapore

1 Introduction

Estimation of 3D hand pose is useful in many human-computer interaction applications such as recovery progress for hand rehabilitation systems [22], social robotics [33], user authentication [18] and virtual reality games [48]. With the availability of affordable depth sensors and high-quality hand pose datasets in recent years [25, 34, 35, 37, 45, 46] and advances in convolutional neural networks (CNNs) [16, 19, 21], there has been significant progress in 3D hand pose estimation. However, the task remains a challenge due to severe finger self-occlusion, poor quality of depth images, variations in viewpoint and complex hand shapes [44].

Discriminative methods for 3D hand pose estimation regress 3D hand joint coordinates directly or output heatmaps from a depth image using 2D CNNs [5, 15, 24, 28, 37]. But these methods do not fully exploit the 3D spatial information in the depth image which is intrinsically 3D data [44]. To address this shortcoming, several works have studied 3D methods for hand pose estimation [7, 10–12, 14, 26, 32, 40]. In a recent work, Xiong et al. [42] proposed a novel anchor-based approach named Anchor-to-Joint (A2J) regression network to regress 3D joint coordinates from depth images. In the anchor proposal procedure, anchor points that are densely set on the depth image are assigned weights to discover informative anchor points for a certain joint.

The introduction of the Transformer neural network [38], which replaces recurrence with self-attention to learn long-range dependencies, has led to the wide adoption of self-attention in natural language processing tasks and the increasing use of self-attention in computer vision tasks in recent years. While there are works that employ other attention mechanisms for 3D hand pose estimation [14, 41], studies that model self-attention for the task are limited.

In this work we propose to extend A2J [42] by augmenting convolution with self-attention [2] for 3D hand pose estimation. Moreover, we extend anchor points to the depth dimension in an attempt to better model the 3D spatial geometric characteristics in the depth image. Using the proposed approach, we developed a prototype system for real-time estimation of hand joint angles. This system can be used to assess the range of hand motion in patients with certain disorders that lead to impairment of the hand, such as stroke or rheumatoid arthritis [4].

To summarize, the contributions in this work are as follows:

1. Anchor points that are set in three-dimensional space are used to regress 3D hand joint locations,
2. Self-attention is modelled for 3D hand pose estimation, and
3. A novel user interface is developed to evaluate the range of motion of the hand in clinical practice.

The rest of the paper is organized as follows. Section 2 briefly summarizes related work in the area of hand pose estimation and attention. Section 3 details the proposed approaches which utilize self-attention and 3D anchor points for 3D hand pose estimation. Section 4 reports the details of the experimental results and Section 5 describes the prototype system for rehabilitation. Finally, Section 6 concludes this paper with a summary of the contributions and limitations of the work.

2 Related work

Deep neural networks are commonly used in 3D hand pose estimation to regress 3D joint locations or heatmaps encoding probability distributions of hand joints. One drawback is that the depth image is treated as 2D data and that spatial information in the depth image is under-utilized. To address this problem, several works converted the depth image into 3D data structures such as points [10, 12] or voxels [7, 14, 26]. Ge et al. [12] processed point clouds directly to obtain point-wise estimations of hand joint locations. Moon et al. [26] used a 3D CNN to estimate the per-voxel likelihood for each hand joint and achieved performance that surpassed existing approaches by a large margin. However, 3D CNN methods incur high memory and computational costs. Other methods have been proposed to capture spatial representations with 2D CNNs [11, 32, 40]. Ge et al. [11] projected the depth image into three orthogonal planes with each projection fed into a 2D CNN to regress a heatmap. The heatmaps were then fused to produce 3D hand joint coordinates. Ren et al. [32] incorporated spatial-aware representations that are based on 3D offsets into a 2D CNN consisting of multiple stacked regression modules. In a recent work, Xiong et al. [42] proposed A2J regression network which uses anchor points that are densely set on a depth image to extract global-local spatial context information for 3D hand and body pose estimation.

The attention mechanism was first proposed in Bahdanu et al. [1] in a neural sequence-to-sequence model for neural machine translation. With the advent of the self-attentional Transformer by Vaswani et al. [38], self-attention has now become an integral component in natural language processing tasks. In self-attention, attention is applied to a single context [33]. By attending to all input positions and computing the contextual information of each output, self-attention captures the dependencies between different positions in the input in a single layer. In contrast, convolutional layers are limited by a restricted receptive field and impose translation invariance through weight sharing [2]. Capturing long-range interactions is a challenge with convolution and the global context of images is typically ignored.

The ability of self-attention to encode long-range dependencies and its parallelizability has led to rapid advances in natural language processing tasks such as machine translation [23]. Although convolutional neural networks have been widely used in computer vision tasks, self-attention models are gaining in popularity in various visual tasks including action recognition [13], video object segmentation [36], semantic segmentation [17] and image generation [29, 47]. Bello et al. [2] combined convolution and self-attention in a visual discriminative task by concatenating convolutional feature maps with a set of convolutional maps produced via self-attention, using multi-head attention to attend to distinct representations of an input. This method achieved competitive results on image classification tasks, obtaining higher accuracy than the ResNet-50 baseline on ImageNet. Ramachandran et al. [31] proposed a fully attentional vision model for image classification, using self-attention layers entirely in place of convolution layers. Thus we hypothesize that the attention mechanism proposed in Bello et al. [2] could improve the accuracy of the 3D hand pose estimation task.

