Signal Processing 142 (2018) 412–422

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Joints kinetic and relational features for action recognition

Xinmei Tian[*], Jiayi Fan

CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application System, University of Science and Technology of China, Anhui, 230027, China

ARTICLE INFO

Article history:
Received 15 April 2017
Revised 3 August 2017
Accepted 4 August 2017
Available online 5 August 2017

Keywords:
Action recognition
Joints kinetic and relational features

ABSTRACT

High-level pose features (HLPF) have been shown to be very effective and efficient for action recognition. However, motion information has not been sufficiently mined in HLPF. In addition, the position relations of joints are limited with respect to orientation, distance and angle in HLPF. To tackle these problems, we propose a set of comprehensive features, termed joints kinetic and relational features (JKRF), for action recognition. Specifically, for each single joint, we propose a group of kinetic features to describe its velocity, speed, acceleration, acceleration rate, angular velocity, angular acceleration, kinetic energy, potential energy and total energy. For each joint pair, we propose a set of corresponding features to describe the correlation relations of velocity, acceleration, angular velocity, angular acceleration and energy change between joints. Additionally, we propose a set of corresponding features to encode distance relations in the horizontal, vertical, orientation cosine, orientation sine, eigenvector and link path directions. For each joint triplet, we also present a joint vector inner product feature, a joint vector cosine similarity feature and an area perimeter rate feature to describe their geometrical relations. We evaluate our JKRF using three datasets, and the experimental results show that JKRF consistently outperform state-of-the-art action recognition methods.

1. Introduction

With a wide range of applications, such as intelligent surveillance, human-computer interaction, sports video analysis and video retrieval, action recognition [1–4] and pose estimation [5–7] are regarded as fundamental problems in the field of computer vision. Although these tasks have different goals, action recognition often uses the result of pose estimation as its input. However, pose estimation is still a difficult task, where errors often arise from small parts of the human body, because of large variation and blending with complex backgrounds. For this reason, recent action recognition studies have begun to investigate the performance of features under the condition that pose estimation is perfect. Thus, annotated joints are used to study the action recognition problem.

Based on the given pose information, Jhuang et al. [1] proposed a set of human pose-based features, termed high-level pose features (HLPF). High-level pose features greatly outperform low-level features (e.g., dense trajectory [8]) and mid-level features (e.g., dense trajectory using ground truth optical flow and segmentation) on joint-annotated datasets, such as the joint-annotated human motion data base (JHMDB) [1] and the sub-JHMDB [1]. Furthermore, HLPF are very efficient since the features are extracted based on the positions (x and y coordinates) of joints.

However, HLPF also have their limitations. Three features in HLPF describe single joint information: normalized joint positions, the trajectories in Cartesian coordinates and the trajectories in polar coordinates. Four features in HLPF describe pairwise joints relations: distance relation, orientation relation and the trajectories of these relations. The other two features, i.e., the angle relation and its trajectory, describe triplet joints relations. Although HLPF perform well on the abovementioned datasets, the motion information of each joint is not mined sufficiently, and the position relations of the joints are limited in distance, orientation and angle in HLPF. To tackle these problems in HLPF, we construct a set of kinetic and relational features that take motion information, correlation relations, distance relations and geometrical relations into consideration, as shown in Fig. 1.

First, motion is an important source of information for classifying human actions [2,8,9]. To capture the motion information, Jhuang et al. [1] used trajectory features to describe the velocity of each joint or the changes in these relations. To further describe the acceleration of each joint, we need to calculate the trajectories of the trajectory features. In addition, to describe the magnitude of the velocity vector, we build the speed feature. In the same manner, we build the acceleration rate feature to denote the magnitude of the acceleration vector. Since each joint of the human body

[*] Corresponding author.
E-mail addresses: xinmei@ustc.edu.cn, xinmeitian@gmail.com (X. Tian), jyfan91@mail.ustc.edu.cn (J. Fan).

http://dx.doi.org/10.1016/j.sigpro.2017.08.005
0165-1684/© 2017 Elsevier B.V. All rights reserved.

