LEARNING INFORMATIVE PAIRWISE JOINTS WITH ENERGY-BASED TEMPORAL PYRAMID FOR 3D ACTION RECOGNITION

Mengyuan Liu†, Chen Chen‡ and Hong Liu†∗

†Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University, China
‡Center for Research in Computer Vision, University of Central Florida, USA
liumengyuan@pku.edu.cn chenchen870713@gmail.com hongliu@pku.edu.cn

ABSTRACT
This paper presents an effective local spatial-temporal descriptor for action recognition from skeleton sequences. The unique property of our descriptor is that it takes the spatial-temporal discrimination and action speed variations into account, intending to solve the problems of distinguishing similar actions and identifying actions with different speeds in one goal. The entire algorithm consists of two stages. First, a frame selection method is used to remove noisy skeletons for a given skeleton sequence. From the selected skeletons, skeleton joints are mapped to a high dimensional space, where each point refers to kinematics, time label and joint label of a skeleton joint. To encode relative relationships among joints, pairwise points from the space are then jointly mapped to a new space, where each point encodes the relative relationships of skeleton joints. Second, Fisher Vector (FV) is employed to encode all points from the new space as a compact feature representation. To cope with speed variations in actions, an energy-based temporal pyramid is applied to form a multi-temporal FV representation, which is fed into a kernel-based extreme learning machine classifier for recognition. Extensive experiments on benchmark datasets consistently show that our method outperforms state-of-the-art approaches for skeleton-based action recognition.

Index Terms— action recognition, skeleton sequence

1. INTRODUCTION

Human action recognition plays an important role in applications involving automatic analysis of human actions, such as intelligent surveillance, human-computer interaction and sign language analysis. An intuitive way to analyse human actions is to estimate human poses from 2D images, facing semantic ambiguities induced by cluttered backgrounds and loss of depth data [1]. The development of RGB-D cameras, in particular the Kinect [2], opens up opportunities in addressing above problems [3, 4, 5, 6, 7]. With the implementation of capturing skeletons from Kinect in realtime [8], recent works focus on skeleton-based action recognition [9, 10].

It remains a key problem to efficiently describe the spatial-temporal skeleton joints for action recognition from skeleton sequences. Xia et al. assigned 3D joints to histograms of 3D joint locations (HOJ3D) by designing a global spherical coordinate system [11]. Vemulapalli et al. explicitly estimated relative 3D geometry between different body parts by the special Euclidean group SE(3) [12]. Generally, these methods encode the spatial distributions of 3D joints. However, the temporal domain is unexplored, leading to the loss of motion information and temporal information of skeleton joints. To encode motion information, Yang et al. adopted the differences of joints in temporal and spatial domains to describe the dynamics of joints [13]. Zanfir et al. provided a non-parametric Moving Pose (MP) framework [14], which considers features including absolute position, speed and acceleration of each joint. To leverage temporal information, Evangelidis et al. divided a skeleton sequence into equal segments, from which the skeletal quads features were extracted [15]. Hussein et al. enhanced the temporal pyramid structure and divided a skeleton sequence into equal segments with overlapping [16]. Recently, Liu et al. [17] applied a Long Short Term Memory (LSTM) network with trust gates to learn the spatial-temporal information of joints.

However, these methods still lack spatial-temporal discrimination of skeleton joints to distinguish similar actions and suffer from the effect of action speed variations. To solve these problems, we propose a descriptive local spatial-temporal descriptor for skeleton joints and design an energy-based temporal pyramid to alleviate the problem of speed variations. The pipeline of extracting representation from a skeleton sequence is shown in Figure 1. Specifically, we extend the concept of global skeleton kinematics in [14] and calculate joint kinematics to describe local dynamics of joints. Besides, our local descriptor provides a complete description of joints, including temporal cue, i.e., time label, spatial cue,
i.e., joint label, and spatial-temporal cue, i.e. joint kinematics. Moreover, our local descriptor encodes relative relationships among joints, therefore it is able to capture contextual information. Our energy-based temporal pyramid differs from traditional temporal pyramid [15, 16] in two aspects. First, we divide the skeleton sequence into segments according to the motion energy, therefore the segments suffer less from speed variations. Second, we further encode segments by FV and Linear Discriminant Analysis (LDA), leading to a compact representation.

The main contributions of our method are two fold. First, a local spatial-temporal descriptor is developed to capture relative relationships of skeleton joints. Second, an adaptive pyramid based on skeleton energy is proposed to encode temporal information and tackle with speed variations.