3 Methods and materials

In this section, we first discuss the self-attention mechanism. Next we introduce the proposed approaches which utilize self-attention and 3D anchor points for 3D hand pose estimation.

3.1 Self-attention

In self-attention, an input tensor of shape (H, W, F_{in}) is flattened to a matrix $X \in \mathbb{R}^{HW \times F_{in}}$ where H, W and F_{in} refer to the height, width and number of input filters respectively. Attention is performed using the matrix and the output of an attention head h is computed as follows [2]:

$$O_h = Softmax\left(\frac{\left(XW_q\right)\left(XW_k\right)^T}{\sqrt{d_k^h}}\right)\left(XW_v\right) \tag{1}$$

where d_k^h refers to the depth of keys/queries of the attention head and d_v^h refers to the depth of values of the attention head. W_q, $W_k \in \mathbb{R}^{F_{in} \times d_k^h}$ and $W_v \in \mathbb{R}^{F_{in} \times d_v^h}$ are learned linear transformations and map X to queries $Q = XW_q$, keys $K = XW_k$ and values $V = XW_v$. In multihead attention, the self-attention mechanism is replicated with multiple attention heads. Each attention head focuses on a different part of the input using different query, key and value matrices. The outputs from all heads are concatenated and projected as follows [2]:

$$MHA\left(X\right) = Concat\left[O_1, ..., O_{N_h}\right]W^o \tag{2}$$

where N_h and d_v refer to the number of heads and depth of values respectively in multi-head attention and $W^o \in \mathbb{R}^{d_v \times d_v}$ is a learned linear transformation. $MHA\left(X\right)$ is reshaped to return a tensor with the original spatial dimensions (H, W, F_{in}). To enable translation equivariance, relative position encoding is implemented by independently adding relative height information and relative width information. The strength of attention between pixel $i = \left(i_x, i_y\right)$ and pixel $j = \left(j_x, j_y\right)$ is computed as [2]:

$$l_{i,j} = \frac{q_i^T}{\sqrt{d_k^h}}\left(k_j + r^W_{j_x - i_x} + r^H_{j_y - i_y}\right) \tag{3}$$

where q_i is the query vector for pixel i, k_j is the key vector for pixel j, $r^W_{j_x - i_x}$ is the learned embedding for relative width $j_x - i_x$, and $r^H_{j_y - i_y}$ is the learned embedding for relative height $j_y - i_y$. The attention head h with relative positional embeddings is [2]:

$$O_h = Softmax\left(\frac{QK^T + S_H^{rel} + S_W^{rel}}{\sqrt{d_k^h}}\right)V \tag{4}$$

where $S_H^{rel}, S_W^{rel} \in \mathbb{R}^{HW \times HW}$ are matrices of relative positional embeddings for each pixel pair that satisfy $S_H^{rel}[i, j] = q_i^T r^H_{j_y - i_y}$ and $S_W^{rel}[i, j] = q_i^T r^W_{j_x - i_x}$. Lastly, the convolutional operator and output from multihead attention are concatenated as follows [2]:

$$AAConv\left(X\right) = Concat\left[Conv\left(X\right), MHA\left(X\right)\right]. \tag{5}$$

$\upsilon = \frac{d^v}{F_{out}}$ is the ratio between the number of attentional channels and number of output filters in the original convolution operator while $\kappa = \frac{d^k}{F_{out}}$ is the ratio between the key depth and number of output filters in the original convolution operator. In this work, the hyperparameters υ and κ are set to 0.1 and 0.65 respectively.

3.2 Proposed approaches

The A2J regression network proposed in Xiong et al. [42] uses anchor points that are densely set on a depth image and it consists of a ResNet-50 backbone pretrained on ImageNet. Three branches extend from the backbone: an in-plane offset estimation branch, a depth estimation branch and an anchor proposal branch. The common trunk of the ResNet-50 backbone passes a feature map to the anchor proposal branch while a feature map from the regression trunk of the backbone is forwarded through the in-plane offset estimation branch and depth estimation branch.

The two proposed approaches in this work involve modifications to A2J. In the first approach, the self-attention mechanism in Bello et al. [2] is incorporated into A2J and this modified network is named AA-A2J. It has the same framework as A2J, except that its three branches are modified to augment convolution with self-attention (Fig. 1). The depth estimation branch regresses the depth position of the hand keypoints following A2J.

In the second approach, aside from incorporating self-attention into A2J, anchor points are extended to the depth dimension. The new network, referred to as AA-3DA2J, has a framework similar to AA-A2J except that it has a depth offset estimation branch instead of depth estimation branch. Figure 2, adapted from Xiong et al. [42], shows the framework of AA-3DA2J. As 3D anchor points are now utilized, the depth offset estimation branch is used to predict the depth offset with respect to a certain joint from each anchor point. The branches in AA-3DA2J and AA-A2J share the same design (Fig. 1). In addition, the ResNet-50 backbones in both AA-A2J and AA-3DA2J are pretrained on ImageNet.

To determine the individual contribution of self-attention and 3D anchor points to the performance of 3D hand pose estimation, a separate regression network named 3DA2J is also investigated in an ablation study. The 3DA2J network is produced by extending anchor points in A2J to the depth dimension.

