# Histogram of Oriented Principal Components for Cross-View Action Recognition

Hossein Rahmani, Arif Mahmood, Du Huynh, Member, IEEE, and Ajmal Mian, Member, IEEE

Abstract—Existing techniques for 3D action recognition are sensitive to viewpoint variations because they extract features from depth images which are viewpoint dependent. In contrast, we directly process pointclouds for cross-view action recognition from unknown and unseen views. We propose the Histogram of Oriented Principal Components (HOPC) descriptor that is robust to noise, viewpoint, scale and action speed variations. At a 3D point, HOPC is computed by projecting the three scaled eigenvectors of the pointcloud within its local spatio-temporal support volume for the detection of Spatio-Temporal Keypoints (STK) in 3D point HOPC descriptors) at these key locations only are used for ac from the normalized spatio-temporal distribution of STKs in 4-D, of our proposed descriptors against nine existing techniques or datasets. The Experimental results show that our techniques private states. The Experimental results show that our techniques private states. The Experimental results show that our techniques private states. The Experimental results show that our techniques private states. The Experimental results show that our techniques private states. The Experimental results show that our techniques private states. The Experimental results show that our techniques private states. The Experimental results show that our techniques private states. The Experimental results show that our techniques private states. The Experimental results show that our techniques private states. The Experimental results show that our techniques private states. The Experimental results show that our techniques private states. The Experimental results show that our techniques private states. The Experimental results show that our techniques private states. The Experimental results show that our techniques private states. The Experimental results show that our techniques private states the experimental results show that our techniques private states. The Experimental results show that our techniques private states the experimental results show that our techniques private states the experimental results show that our techniques private states the experimental results show that our techniques private states the experimental results show that our techniques private states the experimental results show that our techniques private states the experimental results show that our techniques private states the experimental results show that our techniques private states the experimental results show that our techniques private st of the pointcloud within its local spatio-temporal support volume onto the vertices of a regular dodecahedron. HOPC is also used for the detection of Spatio-Temporal Keypoints (STK) in 3D pointcloud sequences so that view-invariant STK descriptors (or Local HOPC descriptors) at these key locations only are used for action recognition. We also propose a global descriptor computed from the normalized spatio-temporal distribution of STKs in 4-D, which we refer to as STK-D. We have evaluated the performance of our proposed descriptors against nine existing techniques on two cross-view and three single-view human action recognition datasets. The Experimental results show that our techniques provide significant improvement over state-of-the-art methods.

Index Terms-Spatio-temporal keypoint, pointcloud, view invariance.

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 se-quences is an important task for many computer vision problems. Most existing techniques (*e.g.* [3]–[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 are not optimal [8], [9]. Instead of processing depth sequences, richer geometric features can be extracted from 3D pointcloud videos.

Action recognition research [4]–[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 viewinvariant approaches [15]-[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]-[19], view-invariant features [20]–[26], and human body joint tracking [27], [28]. More recent approaches transfer

The authors are with the School of Computer Science and Software Engineering, The University of Western Australia, 35 Stirling Highway, Crawley, Western Australia, 6009. E-mail: hossein@csse.uwa.edu.au, {arif.mahmood,du.huynh,ajmal.mian}@uwa.edu.au



Fig. 1: 3D pointcloud sequences of a subject performing the holding head action. Notice how the depth values (color) change significantly with viewpoint.

features across views [29]-[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 videos remains an under explored area. We believe that cross-view action recognition from 3D pointcloud videos holds more promise because viewinvariant 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 Spatio-Temporal Keypoint (STK) detection method so that view-invariant Local HOPC descriptors are extracted from the most discriminative points within a sequence of 3D pointclouds. We also propose another descriptor, STK-D, which is computed from the spatio-temporal distribution of the STKs. Since Local HOPC and STK-D capture complementary information, their combination significantly improves the cross-view action recognition accuracy over existing state-of-the-art.

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 descriptor extracted from this aligned support volume is view-invariant (Fig. 2). Note that this strategy does not necessarity work for other descriptors as shown in Fig. 2. As 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 encodes 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 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.

The proposed descriptors have been evaluated on two multi-view and three single-view human action recognition datasets. The former includes the Northwestern-UCLA Multiview Action3D [29] and the UWA3D Multiview Activity II datasets whereas the latter includes MSR Action3D [38], MSR Daily Activity3D [39], and MSR Gesture3D [40] datasets. Our extensive experimental results show that the proposed descriptors achieved significantly better accuracy compared to the nine existing state-of-the-art techniques [3], [9], [10], [27]–[29], [36], [59], [60].

# 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 different views, 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



Fig. 2: After orientation normalization, the HOPC descriptors are similar for the two views. However, the HON and HOG descriptors are still different.

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]–[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]–[26]. However, these methods have limitations as some of them require access to mocap data while others compromise discriminative power to achieve view invariance [41].

More recently, knowledge transfer based methods [29]– [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 uses 3D pointcloud sequences and does not require skeleton data. 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, multicamera motion capture (MoCap) systems [42] have been used for human action recognition. However, such specialized equipment is marker-based and expensive. On the other hand, some other methods [13], [14], [27], [28], [39], [59] use the human joint positions extracted by the OpenNI tracking framework [43]. 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. [39] extended this approach by computing the histogram of occupancy patterns 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 upright pose. Moreover, when the subject is in a 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 upright positions. In addition to that, as our method performs local orientation normalization at each STK, it is more robust than the single global normalization proposed by [28].

Many of the existing depth-based action recognition methods use global features such as silhouettes and spacetime volume information. For example, Li et al. [38] sampled boundary pixels from 2D silhouettes as a bag of features. Yang et al. [7] added temporal derivatives of 2D projections to get Depth Motion Maps (DMM). Vieira et al. [44] computed silhouettes in 3D by using the spacetime occupancy patterns. Oreifej and Liu [9] extended histogram of oriented 3D normals [45] to 4D by adding the 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 HOPC descriptor is more informative than HON4D [37] because it captures the spread of data in three principal directions. Holistic methods may fail in scenarios where the subject significantly changes her/his spatial position [9]. [10]. 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], degrades the efficiency and effectiveness of these detectors as most interest points are detected at irrelevant locations [8], [9].

Motion trajectory based action recognition methods [20], [49]–[51] 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. [40] 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 PRIN-**CIPAL COMPONENTS

HOPC is extracted at each point within a sequence of 3D pointclouds  $Q = seq(Q_1, \dots, Q_t, \dots, Q_{n_f})$ , where  $n_f$  denotes the number of 3D pointclouds in the sequence and  $Q_t$  is the 3D pointcloud at time t. Consider a point  $\mathbf{p} = (x_t, y_t, z_t)^{\top}$  in  $Q_t$ . We define two different support volumes for  $\mathbf{p}$ : a spatial support volume and a spatio-temporal support volume. The spatial support volume of  $\mathbf{p}$ , denoted by  $\Omega^{S}(\mathbf{p})$ , contains the 3D points in  $Q_t$  that are in a sphere of radius r centered at  $\mathbf{p}$  (Fig. 3(b)). To define the spatio-temporal support volume of  $\mathbf{p}$ , denoted by  $\Omega^{ST}(\mathbf{p})$ , we merge the sequence of pointclouds in the small time interval  $[t-\tau, t+\tau]$ . The 3D points which are in a sphere of radius r centered at  $\mathbf{p}$  are considered as  $\Omega^{ST}(\mathbf{p})$  (Fig. 3(c)).

