3DV: 3D Dynamic Voxel for Action Recognition in Depth Video

Yancheng Wang\(^1\), Yang Xiao\(^1\†\), Fu Xiong\(^2\), Wenxiang Jiang\(^1\), Zhiguo Cao\(^1\), Joey Tianyi Zhou\(^3\), and Junsong Yuan\(^4\)

\(^1\)National Key Laboratory of Science and Technology on Multispectral Information Processing, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China \(^2\)Megvii Research Nanjing, Megvii Technology, China \(^3\)IHPC, A*STAR, Singapore \(^4\)CSE Department, State University of New York at Buffalo

\(^†\)Yang Xiao is corresponding author (Yang_Xiao@hust.edu.cn).

Abstract

To facilitate depth-based 3D action recognition, 3D dynamic voxel (3DV) is proposed as a novel 3D motion representation. With 3D space voxelization, the key idea of 3DV is to encode 3D motion information within depth video into a regular voxel set (i.e., 3DV) compactly, via temporal rank pooling. Each available 3DV voxel intrinsically involves 3D spatial and motion feature jointly. 3DV is then abstracted as a point set and input into PointNet++ for 3D action recognition, in the end-to-end learning way. The intuition for transferring 3DV into the point set form is that, PointNet++ is lightweight and effective for deep feature learning towards point set. Since 3DV may lose appearance clue, a multi-stream 3D action recognition manner is also proposed to learn motion and appearance feature jointly. To extract richer temporal order information of actions, we also divide the depth video into temporal splits and encode this procedure in 3DV integrally. The extensive experiments on 4 well-established benchmark datasets demonstrate the superiority of our proposition. Impressively, we acquire the accuracy of 82.4\% and 93.5\% on NTU RGB+D 120 \cite{33} with the cross-subject and cross-setup test setting respectively. 3DV’s code is available at https://github.com/3huo/3DV-Action.

1. Introduction

During the past decade, due to the emergence of low-cost depth camera (e.g., Microsoft Kinect \cite{52}) 3D action recognition becomes an active research topic, with the wide-range application scenarios of video surveillance, human-machine interaction, etc \cite{45, 46}. The state-of-the-art 3D action recognition approaches can be generally categorized into the depth-based \cite{28, 17, 16, 36, 11, 51, 35, 34} and skeleton-based groups \cite{32, 48, 22, 10, 42, 46}. Since accurate and robust 3D human pose estimation is still challenging \cite{47, 21}, we focus on depth-based avenue in this work.

Since human conducts actions in 3D space, capturing 3D motion pattern effectively and efficiently is crucial for depth-based 3D action recognition. An intuitive way is to calculate dense scene flow \cite{1}. However this can be time consuming \cite{1}, which may not be preferred by the practical applications. Recently, dynamic image \cite{3, 2} able to represent the motion information within RGB video compactly has been introduced to depth domain for 3D action characterization \cite{42, 46}. It can compress RGB video into a single image, while still maintaining the motion characteristics well via temporal rank pooling \cite{6, 5}. Thus dynamic image

Figure 1. A live “Handshaking” 3DV example from NTU RGB+D 60 dataset \cite{33}. 3DV motion value reveals the temporal order of 3DV motion component. The later motion component is of higher value, and vice versa. And, the local region of richer motion information holds higher standard deviation on 3DV motion value.
can fit deep CNN model [8] well for action categorization, which is leveraged by CNN’s strong pattern representation capacity. Nevertheless we argue that the ways of applying dynamic image to 3D field in [42, 46] have not fully exploited 3D descriptive clue within depth video, although normal vector [42] or multi-view projection [46] is applied. The insight is that, both methods in [42, 46] finally encode 3D motion information onto the 2D image plane to fit CNN. Thus, they cannot well answer the question “Where does the certain 3D motion pattern within human action appear in 3D space?” crucial for effective 3D action characterization due to the fact that human actions actually consists of both motion patterns and compact spatial structure [29].

To address the concern above, we propose 3D dynamic voxel (3DV) as a novel 3D motion representation for 3D action representation. To extract 3DV, 3D space voxelization is first executed. Each depth frame will be transformed into a regular voxel set. And the appearance content within it can be encoded by observing whether the yielded voxels have been occupied or not [40], in a binary way. Then, temporal rank pooling [6, 5] is executed towards all the binary voxel sets to compress them into one single voxel set termed 3DV particularly. Thus, 3D motion and spatial characteristics of 3D action can be encoded into 3DV jointly. To reveal this, a live “Handshaking” 3DV example is provided in Fig. 1. As shown, each available 3DV voxel possesses a motion value able to reflect the temporal order of its corresponding 3D motion component. Specifically, the later motion component is of higher value, and vice verse. Meanwhile, the local region of richer 3D motion information possesses higher standard deviation on 3DV motion value (e.g., hand region vs. head region). Meanwhile, 3DV voxel’s location reveals the 3D position of its 3D motion component. Thus, 3DV’s spatial-motion representative ability can essentially leverage 3D action characterization. To involve richer temporal order information, we further divide depth video into finer temporal splits. This is encoded in 3DV integrally by fusing the motion values from all the temporal splits.