The anchor proposal branch in AA-A2J, AA-3DA2J and 3DA2J discovers informative anchors for each joint by assigning weights to the anchor points. These weights are used to

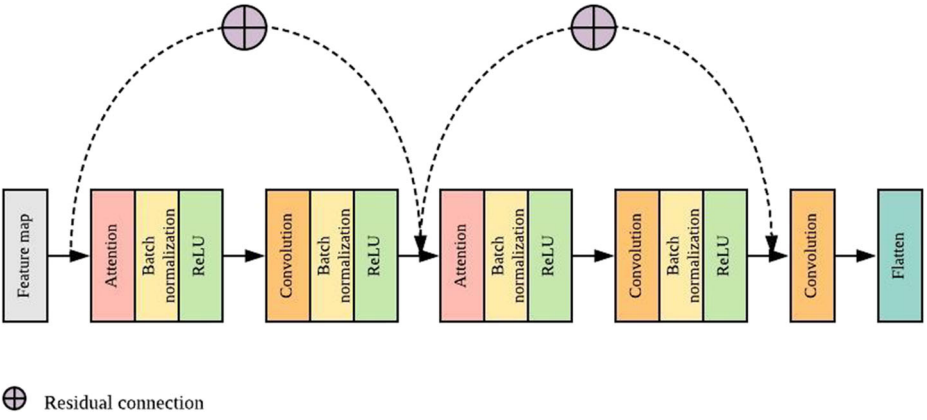

Fig. 1 Architecture of the in-plane offset estimation branch, depth estimation/depth offset estimation branch and anchor proposal branch in AA-A2J and AA-3DA2J

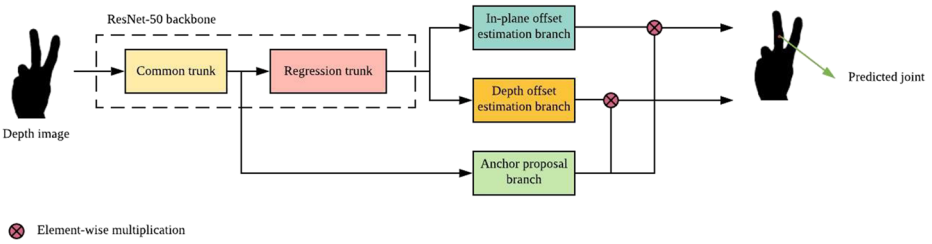

Fig. 2 Framework of AA-3DA2J [42]. The regression network consists of a ResNet-50 backbone which is connected to three branches. The in-plane offset estimation branch and depth offset estimation branches are used to predict the offsets between the anchor points and ground truth while the anchor proposal branch helps discover informative anchor points for a certain joint

predict the contribution of the anchor points to a specific joint. The weights are normalized using the softmax function [42]:

$$\widetilde{P_j}\left(a\right) = \frac{e^{P_j(a)}}{\Sigma_{a\in A} e^{P_j}(a)} \tag{6}$$

where A is the anchor point set and $P_j\left(a\right)$ is the response of anchor point $a \in A$ towards joint j. The estimated in-plane position, estimated depth position and loss functions in AA-A2J are defined according to A2J in Xiong et al. [42].

Next, the estimated in-plane position, estimated depth position and loss functions of the networks which utilize 3D anchor points, AA-3DA2J and 3DA2J, are defined. The estimated in-plane position $\widehat{S_j}$ is formulated as:

$$\widehat{S_j} = \Sigma_{a\in A}\widetilde{P_j}\left(a\right)\left(S^i\left(a\right) + O_j^i\left(a\right)\right) \tag{7}$$

where $S^i\left(a\right)$ and $O_j^i\left(a\right)$ are the in-plane position of anchor point a and predicted in-plane offset towards joint j from anchor point a respectively. The estimated depth position $\widehat{D_j}$ is as follows:

$$\widehat{D_j} = \Sigma_{a\in A}\widetilde{P_j}\left(a\right)\left(S^d\left(a\right) + O_j^d\left(a\right)\right) \tag{8}$$

where $S^d\left(a\right)$ and $O_j^d\left(a\right)$ are the depth position of anchor point a and predicted depth offset towards joint j from anchor point a respectively.

The regression loss function for the in-plane and depth positions is as follows:

$$\begin{aligned} loss_1 = \alpha\Sigma_{j\in J}L_{\tau_1}\left(\Sigma_{a\in A}\widetilde{P_j}\left(a\right)\left(S^i\left(a\right) + O_j^i\left(a\right)\right) - T_j^i\right) \\ +\Sigma_{j\in J}L_{\tau_2}\left(\Sigma_{a\in A}\widetilde{P}\left(a\right)\left(S^d\left(a\right) + O_j^d\left(a\right)\right) - T_j^d\right) \end{aligned} \tag{9}$$

where α is assigned 0.5 according to [42]. Different from A2J and AA-A2J, both the in-plane position and depth position contribute to the informative point surrounding loss in AA-3DA2J and 3DA2J. The informative point surrounding loss is defined as:

$$\begin{aligned} loss_2 = \Sigma_{j\in J}L_{\tau_1}\left(\Sigma_{a\in A}\widetilde{P}\left(a\right)S^i\left(a\right) - T_j^i\right) \\ +\Sigma_{j\in J}L_{\tau_1}\left(\Sigma_{a\in A}\widetilde{P}\left(a\right)S^d\left(a\right) - T_j^d\right) \end{aligned} \tag{10}$$

where T_j^i and T_j^d are the ground-truth in-plane position of joint j and ground-truth depth position of joint j respectively. τ is the smooth L1-like loss and is defined as follows [42]:

$$\tau = \begin{cases} \frac{1}{2\tau}x^2, & for\,|x| < \tau \\ |x| - \frac{\tau}{x}, & otherwise \end{cases} \tag{11}$$

where τ_1 is set to 1 and τ_2 is set to 3 as in [42]. The two loss functions are combined in end-to-end training as follows:

$$loss = \lambda loss_1 + loss_2 \tag{12}$$

where λ is set to 3 following [42].