The covariance matrix  $C^{\alpha}$  of the points  $\mathbf{q} \in \Omega^{\alpha}(\mathbf{p}), \alpha \in \{ST, S\}$  is given by:

$$C^{\alpha} = \frac{1}{n_p} \sum_{\mathbf{q} \in \Omega^{\alpha}(\mathbf{p})} (\mathbf{q} - \mu) (\mathbf{q} - \mu)^{\top}, \qquad (1)$$

where

$$\mu = \frac{1}{n_p} \sum_{\mathbf{q} \in \Omega^{\alpha}(\mathbf{p})} \mathbf{q},$$

and  $n_p = |\Omega^{\alpha}(\mathbf{p})|$  denotes the number of points in the support volume of  $\mathbf{p}$ . Performing eigen decomposition on the covariance matrix  $C^{\alpha}$  gives us:

$$V^{\alpha}E^{\alpha}V^{\alpha\top} = C^{\alpha}, \qquad (2)$$

where  $E^{\alpha}$  is a diagonal matrix containing the eigenvalues  $\lambda_1^{\alpha} \ge \lambda_2^{\alpha} \ge \lambda_3^{\alpha} \ge 0$  of  $C^{\alpha}$  and  $V^{\alpha} = [\mathbf{v}_1^{\alpha} \mathbf{v}_2^{\alpha} \mathbf{v}_3^{\alpha}]$  contains the three corresponding orthonormal eigenvectors.

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* 



Fig. 3: Spatio Temporal Keypoint (STK) detection. (a) A 3D pointcloud sequence corresponding to the holding head action, (b) the spatial support volume of a particular point **p**, (c) the spatio-temporal support volume of p, (d) the HOPC descriptors, (e) STK detection.

dodecahedron which is composed of m = 20 vertices, each of which corresponds to a histogram bin. Let  $\{\mathbf{u}_i\}_{i=1}^m$ be the vertices of a regular dodecahedron and let U = $[\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_m] \in \mathbb{R}^{3 \times m}$ . For a regular dodecahedron centered at the origin, these vertices are given as:

- 8 vertices from  $(\pm 1, \pm 1, \pm 1)$
- 4 vertices from  $(\pm \varphi^{-1}, \pm \varphi)$  4 vertices from  $(\pm \varphi^{-1}, \pm \varphi)$  4 vertices from  $(\pm \varphi^{-1}, \pm \varphi, 0)$  4 vertices from  $(\pm \varphi, 0, \pm \varphi^{-1})$

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. To resolve this orientation ambiguity, we consider the distribution of point vector directions and their magnitudes within the support volume of **p**. That is, for each point  $\mathbf{q} \in \Omega^{\alpha}(\mathbf{p})$ , we compute  $\mathbf{o} = \mathbf{q} - \mathbf{p}$  and we determine the sign of each eigenvector  $\mathbf{v}_i^{\alpha}$  as follows:

$$\mathbf{v}_{j}^{\alpha} = \mathbf{v}_{j}^{\alpha}.\mathrm{sign}\left(\sum_{\mathbf{q}\in\Omega^{\alpha}(\mathbf{p})}\mathrm{sign}(\mathbf{o}^{\top}\mathbf{v}_{j}^{\alpha})(\mathbf{o}^{\top}\mathbf{v}_{j}^{\upsilon})^{2}\right), \quad (3)$$

where the 'sign' function returns the sign of an input number. Note that the squared projection operation ensures that small projected values, which are often due to noise, are suppressed. If the signs of eigenvectors  $\mathbf{v}_1^{\alpha}, \mathbf{v}_2^{\alpha}$ , and  $\mathbf{v}_3^{\alpha}$ disagree, *i.e.*  $\mathbf{v}_1^{\alpha} \times \mathbf{v}_2^{\alpha} \neq \mathbf{v}_3^{\alpha}$ , we switch the sign of the eigenvector whose  $|\sum_{\mathbf{q}\in\Omega^{\alpha}(\mathbf{p})} \operatorname{sign}(\mathbf{o}^{\top}\mathbf{v}_{j}^{\alpha})(\mathbf{o}^{\top}\mathbf{v}_{j}^{\alpha})^{2}|$  value is the smallest. We then project each eigenvector  $\mathbf{v}_{j}^{\alpha}$  onto U to give us:

$$\mathbf{b}_{j}^{\alpha} = U^{\top} \mathbf{v}_{j}^{\alpha} \in \mathbb{R}^{m}, \text{ for } 1 \leq j \leq 3.$$
(4)

If  $\mathbf{v}_i^{\alpha}$  perfectly aligns with  $\mathbf{u}_i \in U$ , it should vote into only the  $i^{th}$  bin. However, as the  $\mathbf{u}_i$ 's are not orthogonal to each other,  $\mathbf{b}_{i}^{\alpha}$  will have non-zero projection values in other bins as well. To overcome this effect, we quantize the projection values of  $\mathbf{b}_{i}^{\alpha}$  by imposing a threshold value  $\psi$ computed as follows:

$$\psi = \mathbf{u}_k^\top \mathbf{u}_l = \varphi + \varphi^{-1}, \text{ for } \mathbf{u}_k, \mathbf{u}_l \in U,$$
 (5)

where  $\mathbf{u}_k$  and  $\mathbf{u}_l$  are any two *neighbouring* vectors in U. The quantized vector is then given by

$$\hat{\mathbf{b}}_{j}^{\alpha}(z) = \begin{cases} 0 & \text{if } \mathbf{b}_{j}^{\alpha}(z) \leqslant \psi \\ \mathbf{b}_{j}^{\alpha}(z) - \psi & \text{otherwise,} \end{cases}$$

where  $1 \leq z \leq m$  denotes a bin number. For the  $j^{\text{th}}$  eigenvector, we define  $\mathbf{h}_{i}^{\alpha}$  to be  $\hat{\mathbf{b}}_{i}^{\alpha}$  scaled by their corresponding eigenvalue  $\lambda_i^{\alpha}$ :

$$\mathbf{h}_{j}^{\alpha} = \frac{\lambda_{j}^{\alpha} \cdot \mathbf{b}_{j}^{\alpha}}{||\mathbf{\hat{b}}_{j}^{\alpha}||_{2}} \in \mathbb{R}^{m}, \text{ for } 1 \leq j \leq 3.$$
(6)

We concatenate the histograms of oriented principal components of the three eigenvectors in decreasing order of magnitudes of their associated eigenvalues to form a descriptor for point p:

$$\mathbf{h}_{\mathbf{p}}^{\alpha} = \begin{bmatrix} \mathbf{h}_{1}^{\alpha \top} \ \mathbf{h}_{2}^{\alpha \top} \ \mathbf{h}_{3}^{\alpha \top} \end{bmatrix} \in \mathbb{R}^{3m}.$$
(7)

The spatial HOPC descriptor  $h_{\mathbf{p}}^{S}$  encodes the shape of the support volume around p. On the other hand, the spatiotemporal HOPC descriptor  $\mathbf{h}_{\mathbf{p}}^{ST}$  encodes information from both shape and motion. Since the smallest principal component of the local surface is the total least squares estimate of the surface normal [52], the surface normals encoded in our descriptor are more robust to noise than gradientbased surface normals used in [9], [45]. Moreover, HOPC additionally encodes the first two eigenvectors which are more dominant compared to the third one. The computation of the spatial and spatio-temporal HOPC descriptors is shown in Fig. 3(d).

# 4 SPATIO-TEMPORAL KEYPOINT (STK) DETECTION

The aim of the STK detection is to find points in 3D pointcloud action sequences that 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  $\mathbf{p} = (x_t, y_t, z_t)^{\mathsf{T}}$  within a sequence of 3D pointclouds. We perform eigen decomposition on the spatial and the spatio-temporal covariance matrices  $C^{\mathsf{S}}$  and  $C^{\mathsf{ST}}$  as described in Section 3. For the first two constraints, we define the following ratios:

$$\delta_{12}^{S} = \frac{\lambda_{1}^{S}}{\lambda_{2}^{S}}, \ \delta_{23}^{S} = \frac{\lambda_{2}^{S}}{\lambda_{3}^{S}}, \ \delta_{12}^{ST} = \frac{\lambda_{1}^{ST}}{\lambda_{2}^{ST}}, \ \delta_{23}^{ST} = \frac{\lambda_{2}^{ST}}{\lambda_{3}^{ST}}.$$
 (8)

For 3D symmetrical surfaces, the ratio between the first two eigenvalues or last two eigenvalues are very close to 1. The principal components at such locations are, therefore, ambiguous. Thus, for a point to be qualified 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  $\epsilon_{\text{STK}}$  is a small margin to cater for noise. This process eliminates ambiguous points and produces a subset of candidate keypoints which can be described uniquely in a local coordinate basis.