With 3DV, the upcoming question is how to choose the adaptive deep learning model to conduct 3D action recognition particularly. Towards voxel set, 3D CNN [20, 7, 21] is often used for 3D visual pattern understanding, and also applicable to 3DV. However, it is difficult to train due to the large number of convolutional parameters. Inspired by the recent success of the lightweighted deep learning models on point set (e.g., PointNet++ [25]), we propose to transfer 3DV into the point set form as the input of PointNet++ to conduct 3D action recognition in end-to-end learning manner. That is, each 3DV voxel will be abstract as a point characterized by its 3D location index and motion value. Our intuition is to alleviate the training difficulty and burden.

Although 3DV can reveal 3D motion information, it still may lose appearance details as in Fig. 1. Since appearance also plays vital role for action recognition [23, 37], only using 3DV may weaken performance. To alleviate this, a multi-stream deep learning model using PointNet++ is also proposed to learn 3D motion and appearance feature jointly. In particular, it consists of one motion stream and multiple appearance streams. The input of motion stream is 3DV. And, the inputs of appearance streams are the depth frames sampled from the different temporal splits. They will also be transformed into the point set form to fit PointNet++.

The experiments on 2 large-scale 3D action recognition datasets (i.e., NTU RGB+D 120 [13] and 60 [33]), and 2 small-scale ones (i.e., N-UCLA [41] and UW A3DII [26]) verify 3DV’s superiority over the state-of-the-art manners.

The main contributions of this paper include:
- 3DV: a novel and compact 3D motion representative manner for 3D action characterization;
- PointNet++ is applied to 3DV for 3D action recognition in end-to-end learning way, from point set perspective;
- A multi-stream deep learning model is proposed to learn 3D motion and appearance feature jointly.

2. Related Works

3D action recognition. The existing 3D action recognition approaches generally falls into the depth-based [23, 28, 48, 22, 10, 42, 46] and skeleton-based groups [15, 17, 16, 36, 11, 51, 35, 34]. Recently the skeleton-based approaches with RNN [15] and GCN [35] has drawn more attention, since using 3D skeleton can help to resist the impact of variations on scene, human attribute, imaging viewpoint, etc. However, one critical issue should not be ignored. That is, accurate and robust 3D human pose estimation is still not trivial [47, 21]. To reveal this, we have checked the 3D skeletons within NTU RGB+D 60 [13] carefully. Actually, even under the constrained condition 3D skeleton extraction still may fail to work as in Fig. 2. Thus, currently for the practical applications depth-based manner seems more preferred and is what we concern.

Most of the paid efforts focus on proposing 3D action representation manner to capture 3D spatial-temporal appearance or motion pattern. At the early stage, the hand-crafted descriptions of bag of 3D points [12], depth mo-
tion map (DMM) [49]. Histogram of Oriented 4D Normals (HON4D) [23], Super Normal Vector (SNV) [19] and binary range sample feature [19] are proposed from the different research perspectives. Recently CNN [8, 50] has been introduced to this field [43, 44, 42, 46], and enhanced performance remarkably. Under this paradigm, the depth video will be compressed into one image using DMM [49] or dynamic image [3, 2] to fit CNN. To better exploit 3D descriptive clue, normal vector or multi-view projection is applied additionally. However, they generally suffer from 2 main defects. First, as aforementioned DMM or dynamic image cannot fully reveal 3D motion characteristics well. Secondly, they tend to ignore appearance information.

**Temporal rank pooling.** To represent action, temporal rank pooling [6, 5] is proposed to capture the frame-level evolution characteristics within video. Its key idea is to train a linear ranking machine towards the frames to arrange them in chronological order. Then, the parameters of the ranking machine can be used as the action representation. By applying temporal rank pooling to the raw frame pixels, dynamic image [3, 2] is proposed with strong motion representative ability and adaptive to CNN. As aforementioned, temporal rank pooling has recently been applied to 3D action recognition [42, 46]. However, how to use it to fully reveal 3D motion property still has not been deeply studied.