4 Experiments and results

Center points are used to crop the hand region from the depth image, following the approach of other works [26, 42]. The cropped image is resized to 176 x 176 and passed as input to the ResNet-50 backbone of the proposed approaches after performing data augmentation. The networks are trained end-to-end under the supervision of two loss functions: joint position estimation loss and informative anchor point surrounding loss.

4.1 Datasets

Experiments are conducted on four public hand pose datasets: NYU dataset, ICVL dataset, MSRA dataset and HANDS 2017 dataset.

NYU Dataset [37] The NYU dataset consists of 72K training and 8.2K testing depth images. In each image, 21 joints are annotated. The dataset has a diverse range of hand poses. In line with previous works [5, 15, 26, 42], 14 joints are used during training and testing.

ICVL Dataset [35] The ICVL dataset has 330K training depth images with in-plane rotation augmented frames. There are 6.5K testing depth images and 16 joints are annotated.

MSRA Dataset [34] The MSRA dataset consists of 76.5K images from nine different subjects and 21 joints are annotated. The leave-one-subject-out cross validation method is used for evaluation and the results are averaged over the nine subjects.

HANDS 2017 Dataset [45] The dataset consists of 957K training and 295K testing depth images sampled from BigHand2.2M dataset [46] and First-Person Hand Action datasets (FHAD) [9]. It is the largest hand pose dataset that is available and provides annotations for 21 hand joints. There are five subjects in the training set and ten subjects in the test set. Five subjects in the test set are seen in the training set.

4.2 Evaluation metrics

We evaluate the performance of our approaches with two standard metrics.

Average 3D distance error This is the average Euclidean distance between the predicted 3D joint coordinates and the ground truth.

Percentage of successful frames This metric measures the fraction of test samples that have all predicted joints below a given maximum Euclidean distance from the ground truth.

4.3 Implementation

The networks are implemented in PyTorch. Data augmentation is performed according to Xiong et al. [42], including rotation, scaling and addition of random gaussian noise to depth values. The images in the NYU, ICVL and HANDS 2017 datasets are normalized using the mean and standard deviation values provided in the A2J GitHub repository at https://github.com/zhangboshen/A2J. A2J is not trained on the MSRA dataset and we normalize images in this dataset separately for each subject by computing the mean and standard deviation values of images from the same subject. Weights are updated by the Adam optimizer and the learning rate is set to 0.00035 with a weight decay of 0.0001 for all datasets. A batch size of 16 is used for the NYU dataset with the learning rate decreased by a factor of 0.2 every 7 epochs for 35 epochs. A batch size of 64 is used and the learning rate is decreased by a factor of 0.2 every 5 epochs for the ICVL dataset, MSRA dataset and HANDS 2017 dataset. The networks are trained for 10, 50 and 16 epochs on the ICVL dataset, MSRA dataset and HANDS 2017 dataset respectively. All networks are trained and validated on a Tesla V100 GPU.

4.4 Comparison with state-of-the-arts methods

We compare AA-A2J and AA-3DA2J with the state-of-the-art 3D hand pose estimation methods [3, 5–8, 10, 12, 15, 20, 24, 26, 27, 30, 32, 34, 35, 39, 40, 42–44, 49] on the four public datasets. Figure 3 shows the performances of various methods on the NYU dataset, ICVL dataset and MSRA dataset.

On the NYU dataset, both approaches achieve better performances than baseline A2J. Moreover, AA-3DA2J achieves a mean 3D distance error of 8.37 mm, lower than other methods except SRN [32] (Table 1). The approaches also produce higher percentages of frames with a mean error under 20 mm, as compared to other methods (Fig. 3). On the ICVL dataset, AA-A2J obtains superior performance to other methods except V2V [26]. Similarly, AA-3DA2J achieves better performance compared to other methods except P2P [12], AA-A2J and V2V [26] (Table 2). On the MSRA dataset, AA-A2J and AA-3DA2J obtain mean errors of 8.08 mm and 8.16 mm respectively and achieve comparable performances to other methods (Table 3). Both approaches outperform other methods on the HANDS 2017 dataset, with AA-3DA2J achieving a mean 3D distance error of 8.13 mm (Table 4).

The runtime speeds of different methods are shown in Table 5. AA-A2J is found to have a faster runtime speed than all other methods except SRN [32] while AA-3DA2J has a slower runtime speed compared to AA-A2J, SRN [32], CrossInfoNet [20] and CrossingNets [39].

4.5 Ablation study

To ascertain the individual contribution of self-attention and 3D anchor points to performance, we perform experiments using the NYU dataset, a challenging dataset with a diversity of hand poses.

