Histogram of Oriented Principal Components for Cross-View Action Recognition

Hossein Rahmani, Arif Mahmood, Du Huynh and Ajmal Mian

School of Computer Science and Software Engineering, The University of Western Australia, 35 Stirling Highway, Crawley, Western Australia, 6009

Abstract

Existing techniques for 3D action recognition are sensitive to viewpoint variations because they extract features from depth images which are viewpoint dependant. In contrast, we directly process pointclouds for cross-view action recognition from novel unknown and unseen views. We propose Histogram of Oriented Principal Components (HOPC) descriptor which is robust to noise, viewpoint, scale and action speed variations. HOPC is extracted at a point by projecting all three eigenvectors of the pointcloud within a local spatio-temporal support volume, scaled by their corresponding eigenvalues, on the vertices of a regular dodecahedron. HOPC is also used to detect Spatio-Temporal Keypoints (STK) in 3D pointcloud sequences so that HOPC descriptors from only the relevant regions are used for action recognition. Finally, we propose a global descriptor that is extracted from the normalized spatio-temporal distribution of STKs in 4-D. STK-Distribution and local HOPC descriptors were evaluated for cross-view human action recognition on the Northwestern-UCLA and UWA3D Multiview Activity II datasets. Comparison with six cross-view action recognition algorithms shows that both descriptors individually outperform all existing techniques. Since HOPC and STK-Distribution capture complimentary information, their combination significantly improves classification accuracy which is over 22% higher than the nearest competitor on both datasets.

Keywords: Spatio-temporal keypoint, pointcloud, view invariance.

1. Introduction

Human action recognition has numerous applications in smart surveillance, human-computer interaction, sports and elderly care [1, 2]. Kinect like depth cameras have become popular for this task because depth sequences are somewhat immune to variations in illumination, clothing color and texture. However, the presence of occlusions, sensor noise, variations in action execution speed and most importantly sensor viewpoint still make action recognition challenging.

Designing an efficient representation for 3D video sequences is an important task for many computer vision problems. Most existing techniques (e.g., [3, 4, 5, 6, 7]) treat depth sequences similar to conventional videos and use color-based action recognition representations. However, simple extensions of color based action recognition techniques to depth sequences is not optimal [8, 9]. Instead of processing depth sequences, richer geometric features can be extracted from 3D pointcloud videos.

Email address: hossein@csse.uwa.edu.au, {arif.mahmood, du.huynh, ajmal.mian}@uwa.edu.au
(Hossein Rahmani, Arif Mahmood, Du Huynh and Ajmal Mian)
Action recognition research [10, 4, 11, 12, 8, 9, 5, 6, 7, 13, 14] has mainly focused on actions captured from a fixed viewpoint. However, a practical human action recognition system should be able to recognize actions from different views. Some view invariant approaches [15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28] have also been proposed for cross-view action recognition where recognition is performed from an unknown and/or unseen view. These approaches generally rely on geometric constraints [15, 16, 17, 18, 19], view invariant features [20, 21, 22, 23, 24, 25, 26], and human body joint tracking [27, 28]. More recent approaches transfer features across views [29, 30, 31, 32, 33, 34, 35, 36]. However, these methods do not perform as good as fixed view action recognition. The majority of cross-view action recognition research has focused on color videos or skeleton data. Cross-view action recognition from 3D video remains an under explored area. We believe that cross-view action recognition from 3D pointcloud videos holds more promise because view invariant features can be extracted from such videos.

We approach the cross-view action recognition problem from a novel perspective by directly processing the 3D pointcloud sequences (Fig. 1). We extend our previous research [37] where we proposed a new descriptor, the Histogram of Oriented Principal Components (HOPC), to capture the local geometric characteristics around each point in a 3D pointcloud sequence. Based on HOPC, we propose a view in-
variant Spatio-Temporal Keypoint (STK) detection method so that HOPC descriptors are extracted from the most discriminative points within a sequence of 3D pointclouds. In [37], we showed how HOPC outperforms existing techniques for front view action recognition. In this work, we focus on cross-view action recognition only and propose another descriptor extracted from the spatio-temporal distribution of STKs. This new descriptor alone outperforms existing techniques for cross view action recognition and when combined with HOPC, achieves state-of-the-art results.

To achieve view invariance for HOPC, all points within an adaptable spatio-temporal support volume of each STK are aligned along the eigenvectors of its spatial support volume. In other words, the spatio-temporal support volume is aligned in a local object centered coordinate basis. Thus, HOPC extracted from this aligned support volume is view invariant (Fig. 2). Note that this strategy does not necessary work for other descriptors as shown in Fig. 2. Humans often perform the same action at different speeds. For speed invariance, we propose automatic temporal scale selection that minimizes the eigenratios over a varying temporal window size independently at each STK.

Our four main contributions are summarized as follows: Firstly, we propose the HOPC descriptor which can encode shape and motion in a robust way. Secondly, we propose a view invariant spatio-temporal keypoint (STK) detector that is integrated with HOPC in the sense that it detects points that are suitable for HOPC. Thirdly, we propose a global action descriptor based on the spatio-temporal distribution of STKs. Finally, we propose a method for viewpoint and action speed invariant action recognition. Moreover, we introduce a new UWA3D Multiview Activity II dataset in addition to [37] which contains 30 actions performed by 10 subjects from four different views. This dataset is larger in number of action classes than existing 3D action datasets and will be made public.

The proposed method was evaluated on the Northwestern-UCLA Multiview Action3D dataset [29] and UWA3D Multiview Activity II dataset. Experimental results show that our method is able to achieve significantly better accuracy than current state-of-the-art methods for cross-view action recognition and is robust to action speed variations.

2. Related Work

Based on the data type, action recognition methods can be divided into three categories including color-based, skeleton-based and depth-based methods. In color videos, a significant portion of the existing work has been proposed for single view action recognition, where the training and test videos are captured from the same view. In order to recognize actions across viewpoint changes, one approach is to collect data from all possible views and train a separate classifier for each view. However, this approach does not scale well due to the requirement of a large number of labeled samples for each view and it becomes infeasible as the number of action categories increases. To overcome this problem, some techniques infer 3D scene structure and use geometric transformations to achieve view invariance [15, 16, 17, 18, 19]. These methods critically rely on accurate detection of the body joints and contours, which are still open problems in real-world settings. Other methods focus on spatio-temporal features which are inherently view invariant [20, 21, 22, 23, 24, 25, 26]. However, these methods have limitations as some of them require access to mocap data while others compromise discriminative power to achieve
Figure 2: After orientation normalization, the HOPC descriptor is similar for the two views. However, the HON and HOG descriptors are still different.
More recently, knowledge transfer based methods [29, 30, 31, 32, 33, 34, 35, 36] have become popular. These methods find a view independent latent space in which features extracted from different views are directly comparable. Such methods are either not applicable or perform poorly when the recognition is performed on videos from unknown and, more importantly from, unseen views. To overcome this problem, Wang et al. [29] proposed cross-view action representation by exploiting the compositional structure in spatio-temporal patterns and geometrical relations among views. Although their method can be applied to action recognition from unknown and unseen views, it requires 3D skeleton data for training which is not always available. Our proposed approach also falls in this category except that it does not require skeleton data. Our method can perform action recognition in videos acquired from unknown and unseen views. Our approach is designed for 3D pointclouds captured by depth sensors. To the best of our knowledge, we are the first to propose cross-view action recognition using 3D pointcloud videos.

In skeleton-based action recognition methods, multi-camera motion capture (MoCap) systems [39] have been used for human action recognition. However, such specialized equipment is marker-based and expensive. On the other hand, some other methods [40, 41, 28, 13, 27, 14] use the human joint positions extracted by the OpenNI tracking framework [42]. For example, Yang and Tian [14] used pairwise 3D joint position differences in each frame and temporal differences across frames to represent an action. Since 3D joints cannot capture all the discriminative information, the action recognition accuracy is compromised. Wang et al. [41] extended this approach by computing the histogram of occupancy pattern of a fixed region around each joint in each frame. In order to make this method more robust to viewpoint variations, they proposed a global orientation normalization using the skeleton data [28]. In this method, a plane is fitted to the joints and a rotation matrix is computed to rotate this plane to the $XY$-plane. However, this method is only applicable if the subject is in an up-right pose. Moreover, when the subject is in non-frontal view, the joint positions may have large errors making the normalization process unreliable. In contrast, our proposed orientation normalization method does not need the joint positions and can efficiently work in non-frontal as well as non up-right positions. In addition to that, our method performs local orientation normalization at each STK, which is also more robust than a single global normalization proposed by [28].