**Deep learning on point set.** Due to the irregularity of 3D point set, typical convolutional architectures (e.g., CNN [8]) cannot handle it well. To address this, deep learning on point set draws the increasing attention. Among the paid efforts, PointNet++ [25] is the representative one. It contributes to ensure the permutation invariance of point sets, and capture 3D local geometric clue. However, it has not been applied to 3D action recognition yet.

Accordingly, 3DV is proposed to characterize 3D motion compactly, via temporal rank pooling. The adaptive multi-stream deep learning model using PointNet++ is also proposed to learn 3D motion and appearance feature jointly.

### 3. 3DV: A Novel Voxel Set based Compact 3D Motion Representative Manner

Our research motivation on 3DV is to seek a compact 3D motion representative manner to characterize 3D action. Accordingly, deep feature learning can be easily conducted on it. The proposition of 3DV can be regarded as the essential effort for extending temporal rank pooling [6, 5] originally for 2D video to 3D domain, to capture 3D motion pattern and spatial clue jointly. The main idea for 3DV extraction is in Fig. 3. The depth frames will be first map in to point clouds to better reveal 3D characteristics. Then, 3D voxelization is executed to further transform the disordered point clouds into the regular voxel sets. Consequently, 3D action appearance clue within the certain depth frame can be described by judging whether the voxels have been occupied or not.

![Figure 3. The main idea for 3DV extraction via temporal rank pooling, towards the 3D voxel sets transformed from depth frames.](image3)

![Figure 4. The point cloud and its corresponding 3D voxel set sampled from “Handshaking”.](image4)

Temporal rank pooling cannot be applied to the yielded point clouds directly, due to their disordered property [25] as in Fig. 4(a). To address this, we propose to execute 3D voxelization towards the point clouds. Then the 3D appearance information can be described by observing whether the voxels have been occupied or not, disregarding the involved point number as

\[
V_t(x, y, z) = \begin{cases} 
1, & \text{if } V_t(x, y, z) \text{ is occupied} \\
0, & \text{otherwise}
\end{cases}
\]

where \(V_t(x, y, z)\) indicates one certain voxel at the \(t\)-th frame; \((x, y, z)\) is the regular 3D position index. This actually holds 2 main profits. First, the yielded binary 3D voxel sets are regular as in Fig. 4(b). Thus, temporal rank pooling...
can be applied to them for 3DV extraction. Meanwhile the binary voxel-wise representation manner is of higher tolerance towards the intrinsic sparsity and density variability problem [25] within point clouds, which essentially helps to leverage generalization power.

3.2. 3DV extraction using temporal rank pooling

With the binary 3D appearance voxel sets above, temporal rank pooling is executed to generate 3DV. A linear temporal ranking score function will be defined for compressing the voxel sets into one voxel set (i.e., 3DV).

Particularly, suppose \( V_1, \ldots, V_T \) indicate the binary 3D appearance voxel sets, and \( \bar{V}_t = \frac{1}{T} \sum_{i=1}^{T} V_i \) is their average till time \( t \). The ranking score function at time \( t \) is given by

\[
S(t|w) = \langle w, \bar{V}_t \rangle ,
\]

where \( w \in \mathbb{R}^d \) is the ranking parameter vector. \( w \) is learned from the depth video to reflect the ranking relationship among the frames. The criteria is that, the later frames are of larger ranking scores as

\[ q > t \Rightarrow S(q|w) > S(t|w). \]

The learning procedure of \( w \) is formulated as a convex optimization problem using RankSVM [38] as

\[
w^* = \arg\min_w \frac{1}{2} || w ||^2 + \frac{1}{T(T - 2)} \sum_{q > t} \max \{0, 1 - S(q|w) + S(t|w)\}. \tag{4}
\]

Specifically, the first term is the often used regularizer for SVM. And, the second is the hinge-loss for soft-counting how many pairs \( q > t \) are incorrectly ranked, which does not obey \( S(q|w) > S(t|w) + 1 \). Optimizing Eqn. 4 can map the 3D appearance voxel sets \( V_1, \ldots, V_T \) to a single vector \( w^* \). Actually, \( w^* \) encodes the dynamic evolution information from all the frames. Spatially reordering \( w^* \) from 1D to 3D in voxel form can construct 3DV for 3D action characterization. Thus, each 3DV voxel can be jointly encoded by the corresponding \( w^* \) item as motion feature and its regular 3D position index \((x, y, z)\) as spatial feature. Some more 3DV examples are shown in Fig. 5. We can intuitively observe that, 3DV can actually distinguish the different actions from motion perspective even human-object or human-human interaction happens. Meanwhile to accelerate 3DV extraction for application, the approximated temporal rank pooling [2] is used by us during implementation.

