A2J: Anchor-to-Joint Regression Network for 3D Articulated Pose Estimation from a Single Depth Image

Fu Xiong1*, Boshen Zhang1*, Yang Xiao1†, Zhiguo Cao1, Taidong Yu1, Joey Tianyi Zhou2, and Junsong Yuan3

1 National Key Laboratory of Science and Technology on Multi-Spectral Information Processing, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
2 IHPC, A*STAR, Singapore
3 CSE Department, State University of New York at Buffalo

xiongfu, zhangbs, Yang_Xiao, zgcao, taidongyu@hust.edu.cn, joey.tianyi.zhou@gmail.com, jsyuan@buffalo.edu

Abstract

For 3D hand and body pose estimation task in depth image, a novel anchor-based approach termed Anchor-to-Joint regression network (A2J) with the end-to-end learning ability is proposed. Within A2J, anchor points able to capture global-local spatial context information are densely set on depth image as local regressors for the joints. They contribute to predict the positions of the joints in ensemble way to enhance generalization ability. The proposed 3D articulated pose estimation paradigm is different from the state-of-the-art encoder-decoder based FCN, 3D CNN and point-set based manners. To discover informative anchor points towards certain joint, anchor proposal procedure is also proposed for A2J. Meanwhile 2D CNN (i.e., ResNet-50) is used as backbone network to drive A2J, without using time-consuming 3D convolutional or deconvolutional layers. The experiments on 3 hand datasets and 2 body datasets verify A2J’s superiority. Meanwhile, A2J is of high running speed around 100 FPS on single NVIDIA 1080Ti GPU.

1. Introduction

With the emergence of low-cost depth camera, 3D hand and body pose estimation from a single depth image draws much attention from computer vision community with wide-range application scenarios (e.g., HCI and AR) [32, 33]. Despite recent remarkable progress [20, 42, 26, 18, 19, 7, 33, 50, 41, 3], it is still a challenging task due to the issues of dramatic pose variation, high similarity among the different joints, self-occlusion, etc [20, 42, 37].

Most of state-of-the-art 3D hand and body pose estimation approaches rely on deep learning technology. Nevertheless, they still suffer from some defects. First, encoder-decoder based FCN manners [2, 43, 27, 4, 42, 41, 26] are generally trained with non-adaptive ground-truth Gaussian heatmap for different joints and with relatively high computational burden. Meanwhile, most of them cannot be fully end-to-end trained towards 3D pose estimation task [35]. Secondly, 3D CNN models [16, 10, 26] are difficult to train with costly voxelizing procedure, due to the large number of convolutional parameters. Additionally, point-set based approaches [14, 17] require some extra time-consuming preprocessing treatments (e.g., point sampling).

Thus, we attempt to address 3D hand and body pose estimation problem using a novel anchor-based approach termed Anchor-to-Joint regression network (A2J). The pro-
A2J network has end-to-end learning ability. The key idea of A2J is to predict 3D joint position by aggregating the estimation results of multiple anchor points, in spirit of ensemble learning to enhance generalization ability. Specifically, the anchor points can be regarded as the local regressors towards the joints from different viewpoints and distances. They are densely set on depth image to capture the global-local spatial context information together. Each of them will contribute to regress the positions of all the joints, but with different weights. The joint is localized by aggregating the outputs of all the anchor points. Since different joints may share the same anchor points, the artiﬁced characteristics among them can be well maintained.

For a speciﬁc joint, not all of the anchor points contribute equally. Accordingly, an anchor proposal procedure is proposed to discover the informative anchor points towards the certain joint by weight assignment. During training, both factors of estimation error of anchor points and spatial layout of informative anchor points are concerned. In particular, the picked up informative anchor points are encouraged to uniformly surround the corresponding joint to alleviate overﬁtting. Accordingly, the main idea of the proposed anchor-based 3D pose estimation paradigm within A2J is shown in Fig. 1. We can see that, generally different joints possess different informative anchor points. Furthermore, the visible “index tip” joint holds few informative anchor points. While, the invisible “index mid” joint and the “palm” joint on the relatively ﬂat area possess much more ones, in order to capture richer spatial contexts. This actually reveals A2J’s adaptive property.