Many of the existing depth-based action recognition methods use global features such as silhouettes and space-time volume information. For example, Li et al. [43] sampled boundary pixels from 2D silhouettes as a bag of features. Yang et al. [7] added temporal derivative of 2D projections to get Depth Motion Maps (DMM). Vieira et al. [44] computed silhouettes in 3D by using the space-time occupancy patterns. Oreifej and Liu [9] extended histogram of oriented 3D normals [45] to 4D by adding time derivative. Recently, Yang and Tian [10] extended HON4D by concatenating the 4D normals in the local neighbourhood of each pixel as its descriptor. Our proposed descriptor HOPC is more informative than HON4D [37], because it captures the spread of data in three principal directions. Holistic methods may fail in scenarios where subjects significantly changes their spatial position [10, 9]. Some other methods use local features where a set of interest points are extracted from the depth sequence and a local feature descriptor is computed for each interest point. For example, Cheng et al. [3] used the Cuboid...
interest point detector [46] and proposed a Comparative Coding Descriptor (CCD). Due to the presence of noise in depth sequences, simply extending color-based interest point detectors such as Cuboid [47], 3D Hessian [48] and 3D Harris [46], degrade the efficiency and effectiveness of these detectors as most interest points are detected at irrelevant locations [8, 9].

Motion trajectory based action recognition methods [49, 50, 51, 20] are also not reliable in depth sequences [9]. Therefore, recent depth based action recognition methods resorted to alternative ways to extract more reliable interest points. Wang et al. [52] proposed Haar features to be extracted from each random subvolume. Xia and Aggarwal [8] proposed a filtering method to extract spatio-temporal interest points. Their approach fails when the action execution speed is faster than the flip of the signal caused by sensor noise. Moreover, both techniques are not robust to viewpoint variations.

3. HOPC: Histogram of Oriented Principal Components

HOPC is extracted at each point within a sequence of 3D pointclouds \( Q = \{Q_t, \cdots, Q_t, \cdots, Q_{n_f}\} \), where \( n_f \) denotes the number of 3D pointclouds in the sequence and \( Q_t \) is the 3D pointcloud at time \( t \), where \( 1 \leq t \leq n_f \). Consider a point \( p = (x_t, y_t, z_t)^\top \) in \( Q_t \). We define two different support volumes for \( p \): spatial support volume and spatio-temporal support volume. The spatial support volume of \( p \), denoted by \( \Omega_p \), contains the 3D points in \( Q_t \) which are in a sphere of radius \( r \) centered at \( p \) (Fig. 3(b)). To define the spatio-temporal support volume of \( p \), denoted by \( \Omega_p \), we make a spatio-temporal accumulated 3D pointcloud by merging the sequence of individual pointclouds in the small time interval \( [t - \tau, t + \tau] \). The 3D points in the spatio-temporal accumulated 3D pointcloud which are in a sphere of radius \( r \) centered at \( p \) are considered as \( \Omega_p \) (Fig. 3(c)).

The scatter matrix \( C_p \) of the points \( q \in \Omega_p \), \( y \in \{ST, S\} \) is given by:

\[
C_p = \frac{1}{n_p} \sum_{q \in \Omega_p} (q - \mu)(q - \mu)^\top,
\]

where

\[
\mu = \frac{1}{n_p} \sum_{q \in \Omega_p} q,
\]

and \( n_p = |\Omega_p| \) denotes the number of points in the support volume of \( p \). Performing eigenvalue decomposition on the scatter matrix \( C_p \) gives us:

\[
V_p E_p V_p^\top = C_p,
\]

where \( E_p \) is a diagonal matrix of the eigenvalues \( \lambda_1 \geq \lambda_2 \geq \lambda_3 \), and \( V_p \) contains three orthonormal eigenvectors \( [v_1^p \ v_2^p \ v_3^p] \) arranged in the order of decreasing magnitude of their associated eigenvalues.

The HOPC descriptor is built by projecting each eigenvector onto \( m \) directions obtained from the vertices of a regular polyhedron. In particular, we consider a regular dodecahedron which is composed of \( m = 20 \) vertices, each of which corresponds to a histogram bin. Let \( U = [u_1, u_2, \cdots, u_i, \cdots, u_m] \in \mathbb{R}^{3 \times m} \) be the matrix of the vertices \( u_1, u_2, \cdots, u_m \) of a regular dodecahedron. For a regular dodecahedron with center at the origin, these vertices are given as:

• 8 vertices obtained by \((\pm 1, \pm 1, \pm 1)\)
Figure 3: Spatio Temporal Keypoint (STK) detection. (a) A 3D pointcloud sequence corresponding to the holding head action, (b) the spatial support volume of the particular point $p$, (c) the spatio-temporal support volume of the particular point $p$, (d) the HOPC descriptors, (e) STK detection.
• 4 vertices obtained by $(0, \pm \varphi^{-1}, \pm \varphi)$
• 4 vertices obtained by $(\pm \varphi^{-1}, \pm \varphi, 0)$
• 4 vertices obtained by $(\pm \varphi, 0, \pm \varphi^{-1})$

where $\varphi = (1 + \sqrt{5})/2$ is the golden ratio.

Each eigenvector is a direction in the 3D space representing the distribution of point positions in the support volume. Therefore, its orientation has a $180^\circ$ ambiguity. Each point in the support volume may also be considered as a vector. To resolve the orientation ambiguity, we consider the distribution of point vector directions and their magnitudes within the support volume of $p$. We determine the sign of each eigenvector $v^y_j$ from the sign of the inner products of $v^y_j$ and $o = q - p$, where $q \in \mathcal{O}(p)$ are point vectors within the support volume of $p$:

$$v^y_j = v^y_j \cdot \text{sign} \left( \sum_{q \in \mathcal{O}(p)} \text{sign}(o^T v^y_j)(o^T v^y_j)^2 \right),$$

where the sign function returns the sign of an input number. Note that the squared projection ensures that small projections, which are often due to noise, are suppressed. If the signs of eigenvectors $v^y_1$, $v^y_2$, and $v^y_3$ disagree i.e. $v^y_1 \times v^y_2 \neq v^y_3$, we switch the sign of the eigenvector whose $|v^y_j|$ value is the smallest. We then project each eigenvector $v^y_j$ onto $U$ to give us:

$$b^y_j = U^T v^y_j \in \mathbb{R}^m, \text{ for } 1 \leq j \leq 3. \quad (4)$$

In case $v^y_j$ is perfectly aligned with $u_i \in U$, it should vote into only the $i$th bin. However, as the $u_i$'s are not orthogonal to each other, $b^y_j$ will have non-zero projection in other bins as well. To overcome this effect, we quantize the projection of $b^y_j$. For this purpose, a threshold value $\psi$ is computed by projecting any two neighbouring vectors $u_k$ and $u_l$ as follows:

$$\psi = u_k^T u_l = \varphi + \varphi^{-1}, \text{ for } u_k, u_l \in U. \quad (5)$$

Note that for any $u_k \in U$, we can find a $u_i \in U$ such that $\psi = \varphi + \varphi^{-1}$. The quantized vector is given by

$$\hat{b}^y_j(z) = \begin{cases} 0 & \text{if } b^y_j(z) \leq \psi \\ b^y_j(z) - \psi & \text{otherwise,} \end{cases}$$

where $1 \leq z \leq m$ denotes a bin number. For the $j$th eigenvector, we define $h^y_j$ to be $\hat{b}^y_j$ scaled by the corresponding eigenvalue $\lambda^y_j$:

$$h^y_j = \frac{\lambda^y_j \cdot \hat{b}^y_j}{||\hat{b}^y_j||_2} \in \mathbb{R}^m, \text{ for } 1 \leq j \leq 3, \quad (6)$$

where $y \in \{S, ST\}$. We concatenate the histograms of oriented principal components of the three eigenvectors in decreasing order of magnitude of their associated eigenvalues to form a descriptor for point $p$:

$$h^y_p = \left[ h^y_1 \, h^y_2 \, h^y_3 \right]^T \in \mathbb{R}^{3m}. \quad (7)$$
The spatial HOPC descriptor, \( h^S_p \), encodes the shape of the support volume around \( p \). On the other hand, the spatio-temporal HOPC descriptor, \( h^{ST}_p \), encodes information from both shape and motion. The smallest principal component of the local surface is the total least squares estimate of the surface normal \([53]\). Therefore, our descriptor inherently encodes the surface normal which is more robust to noise than gradient-based surface normals used in \([45, 9]\). Moreover, HOPC additionally encodes the first two eigenvectors which are more dominant compared to the third one. The spatial and spatio-temporal HOPC are shown in Fig. 3(d). Using the spatial and spatio-temporal HOPC descriptor, we propose a view invariant STK detector and action descriptor.