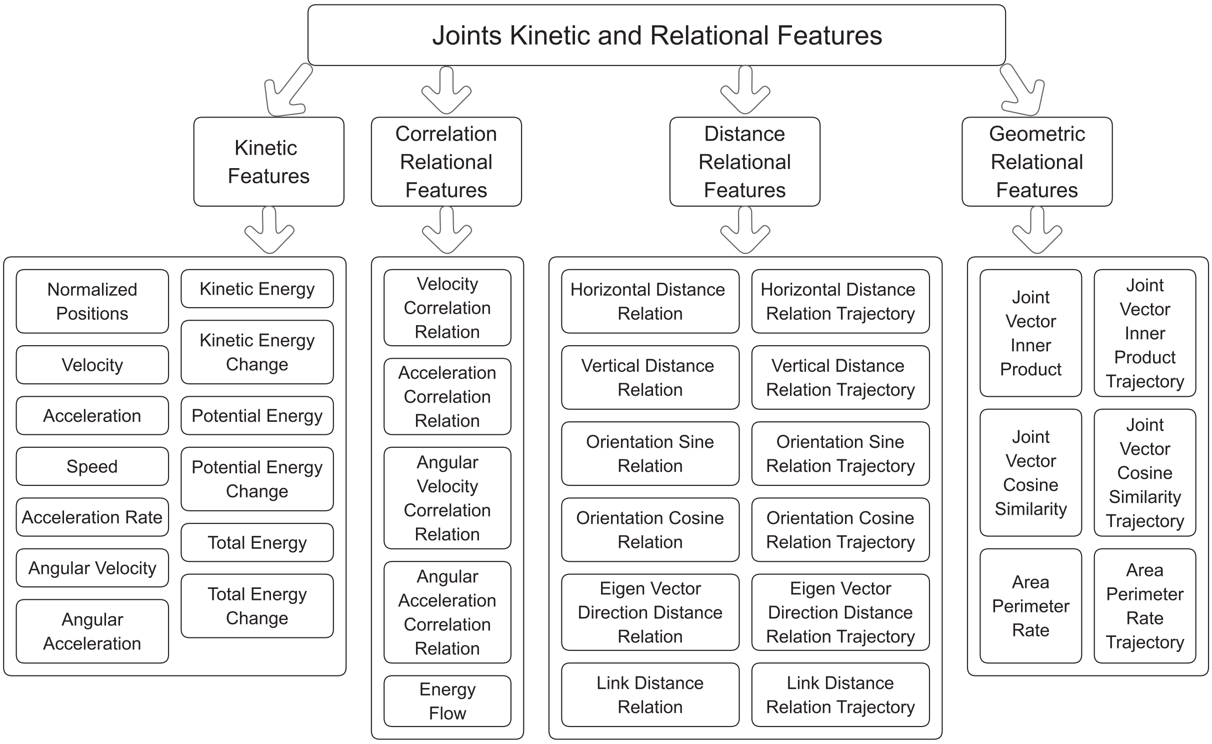

Joints Kinetic and Relational Features
Kinetic Features
Correlation Relational Features
Distance Relational Features
Geometric Relational Features
Normalized Positions
Velocity
Acceleration
Speed
Acceleration Rate
Angular Velocity
Angular Acceleration
Kinetic Energy
Kinetic Energy Change
Potential Energy
Potential Energy Change
Total Energy
Total Energy Change
Velocity Correlation Relation
Acceleration Correlation Relation
Angular Velocity Correlation Relation
Angular Acceleration Correlation Relation
Energy Flow
Horizontal Distance Relation
Horizontal Distance Relation Trajectory
Vertical Distance Relation
Vertical Distance Relation Trajectory
Orientation Sine Relation
Orientation Sine Relation Trajectory
Orientation Cosine Relation
Orientation Cosine Relation Trajectory
Eigen Vector Direction Distance Relation
Eigen Vector Direction Distance Relation Trajectory
Link Distance Relation
Link Distance Relation Trajectory
Joint Vector Inner Product
Joint Vector Inner Product Trajectory
Joint Vector Cosine Similarity
Joint Vector Cosine Similarity Trajectory
Area Perimeter Rate
Area Perimeter Rate Trajectory

Fig. 1. Joints kinetic and relational features.

always revolves around its adjacent ancestor joint(e.g., the hand revolves around the elbow when a person is waving), we use the ratio of the velocity to the length of the limb to describe the angular velocity. Similarly, we calculate the difference in angular velocity to describe the angular acceleration. Rather than the mass of the joint, we simply use the square of the velocity to denote the kinetic energy of the joint. Thus, based on the velocity, we construct the acceleration, speed, acceleration rate, angular velocity, angular acceleration and kinetic energy features. In addition, from the perspective of energy, we use the vertical positions of the joints to construct the potential energy feature and combine it with the kinetic features to construct the total energy feature. These features consist of kinetic features and sufficiently mine the motion information of each single joint.