Technically, A2J network consists of 3 branches driven by 2D CNN backbone network (i.e., ResNet-50 [21]) without deconvolutional layers. In particular, the 3 branches take charges of predicting in-plain offsets between the anchor points and joints, estimating depth value of the joints, and informative anchor point proposal respectively. The main reasons to build A2J on 2D CNN for 3D pose estimation lie in 3 folders: (1) 3D information is already involved in depth image, using 2D CNN can still reveal 3D characteristics of the original depth image data; (2) compared to 3D CNN and point-set network, 2D CNN can be pre-trained on large-scale datasets (e.g., ImageNet [9]), which may help to enhance its visual pattern capturing capacity for depth image; (3) 2D CNN is of high running efﬁciency without time-consuming 3D convolution operation and preprocessing procedures (e.g., voxelizing and point sampling).

A2J is experimented on 3 hand datasets (i.e., HAND-S 2017 [48], NYU [37], and ICVL [36]) and 2 body pose datasets (i.e., ITOP [20] and K2HPD [42]) to verify its superiority. The experiments reveal that, both for 3D hand and body pose estimation tasks A2J generally outperforms the state-of-the-art methods on effectiveness and efﬁciency simultaneously. Meanwhile, A2J can online run with the high speed around 100 FPS on a single NVIDIA 1080Ti GPU.

The main contributions of this paper include:

• A2J: an anchor-based regression network for 3D hand and body estimation from a single depth image. It is of end-to-end learning capacity;

• An informative anchor proposal approach is proposed, concerning the joint position prediction error and anchor spatial layout simultaneously;

• 2D CNN without deconvolutional layers is used to drive A2J to ensure high running efﬁciency.

A2J’s code is available at https://github.com/zhangboshen/A2J.

2. Related Works

The existing 3D hand and body pose estimation approaches can be mainly categorized into non-deep learning and deep learning based groups. The state-of-the-art non-deep learning based ones [33, 22, 13, 46] generally follow the 2-step technical pipeline of ﬁrst extracting hand-crafted feature, and then executing classiﬁcation or regression. One main drawback is that, hand-crafted feature is often not representative enough. This tends to lead non-deep learning based method to be inferior to deep learning based manner. Since the proposed A2J falls into deep learning group, next we will introduce and discuss this paradigm from the perspectives of 2D and 3D deep learning respectively.

2D deep learning based approach. Due to end-to-end working manner, deep learning technology holds strong ﬁtting ability for visual pattern characterization. 2D CNN has already achieved great success for 2D pose estimation [38, 4, 27, 43, 44]. Recently it has also been introduced to 3D domain, resorting to global regression [18, 19, 29, 28, 7, 15, 20] or local detection [37, 25, 42, 41, 39] ways. The global regression manner cannot well maintain local spatial context information due to the global feature aggregation operation within fully-connected layers. Local detection based paradigm of promising performance generally chooses to address this problem via encoder-decoder model (e.g., FCN), setting local heatmap for each joint. Nevertheless, heatmap setting is still not adaptive for the different joints. And, the deconvolution operation is time consuming. Furthermore, most of the encoder-decoder based methods cannot be fully end-to-end trained [44].