3.3. Temporal split

Applying temporal rank pooling to whole depth video may vanish some fine temporal order information. To better maintain motion details, we propose to execute temporal split for 3DV. The depth video will be divided into \( T_1 \) temporal splits with the overlap ratio of 0.5, which is the same as [46]. 3DV will extract from all the temporal splits and the whole depth video simultaneously as in Fig. 6, to involve the global and partial temporal 3D motion clues jointly.

3.4. Action proposal

Since background is generally not helpful for 3D action characterization, action proposal is also conducted by us, following [46] but with some minor modifications. First YOLOv3-Tiny [30] is used for human detection instead of Faster R-CNN [31], concerning running speed. Meanwhile, human and background are separated by depth thresholding. Particularly, depth value histogram is first extracted with the discretization interval of 100 mm. The interval of highest occurrence probability is then found. The threshold is empirically set as its median value plus 200 mm. Then, 3DV will be extracted only from action proposal’s 3D space.

4. Deep learning network on 3DV

After acquiring 3DV, the upcoming problem is how to conduct deep learning on it to conduct feature learning and 3D action type decision jointly. Since 3DV appears in 3D voxel form, an intuitive way is to apply 3D CNN to it as many 3D visual recognition methods [20, 7, 21] does. Nevertheless 3DV is generally hard to train, mainly due to its relatively large number of model parameters. Deep learning on point set (e.g., PointNet [24] and PointNet++ [25]) is the recently emerged research avenue to address the disordered characteristics of point set, with promising performance and lightweight model size. Inspired by this, we propose to apply PointNet++ to conduct
deep learning on 3DV instead of 3D CNN concerning effectiveness and efficiency jointly. To this end, 3DV will be abstracted into point set form. To our knowledge, using PointNet++ to deal with voxel data has not been well studied before. Meanwhile since 3DV tends to lose some appearance information as shown Fig. 4, a multi-stream deep learning model based on PointNet++ is also proposed to learn appearance and motion feature for 3D action characterization.

4.1. Review on PointNet++

PointNet++ [25] is derived from PointNet [24], the pioneer in deep learning on point set. PointNet is proposed mainly to address the disordered problem within point clouds. However, it cannot capture the local fine-grained pattern well. PointNet++ alleviates this in a local-to-global hierarchical learning manner. It declares 2 main contributions. First, it proposes to partition the set of points into overlap local regions to better maintain local fine 3D visual clue. Secondly, it uses PointNet recursively as the local feature learner. And, the local features will be further grouped into larger units to reveal the global shape characteristics. In summary, PointNet++ generally inherits the merits of PointNet but with stronger local fine-grained pattern descriptive power. Compared with 3D CNN, PointNet++ is generally of more light-weight model size and higher running speed. Meanwhile, it tends to be easier to train.

The main intuitions for why we apply PointNet++ to 3DV lie into 3 folders. First, we do not want to trap in the training challenges of 3D CNN. Secondly, PointNet++ is good at capturing local 3D visual pattern, which is beneficial for 3D action recognition. That is, local 3D motion pattern actually plays vital role for good 3D action characterization, as the hand region shown in Fig. 1 towards “Handshaking”. Last, applying PointNet++ to 3DV is not a difficult task. What we need to do is to abstract 3DV into the point set form, which will be illustrated next.

4.2. Abstract 3DV into point set

Suppose the acquired 3DV for a depth video without temporal split is of size $H \times W \times D$, each 3DV voxel $V(x, y, z)$ will possesses a global motion value $m_G$ given by temporal rank pooling as in Fig. 7 where $(x, y, z)$ indicates the 3D position index of $V(x, y, z)$ within 3DV. To fit PointNet++, $V(x, y, z)$ is then abstracted as a 3DV point $P(x, y, z)$ with the descriptive feature of $(x, y, z, m_G)$. Particularly, $(x, y, z)$ denotes the 3D spatial feature and $m$ is the motion feature. Thus, the yielded $P(x, y, z)$ is able to represent the 3D motion pattern and corresponding spatial information integrally. Since $(x, y, z)$ and $m$ are multi-modal feature, feature normalization is executed to balance their effect towards PointNet++ training. Specifically, $m$ is linearly normalized into the range of $[-0.5, 0.5]$. Towards the spatial feature, $y$ is first linearly normalized into the range of $[0, 0.5]$. Then $x$ and $z$ are re-scaled respectively, according to their size ratio towards $y$. In this way, the 3D geometric characteristics can be well maintained to alleviate distortion. As illustrated in Sec. 3.3, temporal split is executed in 3DV to involve multi-temporal motion information. Thus, each 3DV point $P(x, y, z)$ will correspond to multiple global and local motion values. We propose to concatenate all the motion values to describe 3D motion pattern integrally. $P(x, y, z)$ will be finally characterized by the spatial-motion feature as