4. Spatio-Temporal Keypoint (STK) Detector

The aim of STK detection is to find points in 3D pointcloud action sequences which satisfy three constraints:

- **Repeatability**: STKs should be identified with high repeatability in different samples of the same action in the presence of noise and viewpoint changes.

- **Uniqueness**: A unique coordinate basis should be obtained from the neighbourhood of the STKs for the purpose of view invariant description.

- **Significant spatio-temporal variation**: STKs should be detected where the neighbourhood has significant space-time variations.

To achieve these aims, we propose an STK detection technique which has high repeatability, uniqueness and detects points where space-time variation is significant. Consider a point \( p = (x_t, y_t, z_t)^T \) within a sequence of 3D pointclouds. We perform eigenvalue decomposition on both the spatial and the spatio-temporal scatter matrices \( C^S \) and \( C^{ST} \). Let \( \lambda^S_1 \geq \lambda^S_2 \geq \lambda^S_3 \) and \( \lambda^{ST}_1 \geq \lambda^{ST}_2 \geq \lambda^{ST}_3 \) represent the eigenvalues of \( C^S \) and \( C^{ST} \), respectively. To achieve the first two constraints, we define some ratios:

\[
\delta_{12}^S = \frac{\lambda^S_1}{\lambda^S_2}, \quad \delta_{23}^S = \frac{\lambda^S_2}{\lambda^S_3}, \quad \delta_{12}^{ST} = \frac{\lambda^{ST}_1}{\lambda^{ST}_2}, \quad \delta_{23}^{ST} = \frac{\lambda^{ST}_2}{\lambda^{ST}_3}. \tag{8}
\]

For 3D symmetrical surfaces, the ratio between the first two eigenvalues or last two eigenvalues will be very close to 1. Therefore, the principal components at such locations are ambiguous. Thus, for a point to qualify as a potential keypoint, the condition

\[
\{ \delta_{12}^S, \delta_{23}^S, \delta_{12}^{ST}, \delta_{23}^{ST} \} > \theta_{STK} > 1 + \epsilon_{STK}, \tag{9}
\]

must be satisfied, where \( \epsilon_{STK} \) is a small margin to cater for noise. This process prunes ambiguous points and produces a subset of candidate keypoints which can be described uniquely in a local coordinate basis.

Recall that \( h^S_p \) in Eq. (7) represents the spatial HOPC and \( h^{ST}_p \) the spatio-temporal HOPC at point \( p \). To achieve the third constraint, a quality factor \( \eta_p \) is computed for all candidate keypoints:

\[
\eta_p = \frac{1}{2} \sum_{i=1}^{3m} \frac{(h^S_p(i) - h^{ST}_p(i))^2}{(h^S_p(i) + h^{ST}_p(i))}. \tag{10}
\]
When $h_p^S = h_p^{ST}$, the quality factor is at the minimum value of $\eta_p = 0$ which basically means that the candidate point $p$ has a stationary spatio-temporal support volume. On the other hand, significant variations in both space and time change the direction and magnitude of spatio-temporal eigenvectors with respect to the spatial eigenvectors. Thus, $\eta_p$ is high when a significant motion occurs in the spatio-temporal support volume.

Closely located STKs are essentially similar as they describe more or less the same local support volume. We perform a non-maximum suppression to keep a minimum distance between STKs and define a locality of radius $r'$ (with $r' \ll r$) and time interval $2\tau'+1$ (with $\tau' \leq \tau$). The candidate STKs are sorted according to their quality values and starting from the highest quality STK, all STKs within its locality are discarded. The same process is repeated on the remaining STKs until only the desired number of $n_k$ STKs are left. Fig. 3 shows the steps of our STK detection algorithm. Fig. 4 shows the extracted STKs from four different views for a 3D pointcloud sequence corresponding to the two hand waving action.

5. View-Invariant STK Description

The HOPC descriptor discussed in Section 3 is not view-invariant yet. For view-invariance, we perform orientation normalization at each STK using the eigenvectors of its spatial scatter matrix $C^S$. We consider the eigenvectors $V^S = [v_1^S \, v_2^S \, v_3^S]$ of $C^S$ as a local object centered coordinate basis. Note that the matrix $V^S$ is orthonormal and can be used as a valid 3D rotation matrix:

$$v_i^S \cdot v_j^S = \begin{cases} 
1 & \text{if } i = j \\
0 & \text{if } i \neq j
\end{cases}$$

We apply a 3D rotation $R = V^S^T$ on all points within the spatio-temporal support volume of $p$, i.e., $\Omega^{ST}(p)$, and bring them to a canonical coordinates:

$$B = RP^T,$$

where $P \in \mathbb{R}^{n_p \times 3}$ is a matrix of points within $\Omega^{ST}(p)$ and $B \in \mathbb{R}^{3 \times n_p}$ denotes the rotated points in the local object centered coordinate basis. Note that the first, second, and third principal components are now aligned along $X$, $Y$ and $Z$ axes of the Cartesian coordinates. Since the same STKs in two different views have the same canonical representation; therefore, we can do cross-view keypoint matching (Fig. 2). It is important to note that our STK detection algorithm has already pruned ambiguous points to make the local object centered coordinate basis unique, i.e. no two eigenvectors have the same eigenvalues. Therefore, the eigenvector with maximum eigenvalue will always map to the $X$ axis, the second biggest to the $Y$ axis and the smallest to the $Z$ axis.

The final STK descriptor is obtained by computing HOPC for all points within the spatio-temporal support volume of STK and averaging over spatio-temporal cells. In addition to viewpoint normalization, we prune the ambiguous eigenvectors of each point $p'$ in the spatio-temporal support volume of STK $p$ to make the final descriptor more discriminative and robust to noise. To eliminate ambiguous eigenvectors
of each point, we define two eigenratios:

\[
\delta_{12}^{ST} = \frac{\lambda_1^{ST}}{\lambda_2^{ST}}, \quad \delta_{23}^{ST} = \frac{\lambda_2^{ST}}{\lambda_3^{ST}}.
\] (13)

The principal components of symmetric surfaces are ambiguous because the order of eigenvectors is not unique. To get a discriminative \(h_{p}^{ST}\), the values of \(\delta_{12}^{ST}\) and \(\delta_{23}^{ST}\) must be greater than 1. However, to account for noise we use a threshold \(\theta_1 > 1 + \epsilon_1\), where \(\epsilon_1\) is a small margin, and select only the discriminative eigenvectors as:

1. If \(\delta_{12}^{ST} > \theta_1\) & \(\delta_{23}^{ST} > \theta_1\), \(h_{p}^{ST} = [h_1^{ST} \ T \ h_2^{ST} \ T \ h_3^{ST}]^\top;\)

2. If \(\delta_{12}^{ST} \leq \theta_1\) & \(\delta_{23}^{ST} < \theta_1\), \(h_{p}^{ST} = [0 \ T \ h_3^{ST}]^\top;\)

3. If \(\delta_{12}^{ST} < \theta_1\) & \(\delta_{23}^{ST} \leq \theta_1\), \(h_{p}^{ST} = [h_1^{ST} \ T \ 0 \ T]^\top;\)

4. If \(\delta_{12}^{ST} \leq \theta_1\) & \(\delta_{23}^{ST} < \theta_1\), discard \(p\).

After view normalization and ambiguous eigenvectors elimination, we encode the information from both shape and motion at each STK. The orientation normalized spatio-temporal support volume around STK is divided into \(\gamma = n_x \times n_y \times n_t\) spatio-temporal cells along \(X\), \(Y\), and \(T\) dimensions. We use \(c_s\), where \(s = 1 \cdots \gamma\), to denote the \(s\)th cell. The spatio-temporal HOPC descriptor \(h_{p}^{ST}\) in Eq. (7) is computed for each point \(p' \in c_s\). The cell descriptor \(h_{c_s}\) is computed by accumulating

\[
h_{c_s} = \sum_{p' \in c_s} h_{p'}^{ST}
\] (14)

and then normalizing

\[
h_{c_s} \leftarrow h_{c_s} / \|h_{c_s}\|_2.
\] (15)

The final view invariant descriptor \(h_v\) for the given STK is a concatenation of \(h_{c_s}\) obtained from all the cells:

\[
h_v = [h_{c_1}^\top \ h_{c_2}^\top \ \ldots \ h_{c_s}^\top \ \ldots \ h_{c_{\gamma}}^\top]^\top.
\] (16)

Thus, the STK descriptor encodes view-invariant spatio-temporal patterns that will be used for action description.