2. SKELETON SEQUENCE REPRESENTATION

2.1. Local Spatial-Temporal Descriptor

Particle kinematics is the study of a particle trajectory. The kinematics of a particle includes position, velocity and acceleration. The position of a particle is defined to be the coordinate vector from the origin of a coordinate frame to the particle. The velocity of a particle is a vector that describes the direction and the magnitude of the particle’s motion. The acceleration is the rate of change of the magnitude of the velocity vector plus the rate of change of direction of that vector.

To analyze a sequence of skeletons, we treat each joint as a particle and extract particle’s kinematics from the trajectory of each joint. Let function \( f(\cdot) \) represent a mapping from \( \mathbb{R} \) to \( \mathbb{R}^3 \). Given a variable \( t \in \mathbb{R} \), function \( f(t) \) provides a mapping to a position in \( \mathbb{R}^3 \), whose origin locates at point \( o \). The Taylor approximation of \( f(t) \) can be formulated as:

\[
 f(t) = f(t_0) + f'(t_0)(t - t_0) + \frac{1}{2} f''(t_0)(t - t_0)^2 + \beta,
\]

where \( \beta \to 0 \) when \( t \to t_0 \). \( f(t_0) \) means the position of one joint. Accordingly, the velocity and acceleration can be defined as \( f'(t_0) \) and \( f''(t_0) \). Therefore, kinematics of \( j \)-th joint in time \( t \) is denoted as:

\[
 k^j_t = \begin{bmatrix} f_{j,o}(t) & f'_{j,o}(t) & f''_{j,o}(t) \end{bmatrix},
\]

where function \( f_{j,o}(\cdot) \) returns the position of the \( j \)-th joint, whose origin locates at point \( o \).

The \( j \)-th joint on the \( t \)-th skeleton can be mapped to one point in a high dimensional space:

\[
 [k^j_t \ j \ t] \in \Omega,
\]

which involves both spatial and temporal cues of the joint.

Formula 2 treats each joint as an individual particle and calculates the kinematics of each joint separately. As a result, the relationships among different joints are ignored. To address this, we define a concept of relative kinematics, which calculates the kinematics of one joint by referring to another joint. Specifically, the mapping can be revised as:

\[
 f_{j,i}(t) = f_{j,o}(t) - f_{i,o}(t),
\]

which encodes relative coordinates between pairwise joints. Similarly, \( f'_{j,i}(t) \) and \( f''_{j,i}(t) \) can be expressed as:

\[
 f'_{j,i}(t) = f'_{j,o}(t) - f'_{i,o}(t) \quad \text{and} \quad f''_{j,i}(t) = f''_{j,o}(t) - f''_{i,o}(t).
\]

The relative 3D kinematics of joint \( j \) respect to joint \( i \) is denoted by:

\[
 k^i_{j,i} = \begin{bmatrix} f_{j,i}(t) & f'_{j,i}(t) & f''_{j,i}(t) \end{bmatrix}.
\]

Compared with traditional kinematics, relative kinematics jointly encodes both individual state and the relative spatial-temporal distribution of each joint.

Accordingly, the space \( \Omega \) is transformed to space \( \Upsilon \):

\[
 [k^i_{j,i} \ \Psi(j,i) \ t] \in \Upsilon,
\]

which encodes the relative kinematics, joint label and time label of pairwise joints. Function \( \Psi \) is defined as:

\[
 \Psi(j,i) = j + (i - 1) \ast N,
\]

where \( N \) is the total number of joints in each skeleton, and \( \Psi(j,i) \) is an indicator of pairwise joint labels. Till now, a skeleton sequence is treated as a cloud of points in the space \( \Upsilon \in \mathbb{R}^{11} \), where each point refers to a local feature.
2.2. Fisher Vector Representation

Fisher kernel [18] is used to aggregate local features extracted from skeletons as a compact representation. Let \( Q = \{ q_i | i = 1...M \} \) represent a set of \( M \) local features in a skeleton sequence. Assuming statistical independence, \( Q \) can be modeled by a \( k \)-component Gaussian mixture model:

\[
p(Q|\theta) = \prod_{i=1}^{M} \sum_{k=1}^{K} \omega_k \mathcal{N}(q_i|\mu_k, \delta_k), \tag{8}
\]

where \( \theta = \{ \omega_k, \mu_k, \delta_k \} \) represents the GMM mixture parameter set with weight parameter, mean vector and diagonal covariance matrices of Gaussian \( k \). Subsequently, an unseen set \( Q \) can be described by its Fisher score [19], using the Jacobian of the log-probability with respect to the GMM parameters: \( \mathcal{G}_\theta^Q = \nabla_\theta \log p(Q|\theta) \). The Fisher kernel, however, relates any two such vectors through a bilinear form based on the inverse of the Fisher information matrix. Since the decomposition of this matrix is possible, one can write the Fisher kernel as a linear kernel of the so called Fisher Vector (FV), denoted by \( \mathcal{G} \).