Second, when a person performs a particular action, there is invariably a kinetic correlation between each pair of joints. For example, when a man claps, his two hands always move in opposite directions. To describe the correlation between each pair of joints, we calculate the cosine similarity of the velocity, the acceleration, the angular velocity and the angular acceleration of the joint pair. In this way, the corresponding correlation relational features are constructed. In addition, we hold the view that for a particular action, energy from each joint will change due to the motion of other joints. For different actions, the energy changes of different joints vary. Thus, we calculate the inner product of the velocity of one joint relative to the other and the relative acceleration to represent the energy flow from one joint to the other. This process is based on the following: first, velocity represents the displacement per unit time; second, acceleration represents the net force per unit mass. Therefore, the product can represent the energy flow from one joint to the other with unit mass per unit time. Accordingly, we use this feature to describe the energy change correlation relation between joints.

Third, for each pair of joints, the discriminative power of distance relation features and orientation relation features in HLPF [1] is not strong enough. Specifically, the distance between the majority of pairs of joints in the human body varies irregularly and is closely related to the behavior habit of the actor. As a consequence, the intra-class variance of the distance relations feature is not small enough. This conclusion also applies to the orientation relations feature. When different people perform the same action, their action amplitudes are usually different. Thus, the orientation of a pair of joints may vary in a wide range for the same action. Moreover, the viewpoint also greatly affects the orientation. Therefore, the intra-class variance of the orientation relations feature is also not small enough. In other words, the discriminative power of these two features is not strong enough. We believe that the primary reason behind this conclusion is that the discriminative power required for a human to recognize actions differs for different orientations. This point is not well considered in constructing these two features. In detail, a large proportion of human actions are accomplished under the condition that the body of the actor is nearly vertical with respect to the ground and that the limbs are nearly horizontal or vertical with respect to the ground. This means that the discriminative power in the horizontal orientation and vertical orientation is much stronger. Therefore, we present features to describe horizontal and vertical distance relations between pairs of joints. However, in most cases, the body cannot be strictly vertical with respect to the ground. Therefore, we calculate the sine function value and the cosine function value of the orientation relations feature, with the orientation of the body's principal direction being subtracted, to represent horizontal and vertical distance relations after fixing the orientation. Furthermore, to find the most discriminative orientation of each joint, we apply principal component analysis (PCA) to the positions of each joint. Thus, we obtain the position of each joint in the eigenvector direction. After calculating the distance between these positions, the distance relations in the direction of the eigenvector are obtained. In addition, to describe the link distance relation between pairs of joints in the observation plane, we calculate the sum of the distances of adjacent joints between each pair of joints. These features compose our distance relational features.

Fourth, for each triplet of joints, to extend the angular information, we calculate the inner product and the cosine similarity between the vectors of joints in each joint triplet to construct features. The construction of the joint vector inner product feature is inspired by the nature of the horizontal (vertical) distance relation feature, which is essentially the inner product between the horizontal (vertical) unit vector and a vector from one joint to another. In addition, the purpose of designing the joint vector cosine similarity feature is to describe orientational correlation relations between vectors in each joint triplet. Moreover, most triplets of joints form triangles. To describe the geometric information, the area and the perimeter are usually used. However, these two kinds of features are not robust with respect to the viewpoint. Accordingly, we use the ratio of the area and the perimeter as features to present geometric information. Our geometric relational features consist of these three features.

Finally, to utilize temporal information [2,8], we further calculate the trajectories of the distance relational features and geometric relational features.

In summary, to tackle the problems that exist in HLPF [1], we propose the joints kinetic and relational features (JKRF). Specifically, to describe the motion information of each joint, we propose the kinetic features. The correlation relational features are designed to describe the kinetic correlation between pairs of joints. Considering that the discriminative power of distance relations and orientation relations is not strong enough, we propose the distance relational features. Using the method of finding the inner product and adding geometrical information, the geometric relational features are constructed. We conduct experiments on three challenging datasets: JHMDB [1], sub-JHMDB [1] and Penn Action dataset [10]. JKRF outperform state-of-the-art action recognition methods on these three datasets.

2. Related works

Action recognition is a popular topic in computer vision, and there are a number of studies related to this topic. In this section, we introduce some related works conducted in recent years. In general, the methods used for action recognition can be grouped into three categories: pose-based methods, hand-craft methods and deep-learned methods.