Recall that  $\mathbf{h}_{\mathbf{p}}^{\mathbf{S}}$  in (7) represents the spatial HOPC and  $\mathbf{h}_{\mathbf{p}}^{\mathbf{ST}}$  the spatio-temporal HOPC at point **p**. For the third constraint, a *quality factor*  $\eta_{\mathbf{p}}$  is computed for all candidate keypoints:

$$\eta_{\mathbf{p}} = \frac{1}{2} \sum_{i=1}^{3m} \frac{(\mathbf{h}_{\mathbf{p}}^{\mathbf{S}}(i) - \mathbf{h}_{\mathbf{p}}^{\mathbf{ST}}(i))^2}{(\mathbf{h}_{\mathbf{p}}^{\mathbf{S}}(i) + \mathbf{h}_{\mathbf{p}}^{\mathbf{ST}}(i))} .$$
(10)

When  $\mathbf{h}_{\mathbf{p}}^{\mathbf{S}} = \mathbf{h}_{\mathbf{p}}^{\mathbf{ST}}$ , the quality factor is at the minimum value of  $\eta_{\mathbf{p}} = 0$  which basically means that the candidate point  $\mathbf{p}$  has a stationary spatio-temporal support volume. On the other hand, significant variations in space-time change the direction and magnitude of spatio-temporal eigenvectors with respect to the spatial eigenvectors. Thus,  $\eta_{\mathbf{p}}$  is large when a significant motion occurs in the spatio-temporal support volume.

STKs that are in the vicinity of each other are 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. We define radius r' (with r' < r) and time interval  $[t-\tau', t+\tau']$  (with  $\tau' \leq \tau$ ) where t is the frame number being considered. The candidate STKs are firstly sorted according to their quality values. Starting from the highest quality STK, all candidate STKs falling within r' and  $\tau'$  from it are discarded. The same process is repeated on the remaining candidate STKs until only a desired number,  $n_k$ , of STKs are left. Figure. 3 shows the steps of our STK detection algorithm. Figure. 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 (LOCAL HOPC)

The HOPC descriptor discussed in Section 3 is not viewinvariant yet. We compute Local HOPC only at the STKs since it is possible to normalize the orientation of the local region only at these points *i.e.* a unique local coordinate basis can only be defined at these points. We perform orientation normalization at each STK using the eigenvectors of its spatial covariance matrix  $C^{\rm S}$  (see Section 3). We consider the eigenvectors  $V^{\rm S} = [\mathbf{v}_1^{\rm S} \mathbf{v}_2^{\rm S} \mathbf{v}_3^{\rm S}]$  of  $C^{\rm S}$  as a local object centered coordinate basis. Note that the matrix  $V^{\rm S}$  is orthonormal and can be used as a valid 3D rotation matrix, since:

$$\mathbf{v}_{i}^{\mathbf{S}} \cdot \mathbf{v}_{j}^{\mathbf{S}} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j. \end{cases}$$
(11)

We apply the 3D rotation  $R = V^{S^{\top}}$  to all the meancentered points  $\{\mathbf{q}_i\}_{i=1}^{n_p}$  within the spatio-temporal support volume of  $\mathbf{p}$ ,  $\Omega^{\text{ST}}(\mathbf{p})$ , and bring them to a canonical coordinate system:

$$\mathbf{q}'_i = R\mathbf{q}_i, \text{ for } i = 1, \cdots, n_p, \tag{12}$$

where  $\mathbf{q}'_i$  denotes the rotated point in the local object centered coordinate basis. Note that the first, second, and third principal components are now aligned with the X, Y, and Z axes of the Cartesian coordinates. Since the same STKs in two different views have the same canonical representation, 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 the maximum eigenvalue will always map to the X axis, the second largest to the Y axis and the smallest to the Z axis.

After the orientation normalization given in (12), for each point  $\mathbf{q}' \in \Omega^{ST}(\mathbf{p})$ , we inspect the eigenratios  $\delta_{12}^{ST}$ and  $\delta_{23}^{ST}$  (Eq. (9)) computed using neighbouring points of  $\mathbf{q}'$  to determine how the HOPC descriptor at  $\mathbf{q}'$  should contribute to the STK descriptor computation of  $\mathbf{p}$ . When  $\delta_{12}^{ST}$  and  $\delta_{23}^{ST}$  are both larger than  $\theta_1 = 1 + \epsilon_1$ , where  $\epsilon_1$  is a small margin, all the eigenvectors are uniquely defined and, therefore, can all contribute to the STK descriptor. When  $\delta_{12}^{ST} \leq \theta_1$  and  $\delta_{23}^{ST} > \theta_1$ , the first two eigenvectors are ambiguous and so only the third eigenvector should contribute to the STK descriptor. A similar argument



Fig. 4: STKs (shown in red) 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.

applies to the case where  $\delta_{12}^{\text{ST}} > \theta_1$  and  $\delta_{23}^{\text{ST}} \leq \theta_1$ . When both  $\delta_{12}^{\text{ST}} \leq \theta_1$  and  $\delta_{23}^{\text{ST}} \leq \theta_1$ , then **q'** has no contribution to the descriptor computation. In summary, the following three criteria need to be considered for the construction of  $\mathbf{h}_{\mathbf{q}'}^{ST}$ :

1) If 
$$\delta_{12}^{ST} > \theta_1 \& \delta_{23}^{ST} > \theta_1, \mathbf{h}_{\mathbf{q}'}^{ST} = [\mathbf{h}_1^{ST^{\top}} \ \mathbf{h}_2^{ST^{\top}} \ \mathbf{h}_3^{ST^{\top}}]^{\top};$$
  
2) If  $\delta_{12}^{ST} \leq \theta_1 \& \delta_{23}^{ST} > \theta_1, \mathbf{h}_{\mathbf{q}'}^{ST} = [\mathbf{0}^{\top} \ \mathbf{0}^{\top} \ \mathbf{h}_3^{ST^{\top}}]^{\top};$   
3) If  $\delta_{12}^{ST} > \theta_1 \& \delta_{23}^{ST} \leq \theta_1, \mathbf{h}_{\mathbf{q}'}^{ST} = [\mathbf{h}_1^{ST^{\top}} \ \mathbf{0}^{\top} \ \mathbf{0}^{\top}]^{\top}.$ 

Next, the orientation normalized spatio-temporal support volume around the STK **p** is partitioned into  $\gamma = n_x \times n_y \times$  $n_t$  spatio-temporal cells along the X, Y, and T dimensions. We use  $c_s$ , where  $s = 1 \cdots \gamma$ , to denote the  $s^{\text{th}}$  cell. The cell descriptor  $\mathbf{h}_{c_s}$  is computed by accumulating the  $\mathbf{h}_{\mathbf{q}'}^{ST}$ 's

$$\mathbf{h}_{c_s} = \sum_{\mathbf{q}' \in c_s} \mathbf{h}_{\mathbf{q}'}^{ST},\tag{13}$$

and then normalizing

$$\mathbf{h}_{c_s} \leftarrow \frac{\mathbf{h}_{c_s}}{||\mathbf{h}_{c_s}||_2}.$$
(14)

We define the final view-invariant descriptor,  $\mathbf{h}_v$ , of STK **p** to be the concatenation of  $\mathbf{h}_{c_s}$  obtained from all the cells:

$$\mathbf{h}_{v} = \begin{bmatrix} \mathbf{h}_{c_{1}}^{\top} \ \mathbf{h}_{c_{2}}^{\top} \ \cdots \ \mathbf{h}_{c_{\gamma}}^{\top} \end{bmatrix}^{\top}.$$
 (15)

The above steps are repeated for all the STKs. Thus, the STK descriptors encode 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-ofwords approach for object recognition, we build a codebook by clustering the STK descriptors ( $\mathbf{h}_v$ ) with the Kmeans algorithm. Clustering is performed over all action descriptors extracted from all training view samples. Thus, the codebook that we learn is not single action or single view specific. For a 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 STKs in that cluster. One codeword is assigned to each STK descriptor based on the minimum Euclidean distance. The histogram of codewords is used as an action descriptor. For classification, we use an SVM classifier with the histogram intersection kernel [53].