Self-attention AA-A2J, which utilizes self-attention, demonstrates better performance than baseline A2J as shown in Table 6. Compared to 3DA2J which utilizes 3D anchor points but not self-attention, AA-3DA2J which incorporates both self-attention and 3D anchor points achieves a lower mean error. These results show that self-attention improves the performance of 3D hand pose estimation.

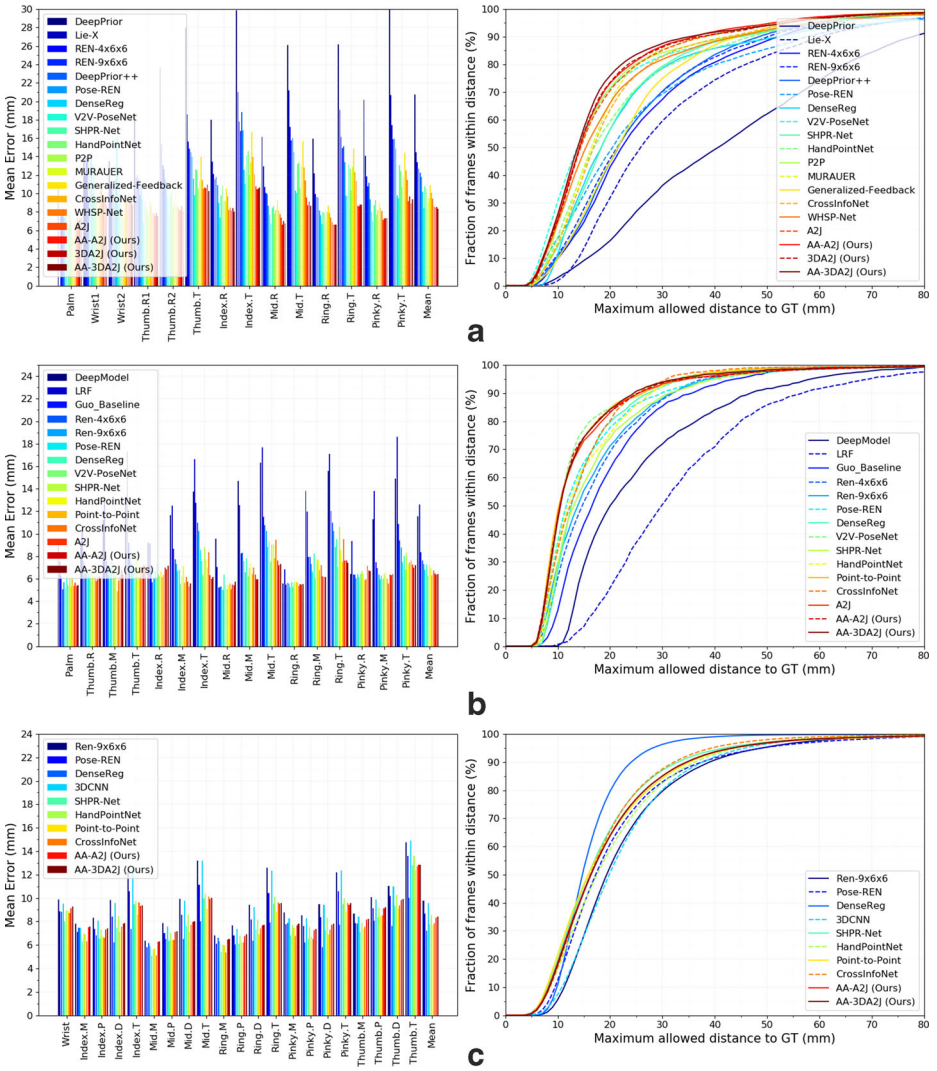

Fig. 3 Evaluation on hand pose datasets. Left: 3D distance errors per hand joint. Right: Percentage of successful frames over different 3D distance error thresholds

3D anchor points A2J and 3DA2J have similar performances as shown in Table 1 and Table 6. In contrast, AA-3DA2J produces superior performance to AA-A2J (Table 6). This suggests that 3D anchor points offer negligible performance advantage over anchor points set in the depth image in the absence of self-attention.

4.6 Runtime analysis

AA-A2J and AA-3DA2J have runtime speeds of 151.06 fps and 79.62 fps respectively on a Tesla V100 GPU (Table 7) whereas A2J has a higher runtime speed of 164.44 fps (Table 7) on the same GPU.

Table 1 Performance of different methods on NYU dataset

Method	Mean error (mm)
DISCO [3]	20.7
Hand3D [7]	17.6
DeepModel [49]	17.04
JSTC [8]	16.8
Global-to-Local [24]	15.60
Lie-X [43]	14.51
REN-4x6x6x [15]	13.39
REN-9x6x6x [15]	12.69
DeepPrior++ [27]	12.24
POSE-REN [5]	11.81
SHPR-Net [6]	10.78
HandPointNet [10]	10.54
DenseReg [40]	10.21
CrossInfoNet [20]	10.08
MURAUER [30]	9.47
P2P [12]	9.05
A2J [42]	8.61
AA-A2J (Ours)	8.45
V2V [26]	8.42
AA-3DA2J (Ours)	8.37
SRN [32]	7.78

Incorporating self-attention to the network leads to a small increase in the number of trainable parameters and decreases runtime speed marginally (Table 7). For instance, AA-A2J has 1.05 times as many trainable parameters as A2J and a slightly slower runtime speed.