3D deep learning based approach. To better reveal the 3D property within depth image for performance enhancement, one recent research trend is to resort to 3D deep learning. The paid efforts can be generally categorized into 3D CNN based and point-set based families. 3D CNN based methods [16, 10, 26] voxelizes the depth image into volumetric representation (e.g., occupancy grid models [24]). 3D convolution or deconvolution operation is then executed to capture 3D visual characteristics. However, 3D CNN is relatively hard to tune due to the large number of conve-
Regression procedures (e.g., point sampling and KNN search) are re-
point cloud as input. Nevertheless some time-consuming
anchor points. Concerning that maybe not all the anchor
it is finally localized by aggregating the outputs of all the
3 branches predict
3 functional branches: in-plain offset prediction branch, depth estimation branch, and anchor proposal branch. The
3 branches are in-plain offset prediction branch, depth estimation branch, and anchor proposal branch.

![Figure 2. The main technical pipeline of A2J. A2J consists of backbone network and 3 functional branches. The backbone network is built on ResNet-50. And, the 3 branches are in-plain offset prediction branch, depth estimation branch, and anchor proposal branch.](image)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Anchor point set.</td>
</tr>
<tr>
<td>a</td>
<td>Anchor point a ∈ A.</td>
</tr>
<tr>
<td>J</td>
<td>Joint set.</td>
</tr>
<tr>
<td>j</td>
<td>Joint j ∈ J.</td>
</tr>
<tr>
<td>K</td>
<td>Number of joints.</td>
</tr>
<tr>
<td>S(a)</td>
<td>In-plain position of anchor point a.</td>
</tr>
<tr>
<td>P_j(a)</td>
<td>Response of anchor a towards joint j.</td>
</tr>
<tr>
<td>O_j(a)</td>
<td>Predicted in-plain offset towards joint j from anchor point a.</td>
</tr>
<tr>
<td>D_j(a)</td>
<td>Predicted depth value of joint j by anchor point a.</td>
</tr>
</tbody>
</table>

Table 1. Symbol definition within A2J.

olutional parameters. Meanwhile, 3D voxelization operation
also leads to high computational burden both on memory s-
storage and running time. Another way for 3D deep learning
is point-set network [6, 30], transferring depth image into
point cloud as input. Nevertheless some time-consuming
procedures (e.g., point sampling and KNN search) are re-
quired [6, 30], which weakens running efficiency.

Accordingly, A2J belongs to 2D deep learning based
group. The dense anchor points capture the global-local s-
patial context information in ensemble way, without using
computationally expensive deconvolutional layers. 2D CN-
N is used as the backbone network for high running efficien-
cy, also aiming to transfer knowledge from RGB domain.

3. A2J: Anchor-to-Joint Regression Network

The main technical pipeline of A2J is shown in Fig. 2.
And, the symbols within A2J are defined in Table 1. A2J
consists of 2D backbone network (i.e., ResNet-50), and
3 functional branches: in-plain offset estimation branch, depth estimation branch, and anchor proposal branch. The
3 branches predict O_j(a), D_j(a), and P_j(a) respectively.

Within A2J, anchor points are densely set up on the input
depth images with stride S_t = 4 pixels to capture the
global-local spatial context information as in Fig. 3. Essenti-
ally, each of them serves as the local regressor to predict the
3D position of all the joints via in-plain offset prediction
branch and depth estimation branch. For certain joint, it is fin-
ally localized by aggregating the outputs of all the anchor
points. Concerning that maybe not all the anchor points contribute equally to certain joint, the anchor points
will be assigned weights via anchor proposal branch to dis-
cover the informative ones. As consequence, the in-plain
position and depth value of joint j can be achieved as the
weighted average of the outputs of all anchor points as:

\[
\begin{align*}
\hat{S}_j &= \sum_{a \in A} P_j(a) (S(a) + O_j(a)) \\
\hat{D}_j &= \sum_{a \in A} P_j(a) D_j(a)
\end{align*}
\]

(1)

where \(\hat{S}_j\) and \(\hat{D}_j\) indicate the estimated in-plain position
and depth value of joint j; \(P_j(a)\) can be regarded as the
normalized weight of anchor a towards joint j across all anchor points, and is acquired using softmax by:

\[
\hat{P}_j(a) = \frac{e^{P_j(a)}}{\sum_{a \in A} e^{P_j(a)}}.
\]