$$F_{P(x,y,z)} = \frac{\text{Spatial}}{\text{Motion}} = (\bar{x}, \bar{y}, \bar{z}, m_G, m_1, ..., m_{T_1}),$$

where $m_G$ is the motion feature extracted from whole video; $m_i$ denotes motion feature from the $i$-th temporal split; and $T_1$ is the number of temporal splits as in Sec. 3.3.

4.3. Multi-stream network

Since 3DV may lose fine appearance clue, a multi-stream network using PointNet++ is proposed to learn motion and appearance feature jointly, following the idea in [37] for RGB video. As in Fig. 8, it consists of 1 motion stream and multiple appearance streams. The input of motion stream is the single 3DV point set from Sec. 4.2. For motion PointNet++ the 3DV points with all the motion features of 0 will not be sampled. And, the inputs of appearance streams are
the raw depth point sets sampled from $T_2$ temporal splits with raw action proposal. Particularly, they share the same appearance PointNet++. Motion and appearance feature is late fused via concatenation at fully-connected layer.

5. Implementation details

3DV voxel is set of size $35 \times 35 \times 35 \times 35 \text{mm}$. $T_1$ and $T_2$ is set to 4 and 3 respectively, for multi-temporal motion and appearance feature extraction. For PointNet++, farthest point sampling is used on the centroids of local regions. The sampled points are grouped with ball query. The group radius at the first and second level is set to 0.1 and 0.2 respectively. Adam [9] is applied as the optimizer with batch size of 32. Learning rate begins with 0.001, and decays with a rate of 0.5 every 10 epochs. Training will end when reaching 70 epochs. During training, we perform data augmentation for 3DV points and raw depth points including random rotation around $Y$ and $X$ axis, jittering and random points dropout. Multi-stream network is implemented using PyTorch. Within each stream, PointNet++ will sample 2048 points for both of motion and appearance feature learning.

6. Experiments

6.1. Experimental setting

Dataset: NTU RGB+D 120 [13]. It is the most recently emerged challenging 3D action recognition dataset, and also of the largest size. Particularly, 114,480 RGB-D action samples of 120 categories captured using Microsoft Kinect v2 are involved in this dataset. These involved action samples are of large variation on subject, imaging viewpoint and background. This imposes essential challenges to 3D action recognition. The accuracy of the state-of-the-art approaches is not satisfactory (i.e., below 70%) both under the cross-subject and cross-view setup evaluation criteria.

Dataset: NTU RGB+D 60 [33]. It is the preliminary version of NTU RGB+D 120. That is, 56,880 RGB-D action samples of 60 categories captured using Microsoft Kinect v2 are involved in this dataset. Before NTU RGB+D 120, it is the largest 3D action recognition dataset. Cross-subject and cross-view evaluation criteria is used for test.

Dataset: N-UCLA [41]. Compared with NTU RGB+D 120 and NTU RGB+D 60, this is a relatively small-scale 3D action recognition dataset. It only contains 1475 action samples of 10 action categories. These samples are captured using Microsoft Kinect v1 from 3 different viewpoints, with relatively higher imaging noise. Cross-view evaluation criteria is used for test.

Dataset: UWA3DII [26]. This is also a small-scale 3D action recognition dataset with only 1075 video samples from 30 categories. One essential challenge of this dataset is the limited number of training samples per action category. And, the samples are captured using Microsoft Kinect v1 with relatively high imaging noise.

Input data modality and evaluation metric. During experiments, the input data of our proposed 3DV based 3D action recognition method is only depth maps. We will not use any other auxiliary information, such as skeleton, RGB image, human mask, etc. The training / test sample splits and testing setups on all the 4 datasets are strictly followed for fair comparison. Classification accuracy on all the action samples is reported for performance evaluation.