Using 3D anchor points increases the number of trainable parameters by a large extent which in turn reduces runtime speed (Table 7). 3D-A2J has 3.68 times as many trainable parameters as A2J and its runtime speed is 0.50 times that of A2J. Similarly, AA-3DA2J has 3.56 times as many trainable parameters as AA-A2J and its runtime speed is 0.53 times that of AA-A2J.

5 Real-time 3D hand pose estimation

Real-time hand pose estimation is useful in assessing the degree of hand impairment for rehabilitation purposes. A real-time 3D hand pose estimation system is implemented using two depth cameras Intel RealSense SR300 and Intel RealSense D415 (Fig. 4). Owing to its faster inference time compared to AA-3DA2J, AA-A2J is used in the system for 3D hand pose estimation. Depth images from both cameras are passed into AA-A2J which has been trained on the HANDS 2017 dataset to estimate the joint locations and the range of motion in terms of flexion. Predicted angles from both cameras are averaged to improve the accuracy. Figure 5 shows the predicted joint locations in the real-time system. Depth images from both

Table 2 Performance of different methods on ICVL dataset

Method	Mean error (mm)
LRF [35]	12.58
DeepModel [49]	11.56
Hand3D [7]	10.9
CrossingNets [39]	10.2
Cascade [34]	9.9
JTSC [8]	9.16
DeepPrior++ [27]	8.1
REN-4x6x6x [15]	7.63
REN-9x6x6x [15]	7.31
DenseReg [40]	7.24
SHPR-Net [6]	7.22
HandPointNet [10]	6.94
POSE-REN [5]	6.79
CrossInfoNet [20]	6.73
A2J [42]	6.46
AA-3DA2J (Ours)	6.39
P2P [12]	6.33
AA-A2J (Ours)	6.30
V2V [26]	6.29

cameras are retrieved and processed simultaneously. The HANDS 2017 dataset annotates the center of the wrist (W), metacarpal phalangeal joint (MCP), proximal interphalangeal joint (PIP), distal interphalangeal joint (DIP) and tip joint (TIP). The annotations are used to compute the flexion hand angles as follows:

$$\widetilde{MCP_x} = arcos\left(\overrightarrow{MCP_x - W}\right).\left(\overrightarrow{PIP_x - MCP_x}\right) \tag{13}$$

Table 3 Performance of different methods on MSRA dataset

Method	Mean error (mm)
CrossingNets [39]	12.2
REN-9x6x6x [15]	9.79
POSE-REN [5]	8.65
HandPointNet [10]	8.51
AA-3DA2J (Ours)	8.16
AA-A2J (Ours)	8.08
CrossInfoNet [20]	7.86
SHPR-Net [6]	7.76
P2P [12]	7.71
V2V [26]	7.49
DenseReg [40]	7.23

Table 4 Performance of different methods on HANDS 2017 dataset

Method	Mean error (mm)
Vanora [44]	11.91
THU VCLab [5]	11.70
Oasis [10]	11.30
RCN-3D [44]	9.97
V2V [26]	9.95
A2J [42]	8.57
SRN [32]	8.39
AA-A2J (Ours)	8.27
AA-3DA2J (Ours)	8.13

Table 5 Runtime of different methods

Method	FPS
SRN [32]	263.1
AA-A2J (Ours)	151.06
CrossInfoNet [20]	124.5
A2J [42]	105.06
CrossingNets [39]	90.9
AA-3DA2J (Ours)	79.62
Global-to-Local [24]	50
HandPointNet [10]	48
P2P [12]	41.8
Hand3D [7]	30
DeepPrior++ [27]	30
DenseReg [40]	27.8
V2V [26]	3.5

Table 6 Effect of self-attention and 3D anchor points on performance on NYU dataset

Model	Mean error (mm)
AA-A2J	8.45
3DA2J	8.59
AA-3DA2J	8.37

Table 7 Number of trainable parameters in different methods

Method	No. of parameters	FPS
A2J	44736424	164.44
AA-A2J	46809304	151.06
3D-A2J	164522664	83.03
AA-3DA2J	166595544	79.62

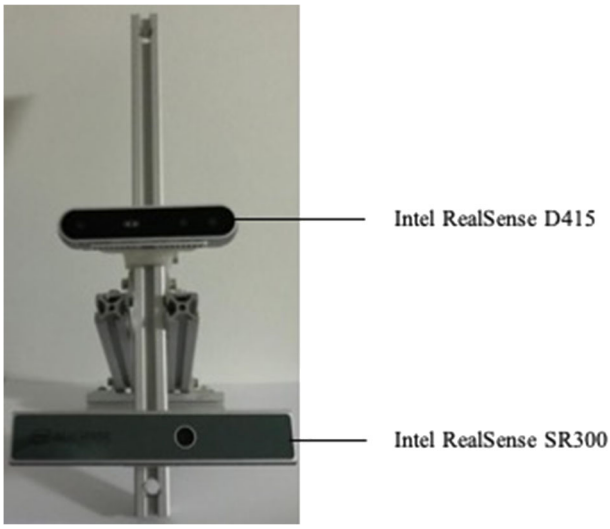

Fig. 4 Setup of depth cameras Intel RealSense D415 and Intel RealSense SR300

$$\widetilde{PIP_x} = arcos\left(\overrightarrow{PIP_x - MCP_x}\right).\left(\overrightarrow{DIP_x - PIP_x}\right) \tag{14}$$

$$\widetilde{DIP_x} = arcos\left(\overrightarrow{DIP_x - PIP_x}\right).a\left(\overrightarrow{TIP_x - DIP_x}\right) \tag{15}$$