The reader is referred to [19] for a detailed description. As in [18], we consider the Jacobians with respect to non-scalar parameters only, so that the FV consists of the concatenation of two vectors \( \mathcal{G}_{\mu_k}^Q \) and \( \mathcal{G}_{\delta_k}^Q \). The \( j \)-th entry for the \( k \)-th mixture component is given by:

\[
\mathcal{G}_{\mu_k}^Q(j) = \frac{1}{M \sqrt{\pi_k}} \sum_{i=1}^{M} \gamma_{k,i} \left( q_i^j - \mu_k^j \right), \tag{9}
\]

\[
\mathcal{G}_{\delta_k}^Q(j) = \frac{1}{M \sqrt{2\pi_k}} \sum_{i=1}^{M} \gamma_{k,i} \left[ \left( q_i^j - \mu_k^j \right)^2 - 1 \right].
\]

where \( \gamma_{k,i} \) is the posterior probability that \( q_i \) belongs to \( k \)-th cluster conditioned by \( Q \). The FV generated from local feature set \( Q \) is formulated as:

\[
FV^Q = [\{(\mathcal{G}_{\mu_k}^Q(j))_{j=1}^{11}\}_{k=1}^{K}, \{(\mathcal{G}_{\delta_k}^Q(j))_{j=1}^{11}\}_{k=1}^{K}] . \tag{10}
\]

Linear Discriminant Analysis (LDA) [20] can find the underlying feature space that best discriminates between different classes. Based on the class specific information, it maximizes the ratio of the intra-class scatter matrix and the within-class scatter matrix. Therefore, we apply LDA to \( FV^Q \) to further extract dominant features from the FV representation.

2.3. Energy-Based Temporal Pyramid

Above FV representation \( FV^Q \) ignores temporal information, which is essential for encoding sequential data. A simple way to involve temporal information is to divide a sequence into equal segments, and then encode all segments in a hierarchical manner [15, 16]. However, this method cannot cope with speed variations. This issue, can be addressed by dividing a sequence into segments according to the motion energy. The observation is that the state of human pose in a sequence is related to motion energy.

For skeleton sequences, we introduce a concept named as

\[
\text{skeleton energy, which is defined as:}
\]

\[
E^t = \sum_{j=1}^{N} ||f_{j,o}(t) - f_{j,o}(t-1)||, \tag{11}
\]

where function \( || \cdot || \) calculates the Euclidean distance, \( E^t \) records the energy from the \((t-1)\)-th frame to the \(t\)-th frame. Similarly, we define the skeleton energy for a segment as the accumulated skeleton energy of consecutive frames.

Since depth data is noisy, the estimated joints may also contain noise, which affects the estimation of skeleton energy. To address this issue, we start by selecting proper frames, which contain less noisy joints, before calculating skeleton energy. The pipeline of frame selection is shown in Figure 2. On the \( t \)-th frame, we calculate the total length of limbs, denoted by \( l^t \). Intuitively, when \( t \) changes from 1 to \( T \), \( l^t \) should remain unchanged to some extent. Therefore, we use the medium value \( ME \) of a set \( \{ l^t \}_{t=1}^{T} \) as a decision making criterion. When \( |l^t - ME| < T \cdot ME \), the \( t \)-th frame is preserved; otherwise, the frame is discarded. The threshold \( T \cdot ME \) is set to 0.05, which works well on different datasets.
Table 1. Recognition accuracies of different methods on MSRAction3D dataset [21]

<table>
<thead>
<tr>
<th>MSRAction3D dataset (protocol of [21])</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Skeleton Joint Features [22]</td>
<td>77.50%</td>
<td></td>
</tr>
<tr>
<td>HOJ3D [11]</td>
<td>78.97%</td>
<td></td>
</tr>
<tr>
<td>Skeletal Shape Trajectories [23]</td>
<td>89.00%</td>
<td></td>
</tr>
<tr>
<td>FV of skeletal quads [15]</td>
<td>89.86%</td>
<td></td>
</tr>
<tr>
<td>Grassmann Manifold [24]</td>
<td>91.21%</td>
<td></td>
</tr>
<tr>
<td>Moving Pose [14]</td>
<td>91.70%</td>
<td></td>
</tr>
<tr>
<td>Lie Group [12]</td>
<td>92.46%</td>
<td></td>
</tr>
<tr>
<td>Attribute Mining [25]</td>
<td>92.97%</td>
<td></td>
</tr>
<tr>
<td>Our method</td>
<td><strong>93.81%</strong></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4. Confuse matrix our method on MSRAction3D dataset