(2)

It is worthy noting that, the anchor point a with \(P_j(a) > 0.02\) will be regarded as the informative anchor points for
joint j. The selected informative anchor points can reveal
A2J’s adaptive characteristics as in Fig. 1. Joint position es-
timation loss and anchor point surrounding loss are used to
supervise A2J’s end-to-end training. Under their joint su-
prevision, informative anchor points with the spatial layout
that surrounds the joint will be picked up to enhance gen-
eralization ability. Next, we will illustrate the proposed A2J
regression network and its learning procedure in details.

3.1. A2J regression network

Here, the 3 functional branches and backbone network
within A2J will be illustrated in details respectively.
3.1.1 In-plain offset and depth estimation branches

Essentially, these 2 branches play the role of predicting the 3D positions of joints. Since in-plain position estimation and depth estimation are of different properties, we choose to execute them separately. Specifically, one is to estimate $O_{j}(a)$ between anchor points and joints. And, the other is to estimate $D_{j}(a)$ towards joints. As in Fig. 4, they are built upon the output feature map of regression trunk within backbone network to involve semantic feature. Four $3 \times 3$ intermediate convolutional layers (with BN and ReLU) are consequently set to aggregate richer local context information without reducing in-plain size. Since the feature map is a $16 \times$ downsampling of the input depth image on in-plain size (illustrated in Sec. 3.1.3) and anchor point setting stride $S_{t} = 4$ as in Fig. 3, one feature map point corresponds to $4 \times 4 = 16$ anchor points on depth image. An output convolutional layer with the feature map in-plain size is then set towards all the 16 corresponding anchor points in column-wise manner. Suppose $K$ joints exist, in-plain offset estimation branch is of $16 \times K \times 2$ output channels. And, depth estimation branch is of $16 \times K \times 1$ output channels.

3.1.2 Anchor proposal branch

This branch discovers informative anchor points for the certain joint by weight assignment as Eqn. 2. As in Fig. 5, anchor proposal branch is built upon the output feature map of common trunk within backbone network to involve relatively fine feature. As the 2 branches introduced in Sec. 3.1.1, 4 intermediate convolutional layers and 1 output convolutional layer are consequently set for predicting $P_{j}(a)$ for the anchor points without losing in-plain size. Accordingly, the output layer of this branch is of $16 \times K \times 1$ channels.

3.1.3 Backbone network architecture

ResNet-50 [21] pre-trained on ImageNet is used as the backbone network. In particular, layers 0-3 correspond to the common trunk in Fig. 2. And, layer 4 corresponds to regression trunk. Some modifications are executed to make ResNet-50 more suitable for pose estimation. First, the convolutional stride in layer 4 is set to 1. Consequently, the output feature map of layer 4 is a $16 \times$ downsampling of the input depth image on in-plain size. Compared with the raw ResNet-50 with $32 \times$ downsampling, more fine spatial information can be maintained in this way. Meanwhile, the convolution operation within layer 4 is revised as the dilated convolution with a dilation of 2 to enlarge receptive field.

3.2. Learning procedure of A2J

To generate input of A2J, we follow [26] and use center points to crop the hand region from depth image. For body pose, we follow [11] and use bounding box to crop the body region. For joint $j$, in-plain target $T_{j}^{d}$ denotes the 2D ground-truth in pixel coordinate transformed according to the cropped region. To make $T_{j}^{d}$ and depth target $T_{j}^{d}$ be in comparable magnitude, we transform the ground-truth depth $Z_{j}$ of joint $j$ as:

$$T_{j}^{d} = \mu(Z_{j} - \theta),$$

where $\mu$ and $\theta$ are the transformation parameters. For hand pose $\mu$ is set to 1, and $\theta$ is set to the depth of center points. For body pose $\mu$ is set to 50 and $\theta$ is set as 0, since we do not have depth center. During test, the prediction result will be warped back to world coordinate. A2J is then trained under the joint supervision of 2 loss functions: joint position estimation loss and informative anchor point surrounding loss. Next, we will illustrate these 2 loss functions in details.