	Joint	Camera 1 (°)	Camera 2 (°)	Average angle (°)
1	TMCP	116.1	73.8	94.9
2	TPIP	156.6	173.2	164.9
3	TDIP	163.9	165.2	164.6
4	IMCP	167.7	178.6	173.1
5	IPIP	172.1	177.6	174.8
6	IDIP	179.8	177.4	178.6
7	MMCP	169.5	168.0	168.7
8	MPIP	178.9	171.0	174.9
9	MDIP	174.2	177.6	175.9
10	RMCP	177.2	173.5	175.4
11	RPIP	178.0	157.8	167.9
12	RDIP	101.0	100.4	100.7
13	PMCP	172.4	161.9	167.2
14	PPIP	175.9	168.0	171.9
15	PDIP	146.4	158.3	152.4

Fig. 5 Real-time hand pose estimation

where for each digit x, MCP_x is the angle between W, MCP_x and $PIPx$, DIP_x is the angle between MCP_x, PIP_x and DIP_x, and DIP_x is the angle between PIP_x, DIP_x and TIP_x.

The processing speed of the system is 12.2 fps on a 2070 Super GPU.

6 Conclusion

In this work, two networks are proposed to recover 3D hand poses from a single depth image. The first network, AA-A2J, uses a self-attention mechanism for 3D hand pose estimation whereas the second network, AA-3DA2J, utilizes 3D anchor points in addition to self-attention. The two approaches AA-A2J and AA-3DA2J obtain performances that are comparable to the other state-of-the-art methods and are superior to the baseline A2J regression network. In addition, both AA-A2J and A2J have similar runtime speeds. Modelling self-attention helps capture spatial context information from depth images and is beneficial for 3D hand pose estimation. This advantage is demonstrated by the performances of AA-A2J on the four hand pose datasets. The use of both self-attention and 3D anchor points in AA-3DA2J further boosts performance over AA-A2J on the NYU dataset and HANDS 2017 dataset. However, this comes at the expense of runtime speed.

There are several limitations in our work. First, the approach with 3D anchor points, 3D-A2J, is evaluated on the NYU dataset but not on the other hand pose datasets. In future work, this approach should be evaluated on the other datasets to determine whether extending anchor points to the depth dimension has a marginal effect on performance across all datasets. Second, a relatively small amount of data is used to train and evaluate the proposed approaches. Future work includes evaluation on a bigger dataset comprised of normal subjects and subjects with hand impairment due to stroke. This would enable further studies on the robustness of the proposed approaches in 3D hand pose estimation for stroke rehabilitation.

Acknowledgements This study was funded by Tote Board Enabling Lives Initiative Grant (Grant Number: GC62018NUSISS) and supported by SG Enable.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. arXiv:1409.0473
2. Bello I, Zoph B, Vaswani A, Shlens J, Le QV (2019) Attention augmented convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp 3286–3295
3. Bouchacourt D, Mudigonda PK, Nowozin S (2016) Disco nets: Dissimilarity coefficients networks. In: Advances in neural information processing systems. pp 352–360
4. Cejnog LWX, Cesar RM, de Campos TE, Elui VMC (2019) Hand range of motion evaluation for rheumatoid arthritis patients. In: 2019 14th IEEE international conference on automatic face & gesture recognition (FG 2019). IEEE, pp 1–5
5. Chen X, Wang G, Guo H, Zhang C (2020) Pose guided structured region ensemble network for cascaded hand pose estimation. Neurocomputing 395:138–149