6. Action Description

6.1. Bag of STK Descriptors

We represent each sequence of 3D pointclouds by a set of STK descriptors. Inspired by the successful bag-of-words approach for object recognition, we build a codebook by clustering the STK descriptors \((h_v)\) with K-means. Clustering is performed over all action descriptors extracted from all training view samples. Thus, the codebook we learn is not single action specific or single view specific. For fair evaluation, we do not use the target test views in codebook learning or any other training task. We consider each cluster as a codeword that represents a specific spatio-temporal pattern shared by the
Figure 4: STKs projected onto $XYZ$ dimensions of all points of a 3D pointcloud sequence corresponding to the two hand waving action. Four different views are shown. Note that the distribution of STKs encodes the action globally as they are detected only where movement is performed.

STKs in that cluster. One codeword is assigned to each STK descriptor based on the minimum Euclidean distance. Histogram of codewords is used as an action descriptor. For classification, we use an SVM classifier with histogram intersection kernel [54].

6.2. Mining Discriminative Codebooks

Not all codewords have the same level of discrimination. Some codewords may encode movements that do not offer good discrimination among different actions. We use $F$-score to find the most discriminative codewords. $F$-score [55] measures the discrimination of two sets of real numbers. For more than two sets of real numbers, we use the multiset F-score [56] to measure their discrimination. Given the training histogram of codewords $x_k$, $k = 1, \cdots, m$, and the number of action classes $l$ ($l \geq 2$). If the number of the samples in $j$th ($1 \leq j \leq l$) class is $n_j$, then the $F$-score of the $i$th histogram bin is defined as:

$$F_i = \frac{\sum_{j=1}^{l} \left( \bar{x}_i^{(j)} - \bar{x}_i \right)^2}{\sum_{j=1}^{l} \frac{1}{n_j-1} \sum_{k=1}^{n_j} \left( \bar{x}_{k,i}^{(j)} - \bar{x}_i \right)^2}$$

(17)

where $\bar{x}_i$ and $\bar{x}_i^{(j)}$ are the average of the $i$th histogram bin of all samples and the $j$th class samples, respectively. $\bar{x}_{k,i}^{(j)}$ is the $i$th histogram bin of the $k$th sample in the $j$th class. The larger the $F$-score, the more discriminative is the corresponding histogram bin. Therefore, we rank the codewords by their $F$-scores and select the codewords with $F$-score higher than a threshold.

6.3. Encoding Spatio-Temporal STK Distribution

The bag-of-words approach efficiently encodes the local spatio-temporal information in a 3D pointcloud sequence. However, it ignores the spatio-temporal relationship among the STKs. We observed that
the distribution of STKs in space-time (see Fig. 4) can further improve discrimination between different actions in addition to the bag-of-words based descriptors. To incorporate the space-time positional information of STKs, we propose a method that encodes this information.

Let \( Q = \{ Q_1, Q_2, \cdots, Q_{t}, \cdots, Q_n \} \) represent a sequence of 3D pointclouds and \( P \in \mathbb{R}^{n_k \times 4} \) represent a set of all selected STKs within \( Q \), where \( n_k \) denotes the number of STKs. Let \( p_i = (x, y, z, t)^T \) denotes a position of an STK in the 4D space where \( X \) and \( Y \) are spatial coordinates, \( Z \) represents depth and \( T \) is time. The first three dimensions are in millimetres and the last one is in seconds. To remove this inconstancy, we use feature normalization to make the values of each variable zero-mean

\[
p_i \leftarrow p_i - \mu_p, \quad \text{for } i = 1 \cdots n_k, \quad (18)
\]

where

\[
\mu_p = \frac{1}{n_k} \sum_{i=1}^{n_k} p_i, \quad (19)
\]

and unit-variance

\[
p_i \leftarrow p_i \oplus \sigma_p, \quad \text{for } i = 1 \cdots n_k, \quad (20)
\]

where

\[
\sigma_p = \sqrt{\frac{1}{n_k} \sum_{i=1}^{n_k} (p_i - \mu_p) \otimes (p_i - \mu_p)}, \quad (21)
\]

\( \oplus \) and \( \otimes \) denote element-wise matrix division and multiplication, respectively.

We make an accumulated 3D pointcloud of STKs by merging all normalized 4D STKs in \( P \) i.e. removing the time axis of STKs in \( P \). Let \( P' \in \mathbb{R}^{n_k \times 3} \) represent a set of normalized STKs after removing time axis and \( C \) be the scatter matrix of 3D STKs in \( P' \). Performing eigenvalue decomposition on the scatter matrix \( C \) gives us \( VEV^T = C \), where \( E \) is a diagonal matrix of the eigenvalues \( \lambda_1 \geq \lambda_2 \geq \lambda_3 \), and \( V \) contains three orthogonal eigenvectors \( [v_1, v_2, v_3] \) arranged in the order of decreasing magnitude of their associated eigenvalues. To eliminate the orientation ambiguity of the eigenvectors \( v_1, v_2, v_3 \), the sign disambiguation technique (Eq. 3) is used to find a unique orientation for each eigenvector.

For a unique orientation of STK distribution in space-time, the eigenratios \( \lambda_1/\lambda_2 \) and \( \lambda_2/\lambda_3 \) must be greater than 1. We ensure that \( \{ \lambda_1/\lambda_2, \lambda_2/\lambda_3 \} > \theta_g > 1+\epsilon_g \) (where \( \epsilon_g \) is a small constant) constraints are satisfied for all STK distributions. In case the ratios \( \lambda_1/\lambda_2 \) or \( \lambda_2/\lambda_3 \) do not satisfy these constraints, we perform an iterative refinement of STKs. Given \( n_k \) total STKs, in each iteration, \( m_k \) (where \( m_k < n_k \)) STKs with lowest quality factor (Eq. (10)) are removed until the eigenratio constraints are satisfied. To achieve view invariant representation, all points in \( P' \) are aligned along \( V \):

\[
P' \leftarrow P'V, \quad (22)
\]

and the temporal dimension is reattached to each point in \( P' \):

\[
\hat{p}_i \leftarrow [p'_i, t_i], \quad (23)
\]

where \( t_i \) is the normalized frame number of point \( p'_i \).
Table 1: The 600 vertices of a 120-cell regular polychoron centered at the origin generated from all and even permutations of these coordinates [57].

| Vertices | Perm. | Coordinate points |
|----------|-------|-------------------|
| 24       | all   | 0, 0, ±2, ±2      |
| 64       | all   | ±1, ±1, ±1, ±√5   |
| 64       | all   | ±φ⁻², ±φ, ±φ, ±φ  |
| 64       | all   | ±φ⁻¹, ±φ⁻¹, ±φ⁻¹, ±φ² |
| 96       | even  | 0, ±φ⁻², ±1, ±φ⁺² |
| 96       | even  | 0, ±φ⁻¹, ±φ, ±√5  |
| 192      | even  | ±φ⁻¹, ±1, ±φ, ±2  |

To encode the distribution of STKs in 4D space, we consider a 4D regular geometric object called polychoron [57] which is a 4D extension of the 2D polygon. The vertices of a regular polychoron divide the 4D space uniformly and therefore, each vertex can be considered as a histogram bin. In particular, from the set of regular polychorons, we consider the 120-cell regular polychoron with 600 vertices as given in Table 1 [57].