Pose-based methods Pose estimation and pose-based description are two main classes of these methods. Yang et al. [5] used the flexible mixtures-of-parts model to estimate joint positions. Hong et al. [11] used nonlinear mapping with a multilayer deep neural network to recover a pose. Hong et al. [12] used multiview locality-sensitive sparse retrieval to recover a three-dimensional human pose. Lu et al. [13] proposed a hierarchical MRF model for human action segmentation. Yao et al. [14] proved that actions could be recognized reliably from multiple camera views when using pose estimation. Singh et al. [7] demonstrated that estimated poses were reliable for the action recognition task using relatively simple datasets composed of monocular videos. Nie et al. [15] combined action recognition and pose estimation in a unified framework with a spatial-temporal and-or graph model. To study the influence of the pose estimation error and the effectiveness of the pose-based features on the performance of action recognition, Jhuang et al. [1] constructed two challenging datasets: the JHMDB, in which all the joints are annotated as shown in Fig. 2 (a) and (b), and the sub-JHMDB [1], which is a subset of the JHMDB. In addition, they proposed the HLPF [1], which achieved excellent performance on the two datasets when using annotated joint positions. Cheron et al. [16] also used annotated joint positions and designed pose-based CNN (P-CNN) features. This method performed better than the improved dense trajectory features [2] encoded using the Fisher vector [17].

Hand-craft methods In these methods, interest point detection and feature description are two indispensable procedures for action recognition. The space time interest points [18–20] or dense sampling points [2,8] in a video are detected first. Then, the descriptors [2,8,9,21,22] are computed at these key points or along the trajectory of the dense sampled points. After the descriptors or features are extracted using feature mining approaches [23,24], the video representation [4,17,25,26] is built. Finally, a proper distance metric [27–30] is learned, and the support vector machines are trained to be classifiers [31,32].

Deep-learned methods Inspired by the success of deep learning techniques in image classification [33–36], much effort has been made to develop deep architectures for video action recognition. To describe both appearance and motion information, Simonyan et al. [37] designed a two-stream CNN. Ji et al. [38] extended the 2D CNN to videos. Karpathy et al. [39] tested the CNN with deep structures. By using dense trajectory points to pool 2D CNN feature maps, Wang et al. [40] constructed trajectory-pooled deep convolutional descriptors.

In this work, we use pose-based methods to construct a set of JKRF. JKRF solve the problems present in [1], which is closely related to our work, and our features achieve consistent improvement over state-of-the-art methods on three challenging datasets.