6. Chen X, Wang G, Zhang C, Kim Tae-Kyun, Ji X (2018) Shpr-net: Deep semantic hand pose regression from point clouds. IEEE Access 6:43425–43439
7. Deng X, Yang S, Zhang Y, Tan P, Chang L, Wang H (2017) Hand3d: Hand pose estimation using 3d neural network. arXiv:1704.02224
8. Fourure D, Emonet Rémi, Fromont E, Muselet D, Neverova N, Trémeau A., Wolf C (2017) Multi-task, multi-domain learning: application to semantic segmentation and pose regression. Neurocomputing 251:68–80
9. Garcia-Hernando G, Yuan S, Baek S, Kim T-K (2018) First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 409–419
10. Ge L, Cai Y, Weng J, Yuan J (2018) Hand pointnet: 3d hand pose estimation using point sets. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 8417–8426
11. Ge L, Liang H, Yuan J, Thalmann D (2016) Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 3593–3601
12. Ge L, Ren Z, Yuan J (2018) Point-to-point regression pointnet for 3d hand pose estimation. In: Proceedings of the European conference on computer vision (ECCV). pp 475–491
13. Girdhar R, Carreira J, Doersch C, Zisserman A (2019) Video action transformer network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 244–253
14. Guo F, He Z, Zhang S, Zhao X, Tan J (2020) Attention-based pose sequence machine for 3d hand pose estimation. IEEE Access 8:18258–18269
15. Guo H, Wang G, Chen X, Zhang C (2017) Towards good practices for deep 3d hand pose estimation. arXiv:1707.07248
16. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp 2961–2969
17. Huang L, Yuan Y, Guo J, Zhang C, Chen X, Wang J (2019) Interlaced sparse self-attention for semantic segmentation. arXiv:1907.12273
18. Imura S, Hosobe H (2018) A hand gesture-based method for biometric authentication. In: International conference on human-computer interaction. Springer, pp 554–566
19. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25:1097–1105
20. Kuo Du, Lin X, Yi S, Ma X (2019) Crossinfonet: Multi-task information sharing based hand pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 9896–9905
21. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551
22. Li W-J, Hsieh C-Y, Lin L-F, Chu W-C (2017) Hand gesture recognition for post-stroke rehabilitation using leap motion. In: 2017 international conference on applied system innovation (ICASI). IEEE, pp 386–388
23. Luong M-T, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. arXiv:1508.04025
24. Madadi M, Escalera S, Baró X, Gonzalez J (2017) End-to-end global to local cnn learning for hand pose recovery in depth data. arXiv:1705.09606
25. Madadi M, Escalera S, Carruesco A, Andujar C, Baró X, Gonzalez J (2017) Occlusion aware hand pose recovery from sequences of depth images. In: 2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017). IEEE, pp 230–237
26. Moon G, Ju YC, Lee KM (2018) V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In: Proceedings of the IEEE conference on computer vision and pattern Recognition. pp 5079–5088
27. Oberweger M, Lepetit V (2017) Deepprior++: Improving fast and accurate 3d hand pose estimation. In: Proceedings of the IEEE international conference on computer vision workshops. pp 585–594
28. Oberweger M, Wohlhart P, Lepetit V (2015) Hands deep in deep learning for hand pose estimation. arXiv:1502.06807
29. Parmar N, Vaswani A, Uszkoreit J, Kaiser Łukasz, Shazeer N, Alexander Ku, Tran D (2018) Image transformer. arXiv:1802.05751
30. Poier G, Opitz M, Schinagl D, Bischof H (2019) Murauer: Mapping unlabeled real data for label austerity. In: 2019 IEEE winter conference on applications of computer vision (WACV). IEEE, pp 1393–1402
31. Ramachandran P, Parmar N, Vaswani A, Bello I, Levskaya A, Shlens J (2019) Stand-alone self-attention in vision models. arXiv:1906.05909

32. Ren P, Sun H, Qi Qi, Wang J, Huang W (2019) Srn: Stacked regression network for real-time 3d hand pose estimation. In: BMVC, page 112
33. Showers A, Si M (2018) Pointing estimation for human-robot interaction using hand pose, verbal cues, and confidence heuristics. In: International conference on social computing and social media. Springer, pp 403–412
34. Sun X, Wei Y, Liang S, Tang X, Sun J (2015) Cascaded hand pose regression. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 824–832
35. Tang D, Chang HJ, Tejani A, Kim T-K (2014) Latent regression forest: Structured estimation of 3d articulated hand posture. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 3786–3793
36. Tian Y, Zhang Y, Di Z, Cheng G, Chen W-G, Wang R (2020) Triple attention network for video segmentation. Neurocomputing 417:202–211
37. Tompson J, Stein M, Lecun Y, Perlin K (2014) Real-time continuous pose recovery of human hands using convolutional networks. ACM Trans Graph (ToG) 33(5):1–10
38. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems. pp 5998–6008
39. Wan C, Probst T, Gool LV, Yao A (2017) Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 680–689
40. Wan C, Probst T, Gool LV, Yao A (2018) Dense 3d regression for hand pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 5147–5156
41. Wang X, Jiang J, Guo Y, Kang L, Wei Y, Li D (2020) Cfam: Estimating 3d hand poses from a single rgb image with attention. Appl Sci 10(2):618
42. Xiong F, Zhang B, Xiao Y, Cao Z, Yu T, Zhou JT, Yuan J (2019) A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In: Proceedings of the IEEE international conference on computer vision. pp 793–802
43. Xu C, Govindarajan LN, Yu Z, Li C (2017) Lie-x: Depth image based articulated object pose estimation, tracking, and action recognition on lie groups. Int J Comput Vis 123(3):454–478
44. Yuan S, Garcia-Hernando G, Stenger B, Moon G, Ju YC, Kyoung ML, Molchanov P, Kautz J, Honari S, Ge L et al (2018) Depth-based 3d hand pose estimation: From current achievements to future goals. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 2636–2645
45. Yuan S, Qi Y, Garcia-Hernando G, Kim T-K (2017) The 2017 hands in the million challenge on 3d hand pose estimation. arXiv:1707.02237
46. Yuan S, Ye Q, Stenger B, Jain S, Kim T-K (2017) Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 4866–4874
47. Zhang H, Goodfellow I, Metaxas D, Odena A (2019) Self-attention generative adversarial networks. In: International conference on machine learning. PMLR, pp 7354–7363
48. Zhang Y, Meruvia-Pastor O (2017) Operating virtual panels with hand gestures in immersive vr games. In: International conference on augmented reality, virtual reality and computer graphics. Springer, pp 299–308
49. Zhou X, Wan Q, Zhang W, Xue X, Wei Y (2016) Model-based deep hand pose estimation. arXiv:1606.06854