#### 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, *e.g.* the sway of the human body. We use the *F*-score to find the most discriminative features in the codebook and discard non-discriminative features. The *F*-score [54] measures the discrimination of two sets of real numbers. For more than two sets of real numbers, we use the multiset F-score [55] to measure their discrimination. Given the training histogram of codewords  $x_k$ , for  $k = 1, \dots, m$ , and  $l \ge 2$  action classes, if the number of the samples in the *j*th  $(1 \le j \le l)$  class is  $n_j$ , then the *F*-score of the *i*th histogram bin is defined as:

$$F_{i} = \frac{\sum_{j=1}^{l} \left(\bar{x}_{i}^{(j)} - \bar{x}_{i}\right)^{2}}{\sum_{j=1}^{l} \frac{1}{n_{j}-1} \sum_{k=1}^{n_{j}} \left(\bar{x}_{k,i}^{(j)} - \bar{x}_{i}^{(j)}\right)^{2}},$$
(16)

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, and  $\bar{x}_{k,i}^{(j)}$ is the *i*th histogram bin of the *k*th sample in the *j*th class. The larger is the *F*-score, the more discriminative is the corresponding histogram bin. Therefore, we rank the codewords by their *F*-scores and select the codewords whose *F*-scores are higher than a threshold. In our experiments, up to 1.5% improved accuracy was observed by selecting the top 98% discriminative features out of the total 1500.

#### 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 encoding the distribution of STKs in space-time (Fig. 4) can further improve the 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  $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^4, i = 1, \cdots, n_k\}$  represent the set of all selected STKs within a sequence of 3D pointclouds  $\mathcal{Q}$ , where  $n_k$  is the number of STKs and  $\mathbf{p}_i = (x, y, z, t)^{\top}$  are the coordinates of an STK in the 4D space with x and y being the spatial coordinates, z being depth and t being time. To cope with the heterogeneity in the vectors, we normalize the vectors so that all their components have zero-mean and unit variance.

To simplify the description, let us assume that the set  $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^4\}$  now have all the normalized vectors as described above. By dropping the time axis, we have

TABLE 1: The 600 vertices of a *120-cell regular polychoron* centered at the origin generated from *all* and *even* permutations of these coordinates [56].

Vertices	Permutation	Coordinate points
24	all	$0,0,\pm 2,\pm 2$
64	all	$\pm 1,\pm 1,\pm 1,\pm \sqrt{5}$
64	all	$\pm arphi^{-2}, \pm arphi, \pm arphi, \pm arphi$
64	all	$\pm \varphi^{-1}, \pm \varphi^{-1}, \pm \varphi^{-1}, \pm \varphi^{+2}$
96	even	$0,\pm\varphi^{-2},\pm1,\pm\varphi^{+2}$
96	even	$0,\pm\varphi^{-1},\pm\varphi,\pm\sqrt{5}$
192	even	$\pm \varphi^{-1}, \pm 1, \pm \varphi, \pm 2$

a set of normalized 3D STKs:  $\mathcal{P}' = \{\mathbf{p}'_i \in \mathbb{R}^3\}$ . Eigen decomposition is then applied to the covariance matrix of points in  $\mathcal{P}'$  to yield two eigenratios  $\bar{\delta}_{12} = \lambda_1/\lambda_2$  and  $\bar{\delta}_{23} = \lambda_2/\lambda_3$ . To get a unique coordinate basis, we require that  $\bar{\delta}_{12}, \bar{\delta}_{23} > \theta_g = 1 + \epsilon_g$ , where  $\epsilon_g$  is a small constant. If these constraints are not satisfied, we perform an iterative refinement of STKs as follows. Given  $n_k$  initial STKs, in each iteration,  $m_k$  (where  $m_k \ll n_k$ ) STKs with the lowest quality factor (Eq. (10)) are removed. Eigen decomposition is applied to the remaining points to yield two new eigenratios. This process is iterated until the eigenratio constraints are satisfied. Generally, three iterations are sufficient.

To achieve a view-invariant representation, all points in  $\mathcal{P}'$  are aligned along V, *i.e.*, for all  $\mathbf{p}' \in \mathcal{P}'$ , we set  $\mathbf{p}' \leftarrow V^{\top}\mathbf{p}'$  where V is the eigenvector matrix obtained from the eigen decomposition in the last iteration. The normalized temporal dimension is then reattached to each point in  $\mathcal{P}'$ :

$$\widehat{\mathbf{p}} \leftarrow [\mathbf{p}', t], \tag{17}$$

to form the set  $\hat{\mathcal{P}} = { \hat{\mathbf{p}}_i \in \mathbb{R}^4 }$ . To encode the distribution of STKs in the 4D space, we consider a 4D regular geometric object called *polychoron* [56] 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 [56].

Given the set  $\hat{\mathcal{P}}$  constructed above, we project each orientation normalized  $\hat{\mathbf{p}}_i$  onto the 600 vertices of the *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 in  $\hat{\mathcal{P}}$  and the final histogram is a 600 dimensional STK Distribution (STK-D) descriptor which encodes the global spatio-temporal distribution of STKs of the sequence Q in a compact and discriminative form.

## 7 ADAPTABLE SUPPORT VOLUME

So far, for STK detection and description, we have used a fixed spatio-temporal support volume with spatial radius r and temporal scale  $\tau$ . However, subjects may have different scales (height and width) and may perform actions at

different speeds. Therefore, simply using a fixed spatial radius r and temporal scale  $\tau$  is not optimal. Moreover, a larger 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 [57]. We adapt the method proposed by Mian et al. [58] in object retrieval for action recognition in 3D pointcloud sequences. Note that in the human action recognition problem, the subject's height is available in most cases (which is not the case for object retrieval). Where available, we use the subject's height  $(h_s)$ to find an appropriate spatial scale. We select the ratios as  $r = \sigma h_s$ , where  $0 < \sigma < 1$  is a constant factor. We have empirically selected the value of  $\sigma$  to maximize the descriptiveness and robustness of our descriptor to occlusions. In all experiments, we use a fixed value of  $\sigma$ for all actions, views and datasets. In our experiments in Section 8, we observe that this simple approach achieves almost the same accuracy as the automatic spatial scale selection method adapted from [58]. 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 techniques [5], [6], [8], [9], [44], [47] use a fixed temporal scale. We observe that variations in action execution speed cause 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  $\mathcal{Q} = \operatorname{seq}(Q_1, \dots, Q_t, \dots, Q_{n_f})$  be a sequence of 3D pointclouds. For a given point  $\mathbf{p} = (x_t, y_t, z_t)^\top \in Q_t$ and a given temporal scale  $\tau$ , we can define the spatial support volume  $\Omega_{\tau}^{\mathrm{ST}}(\mathbf{p})$  of  $\mathbf{p}$ . The covariance matrix of the points falling within  $\Omega_{\tau}^{\mathrm{ST}}(\mathbf{p})$  can be eigen-decomposed to yield the eigenvalues  $\lambda_1^{\tau} \geq \lambda_2^{\tau} \geq \lambda_3^{\tau} \geq 0$ . These steps are similar to those described in Section 3, except that, in this Section, we repeat these steps for each temporal scale  $\tau = 1, \dots, \tau_m$ , where  $\tau_m$  is a fixed upper threshold. For each  $\tau$  value, we calculate:

$$A_{\mathbf{p}}(\tau) = \frac{\lambda_2^{\tau}}{\lambda_1^{\tau}} + \frac{\lambda_3^{\tau}}{\lambda_2^{\tau}}.$$
(18)

The optimal temporal scale  $\tau^*(\mathbf{p})$  for the given point  $\mathbf{p}$  is chosen to be the one that minimizes  $A_{\mathbf{p}}$  over the range  $1 \leq \tau \leq \tau_m$ , *i.e.*,

$$\tau^*(\mathbf{p}) = \operatorname*{argmin}_{\tau} A_{\mathbf{p}}(\tau). \tag{19}$$



Fig. 5: The same action (hand waving) 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, the optimal temporal scale is found to be  $\tau^* = 3$ , for moderate movement  $\tau^* = 2$ , and for fast movement  $\tau^* = 1$ .