3. High-Level Pose Features (HLPF)

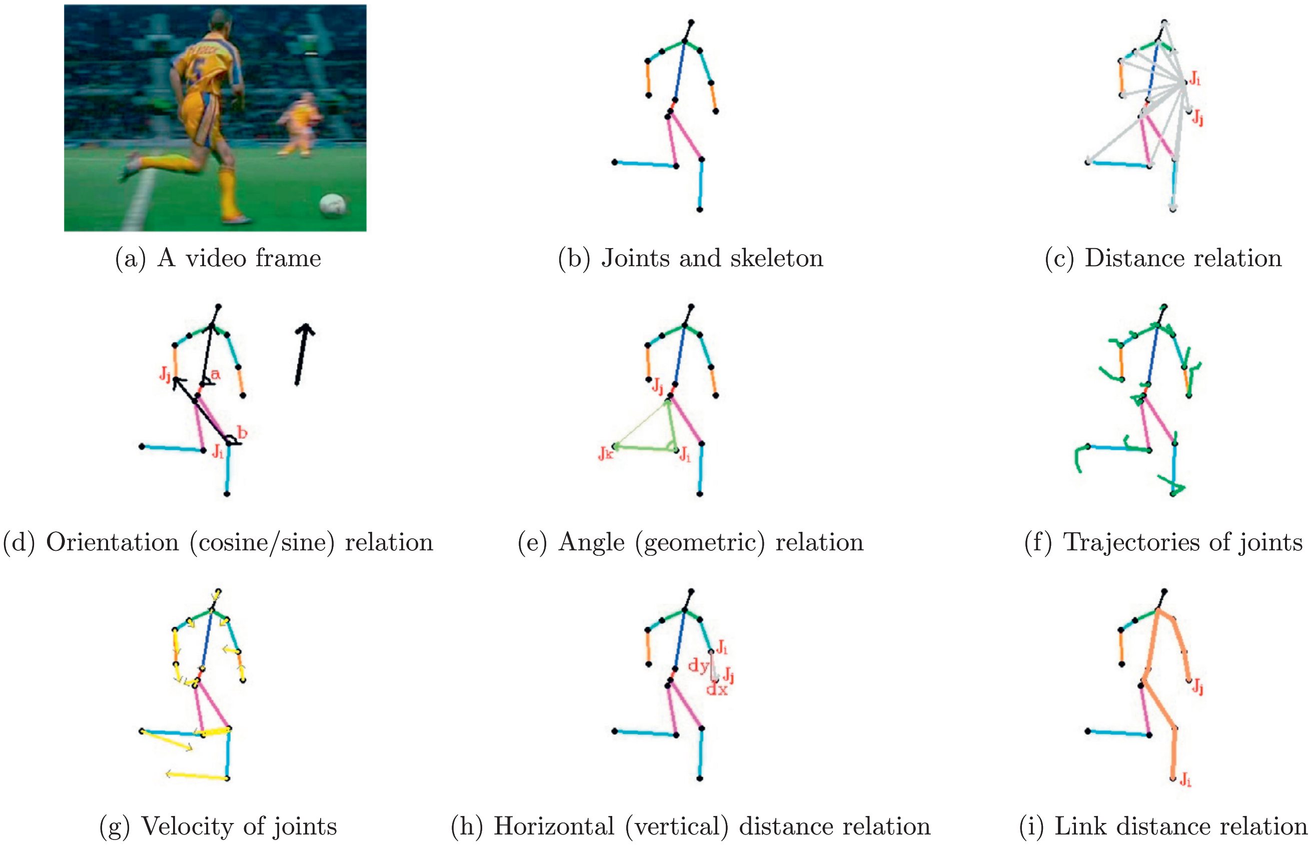
In this section, we briefly introduce HLPF [1], which are used as a baseline in our algorithm. HLPF consist of 9 features, which can be grouped into 3 categories according to the number of joints involved in the process of feature design. Both HLPF and JKRF can be extracted from an arbitrary skeleton. In this paper, we describe these features for a skeleton with 15 joints, for which the x and y coordinates are annotated as shown in Fig. 2 (b).

For each single joint, the position is first normalized with respect to the human scale. The scale normalization process will be introduced in detail in Section 5.1. Then, the position of each joint with respect to the center of the human body is computed to form the **normalized positions feature**. It uses the translation of the joint position along the x and y coordinates $(x_{t_2} - x_{t_1}, y_{t_2} - y_{t_1})$, called the **Cartesian trajectory feature**, as shown in Fig. 2 (f). In addition, the translation of the orientation $arctan(\frac{y_{t_2}-y_{t_1}}{x_{t_2}-x_{t_1}})$ is computed as the **radial trajectory feature**.

For each pair of joints, the **distance relation feature** is obtained by calculating the Euclidean distance between the joints. Fig. 2 (c) shows the distance relation between joint J_i and other joints. The **orientation relation feature** is obtained by calculating the orientation of the vector connecting each pair of joints. Notice that the orientation is normalized by subtracting the orientation from the belly to the neck, which can be regarded as the principal direction of a person. For example, as shown in Fig. 2 (d), the orientation from J_i to J_j is $\angle b - \angle a$. The trajectories of these two features are calculated as the corresponding trajectory features.

For each triplet of joints, the **angle relation feature** is constructed by calculating the inner angles that span the vectors connecting this joint triplet. For instance, the angle relation among joints J_i, J_j and J_k in Fig. 2 (e) is obtained by calculating $\angle J_iJ_jJ_k$, $\angle J_jJ_iJ_k$ and $\angle J_iJ_kJ_j$, respectively. The trajectories of this feature are then extracted as its trajectory feature.

In general, the trajectory features are considered as the differences between spatial features along the trajectory at frame t and frame $t+k$, i.e., the feature of dimension f is a sequence $(f_{t+s} - f_t, \cdots, f_{t+ks} - f_{t+(k-1)s})$, where k is the trajectory length and s is the step size. Fig. 2 (f) shows the trajectories of each joint for $k = 4$ and $s = 1$. We use $k = 2$ and $s = 3$ in this paper, which are the same as the settings used in [1].



(a) A video frame (b) Joints and skeleton (c) Distance relation

(d) Orientation (cosine/sine) relation (e) Angle (geometric) relation (f) Trajectories of joints

(g) Velocity of joints (h) Horizontal (vertical) distance relation (i) Link distance relation

Fig. 2. Overview of the construction of some features in HLPF and JKRF. (a) A video frame from JHMDB. (b) Annotated joints and skeleton. (c) Distance relation between one joint and other joints. (d) Orientation (cosine/sine) relation and the principal direction. (e) Angle (geometric) relation. (f) Trajectories of joints. (g) Velocity of joints. (h) Horizontal (vertical) distance relation. (i) Link distance relation.