Given \( \mathbf{p} \in \mathbb{R}^{n \times 4} \) of a 3D pointcloud sequence, we project each view normalized \( \mathbf{p}_i \) on the 600 vertices of polychoron and select the vertex with the highest projection value. The histogram bin corresponding to the selected vertex is incremented by one. We repeat this process for all STKs and the final histogram is a descriptor which encodes the global spatio-temporal distribution of STKs of sequence \( Q \) in a compact and discriminative form.

7. Adaptable Support Volume

So far, for STK detection and description, we used a fixed spatio-temporal support volume with spatial radius \( r \) and temporal scale \( \tau \). However, subjects may have different scales (in height and width) and may perform an action at different speeds. Therefore, simply using a fixed spatial radius \( r \) and temporal scale \( \tau \) is not optimal. Moreover, a large value of \( r \) enables the proposed descriptor to encapsulate more information about shape but makes the descriptor vulnerable to occlusions. Similarly, a small \( \tau \) is preferable over large \( \tau \) for better temporal action localization. However, a small \( \tau \) may not capture sufficient information about an action if it is performed slowly.

7.1. Spatial Scale Selection

Several automatic spatial scale selection methods have been proposed for 3D object retrieval [58]. We adapt the method proposed by Mian et al. [59] for object retrieval to action recognition in 3D pointcloud sequences. Note that in case of human action recognition, the subject’s height is available in most cases (which is not the case for the object retrieval). When available, we use the subject’s height (\( h_s \)) to find an appropriate spatial scale. We select the ratios as \( r = e h_s \), where \( 0 < e < 1 \) is a constant factor. We have empirically selected the value of \( e \) to maximize the descriptiveness and robustness of our descriptor. 
to occlusions. In all experiments, we use a fixed value of \( e \) for all actions, views and datasets. In our experiment in section 8 we observe that this simple approach achieves almost the same accuracy as the automatic spatial scale selection method adapted from [59]. Once we have selected an appropriate spatial scale \( r \), then we proceed to select an appropriate temporal scale \( \tau \).

7.2. Automatic Temporal Scale Selection

Most existing action recognition technique use a fix temporal scale \([5, 9, 8, 6, 44, 47]\). We observed that variations in action execution speed causes significant disparity among the descriptors from the same action (Fig. 5). To make our descriptor robust to action speed variations, we propose an automatic temporal scale selection technique. Let \( Q = \{Q_1, Q_2, \cdots, Q_t, \cdots, Q_n\} \) represent a sequence of 3D pointclouds. For a given point \( p = [x, y, z]^T \in Q_t \), we select all points in the neighbouring pointclouds \([Q_{t-\tau}, \cdots, Q_t, \cdots, Q_{t+\tau}]\) which are within the spatial radius \( r \) of \( p \):

\[
\Omega_r(p) = \bigcup_{t-\tau}^{t+\tau} \Omega^S(p) \in Q_{t-\tau},
\]

where \( \Omega^S(p) \in Q_{t-\tau} \) denotes the points in the 3D pointcloud \( Q_{t-\tau} \) which are within the radius \( r \) of \( p \), and \( \Omega_r(p) \) is the union of all the points within the spatial radius \( r \) of \( p \) over all temporal pointclouds in \([Q_{t-\tau}, \cdots, Q_t, \cdots, Q_{t+\tau}]\). In other words, we map the same coordinates of \( p \in Q_t \) to multiple pointcloud video frames \([Q_{t-\tau}, \cdots, Q_t, \cdots, Q_{t+\tau}]\) and take the union of all points within spatial radius \( r \) in each pointcloud. The scatter matrix from the accumulated points is computed and eigenvalue decomposition is performed to find the eigenvalues (\( \lambda_1 \geq \lambda_2 \geq \lambda_3 \)). Then we calculate:

\[
A_p(\tau) = \frac{\lambda_2}{\lambda_1} + \frac{\lambda_3}{\lambda_2}.
\]

We start temporal scale selection with \( \tau = 1 \) and repeat until \( \tau = \tau_m \), which is a fixed upper threshold. An appropriate temporal scale \( \tau_o \) for the point \( p \) corresponds to the global minimum value of \( A_p \) over the range \( 1 \leq \tau \leq \tau_m \). Fig. 5 shows the same action performed at three different speeds. Fig. 5(a)-(c) shows temporal sequences of pointclouds. The dotted circle shows the sphere define by the spatial radius \( r \) in each pointcloud. We obtain the same spatial radius \( r \) in three cases because of similar geometry. The aim is to select an appropriate temporal scale for a point \( p \) in the middle pointcloud \( (Q_t) \) shown in black and dotted outline. Fig. 5(d) shows the union of points from \([Q_{t-3}, \cdots, Q_{t+3}]\) which are within the radius \( r \) measured from the coordinate \((x, y, z)\) of point \( p \). Fig. 5(e) and (f) show the union of points in the same way for \( \tau = 2 \) and \( \tau = 1 \), respectively. Fig. 5(g)-(i) show the plots of \( A_p \) with the variation of \( \tau \). Increasing \( \tau \) beyond a certain value does not affect the accumulated pointcloud as the value of \( A_p \) becomes constant. In most cases, increasing \( \tau \) decreases \( A_p \) until a fixed value is reached. We compute \( A_p \) for all values of \( \tau \) and find the global minimum \( \tau_o \) which is the smallest value of \( \tau \) at which \( A_p \) achieves its minimum value.

For each STK, temporal scale is selected independently and may vary from one STK to the other in the same 3D pointcloud sequence. The proposed temporal scale selection is detailed in Algorithm 1. The algorithm outputs two variables \( \tau_o \) and \( flag \in \{0, 1\} \). \( flag = 0 \) means the minimum value of \( A_p \).
Figure 5: Same action (hand movement) is shown at three different speeds: (a) slow, (b) moderate, and (c) fast. The number of frames reduces as the action speed increases. For the slow movement, an appropriate temporal scale is found to be $\tau_o = 3$, for moderate movement $\tau_o = 2$, and for fast movement $\tau_o = 1$.

corresponds to $\tau_m$, therefore point $p$ should be discarded. $flag = 1$ means that $\tau_o$ is the appropriate temporal scale for point $p$.

8. Experiments

The proposed algorithm was evaluated on the Northwestern-UCLA Multiview Action3D [29], and UWA3D Multiview Activity II datasets. We compare our performance to the state-of-the-art cross-view action recognition methods including Comparative Coding Descriptor (CCD) [3], Virtual Views (VV) [60], Histogram of 3D Joints (HOJ3D) [27], Discriminative Virtual Views (DVV) [36], Actionlet Ensemble (AE) [28], and AND-OR graph (AOG) [29]. HOJ3D [27] uses only skeleton data, while AE [28] uses both skeleton and depth data. AOG [29] uses skeleton data and depth videos in training phase and only depth videos in test phase. For CCD [3], VV [60] and DVV [36], we use DSTIP [8] to extract and describe the spatio-temporal interest points, which is more robust to 3D sensor noise as compared to color-based interest point detector [8].

Although our algorithm is not sensitive to parameter values, we provide these values in Table 2 for reproducing the reported results. To compute the HOPC descriptor, the spatio-temporal support volume around each STK is divided into $n_x \times n_y \times n_t (2 \times 2 \times 3)$ spatio-temporal cells as discussed in Section 5.

8.1. Northwestern-UCLA Multiview Action3D Dataset

This dataset contains RGB, depth and human skeleton positions captured simultaneously by three Kinect cameras. It consists of 10 action categories: (1) pick up with one hand, (2) pick up with two hands, (3) drop trash, (4) walk around, (5) sit down, (6) stand up, (7) donning, (8) doffing, (9) throw,
Algorithm 1: Automatic Temporal Scale Selection

input : $Q$, $p$, $r$, and $\tau_m$.

output: $\tau_o$, flag.