As an example to illustrate this automatic temporal scale selection process, Fig. 5(a)-(c) show the temporal sequences of pointclouds for the hand waving action performed at three different speeds. The dotted circle shows the sphere defined by the spatial radius r in each pointcloud. The spatial radius r in the three cases is the same because of similar geometry. Our aim here is to select the optimal temporal scale for the point **p** in the pointcloud  $Q_t$  shown in black and dotted outline. Figure 5(d) shows the union of points in the range  $Q_{t-3} \cdots Q_{t+3}$  which are within the radius r measured from the coordinate (x, y, z) of point **p**. Figure 5(e) and (f) show the union of points in the same way for  $Q_{t-2} \cdots Q_{t+2}$  and  $Q_{t-1} \cdots Q_{t+1}$ , respectively. Figure 5(g)-(i) show the plots of  $A_{\mathbf{p}}$  with the variation of  $\tau$ . Increasing  $\tau$  beyond a certain value does not affect the accumulated pointcloud as the value of  $A_{\mathbf{p}}$  becomes constant. In most cases, increasing  $\tau$  decreases  $A_{\mathbf{p}}$  until a fixed value is reached. We compute  $A_{\mathbf{p}}(\tau)$  for all values of  $\tau$  and find the global minimum  $\tau^*$ . When more than one  $\tau^*$  exist, the smallest value of  $\tau^*$  is chosen.

For each STK, the 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  $\tau^*$  and  $flag \in \{0, 1\}$ . If the optimal  $\tau^*$  is equal to  $\tau_m$  then the flag is set to 0, indicating that the STK **p** should be discarded. If the computed optimal  $\tau^*$  is smaller than  $\tau_m$  then the flag is set to 1, indicating that the return  $\tau^*$  value is the optimal temporal scale for **p**.

Algorithm 1: Automatic Temporal Scale Selection	l
<b>input</b> : $Q$ , <b>p</b> , $r$ , and $\tau_m$ . <b>output</b> : $\tau^*$ , flag.	
1 for $\tau = 1 : \tau_m$ do	
2 Construct $\Omega_{\tau}^{\text{ST}}(\mathbf{p});$	
3 $ \mu_{\tau} \leftarrow \frac{1}{n_p} \sum_{\mathbf{q} \in \Omega_{\tau}(\mathbf{p})} \mathbf{q}; $	
$ C_{\tau} \leftarrow \frac{1}{n_p} \sum_{\mathbf{q} \in \Omega_{\tau}(\mathbf{p})} (\mathbf{q} - \mu_{\tau}) (\mathbf{q} - \mu_{\tau})^{\top}; $	
$4  V_{\tau} \begin{bmatrix} \lambda_{1}^{\tau} & 0 & 0 \\ 0 & \lambda_{2}^{\tau} & 0 \\ 0 & 0 & \lambda_{3}^{\tau} \end{bmatrix} V_{\tau}^{\top} = C_{\tau};$	
5 $A_{\mathbf{p}}(\tau) \leftarrow \frac{\lambda_2^{\tau}}{\lambda_1^{\tau}} + \frac{\lambda_3^{\tau}}{\lambda_2^{\tau}};$	
6 end	
$\tau \ \tau^* = \underset{\tau}{\operatorname{argmin}} \ A_{\mathbf{p}}(\tau);$	
s if $\tau^* = \tau_m$ then	
9   $flag \leftarrow 0;$	
10 else	
11 $flag \leftarrow 1;$	
12 end	

### 8 EXPERIMENTS

We evaluate the proposed algorithm on five benchmark datasets, including two multi-view (Northwestern-UCLA Multiview Action3D [29], and UWA3D Multiview Activity II) and three single-view (MSR Action3D [38], MSR Daily Activity3D [39], and MSR Gesture3D [40]) datasets. Performance is compared to nine existing action recognition methods including Histogram of Oriented 4D Normals (HON4D) [9], Super Normal Vector (SNV) [10], Lie Algebra Relative Pairs (LARP) [59], 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]. The baseline results are obtained using publicly available implementations of CCD [3], VV [60], DVV [36], HON4D [9], SNV [10], and LARP [59] from the respective authors' websites. For the remaining three methods AOG [29], HOJ3D [27], and AE [28], we use our implementations because their codes are not publicly available. For CCD [3], VV [60] and DVV [36], we use DSTIP [8], which is more robust to 3D sensor noise compared to color-based interest point detectors, to extract and describe the spatio-temporal interest points. Our algorithm is robust to many different parameter settings (see Section 8.7). To help the reader reproduce our results, we provide the parameter values that we used in Table 2. The UWA3D Multiview Activity II dataset and code will be made publicly available.

To evaluate individual components of the proposed algorithm, we report results for the following four settings:

**Holistic HOPC:** A sequence of 3D pointclouds is divided into  $\gamma = 6 \times 5 \times 3$  spatio-temporal cells along the X, Y, and T dimensions. The spatio-temporal HOPC descriptor

TABLE 2: Parameters and their values: K: number of codewords,  $n_k$ : number of STKs ,  $\theta_{\text{STK}}$ ,  $\theta_1$ ,  $\theta_g$ : eigenratio thresholds,  $n_x \times n_y \times n_t$ : spatio-temporal cells (Section 5),  $\tau_m$ : maximum temporal scale,  $m_k$ : iterative refinement (Section 6.3).



Fig. 6: Sample pointclouds from the Northwestern-UCLA Multiview Action3D dataset [29] captured by 3 cameras.

 $h_p^{ST}$  in (7) is computed for each point **p** within the sequence. The cell descriptor is computed using (13) and then normalized using (14). The final descriptor for the given sequence is a concatenation of all the cell descriptors. We use SVM for classification. Similar to HON4D [9] and SNV [10], our Holistic HOPC is suitable for single-view action recognition [37] and can handle more inter-class similarities of local motions compared to local methods [9].

**STK-D**: For each sequence of 3D pointclouds, the histogram of spatio-temporal distribution of STKs is used as the sequence descriptor (Section 6.3).

**Local HOPC**: For each sequence, STKs are detected using the method proposed in Section 4. The proposed orientation normalization is then applied at each STK neighborhood to extract its view-invariant HOPC descriptor (Section 5). The BoW approach is used to describe the sequence.

**Local HOPC+STK-D**: The bag of STK descriptors and the histogram of spatio-temporal distribution of STKs are concatenated to form the sequence descriptor.

#### 8.1 N-UCLA Multiview Action3D Dataset

The Northwestern-UCLA dataset [29] 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*, and (10) *carry*. Each action was performed by 10 subjects 1 to 6 times. Figure 6 shows 12 sample 3D pointclouds of four actions captured by the three cameras.

TABLE 3: Comparison of action recognition accuracy (%) on the Northwestern-UCLA Multiview Action3D dataset where the samples from the first two cameras are used as training data, and the samples from the third camera are used as test data.