3.2.1 Joint position estimation loss

Within A2J, the anchor points serve as the local regressors to predict the 3D position of joints in ensemble way. This
objective loss can be formulated as:

$$\text{loss}_1 = \alpha \sum_{j \in J} L_{\tau_1}(\sum_{a \in A} \tilde{P}_j(a)(S(a) + O_j(a)) - T^i_j)$$

$$+ \sum_{j \in J} L_{\tau_2}(\sum_{a \in A} \tilde{P}_j(a)D_j(a) - T^d_j),$$

(4)

where $\alpha = 0.5$ is the factor to balance in-plane offset and depth estimation task; $T^i_j$ and $T^d_j$ are the in-plane and depth targets position of joint $j$; and $L_{\tau}(\cdot)$ is the smooth$L_1$ like loss function [31] given by:

$$L_{\tau}(x) = \begin{cases} \frac{1}{2\tau}x^2, & \text{for } |x| < \tau, \\ |x| - \frac{\tau}{2}, & \text{otherwise}. \end{cases}$$

In Eqn. 4, $\tau_1$ is set to 1 and $\tau_2$ is set to 3 since the depth value is relatively noisy.

### 3.2.2 Informative anchor point surrounding loss

To enhance the generalization ability of A2J, we intend to let the picked up informative anchor points locate around the joints, in spirit of observing the joints from multiple viewpoints simultaneously. Hence, the informative anchor point surrounding loss is defined by us as:

$$\text{loss}_2 = \sum_{j \in J} L_{\tau_1}(\sum_{a \in A} \tilde{P}_j(a)S(a) - T^i_j).$$

(6)

To reveal its effectiveness, we show the informative anchor point spatial layouts with and without using it both for hand and body pose cases in Fig. 6. It can be seen that, informative anchor point surrounding loss can essentially help to alleviate viewpoint bias. Its quantitative effectiveness will also be verified in Sec. 4.3.1.

### 3.2.3 End-to-end training

The 2 loss functions above jointly supervise the end-to-end learning procedure of A2J, which is formulated as:

$$\text{loss} = \lambda \text{loss}_1 + \text{loss}_2,$$

(7)

where $\text{loss}$ is the loss in all; and $\lambda = 3$ is the weight factor to balance $\text{loss}_1$ and $\text{loss}_2$.

### 4. Experiments

#### 4.1. Experimental setting

##### 4.1.1 Datasets

**HANDS 2017 dataset** [48]. It contains 957K training and 295K test depth images sampled from BigHand 2.2M [49] and First-Person Hand Action [48] datasets. The ground-truth is the 3D coordinates of 21 hand joints.

**NYU Hand Pose Dataset** [37]. It contains 72K training and 8.2K test depth images with 3D annotation on 36 hand joints. Following [7, 19, 18, 26], we pick 14 of the 36 joints from frontal view for evaluation.

**ICVL Hand Pose Dataset** [36]. It contains 22K training and 1.5k test depth images. It is augmented to 330K samples by in-plane rotations. 16 hand joints are annotated.

**ITOP Body Pose Dataset** [20]. It contains 40K training and 10K test depth images both for the front-view and top-view tracks. Each depth image is labelled with 15 3D joint locations of human body.

**K2HPD Body Pose Dataset** [42]. It contains about 100K depth images. 19 human body joints are annotated with the in-plane manner.

#### 4.1.2 Evaluation metric

For hand, the average 3D distance error and percentage of success frame metrics [26] are used as the evaluation criteria. For body, Percent of Detected Joints (PDJ) [42, 20] and mean average precision (mAP) with 10-cm rule [42, 20] are used for evaluation.