As shown in Figure 3, we segment a skeleton sequence into two segments with equal skeleton energy. Correspondingly, the total feature set \( Q \) is divided into \( Q_1 \) and \( Q_2 \). Let \( FV^{Q_1} \) and \( FV^{Q_2} \) denote FV representation of \( Q_1 \) and \( Q_2 \), respectively. A hierarchical FV representation of a skeleton sequence is formed as:

\[
FV = [FV^{Q_1}, FV^{Q_2}],
\]

which is used as input to the Kernel-based Extreme Learning Machine (KELM) classifier [26] with a Radial Basis Function (RBF) kernel for classification.

3. EXPERIMENTS AND DISCUSSIONS

The proposed method is evaluated on public benchmark datasets: MSRAction3D [21] and UTKinect-Action [11]. To test the robustness to speed variations, a MSRAction3DSpeed dataset is collected based on MSRAction3D dataset. Our method is implemented by a three-level structure of hierarchical FV representation.

3.1. MSRAction3D Dataset

The MSRAction3D dataset [21] contains 20 actions, where each action is performed 2 or 3 times by 10 subjects facing the depth camera, resulting in 567 sequences. This dataset is challenging due to large inter-class similarities, e.g. actions “hammer” and “high throw” are similar except for a slight difference in the movement of one hand. We follow the same experimental setting as [21], i.e. the cross subject setting: subjects 1,3,5,7,9 for training and subjects 2,4,6,8,10 for testing. When the cluster center \( K \) is set to 50, our hierarchical FV representation achieves an overall accuracy of 93.81%, which is 14.84% higher than [11]. From this it is evident that incorporating dynamic and temporal information in our method has led to considerable improvements in effective recognition of similar actions. We compare our method with state-of-the-art approaches in Table 1, where our method outperforms all skeleton-based methods, including the most related works [14], [15]. This result shows that our method involves more spatial-temporal discrimination of joints. The confusion matrix of our method is shown in Figure 4, where high recalls and precisions are achieved.

3.2. UTKinect-Action Dataset

The UTKinect-Action dataset [11] contains 10 actions, where each action is performed 2 times by 10 subjects, resulting in 200 sequences. Some snaps are shown in Figure 5, where action “pull” and action “push” share similar joint movements. Moreover, same type of action performed by different subjects may look different (see Figure 5 (c) and (d)). We use the leave-one-out cross validation [11] and the cross subject setting, that is subjects 1,2,3,4,5 for training and subjects 6,7,8,9,10 for testing [22]. When the cluster center \( K \) is set to 30, our hierarchical FV representation achieves an overall accuracy of 97.47% using the leave-one-out setting and 94.44% using the cross subject setting. In Table 2, we compare our method with state-of-the-art approaches, where our method outperforms all hand-crafted features and achieves compatible results with recent deep learning methods, e.g. 3D\(^2\)CNN+Joint Vector [28] and ST-LSTM [17].

![Figure 5. Skeleton sequences from UTKinect-Action dataset, where intra-similarities between (a) and (b), and inter-variations between (c) and (d) are observed. To facilitate observation, each skeleton is colored according to the time label.](image-url)
3.3. Evaluation of Individual Components

Frame selection. Skeleton sequences from MSRAction3D dataset are usually noisy, which is shown in Figure 6. Frame selection method is designed to remove these skeletons which contain large amount of noise. Table 3 shows the performances of our method with and without frame selection method. An improvement of 3.63% in accuracy is obtain when frame selection method is applied.

Relative relationship. Table 4 shows the performance of local features generated from space Ω and Υ. The reason is that each point from space Υ not only encode the spatial-temporal distribution of skeleton joints but only captures the relative relationships among joints. While, point from space Ω ignores the constrains among joints, therefore lacking discriminative power to distinguish similar actions.

Energy-based temporal pyramid. To encode the time information of a skeleton sequence, we propose an adaptive pyramid structure based on skeleton energy. Table 5 compares our method with that of the traditional temporal pyramid, which encodes time cue by dividing a sequence into equal segments and concatenating them in a hierarchical manner. Table 5 shows that using an energy-based approach rather than the traditional approach leads in further improvement in accuracy. It is noted that “1 level” means directly applying methods to original sequence.