Data type	RGB	Skeleton	Depth
CCD [3]	-	-	34.4
VV [60]	43.5	-	48.8
HOJ3D [27]	-	54.5	-
DVV [36]	47.8	-	52.1
AE [28]	-	69.9	-
AOG [29]	73.3	-	53.6
HON4D [9]	-	-	39.9
SNV [10]	-	-	42.8
LARP [59]	-	74.2	-
Holistic HOPC	-	-	43.4
STK-D	-	-	53.9
Local HOPC	-	-	71.9
Local HOPC+STK-D	-	-	80.0

To compare our method with state-of-the-art algorithms, we use the same experimental setting as [29], using the samples from the first two cameras as training data, and the samples from the third camera as test data. Results are given in Table 3. Holistic approaches such as HON4D [9], SNV [10], and the proposed Holistic HOPC achieved low recognition accuracy since they are not designed to handle viewpoint changes. Similarly, since depth is a function of viewpoint, CCD [3] achieved low accuracy by encoding the differences between the depth values of an interest point and its neighbourhood points.

Among the knowledge transfer based methods, VV [60] and DVV [36] did not perform well; however, AOG [29] obtained high accuracy on only RGB videos. On depth videos, AOG also did not perform well. A possible reason is that depth videos have higher noise levels and interpolating noisy features across views can compromise discrimination ability.

Skeleton based methods such as AE [28] and LARP [59] achieved high accuracy. We used the scale and orientation normalization of skeletons proposed in LARP [59] for AE [28] as well which improved the results of AE. However, skeleton data may not be reliable, or even available, when the subject is not in an upright position or is occluded [8]. More importantly, the application of these methods is limited to human activity recognition where the human skeleton is generally estimated by [43].

STK-D alone achieved higher accuracy compared to all depth image based methods. This confirms the repeatability of STKs and the robustness of STK-D to viewpoint changes. The Local HOPC descriptor achieved higher accuracy than STK-D and all depth based methods. Since HOPC and STK-D capture complementary information, their combination (Local HOPC+STK-D) further improved the performance by 8.1% achieving the overall best accuracy of 80%. Note that this is about 6% higher than the nearest competitor LARP [59] which requires skeleton data whereas our method does not.

The confusion matrix of our proposed view-invariant Local HOPC+STK-D method is shown in Fig. 7. The action



Fig. 7: Confusion matrix of our algorithm on the Northwestern-UCLA Multiview Action3D dataset [29].

(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 (3) *drop trash* have high confusion due to similarity in motion and appearance.

#### 8.2 UWA3D Multiview Activity II Dataset

This dataset was collected in our lab using Kinect to emphasize three points: (1) Larger number of human activities. (2) Each subject performed all actions in a 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 organized 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



Fig. 8: Sample pointclouds from the UWA3D Multiview Activity II dataset captured by one camera from 4 different views.

*back*, have self-occlusion. Moreover, in the top view, the lower part of the body was not properly captured because of occlusion. Figure 8 shows 16 sample pointclouds of five actions from 4 views.

For cross-view action recognition, we use the samples from two views as training data, and the samples from the two remaining views as test data. Table 4 summarizes our results. Since this dataset is more challenging compared to the N-UCLA dataset, the performance of all methods drops significantly. Our Holistic HOPC descriptor achieved higher average recognition accuracy than the depth based methods but lower than the methods which use normalized skeleton data. Among the depth based methods, HON4D [9] and SNV [10] are the nearest competitors to the Holistic HOPC. The Local HOPC achieved higher accuracy than STK-D and the Holistic HOPC. Combining STK-D with Local HOPC again improved performance by 8.2% achieving the overall best performance of 52.2%. Note that this is about 9% higher than the nearest competitor LARP [59] which uses skeleton data. Local HOPC+STK-D achieved the highest accuracy in all combinations of training and test views except one. The accuracy of skeleton based methods is significantly lower on this dataset because the skeleton data is not accurate for some actions such as *drinking*, phone answering, sneezing or is not available for some actions such as *falling down* and *lying down*.

Moreover, the overall accuracy of the knowledge transfer based methods VV [60], DVV [36], and AOG [29] when depth videos are used as input data 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 between actions in the presence of noise.

Figure 9 shows the confusion matrix of our proposed view-invariant Local HOPC+STK-D 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 causes the most confusion are (9) *holding chest* versus

Training views	$V_1 \delta$	$\& V_2$	$V_1 \delta$	$\& V_3$	$V_1 \delta$	$\& V_4$	$V_2 \delta$	$k V_3$	$V_2$ &	$\gtrsim V_4$	$V_3$ &	$\gtrsim V_4$	Maan
Test view	$V_3$	$V_4$	$V_2$	$V_4$	$V_2$	$V_3$	$V_1$	$V_4$	$V_1$	$V_3$	$V_1$	$V_2$	Weall
AOG [29] (RGB)	47.3	39.7	43.0	30.5	35.0	42.2	50.7	28.6	51.0	43.2	51.6	44.2	42.3
HOJ3D [27] (Skeleton)	15.3	28.2	17.3	27.0	14.6	13.4	15.0	12.9	22.1	13.5	20.3	12.7	17.7
AE [28] (Norm. Skeleton)	45.0	40.4	35.1	36.9	34.7	36.0	49.5	29.3	57.1	35.4	49.0	29.3	39.8
LARP [59] (Norm. Skeleton)	49.4	42.8	34.6	39.7	38.1	44.8	53.3	33.5	53.6	41.2	56.7	32.6	43.4
CCD [3] (Depth)	10.5	13.6	10.3	12.8	11.1	8.3	10.0	7.7	13.1	13.0	12.9	10.8	11.2
VV [60] (Depth)	20.2	22.0	19.9	22.3	19.3	20.5	20.8	19.3	21.6	21.2	23.1	19.9	20.9
DVV [36] (Depth)	23.5	25.9	23.6	26.9	22.3	20.2	22.1	24.5	24.9	23.1	28.3	23.8	24.1
AOG [29] (Depth)	29.3	31.1	25.3	29.9	22.7	21.9	25.0	20.2	30.5	27.9	30.0	26.8	26.7
HON4D [9] (Depth)	31.1	23.0	21.9	10.0	36.6	32.6	47.0	22.7	36.6	16.5	41.4	26.8	28.9
SNV [10] (Depth)	31.9	25.7	23.0	13.1	38.4	34.0	43.3	24.2	36.9	20.3	38.6	29.0	29.9
Holistic HOPC (Depth)	32.3	25.2	27.4	17.0	38.6	38.8	42.9	25.9	36.1	27.0	42.2	28.5	31.8
STK-D (Depth)	32.8	25.1	38.7	22.7	23.7	23.4	29.0	19.2	27.9	28.0	24.5	30.1	27.1
Local HOPC (Depth)	42.3	46.5	39.1	49.8	35.0	39.3	51.9	34.4	57.9	35.3	60.5	36.5	44.0
Local HOPC+STK-D (Depth)	52.7	51.8	59.0	57.5	42.8	44.2	58.1	38.4	63.2	43.8	66.3	48.0	52.2

TABLE 4: Comparison of action recognition accuracy (%) on the UWA3D Multiview Activity II dataset. Each time two views are used for training and the remain two views are individually used for testing.



Fig. 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.

(11) *holding back* and (12) *walking* versus (13) *irregular walking*, because the motion and appearance of these two actions are very similar.

### 8.3 MSR Action3D Dataset

The MSR Action3D dataset [38] consists of 20 actions performed 2 to 3 times by 10 subjects. This dataset is challenging due to high inter-action similarities. Following the protocol of [9], we use 5 subjects for training and the remaining 5 for testing and exhaustively repeated the experiments 252 folds. Table 5 compares our algorithms with existing methods. The proposed Holistic HOPC outperformed all methods and achieved 86.5% average accuracy which is more than 2% higher than its nearest competitor SNV [10] and significantly higher than the skeleton based methods such as HOJ3D [27] and LARP [59]. The average accuracy of our view-invariant Local HOPC+STK-D method is 82.9% which is still higher than HOJ3D [27], AE [28], HON4D [9], and LARP [59].

#### 8.4 MSR Daily Activity3D Dataset

This dataset [39] contains 16 daily activities performed twice by 10 subjects, once in standing position and once while sitting. Most activities involve human-object interactions which makes this dataset challenging. We follow the experimental setting of [39] and use samples from half of the subjects as training data, and the rest as test data. As shown in Table 5, the proposed Holistic HOPC outperformed all techniques achieving an average accuracy of 88.8%. The view-invariant Local HOPC+STK-D outperformed AOG [29], HOJ3D [27] and LARP [59]; however, it achieved lower accuracy than HON4D [9] and SNV [10], because these methods assume that the training and test samples are obtained from the same viewpoint.