1 for $\tau = 1 : \tau_m$ do

2 $\Omega_\tau(p) = \bigcup_{\tau'} \Omega^S_\tau(p) \in Q_{\tau'}$;

3 $\mu_\tau = \frac{1}{n_p} \sum_{q_r \in \Omega_\tau(p)} q_\tau; C_\tau = \frac{1}{n_p} \sum_{q_r \in \Omega_\tau(p)} (q_\tau - \mu_\tau)(q_\tau - \mu_\tau)^T$;

4 $V_\tau = \begin{pmatrix} \lambda_1^\tau & 0 & 0 \\ 0 & \lambda_2^\tau & 0 \\ 0 & 0 & \lambda_3^\tau \end{pmatrix};$

5 $A_p(\tau) = \frac{\lambda_1^\tau}{\lambda_1^\tau} + \frac{\lambda_2^\tau}{\lambda_2^\tau};$

6 end

7 $\tau^* = \arg\min_{\tau} A_p(\tau)$;

8 if $\tau^* = \tau_m$ then

9 $flag = 0;$

10 else

11 $flag = 1;$

12 $\tau_o = \tau^*(1);$  

13 end

Table 2: Parameter values: number of codewords $K$, number of STKs $n_k$, thresholds $\theta_{STK}$, $\theta_l$, $\theta_g$, maximum temporal scale $\tau_m$, and iterative refinement process parameters $m_k$.

| Parameter | $K$ | $n_k$ | $\theta_{STK}$ | $\theta_l$ | $\theta_g$ | $\tau_m$ | $m_k$ |
|-----------|-----|------|----------------|-----------|-----------|---------|------|
| Value     | 1500| 400  | 1.3            | 1.3       | 1.3       | 0.25$n_k$| 0.05$n_k$ |
Figure 6: Sample pointclouds from the Northwestern-UCLA Multiview Action3D dataset [29] captured by cameras (top row) $C_1$, (middle row) $C_2$, and (last row) $C_3$. Each column shows a different action namely, drop trash, pick up with one hand, carry and pick up with two hands.

and (10) carry. Each action is performed by 10 subjects from 1 to 6 times. Fig. 6 shows 12 sample 3D pointclouds of four actions captured from three views.

To compare our method with state-of-the-art algorithms, we use the samples from one camera as training data, and the samples from the two remaining cameras as test data. The comparison of the recognition accuracy for six possible combinations of training and test cameras is shown in Table [3]. We report the recognition accuracy of our algorithm in three different settings. In the first setting (STK-D), for each sequence of 3D pointclouds we only use the histogram of spatio-temporal distribution of STKs as the sequence descriptor. In the second setting (HOPC), we only use the bag of STK descriptors. In the third setting (HOPC+STK-D), we concatenate the bag of STK descriptors and the histogram of spatio-temporal distribution of STKs.

We observe a higher average accuracy by only using STK-D. This proves the spatio temporal keypoints are repeatable and robust to viewpoint changes. Despite significant view changes, a large number of STKs were detected on the correct corresponding positions in the sequences. Moreover, the action descriptors obtained from the histograms of STK positions capture more discriminative action information compared to the other methods. In the second experiment with HOPC descriptor, we consistently achieve
Table 3: Comparison with state-of-the-art methods on the Northwestern-UCLA Multiview Action3D dataset. Note that our STK-Distribution based global descriptor alone outperforms existing methods.

| Training camera | $C_1$ | $C_2$ | $C_3$ | Mean |
|-----------------|-------|-------|-------|------|
| Test camera     | $C_2$ | $C_3$ | $C_1$ | $C_3$ | $C_1$ | $C_2$ |
|-----------------|-------|-------|-------|------|-------|-------|
| CCD [3]         | 22.0  | 28.1  | 20.8  | 27.7 | 15.3  | 15.9  | 21.6  |
| VV [60]         | 26.2  | 27.4  | 25.0  | 38.6 | 23.2  | 35.7  | 29.4  |
| HOJ3D [27]      | 28.8  | 25.8  | 30.0  | 40.5 | 26.5  | 37.9  | 31.6  |
| DVV [36]        | 28.7  | 30.6  | 30.0  | 40.4 | 27.1  | 36.9  | 32.3  |
| AE [28]         | 30.3  | 30.5  | 32.3  | 44.9 | 26.6  | 40.0  | 34.1  |
| AOG [29]        | 25.6  | 36.7  | 25.9  | 37.0 | 33.5  | 34.1  | 32.1  |
| STK-D           | 32.9  | 37.6  | 32.1  | 48.5 | 33.2  | 42.3  | 39.8  |
| HOPC            | 50.8  | 54.5  | 48.5  | 60.0 | 46.0  | 51.2  | 44.2  |
| HOPC+STK-D      | 57.2  | 58.4  | 52.7  | 62.4 | 51.3  | 56.5  | 56.4  |

Higher accuracy than STK-D and other methods. Note that HOPC and STK-D capture complementary information which explains the increased performance obtained by their combination (HOPC+STK-D).

Among the current approaches, CCD [3] achieved quite low accuracy because it encodes the differences between the depth values of an interest point and its neighbourhood points which is sensitive to viewpoint variation as depth is a function of viewpoint. Skeleton based methods HOJ3D [27] and AE [28] have achieved high accuracy on single view action recognition datasets. However, they could not achieve similar performance on this multiview dataset because the skeleton data is not accurate when the subject is not in frontal view and upright position. The overall accuracy of the knowledge transfer based methods VV [60] and DVV [36] have also remained low. This could be because the viewpoint dependent local features were not able to discriminate intra action differences induced by viewpoint variations from inter action differences. Another knowledge transfer method AOG [29] obtained excellent accuracy in cross-view action recognition on color videos. However, its performance was similar to other knowledge transfer methods on depth videos. This may be due to the high level of noise in depth videos compared to color videos. Interpolating noisy features across views can compromise discrimination ability.

The confusion matrix of the proposed method when samples from camera $C_2$ are used as training and samples from camera $C_3$ are used as test data is shown in Fig. 7. The action (7) donning and action (8) doffing have maximum confusion with each other because the motion and appearance of these actions are very similar. Similarly, action (1) pick up with one hand and action (9) throw have high confusion due to similarity in motion and appearance. Also, action (1) pick up with one hand has some confusion with action (4) walk around due to some walking performed within the pick up with one hand action.
8.2. UWA3D Multiview Activity II Dataset

This dataset has been collected in our lab using Kinect to emphasize three points: (1) Larger number of human activities. (2) Each subject performed all actions in continuous manner with no breaks or pauses. Therefore, the start and end positions of body for the same actions are different. (3) Each subject performed the same actions four times while imaged from four different views: front view, left and right side views, and top view.

This dataset consists of 30 human activities performed by 10 subjects with different scales: (1) one hand waving, (2) one hand Punching, (3) two hand waving, (4) two hand punching, (5) sitting down, (6) standing up, (7) vibrating, (8) falling down, (9) holding chest, (10) holding head, (11) holding back, (12) walking, (13) irregular walking, (14) lying down, (15) turning around, (16) drinking, (17) phone answering, (18) bending, (19) jumping jack, (20) running, (21) picking up, (22) putting down, (23) kicking, (24) jumping, (25) dancing, (26) moping floor, (27) sneezing, (28) sitting down (chair), (29) squatting, and (30) coughing. To capture depth videos, each subject performed 30 activities 4 times in a continuous manner. Each time, the Kinect was moved to a different angle to capture the actions from four different views. Note that this approach generates more challenging data than when actions are captured simultaneously from different viewpoints. We organize our dataset by segmenting the continuous sequences of activities. The dataset is challenging because of varying viewpoints, self-occlusion and high similarity among activities. For example, the actions (16) drinking and (17) phone answering have very similar motion, but the location of hand in these two actions is slightly different. Also, some actions such as: (10) holding head and (11) holding back, have self-occlusion. Moreover, in the top view, the lower part of the body is not properly captured because of occlusion. Fig. 8 shows 20 sample pointclouds of five actions from four views.