### Table 3. Evaluation of frame selection method

<table>
<thead>
<tr>
<th>Feature components</th>
<th>MSRAction3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>With frame selection</td>
<td>93.81%</td>
</tr>
<tr>
<td>Without frame selection</td>
<td>90.18%</td>
</tr>
</tbody>
</table>

### Table 4. Evaluation of different high dimensional spaces

<table>
<thead>
<tr>
<th>Feature components</th>
<th>MSRAction3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k_{j, t}^{\psi}(j, i) \in \Omega)</td>
<td>93.81%</td>
</tr>
<tr>
<td>(k_{j, t}^{\psi}(j, i) \in \Omega)</td>
<td>89.45%</td>
</tr>
</tbody>
</table>

### Table 5. Evaluation of Energy-based temporal pyramid and traditional temporal pyramid on MSRAction3D dataset

<table>
<thead>
<tr>
<th>Temporal levels</th>
<th>Energy-based pyramid</th>
<th>Temporal pyramid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 level</td>
<td>86.54%</td>
<td>86.54%</td>
</tr>
<tr>
<td>2 levels</td>
<td>92.36%</td>
<td>90.54%</td>
</tr>
<tr>
<td>3 levels</td>
<td>93.81%</td>
<td>91.63%</td>
</tr>
</tbody>
</table>

may change dramatically in a non-linear manner. Table 6 shows that using the energy-based approach rather than temporal pyramid results in 4.72% improvement in accuracy. We achieve 96.81% on this challenging dataset, which verifies the robustness of our method to speed variations.

3.5. Parameters Analysis and Computational Cost

Figure 8 evaluates the effect of parameter \(K\) on MSRAction3D dataset, where the accuracy first rises up and then drops down as the value of \(K\) increases. When \(K\) equals to 50 or 70, we achieve the highest accuracy of 93.81%. Since larger number of \(K\) corresponds to higher dimension of feature, in the experiments we set \(K\) to 50 taking time cost into consideration. When performing on a given dataset, we select the proper value of \(K\) which ensures highest accuracy. Thus, we respectively set \(K\) to 50 and 30 for MSRAction3DSpeed and UTKinect-Action datasets. As shown in Table 5 and Table 6, the accuracy of our method increases along with temporal levels. In this work, we use three levels as default value, taking both accuracy and time cost into account.

To evaluate the computational complexity of our method, we test it on the MSRAction3D dataset with the default parameter of \(K = 50\). The average time required for extracting a local descriptor is 0.0021 second on a 2.5GHz machine with 8GB RAM, using Matlab R2012a. The average computation time for calculating a skeleton sequence descriptor by Fisher kernel encoding method is about 0.0496 second. Applying LDA to a proposed descriptor costs 0.0054 second.

4. CONCLUSIONS

This paper presents an effective and compact representation for skeleton-based action recognition, which is detailed in four stages. First, a local spatial-temporal descriptor is designed to encode relative spatial-temporal relationships among skeleton joints. Second, to make our proposed descriptor robust against the noisy skeleton data, a frame selection method is applied to original skeletons. Third, to cope with speed variations induced by different performers or habits, a skeleton sequence is adaptively partitioned into segments.

![Fig. 6. Noisy skeleton sequences from MSRAction3D dataset](image-url)
Fig. 7. Comparison between linear sampling and random sampling for building the MSRAction3DSpeed dataset

Table 6. Evaluation of Energy-based temporal pyramid and traditional temporal pyramid on MSRAction3DSpeed dataset

<table>
<thead>
<tr>
<th>Temporal levels</th>
<th>Energy-based pyramid</th>
<th>Traditional pyramid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 level</td>
<td>71.27%</td>
<td>71.27%</td>
</tr>
<tr>
<td>2 levels</td>
<td>82.05%</td>
<td>80.95%</td>
</tr>
<tr>
<td>3 levels</td>
<td><strong>86.81%</strong></td>
<td>82.09%</td>
</tr>
</tbody>
</table>

Fig. 8. Evaluation of parameter K on MSRAction3D dataset

with equal skeleton energy. Finally, local descriptors are respectively extracted from segments and represented by FVs in a hierarchical manner. Experimental results on two benchmark datasets, i.e. MSRAction3D and UTKinect-Action, show that our method can effectively encode spatial-temporal joints, therefore outperforms most state-of-the-art skeleton-based action recognition approaches. Results on our collected MSRAction3DSpeed dataset confirms that our method is robust to speed variations.

5. REFERENCES