#### 8.5 MSR Gesture3D Dataset

The MSR Gesture3D dataset [40] contains 12 American sign language gestures performed 2 to 3 times by 10 subjects. For comparison with previous techniques, we use the leave-one-subject-out cross validation scheme proposed by [40]. Table 5 compares our methods to existing ones excluding AE [28], LARP [59], AOG [29] and HOJ3D [27] since they require 3D joint positions which are not present in this dataset. Our Holistic HOPC outperformed all techniques and achieved an average accuracy of 96.2%. The Local HOPC+STK-D achieves an accuracy of 93.6% which is higher than HON4D [9].

#### 8.6 Effects of Adaptable Support Volume

#### 8.6.1 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

TABLE 5: Comparison of average action recognition accuracy (%) on the MSR Action3D [38], MSR Daily Activity3D [39], and MSR Gesture3D [40] datasets. NA: RGB or skeleton data Not Available.

Method	Action	DailyActivity	Gesture
AOG [29] (RGB) HOJ3D [27] (Skeleton) AE [28] (Norm. Skeleton) LARP [59] (Norm. Skeleton) AOG [29] (Depth) HON4D [9] (Depth) SNV [10] (Depth)	NA 63.6 81.6 78.8 NA 82.2 84.4	73.1 66.8 85.8 69.4 53.8 80.0 86.3	NA NA NA NA 92.5 94 7
Holistic HOPC (Depth) Local HOPC+STK-D (Depth)	<b>86.5</b> 82.9	88.8 78.8	<b>96.2</b> 93.6

TABLE 6: Average recognition accuracy of the proposed method in three different settings on the Northwestern-UCLA Multiview Action3D [29] and the UWA3D 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 [58].

Dataset	Spatial scale selection method					
Dataset	Constant	Subject height	Automatic			
N-UCLA	77.9	80.0	79.5			
UWA3DII	48.0	52.2	50.9			

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. [58]. Table 6 shows the average accuracy of the proposed method in the three settings. Using the subject's height to find a subject specific scale for the STKs turns out to be the best approach. Automatic scale selection performs closely and can be a good option if the subject's height cannot be measured due to occlusions. Constant scale for all subjects performs the worst. However, the performance of our algorithm is better than existing techniques in all three settings.

#### 8.6.2 Automatic Temporal Scale Selection

We evaluate the improvement gained by our method using automatic temporal scale selection by repeating our experiments with constant temporal scale for STK detection and Local HOPC descriptor extraction. Table 7 shows the average recognition accuracy of our proposed method using a constant temporal scale ( $\tau = 2$ ) and automatic temporal scale selection. The proposed automatic temporal scale selection technique achieved higher accuracy which shows the robustness of our method to action speed variations.

# 8.7 Evaluation of Parameters and Computation Time

#### 8.7.1 Number of STKs

To study the effect of the total number of STKs  $(n_k)$ , we select STKs with the top  $n_k = 100, 400, 700, 1000$  quality factors as shown in Fig. 10. Note how the STK detector effectively captures the movement of the hands in the highest quality STKs, and noisy points begin to appear as

Dataset	Constant	Automatic
N-UCLA	78.3	80.0
UWA3DII	49.2	52.2



Fig. 10: STKs (shown in red) 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  $n_k = 1000$ .

late as  $n_k = 1000$ . Figure 11(a) shows the influence of the number of STKs on the average recognition accuracy. The proposed method achieves the best recognition accuracy when  $n_k = 400$ ; however, the performance remains stable up to  $n_k = 700$ .

#### 8.7.2 Threshold Values

We evaluate the effect of the eigenratio thresholds  $\theta_{\rm STK}$  for STK detection in (9),  $\theta_1$  for view-invariant Local HOPC in Section 5, and  $\theta_g$  for STK-D in Section 6.3 on the average recognition accuracy of our proposed method. Figures 11(b)-(d) show our results. Notice that there is a large range ( $1.1 \le \theta_{\rm STK} \le 1.5$ ) over which the recognition accuracy remains stable. For very small values of  $\theta_{\rm STK}$ ,



Fig. 11: Average recognition accuracy of Local HOPC+STK-D versus (a) the number of STKs, (b)  $\theta_{STK}$ , (c)  $\theta_{l}$ , and (d)  $\theta_{g}$  on the Northwestern-UCLA [29] and the UWA3D Multiview Activity II datasets.

a unique coordinate basis can not be obtained and for larger values of  $\theta_{\text{STK}}$ , the number of detected STKs is not sufficient.

A more stable trend in recognition accuracy can be observed for varying the thresholds  $\theta_1$  and  $\theta_g$ . The recognition accuracy starts to decrease when  $\theta_1 > 1.5$  because the number of points within the spatio-temporal support volume of STKs which have unique eigenvectors starts 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 (Section 6.3) almost always finds an unambiguous orientation for all values of  $\theta_g$ .

#### 8.7.3 Computation Time

The average computational time of the STK detection is 1.7 seconds per frame on a 3.4GHz machine with 24GB RAM using Matlab. The calculation of Local HOPC at STKs takes 0.2 seconds per frame. The overall computational time of the proposed view-invariant method is about 2 seconds per frame. However, the proposed view-dependent Holistic HOPC is faster and take only 0.6 seconds per frame. The average computation time of the nearest competitor AOG [29] that uses depth images is 1.4 seconds per frame. However, our method outperforms AOG [29] on single-view and multi-view datasets by significant margins. Moreover, the calculation of STK and HOPC are individually parallel in nature and can be implemented on a GPU.

### 9 DISCUSSION AND CONCLUSION

Performance of the current 3D action recognition techniques degrades under viewpoint variations because they treat 3D videos as depth image sequences. Depth images are defined with respect to a particular viewpoint and are thus highly dependent on the viewpoint. We have proposed an algorithm for cross-view action recognition which directly processes 3D pointcloud videos to achieve robustness to variations in viewpoint, subject scale and action speed. We have also proposed the HOPC descriptor that is well integrated with our proposed spatio-temporal keypoint (STK) detection algorithm. Local HOPC descriptor combined with global STK-Distribution achieve state-of-the-art results on two standard cross-view action recognition datasets.

Unlike HOJ3D [27], LARP [59], AE [39], and AOG [29], our method does not require skeleton data. Skeleton or joint estimation methods such as [43] do not work well when the human body is only partially visible. Moreover, joint estimation may not be reliable when the subject is not in an upright position (e.g. patient lying on bed) [8] or touches the background. Finally, surveillance cameras are usually mounted at elevated angles which causes further difficulties in joint estimation [8]. Thus, our proposed methods (and other non-skeleton based methods) are more general in the sense that they can be applied to a wider variety of action recognition problems where skeletonization of the data is either not possible or has not been achieved yet.

### 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 [3], [8]–[10], [36], [59], [60] for making their codes publicly available. This research is supported by ARC Discovery grant DP110102399.