6.2. Comparison with state-of-the-art methods

NTU RGB+D 120: Our 3DV based approach is compared with the state-of-the-art skeleton-based and depth-based 3D action recognition methods [13, 17, 16, 18, 4] on this dataset. The performance comparison is listed in Table 1. We can observe observed that:

- It is indeed impressive that, our proposition achieves the breaking-through results on this large-scale challenging dataset both towards the cross-subject and cross-setup test settings. Particularly we achieve 82.4% and 93.5% on these 2 settings respectively, which outperforms the state-of-the-art manners by large margins (i.e., 14.7% at least on cross-subject, and 26.6% at least on cross-setup). This essentially verifies the superiority of our proposition;
- The performance of the other methods is poor. This reveals the great challenges of NTU RGB+D 120 dataset;

---

**Table 1. Performance comparison on action recognition accuracy (%) among different methods on NTU RGB+D 120 dataset.**

<table>
<thead>
<tr>
<th>Methods</th>
<th>Cross-subject</th>
<th>Cross-setup</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTU RGB+D baseline [13]</td>
<td>55.7</td>
<td>57.9</td>
</tr>
<tr>
<td>GC-LSTM [16]</td>
<td>58.3</td>
<td>59.3</td>
</tr>
<tr>
<td>FSNet [14]</td>
<td>59.9</td>
<td>62.4</td>
</tr>
<tr>
<td>Two-stream attention LSTM [16]</td>
<td>61.2</td>
<td>63.3</td>
</tr>
<tr>
<td>Body Pose Evolution Map [18]</td>
<td>64.6</td>
<td>66.9</td>
</tr>
<tr>
<td>SkeletonMotion [9]</td>
<td>67.7</td>
<td>66.9</td>
</tr>
<tr>
<td>Input: 3D Skeleton</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTU RGB+D 120 baseline [13]</td>
<td>88.7</td>
<td>90.1</td>
</tr>
<tr>
<td>3DV-PointNet++ (ours)</td>
<td>82.4</td>
<td>93.5</td>
</tr>
</tbody>
</table>

**Table 2. Performance comparison on action recognition accuracy (%) among different methods on NTU RGB+D 60 dataset.**

<table>
<thead>
<tr>
<th>Methods</th>
<th>Cross-subject</th>
<th>Cross-view</th>
</tr>
</thead>
<tbody>
<tr>
<td>SkeletonMotion [9]</td>
<td>80.6</td>
<td>80.1</td>
</tr>
<tr>
<td>GC-LSTM [16]</td>
<td>74.4</td>
<td>82.8</td>
</tr>
<tr>
<td>Two-stream attention LSTM [16]</td>
<td>77.1</td>
<td>85.1</td>
</tr>
<tr>
<td>AGC-LSTM [35]</td>
<td>89.2</td>
<td>95.0</td>
</tr>
<tr>
<td>AS-GCN [11]</td>
<td>86.8</td>
<td>94.2</td>
</tr>
<tr>
<td>VA-fusion [51]</td>
<td>89.4</td>
<td>95.0</td>
</tr>
<tr>
<td>2s-AGCN [35]</td>
<td>88.5</td>
<td>95.1</td>
</tr>
<tr>
<td>DONS [3]</td>
<td>89.9</td>
<td>96.1</td>
</tr>
<tr>
<td>Input: Depth maps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HRNN [26]</td>
<td>70.6</td>
<td>73.6</td>
</tr>
<tr>
<td>SNV [19]</td>
<td>71.5</td>
<td>73.6</td>
</tr>
<tr>
<td>HOG [25]</td>
<td>32.2</td>
<td>22.1</td>
</tr>
<tr>
<td>Li. [10]</td>
<td>68.1</td>
<td>83.4</td>
</tr>
<tr>
<td>Wang. [42]</td>
<td>87.1</td>
<td>84.2</td>
</tr>
<tr>
<td>MYDT [40]</td>
<td>84.6</td>
<td>87.3</td>
</tr>
<tr>
<td>3DV-PointNet++ (ours)</td>
<td>88.8</td>
<td>96.3</td>
</tr>
</tbody>
</table>
Our method achieves better performance on cross-setup case than cross-subject. This implies that, 3DV is more sensitive to subject variation.

**NTU RGB+D 60**: The proposed method is compared with the state-of-the-art approaches [17, 16, 36, 11, 51, 35, 34, 23, 48, 22, 10, 42, 46] on this dataset. The performance comparison is listed in Table 2. We can see that:

- Our proposition still significantly outperforms all the depth-based manners, both on the cross-subject and cross-view test settings.
- On cross-view setting, the proposed method is also superior to all the skeleton-based manners. And, it is only slightly inferior to DGNN [34] on cross-subject setting; This reveals that, only using depth maps can still achieve the promising performance.
- By comparing Table 1 and 2, we can find that the performance of some methods (i.e., GCA-LSTM [17], Two stream attention LSTM [16],) significantly drops. Concerning the shared cross-subject setting, GCA-LSTM drops 16.1% and Two stream attention LSTM drops 15.9%. However, our manner only drops 6.4%. This demonstrates 3D-V’s strong adaptability and robustness.