#### 4.1.3 Implementation details

A2J network is implemented using PyTorch. The input depth image is cropped and resized to a fixed resolution (i.e., $176 \times 176$ for hand, and $288 \times 288$ for body). Random
in-plain rotation and random scaling for both in-plain and depth dimension are executed for data augment. Random Gaussian noise is also randomly added with the probability of 0.5 for data augment. We use Adam as the optimizer. The learning rate is set to 0.00035 with a weight decay of 0.0001 in all cases. A2J is trained on NYU for 34 epochs with a learning rate decay by 0.1 every 7 epoch. For 2 human body datasets, the learning rate decay by 0.1 every 10 epoch. And, “AVG” indicates the result over all subjects. And, “∗” means the ensemble of 10 models.

### 4.2. Comparison with state-of-the-art methods

**HANDS 2017 dataset**: A2J is compared with the state-of-the-art 3D hand pose estimation methods [47, 8, 14, 26], particularly. The performance comparison is listed in Table 2. It can be observed that:

- On this challenging million-scale dataset, A2J consistently outperforms the other approaches both from the perspectives of effectiveness and efficiency. This essentially verifies the superiority of our proposition;
- It is worthy noting that, A2J is significantly superior to the others with the remarkable margin (2.05 at least) towards the “UNSEEN” test case. This phenomenon essentially demonstrates the generalization ability of A2J;
- V2V∗ is the strongest competitor of A2J, but with 10 models ensemble. As a consequence, it is much slower than A2J with only a single model.

**NYU and ICVL datasets**: We compare A2J with the state-of-the-art 3D hand pose estimation methods [36, 1, 34, 28, 51, 12, 10, 45, 19, 18, 40, 7, 23, 39, 14, 17, 26] on this 2 datasets specifically. The experimental results are given in Table 3, 4 on the average 3D distance error. Meanwhile, the percentage of success frames over different error thresholds and the error of each joint are also given in Fig. 7. We can summarize that:

- A2J is superior to the other methods in most cases both on accuracy and efficiency. The exceptional case is that, A2J is slightly inferior to V2V∗ and P2P on ICVL dataset on accuracy but with much higher running efficiency;
- Concerning the good tradeoff between effectiveness and efficiency, A2J essentially takes advantage over the state-of-the-art 3D hand pose estimation approaches.
Table 5. Performance comparison on ITOP 3D body pose estimation dataset [20].

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset Component error / mAP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NYU (hand)</td>
<td>w/o anchor proposal branch</td>
<td>10.08</td>
<td>w/o informative anchor point surrounding loss</td>
<td>9.00</td>
<td>Estimate IPO and DV using one branch</td>
<td>8.95</td>
</tr>
<tr>
<td></td>
<td>A2J (Ours)</td>
<td>8.61</td>
<td>w/o informative anchor point surrounding loss</td>
<td>8.64</td>
<td>Estimate IPO and DV using one branch</td>
<td>8.74</td>
</tr>
<tr>
<td>A2J (Ours)</td>
<td>88.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6. Performance comparison on K2HPD dataset [42].

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset Component error / mAP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITOP (body pose)</td>
<td>w/o anchor proposal branch</td>
<td>80.1</td>
<td>w/o informative anchor point surrounding loss</td>
<td>86.4</td>
<td>Estimate IPO and DV using one branch</td>
<td>87.4</td>
</tr>
<tr>
<td></td>
<td>A2J (Ours)</td>
<td>88.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.3. Ablation study

4.3.1 Component effectiveness analysis

The component effectiveness analysis within A2J is executed on NYU [37] (hand), and ITOP [20] dataset (body). We will investigate the effectiveness of anchor proposal branch, informative anchor point surrounding loss, and configuration of in-plane offset and depth estimation branches. The results are listed in Table 7. It can be observed that:

- Without using anchor proposal branch, performance will drop remarkably especially for body pose. This verifies our point that, not all the anchor points contribute equally to the certain joints. Actually, anchor point adaptivity is A2J’s essential property to leverage performance;
- Without using informative anchor point surrounding loss, performance will drop especially for body pose. This demonstrates that, informative anchor point spatial layout is an essential issue that should be concerned towards generalization ability;
- When estimating in-plane offset and depth value in one branch, performance will drop to some degree. This may be caused by the fact that, in-plane offset and depth value holds different physical characteristics.