#### REFERENCES

- J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," in ACM Computing Survey, 2011.
- [2] D. Weinland, R. Ronfard, and 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, and Q. Tian, "Human daily action analysis with multi-view and color-depth data," in *ECCVW*, 2012.
- [4] C. Lu, J. Jia, and C. keung Tang, "Range-sample depth feature for action recognition," in CVPR, 2014.
- [5] H. Rahmani, A. Mahmood, D. Huynh, and A. Mian, "Action classification with locality-constrained linear coding," in *ICPR*, 2014.
- [6] H. Rahmani, A. Mahmood, A. Mian, and D. Huynh, "Real time action recognition using histograms of depth gradients and random decision forests," in WACV, 2014.
- [7] X. Yang, C. Zhang, and Y. Tian, "Recognizing actions using depth motion maps-based histograms of oriented gradients," in ACM ICM, 2012.
- [8] L. Xia and J. Aggarwal, "Spatio-temporal depth cuboid similarity feature for activity recongition using depth camera," in CVPR, 2013.
- [9] O. Oreifej and Z. Liu, "HON4D: histogram of oriented 4D normals for activity recognition from depth sequences," in CVPR, 2013.
- [10] X. Yang and Y. Tian, "Super normal vector for activity recognition using depth sequences," in *CVPR*, 2014.
- [11] H. Zhang, W. Zhou, C. Reardon, and L. E. Parker, "Simplex-based 3D spatio-temporal feature description for action recognition," in *CVPR*, 2014.
- [12] Y. Tian, R. Sukthankar, and M. Shah, "Spatiotemporal deformable part models for action detection," in *CVPR*, 2013.
- [13] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The Moving Pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection," in *ICCV*, 2013.
- [14] X. Yang and Y. Tian, "EigenJoints-based action recognition using naive bayes nearest neighbor," in CVPRW, 2012.
- [15] A. Yilmaz and M. Shah, "Action sketch: a novel action representation," in CVPR, 2005.
- [16] T. Syeda-Mahmood, A. Vasilescu, and S. Sethi, "Action recognition from arbitrary views using 3D exemplars," in *ICCV*, 2007.
- [17] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and viterbi path searching," in CVPR, 2007.
- [18] D. Gavrila and L. Davis, "3D model-based tracking of humans in action: a multi-view approach," in *CVPR*, 1996.
- [19] T. Darrell, I. Essa, and A. Pentland, "Task-specific gesture analysis in real-time using interpolated views," in *PAMI*, 1996.
- [20] B. Li, O. Camps, and M. Sznaier, "Cross-view activity recognition using hankelets," in CVPR, 2012.
- [21] V. Parameswaran and R. Chellappa, "View invariance for human action recognition," in *IJCV*, 2006.
- [22] C. Rao, A. Yilmaz, and M. Shah, "View-invariant representation and recognition of actions," in *IJCV*, 2002.
- [23] T. Syeda-Mahmood, A. Vasilescu, and S. Sethi, "Recognizing action events from multiple viewpoints," in *IEEE Workshop on Detection* and Recognition of Events in Video, 2001.
- [24] S. Seitz and C. Dyer, "View-invariant analysis of cyclic motion," in *IJCV*, 1997.
- [25] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *ICCV*, 2005.
- [26] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," in CVIU, 2006.
- [27] L. Xia, C. C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3D joints," in CVPRW, 2012.
- [28] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Learning actionlet ensemble for 3D human action recognition," in *PAMI*, 2013.

- [29] J. Wang, X. Nie, Y. Xia, Y. Wu, and S. Zhu, "Cross-view action modeling, learning and recognition," in CVPR, 2014.
- [30] A. Gupta, J. Martinez, J. J. Little, and R. J. Woodham, "3D pose from motion for cross-view action recognition via non-linear circulant temporal encoding," in CVPR, 2014.
- [31] J. Zheng and 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, and C. Shi, "Crossview action recognition via a continuous virtual path," in *CVPR*, 2013.
- [33] A. Farhadi and M. K. Tabrizi, "Learning to recognize activities from the wrong view point," in ECCV, 2008.
- [34] A. Farhadi, M. K. Tabrizi, I. Endres, and D. A. Forsyth, "A latent model of discriminative aspect," in *ICCV*, 2009.
- [35] J. Liu, M. Shah, B. Kuipersy, and S. Savarese, "Cross-view action recognition via view knowledge transfer," in CVPR, 2011.
- [36] R. Li and T. Zickler, "Discriminative virtual views for cross-view action recognition," in *CVPR*, 2012.
- [37] H. Rahmani, A. Mahmood, D. Q Huynh, and A. Mian, "HOPC: Histogram of oriented principal components of 3D pointclouds for action recognition," in *ECCV*, 2014.
- [38] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in CVPRW, 2010.
- [39] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in CVPR, 2012.
- [40] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3D action recognition with random occupancy patterns," in ECCV, 2012.
- [41] X. Yang and Y. Tian, "A survey of vision-based methods for action representation, segmentation and recognition," 2011.
- [42] L. Campbell and A. Bobick, "Recognition of human body motion using phase space constraints," in *ICCV*, 1995.
- [43] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR*, 2011.
- [44] A. W. Vieira, E. Nascimento, G. Oliveira, Z. Liu, and 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, and 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, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *ICCV*, 2005.
- [48] G. Willems, T. Tuytelaars, and L. Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in ECCV, 2008.
- [49] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *ICCV*, 2013.
- [50] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in CVPR, 2011.
- [51] S. Wu, O. Oreifej, and M. Shah, "Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories," in *ICCV*, 2011.
- [52] N. J. Mitra and A. Nguyen, "Estimating surface normals in noisy point clouds data," in SCG, 2003.
- [53] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in CVPR, 2008.
- [54] Y. W. Chen and C. J. Lin, "Combining SVMs with various feature selection strategies," in *Feature Extraction*, ser. Studies in Fuzziness and Soft Computing. Springer Berlin Heidelberg, 2006.
- [55] J. Xie and C. Wang, "Using support vector machines with a novel hybrid feature selection method for diagnosis of erythematosquamous diseases," in *Expert Systems with Applications*, 2011.
- [56] H. S. M. Coxeter, "Regular polytopes." Dover Publications, 1973.
- [57] F. Timbari and L. D. Stefano, "Performance evaluation of 3D keypoint detectors," in *IJCV*, 2013.
- [58] A. Mian, M. Bennamoun, and R. Owens, "On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes," in *IJCV*, 2010.
- [59] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a Lie group," in *CVPR*, 2014.
- [60] R. Gopalan, R. Li, and R. Chellapa, "Domain adaption for object recognition: An unsupervised approach," in *ICCV*, 2011.



Hossein Rahmani received his B.Sc. degree in Computer Software Engineering in 2004 from Isfahan University of Technology (IUT), Isfahan, Iran and the M.S. degree in Software Engineering in 2010 from Shahid Beheshti University (SBU), Tehran, Iran. His research interests include computer vision, 3D shape analysis, and pattern recognition. He is currently working towards his PhD degree in computer science from The University of Western Australia. His current research is

focused on RGB-Depth based human activity recognition.



Arif Mahmood obtained Gold Medal in MS and completed PhD with distinction from Lahore University of Management Sciences in 2011. He is currently PostDoc Researcher in Qatar University. Previously he was Research Assistant Professor in The University of Western Australia. His major research in terests include data clustering and classification, action and object recognition, crowd analysis, community detection and content based image search and matching.



**Du Huynh** is an Associate Professor at the School of Computer Science and Software Engineering, The University of Western Australia. She obtained her Ph.D in Computer Vision, in 1994, at the same university. Since then, she has worked for the Australian Cooperative Research Centre for Sensor Signal and Information Processing (CSSIP) and Murdoch University. She has been a visiting scholar at Lund University, Malmo University, Chinese University of Hong Kong, Nagoya

University, Gunma University, and the University of Melbourne. Associate Professor Huynh has won several grants funded by the Australian Research Council. Her research interests include shape from motion, multiple view geometry, video image processing, and visual tracking.



**Ajmal Mian** completed his PhD from The University of Western Australia in 2006 with distinction and received the Australasian Distinguished Doctoral Dissertation Award from Computing Research and Education Association of Australasia. He received two prestigious nationally competitive fellowships namely the Australian Postdoctoral Fellowship in 2008 and the Australian Research Fellowship in 2011. He received the UWA Outstanding Young Investigator Award in

2011, the West Australian Early Career Scientist of the Year award in 2012 and the Vice-Chancellors Mid-Career Research Award in 2014. He has secured five Australian Research Council grants worth over \$2.3 Million. He is currently with the School of Computer Science and Software Engineering, The University of Western Australia. His research interests include computer vision, action recognition, 3D shape analysis, hyperspectral image analysis, machine learning, and multimodal biometrics.