**N-UCLA and UWA3DII**: We compared the proposed manner with the state-of-the-art depth-based approaches [23, 48, 41, 27, 46] on these 2 small-scale datasets. The performance comparison is given in Table 3 and 4 respectively. To save space, the average accuracy of the different viewpoint combinations is reported on UWA3DII. It can be summarized that:

- On these 2 small-scale datasets, the proposed approach still consistently outperforms the other depth-based manners. This demonstrates that, our proposition takes advantages over both of the large-scale and small-scale test cases;
- 3DV does not perform well on UWA3DII, with the accuracy of 73.2%. In our opinion, this may be caused by the fact that the training sample amount per class is limited on this dataset. Thus, deep learning cannot be well conducted.

### Table 3. Performance comparison on action recognition accuracy (%) among different depth-based methods on N-UCLA dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>HON4D [21]</td>
<td>99.9</td>
</tr>
<tr>
<td>SNV [33]</td>
<td>42.8</td>
</tr>
<tr>
<td>AoG [41]</td>
<td>33.6</td>
</tr>
<tr>
<td>HOPC [27]</td>
<td>80.0</td>
</tr>
<tr>
<td>MVDI [36]</td>
<td>84.2</td>
</tr>
<tr>
<td>3DV-PointNet++ (ours)</td>
<td>95.3</td>
</tr>
</tbody>
</table>

### Table 4. Performance comparison on action recognition accuracy (%) among different depth-based methods on UWA3DII dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Mean accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>HON4D [21]</td>
<td>28.9</td>
</tr>
<tr>
<td>SNV [33]</td>
<td>29.7</td>
</tr>
<tr>
<td>AoG [41]</td>
<td>26.7</td>
</tr>
<tr>
<td>HOPC [27]</td>
<td>52.2</td>
</tr>
<tr>
<td>MVDI [36]</td>
<td>88.1</td>
</tr>
<tr>
<td>3DV-PointNet++ (ours)</td>
<td>93.2</td>
</tr>
</tbody>
</table>

### 6.3. Ablation study

**Effectiveness of 3DV motion feature**: To verify this, we choose to remove 3DV motion feature from the sampled 3DV points within PointNet++ to observe the performance change. The comparison results on NTU RGB+D 120 dataset are given in Table 5. We can see that, without the motion feature 3DV’s performance will significantly drop (i.e., 18.5% at most).

**Effectiveness of temporal split for 3DV extraction**: Towards this, the temporal split number \( T \) is set to 1, 2, and 4 respectively on NTU RGB+D 120 dataset. The comparison results are listed in Table 6. Obviously, temporal split can essentially leverage the performance in all test cases.

**Effectiveness of appearance stream**: This is verified on NTU RGB+D 60 and 120 dataset simultaneously, as listed in Table 7. We can observe that:

- The introduction of appearance stream can consistently enhance the performance of 3DV on these 2 datasets, towards all the 3 test settings;
- 3DV stream significantly outperforms the appearance stream consistently, especially on cross-setup and cross-view settings. This verifies 3DV’s strong discriminative power for 3D action characterization in motion way.

**Effectiveness of Action proposal**: The performance comparison of our method with and without action proposal on N-UCLA dataset is listed in Table 8. Actually, action proposal can help to enhance performance.

**PointNet++ vs. 3D CNN**: To verify the superiority of PointNet++ for deep learning on 3DV, we compare it with
3D CNN, particularly, the well-established 3D CNN model (i.e., C3D [39]) for video classification is used with some modification. That is, the number of C3D’s input channels is reduced from 3 to 1. And, 3DV is extracted with the fixed size of $50 \times 50 \times 50$ grids as the input of C3D. Without data augmentation and temporal split, the performance and model complexity comparison on N-UCLA and NTU RGB+D 60 dataset (cross-view setting) is given in Table 9. We can see that, PointNet++ essentially takes advantage both on effectiveness and efficiency.

### 6.4. Parameter analysis

**Sampling point number on 3DV:** Before inputting 3DV point set into PointNet++, farthest point sampling is executed first. To investigate the choice of sampling point number, we compare the performance of 3DV stream with the different sampling point number values on NTU RGB+D 120 dataset. The results are listed in Table 10. That is, 2048 can achieve the best performance on 3DV.