4.3.2 Effectiveness of anchor-based paradigm

To verify the effectiveness of anchor-based 3D pose estimation paradigm, we compare A2J with the global regression based manner [38] and FCN-based approach [44]. Since FCN model is generally used to predict in-plane joint position, this ablation study is executed on K2HPD.
Global regression manner encodes depth image with 2D C-NN, and then regresses in-plain human joint position using fully-connected layers. FCN model is built following [44]. ResNet-50 [21] is employed as the backbone network for them, which is the same as A2J for fair comparison. PDJ (0.05) is used as the evaluation criteria. The performance comparison is listed in Table 8. We can see that:

- Our proposed anchor-based paradigm significantly outperforms the other 2 ones, when using the same ResNet-50 backbone network. We think 2 main reasons lie. First, compared with global regression based manner local spatial context information can be better maintained within A2J. Meanwhile, compared with FCN model A2J possess anchor point adaptivity towards the certain joint;
- A2J runs faster than FCN model, but slower than global regression way. However, its performance advantage over global regression paradigm is significant, actually with better tradeoff between effectiveness and efficiency.

### 4.3.4 Backbone network comparison

The comparison among the different backbone networks is further studied. As shown in Table 10, we compare the performance of 3 backbone networks (i.e., ResNet-18, ResNet-34 and ResNet-50). It can be summarized that:

- Deeper network can achieve better results, but with relatively slower running efficiency. However, the performance gap among the different backbones is not huge;
- It is worthy noting that, even using ResNet-18 A2J still can generally achieve the state-of-the-art performance and with extremely fast running speed of 192.25 FPS. This reveals the applicability of A2J towards high real-time running demanding application scenarios.

### 4.4. Running speed analysis

The average online running speed of A2J for 3D hand pose estimation is 105.06 FPS, including 1.5 ms for reading and warpping image, and 8.0 ms for network forward propagation and post-processing on a single NVIDIA 1080Ti GPU. The running speed for 3D body pose estimation is 93.78 FPS, including 0.4 ms for reading and warpping image, and 10.2 ms for network forward propagation and post-processing. This reveals A2J’s real-time running capacity.

### 5. Conclusions

In this paper, an anchor-based 3D articulated pose estimation approach for single depth image termed A2J is proposed. Within A2J anchor points are densely set up on depth image to capture the global-local spatial context information, and predict joint’s position in ensemble way. Meanwhile, informative anchor points are extracted to reveal A2J’s adaptive characteristics towards the different joints. A2J is built on 2D CNN without using computational expensive deconvolutional layers. The wide-range experiments demonstrate A2J’s superiority both from the perspectives of effectiveness and efficiency. In future work, we will seek the more effective way to fuse the anchor points.

### Acknowledgment

This work is jointly supported by the National Key R&D Program of China (No. 2018YFB1004600), National Natural Science Foundation of China (Grant No. 61876211 and 61602193), the Fundamental Research Funds for the Central Universities (Grant No. 2019kyfYXKJC024), the International Science & Technology Cooperation Program of Hubei Province, China (Grant No. 2017AHB051), the start-up funds from University at Buffalo. Joey Tianyi Zhou is supported by Singapore Government’s Research, Innovation and Enterprise 2020 Plan (Advanced Manufacturing and Engineering domain) under Grant A1687b0033 and Grant A18A1b0045. We also thank the anonymous reviewers for their suggestions to enhance the quality of this paper.
References