For cross-view action recognition, we use videos from two views as training and videos from the two remaining views as test data. Video from only one view is provided at test time. Table 4 summarizes our
Figure 8: Sample pointclouds from the UWA3D Multiview Activity II dataset. Row wise from top: front view \((V_1)\), left view \((V_2)\), right view \((V_3)\), and top view \((V_4)\). Each column shows a different action namely (2) one hand punching, (12) walking, (16) drinking, (17) phone answering, and (26) moping floor.
results. Our STK distribution encoding technique (STK-D) achieves higher average recognition accuracy than current methods. By only using the proposed HOPC descriptor, we observe higher accuracy than STK-D. Concatenating HOPC with STK-D significantly increase the average recognition accuracy by 8.2%. The proposed HOPC and HOPC+STK-D methods outperform state-of-the-art algorithms for all training-test data combinations. The recognition accuracy of skeleton based methods HOJ3D [27] and AE [28] are low because the skeleton data is not accurate for many actions (e.g. (16) drinking, (17) phone answering, (27) sneezing) or is not available (e.g. (8) falling down, (14) lying down). Moreover, the overall accuracy of the knowledge transfer based methods VV [60], DVV [36], AOG [29] is low because motion and appearance of many actions are very similar and the depth sequences have a high level of noise. Therefore, the view dependent local features used in VV [60], DVV [36] and the appearance and motion interpolation based method used in AOG [29] are not enough to discriminate subtle differences between actions in the presence of noise.

Fig. 9 shows the confusion matrix of the proposed method when videos from view $V_1$ and view $V_2$ are used for training and videos from view $V_3$ are used as test data. The actions that cause most confusion are (9) holding chest versus (11) holding back and (12) walking versus (13) irregular walking, because the motion and appearance of these two actions are very similar.

8.3. Effects of Adaptable Support Volume

8.3.1. Spatial Scale Selection

In this experiment, we evaluate the influence of three different approaches for spatial scale selection at each STK. In the first approach, we use a constant spatial scale for all subjects. In the second approach, we select a scale for each subject relative to the subject’s height. In the third one, we use the automatic spatial scale selection method proposed by Mian et al. [59]. Table 5 shows the average accuracy of

Figure 9: Confusion matrix of our algorithm on the UWA3D Multiview Activity II dataset when view $V_1$ and view $V_2$ are used for training and view $V_3$ is used for test.
Table 4: Comparison with state-of-the-art methods under all combinations of selecting the training and test views on the UW A3D Multiview Activity II dataset.

| training views | $V_1$ & $V_2$ | $V_1$ & $V_3$ | $V_1$ & $V_4$ | $V_2$ & $V_3$ | $V_2$ & $V_4$ | $V_3$ & $V_4$ | Mean |
|----------------|---------------|---------------|---------------|---------------|---------------|---------------|------|
| test view      | $V_1$         | $V_2$         | $V_3$         | $V_4$         | $V_1$         | $V_2$         | $V_3$ |
| CCD [3]        | 10.5          | 13.6          | 10.3          | 12.8          | 11.1          | 8.3           | 10.0 |
| VV [60]        | 20.2          | 22.0          | 19.9          | 22.3          | 19.3          | 20.5          | 20.8 |
| HOJ3D [27]     | 15.3          | 28.2          | 17.3          | 27.0          | 14.6          | 13.4          | 15.0 |
| DVV [36]       | 23.5          | 25.9          | 23.6          | 26.9          | 22.3          | 20.2          | 22.1 |
| AE [28]        | 18.4          | 27.2          | 21.1          | 30.3          | 16.8          | 16.0          | 15.0 |
| AOG [29]       | 29.3          | 31.1          | 25.3          | 29.9          | 22.7          | 21.9          | 25.0 |
| STK-D          | 32.8          | 25.1          | 38.7          | 22.7          | 23.4          | 29.0          | 19.2 |
| HOPC           | 42.3          | 46.5          | 39.1          | 49.8          | 35.0          | 39.3          | 51.9 |
| HOPC+STK-D     | **52.7**      | **51.8**      | **59.0**      | **57.5**      | **42.8**      | **44.2**      | **58.1** |

Table 5: Average recognition accuracy of the proposed method in three different settings on the Northwestern-UCLA Multiview Action3D [29] and the UW A3D Multiview Activity II datasets. (1) Constant spatial scale for all subjects, (2) ratio of subject’s height as the spatial scale, and (3) automatic spatial scale selection [59].

| Dataset   | Spatial Scale Selection Method | Constant | Subject height | Automatic |
|-----------|--------------------------------|----------|----------------|-----------|
| N-UCLA    | 53.9                           | 56.4     | 55.5           |
| UW3A3DII  | 48.0                           | 52.2     | 50.9           |

8.3.2. **Automatic Temporal Scale Selection**

We evaluate the improvement gained in our method by using automatic temporal scale selection by repeating our experiments with constant temporal scale for STK detection and HOPC descriptor extraction. Table 6 shows the average recognition accuracy of our proposed method using constant temporal scale ($\tau = 2$) and using automatic temporal scale selection. The proposed automatic temporal scale selection technique achieves higher accuracy which shows the robustness of our method to action speed variations.
Table 6: Average recognition accuracy of the proposed method in two different settings on both datasets. (1) Constant temporal scale ($\tau = 2$) and (2) our automatic temporal scale selection technique.

| Dataset    | Temporal Scale Selection Method | Constant | Automatic |
|------------|---------------------------------|----------|-----------|
| N-UCLA     | 54.0                            | 56.4     |
| UWA3DII    | 49.2                            | 52.2     |

Figure 10: Parameter evaluation around optimum value of (a) the number of STKs, (b) $\theta_{STK}$, (c) $\theta_1$, and (d) $\theta_2$ on the Northwestern-UCLA Multiview Action3D [29] and the UWA3D Multiview Activity II datasets.

8.4. Parameters Evaluation

8.4.1. Number of STKs

To study the effect of the total number of STKs ($n_k$), we select STKs with the top 100, 400, 700, 1000 quality factors as shown in Fig. 11. Note how our proposed STK detector effectively captures the movement of the hands in the highest quality STKs and noisy points begin to appear as late as $n_k = 1000$. This demonstrates the effectiveness of our STK detector. Fig. 10(a) shows the influence of number of STKs on the average recognition accuracy of the proposed method. Our proposed method achieves the best recognition accuracy, when $n_k = 400$.

8.4.2. Threshold Values

We evaluate the effects of various thresholds ($\theta_{STK}$ in Eq. 9, $\theta_1$ in Section 5, and $\theta_2$ in Section 6.3) on the average recognition accuracy of our proposed method. Fig. 10(b) shows the recognition accuracy versus $\theta_{STK}$. Notice that there is a large range ($1.1 \geq \theta_{STK} \leq 1.5$) over which the recognition accuracy remains stable. For very small values of $\theta_{STK}$, a unique coordinate basis is not obtained and for larger values of $\theta_{STK}$, sufficient number of STKs are not detected.
Figure 11: Spatio-Temporal Keypoints (STK) extracted using our proposed detector. STKs are projected on $XYZ$ dimensions of all points within a 3D pointcloud sequence corresponding to the action *two hand waving*. The top $n_k = 100, 400, 700, 1000$ with the best quality are shown in (a)-(d), respectively. Note that the highest quality STKs are detected where significant movement is performed. Noisy points begin to appear as late as $n_k = 1000$.

Next, we varied $\theta_l$ and $\theta_g$ from 1 to 2. Fig. 10(c) and (d) show the average recognition accuracy versus $\theta_l$ and $\theta_g$. A more stable trend in recognition accuracy is observed for varying values of these two parameters. The recognition accuracy starts to decrease when $\theta_l > 1.5$ because the number of points within the spatio-temporal support volume of STKs which have unique eigenvectors start to decrease. Finally, varying $\theta_g$ does not change the accuracy significantly because the extracted STKs from most actions already have a unique orientation and the proposed iterative refinement process discussed in Section 6.3 almost always finds an unambiguous orientation for all values of $\theta_g$.

9. Conclusion

Performance of current 3D action recognition techniques degrades in the presence of viewpoint variations because they tread 3D videos as depth image sequences. Depth images are defined with respect to some viewpoint and their pixel values change significantly under viewpoint variations. We proposed an algorithm for cross-view action recognition which directly processes 3D pointcloud videos. Our method is robust to viewpoint and action speed variations. We also proposed a HOPC descriptor that is well integrated with our spatio-temporal keypoint (STK) detection algorithm. The distribution of STKs alone outperforms existing techniques on cross-view action recognition. Local HOPC descriptors combined with global STK-Distribution achieves state of the art results on two standard cross-view 3D action recognition datasets.
Acknowledgment

We thank the authors of [29] for providing the Northwestern-UCLA Multiview Action3D dataset and especially Dr Jiang Wang for answering our questions about the implementation of AOG [29] and AE [28] methods. We also thank the authors of [8, 36, 60] for making their codes publicly available. This research is supported by ARC Discovery grant DP110102399.