**3DV voxel size:** To investigate the choice of 3D voxel size, we compare the performance of 3DV stream with the different 3DV voxel sizes on NTU RGB+D 120 dataset. The results are listed in Table 11. Particularly, $35mm \times 35mm \times 35mm$ is the optimal 3D voxel size.

### 6.5. Other issues

**Running time:** On the platform with CPU: Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.6GHz (only using 1 core), and GPU: 1 Nvidia RTX 2080Ti, 3DV’s overall online running time is 2.768s/video as detailed in Table 12. Particularly, 100 samples with the average length of 97.6 frames are randomly selected from NTU RGB+D 60 dataset for test.

**Approximated temporal rank pooling:** In our implementation, the approximated temporal rank pooling is used for 3DV extraction due to its high running efficiency. We compare it with the original one on N-UCLA data, using

**Table 9.** Comparison on performance and complexity between PointNet++ and C3D on N-UCLA and NTU RGB+D 60 dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
<th>FLOPs</th>
<th>N-UCLA</th>
<th>NTU RGB+D 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>C3D</td>
<td>29.2M</td>
<td>10.99G</td>
<td>64.5</td>
<td>85.0</td>
</tr>
<tr>
<td>PointNet++</td>
<td>1.24M</td>
<td>1.24G</td>
<td>71.3</td>
<td>90.0</td>
</tr>
</tbody>
</table>

**Table 10.** Performance comparison among the different sampling point numbers for 3DV, on NTU RGB+D 120 dataset.

<table>
<thead>
<tr>
<th>Sampling point number</th>
<th>Cross-subject</th>
<th>Cross-view</th>
</tr>
</thead>
<tbody>
<tr>
<td>512</td>
<td>74.9</td>
<td>89.0</td>
</tr>
<tr>
<td>1024</td>
<td>75.7</td>
<td>90.9</td>
</tr>
<tr>
<td>2048</td>
<td>76.9</td>
<td>92.5</td>
</tr>
</tbody>
</table>

**Table 11.** Performance comparison among the different 3DV voxel sizes, on NTU RGB+D 120 dataset.

<table>
<thead>
<tr>
<th>Voxel size (mm)</th>
<th>Cross-subject</th>
<th>Cross-view</th>
</tr>
</thead>
<tbody>
<tr>
<td>$25 \times 25 \times 25$</td>
<td>75.9</td>
<td>92.0</td>
</tr>
<tr>
<td>$35 \times 35 \times 35$</td>
<td>76.9</td>
<td>92.5</td>
</tr>
<tr>
<td>$50 \times 50 \times 50$</td>
<td>76.0</td>
<td>91.6</td>
</tr>
<tr>
<td>$75 \times 75 \times 75$</td>
<td>74.1</td>
<td>90.4</td>
</tr>
</tbody>
</table>

**Table 12.** Item-wise time consumption of 3DV per video.

<table>
<thead>
<tr>
<th>Unit</th>
<th>Item</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU: Human detection</td>
<td>2.31ms</td>
<td></td>
</tr>
<tr>
<td>GPU: 3DV pointization</td>
<td>10.9ms</td>
<td></td>
</tr>
<tr>
<td>CPU: PointNet++ forward</td>
<td>1.24ms</td>
<td></td>
</tr>
</tbody>
</table>

**CPU:** As shown in Table 13, the approximated temporal rank pooling runs much faster than the original one with the similar performance.

**3DV failure cases:** Some classification failure cases of 3DV are shown in Fig. 9. We find that, the failures tend to be caused by the tiny motion difference between the actions.

**7. Conclusions**

In this paper, 3DV is proposed as a novel and compact 3D motion representation for 3D action recognition. PointNet++ is applied to 3DV to conduct end-to-end feature learning. Accordingly, a multi-stream PointNet++ based network is also proposed to learn the 3D motion and depth appearance feature jointly to better characterize 3D actions. The experiments on 4 challenging datasets demonstrate the superiority of our proposition both for the large-scale and small-scale test cases. How to further enhance 3DV’s discriminative power is what we mainly concern about in future, especially towards the tiny motion patterns.

**Acknowledgment**

This work is jointly supported by the National Natural Science Foundation of China (Grant No. 61502187 and 61876211), Equipment Pre-research Field Fund of China (Grant No. 61403120405), the Fundamental Research Funds for the Central Universities (Grant No. 2019kfyXKJC024), National Key Laboratory Open Fund of China (Grant No. 6142113180211), 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.
References


[34] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqin g Lu. Skeleton-based action recognition with directed graph neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7912–7921, 2019. 1, 2, 6, 7