References

[1] J. K. Aggarwal, M. S. Ryoo, Human activity analysis: A review, in: ACM Computing Survey, 2011.

[2] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition, in: CVIU, 2011.

[3] Z. Cheng, L. Qin, Y. Ye, Q. Huang, Q. Tian, Human daily action analysis with multi-view and color-depth data, in: ECCVW, 2012.

[4] C. Lu, J. Jia, C. keung Tang, Range-sample depth feature for action recognition, in: CVPR, 2014.

[5] H. Rahmani, A. Mahmood, D. Huynh, A. Mian, Action classification with locality-constrained linear coding, in: ICPR, 2014.

[6] H. Rahmani, A. Mahmood, A. Mian, D. Huynh, Real time action recognition using histograms of depth gradients and random decision forests, in: WACV, 2014.

[7] X. Yang, C. Zhang, Y. Tian, Recognizing actions using depth motion maps-based histograms of oriented gradients, in: ACM ICM, 2012.

[8] L. Xia, J. Aggarwal, Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera, in: CVPR, 2013.

[9] O. Oreifej, Z. Liu, HON4D: histogram of oriented 4D normals for activity recognition from depth sequences, in: CVPR, 2013.

[10] X. Yang, Y. Tian, Super normal vector for activity recognition using depth sequences, in: CVPR, 2014.

[11] H. Zhang, W. Zhou, C. Reardon, L. E. Parker, Simplex-based 3d spatio-temporal feature description for action recognition, in: CVPR, 2014.

[12] Y. Tian, R. Sukthankar, M. Shah, Spatiotemporal deformable part models for action detection, in: CVPR, 2013.

[13] M. Zanfir, M. Leordeanu, C. Sminchisescu, The Moving Pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection, in: ICCV, 2013.
[14] X. Yang, Y. Tian, EigenJoints-based action recognition using naive bayes nearest neighbor, in: CVPRW, 2012.

[15] A. Yilmaz, M. Shah, Action sketch: a novel action representation, in: CVPR, 2005.

[16] T. Syeda-Mahmood, A. Vasilescu, S. Sethi, Action recognition from arbitrary views using 3D exemplars, in: ICCV, 2007.

[17] F. Lv, R. Nevatia, Single view human action recognition using key pose matching and viterbi path searching, in: CVPR, 2007.

[18] D. Gavrila, L. Davis, 3D model-based tracking of humans in action: a multi-view approach, in: CVPR, 1996.

[19] T. Darrell, I. Essa, A. Pentland, Task-specific gesture analysis in real-time using interpolated views, in: PAMI, 1996.

[20] B. Li, O. Camps, M. Sznaier, Cross-view activity recognition using hankelets, in: CVPR, 2012.

[21] V. Parameswaran, R. Chellappa, View invariance for human action recognition, in: IJCV, 2006.

[22] C. Rao, A. Yilmaz, M. Shah, View-invariant representation and recognition of actions, in: IJCV, 2002.

[23] T. Syeda-Mahmood, A. Vasilescu, S. Sethi, Recognizing action events from multiple viewpoints, in: IEEE Workshop on Detection and Recognition of Events in Video, 2001.

[24] S. Seitz, C. Dyer, View-invariant analysis of cyclic motion, in: IJCV, 1997.

[25] M. Blank, L. Gorelick, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, in: ICCV, 2005.

[26] D. Weinland, R. Ronfard, E. Boyer, Free viewpoint action recognition using motion history volumes, in: CVIU, 2006.

[27] L. Xia, C. C. Chen, J. K. Aggarwal, View invariant human action recognition using histograms of 3D joints, in: CVPRW, 2012.

[28] J. Wang, Z. Liu, Y. Wu, J. Yuan, Learning actionlet ensemble for 3d human action recognition, in: PAMI, 2013.

[29] J. Wang, X. Nie, Y. Xia, Y. Wu, S. Zhu, Cross-view action modeling, learning and recognition, in: CVPR, 2014.

[30] A. Gupta, J. Martinez, J. J. Little, R. J. Woodham, 3d pose from motion for cross-view action recognition via non-linear circulant temporal encoding, in: CVPR, 2014.

[31] J. Zheng, Z. Jiang, Learning view-invariant sparse representations for cross-view action recognition, in: ICCV, 2013.
[32] Z. Zhang, C. Wang, B. Xiao, W. Zhou, S. Liu, C. Shi, Cross-view action recognition via a continuous virtual path, in: CVPR, 2013.

[33] A. Farhadi, M. K. Tabrizi, Learning to recognize activities from the wrong view point, in: ECCV, 2008.

[34] A. Farhadi, M. K. Tabrizi, I. Endres, D. A. Forsyth, A latent model of discriminative aspect, in: ICCV, 2009.

[35] J. Liu, M. Shah, B. Kuipersy, S. Savarese, Cross-view action recognition via view knowledge transfer, in: CVPR, 2011.

[36] R. Li, T. Zickler, Discriminative virtual views for cross-view action recognition, in: CVPR, 2012.

[37] H. Rahmani, A. Mahmood, D. Q Huynh, A. Mian, HOPC: Histogram of oriented principal components of 3D pointclouds for action recognition, in: ECCV, 2014.

[38] X. Yang, Y. Tian, A survey of vision-based methods for action representation, segmentation and recognition.

[39] L. Campbell, A. Bobick, Recognition of human body motion using phase space constraints, in: ICCV, 1995.

[40] R. Vemulapalli, F. Arrate, R. Chellappa, Human action recognition by representing 3d skeletons as points in a lie group, in: CVPR, 2014.

[41] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, in: CVPR, 2012.

[42] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, in: CVPR, 2011.

[43] W. Li, Z. Zhang, Z. Liu, Action recognition based on a bag of 3D points, in: CVPRW, 2010.

[44] A. W. Vieira, E. Nascimento, G. Oliveira, Z. Liu, M. Campos, STOP: space-time occupancy patterns for 3D action recognition from depth map sequences, in: CIARP, 2012.

[45] S. Tang, X. Wang, X. Lv, T. Han, J. Keller, Z. He, M. Skubic, S. Lao, Histogram of oriented normal vectors for object recognition with a depth sensor, in: ACCV, 2012.

[46] I. Laptev, On space-time interest point, in: IJCV, 2005.

[47] P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: ICCV, 2005.

[48] G. Willems, T. Tuyltelaars, L. Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, in: ECCV, 2008.

[49] H. Wang, C. Schmid, Action recognition with improved trajectories, in: ICCV, 2013.
[50] H. Wang, A. Klaser, C. Schmid, C.-L. Liu, Action recognition by dense trajectories, in: CVPR, 2011.

[51] S. Wu, O. Oreifej, M. Shah, Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories, in: ICCV, 2011.

[52] J. Wang, Z. Liu, J. Chorowski, Z. Chen, Y. Wu, Robust 3D action recognition with random occupancy patterns, in: ECCV, 2012.

[53] N. J. Mitra, A. Nguyen, Estimating surface normals in noisy point clouds data, in: SCG, 2003.

[54] S. Maji, A. C. Berg, J. Malik, Classification using intersection kernel support vector machines is efficient, in: CVPR, 2008.

[55] Y. W. Chen, C. J. Lin, Combining svms with various feature selection strategies, in: Feature Extraction, Studies in Fuzziness and Soft Computing, Springer Berlin Heidelberg, 2006.

[56] J. Xie, C. Wang, Using support vector machines with a novel hybrid feature selection method for diagnosis of erythemato-squamous diseases, in: Expert Systems with Applications, 2011.

[57] H. S. M. Coxeter, Regular polytopes, Dover Publications, 1973.

[58] F. Timbari, L. D. Stefano, Performance evaluation of 3D keypoint detectors, in: IJCV, 2013.

[59] A. Mian, M. Bennamoun, R. Owens, On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes, in: IJCV, 2010.

[60] R. Gopalan, R. Li, R. Chellapa, Doamin adaption for object recognition: An unsupervised approach, in: ICCV, 2011.