HLPF consist of these nine kinds of features. The dimensionality of HLPF is $30+60+30+6\times C_{15}^2+3\times 3\times C_{15}^3=4845$. For each of these nine kinds of features in HLPF, a codebook is generated using k-means for quantization. After each video clip is described by a histogram, the SVM with an RBF-χ^2 kernel is used as the classifier. More details can be found in [1].

4. Joints Kinetic and Relational Features (JKRF)

JKRF consist of 36 kinds of features. In this section, we will introduce these features in detail.

4.1. Kinetic features

To capture the motion information of each single joint, we propose a set of kinetic features. Notice that we only describe the actions in videos and that the following features are independent of the physical features of the subject performing the action.

The **velocity** describes the change in the position of each joint with time, as shown in Fig. 2(g). For joint J_i, the position of which at frame t is $(x_{i,t}, y_{i,t})$, its velocity (V) is:

$$V(i,t)=(x_{i,t+k}-x_{i,t}, y_{i,t+k}-y_{i,t}). \tag{1}$$

Hereafter, we use the setting $k=3$, which yields the best performance.

The **acceleration** describes the change rate of the position of each joint with time. The acceleration (A) of joint J_i at frame t is:

$$A(i,t)=V(i,t+k)-V(i,t). \tag{2}$$

The **speed** describes the magnitude of the velocity. The speed (S) of joint J_i at frame t is:

$$S(i,t)=\sqrt{V(i,t)_x^2+V(i,t)_y^2}. \tag{3}$$

The **acceleration rate** describes the magnitude of the acceleration. The acceleration rate $(ACCR)$ of joint J_i at frame t is:

$$ACCR(i,t)=\sqrt{A(i,t)_x^2+A(i,t)_y^2}. \tag{4}$$

Because the human skeleton can be regarded as a tree structure, to describe the rotation velocity of the child joint with respect to its father joint, we design the **angular velocity**. The distance between joints J_i and J_j at frame t is:

$$DST(i,j,t)=\sqrt{(x_{i,t}-x_{j,t})^2+(y_{i,t}-y_{j,t})^2}. \tag{5}$$

Suppose that the father joint of joint J_i is J_f; then, the angular velocity (AV) of joint J_i at frame t is:

$$AV(i,t)=\frac{V(i,t)}{DST(i,f,t)}. \tag{6}$$

The **angular acceleration** describes the rotation acceleration of the child joint with respect to its father joint. The angular acceleration (AA) of joint J_i at frame t is:

$$AA(i,t)=\frac{A(i,t)}{DST(i,f,t)}. \tag{7}$$

The **kinetic energy** describes the kinetic energy of the joint with unit mass. The kinetic energy (KE) of joint J_i at frame t is:

$$KE(i,t)=\frac{1}{2}(V(i,t)_x^2+V(i,t)_y^2). \tag{8}$$

The **kinetic energy change** describes the change in the kinetic energy with time. The kinetic energy change (*KEC*) of joint J_i at frame t is:

$$KEC(i,t) = KE(i,t+k) - KE(i,t). \quad (9)$$

The **potential energy** describes the gravitational potential energy of the joint with unit mass. The potential energy (*PE*) of joint J_i at frame t is:

$$PE(i,t) = g \cdot y_{i,t}. \quad (10)$$

Here, $g = 10$.

The **potential energy change** describes the change in the potential energy with time. The potential energy change (*PEC*) of joint J_i at frame t is:

$$PEC(i,t) = PE(i,t+k) - PE(i,t). \quad (11)$$

The **total energy** describes the sum of the kinetic energy and the potential energy of the joint with unit mass. The total energy (*TE*) of joint J_i at frame t is:

$$TE(i,t) = KE(i,t) + PE(i,t). \quad (12)$$

The **total energy change** describes the change in the total energy with time. The total energy change (*TEC*) of joint J_i at frame t is:

$$TEC(i,t) = TE(i,t+k) - TE(i,t). \quad (13)$$

In addition, we use the **normalized positions feature** (*NP*) in HLPF [1] to add position information.

For each of the 15 joints, we extract the 2-dimensional normalized positions feature, 2-dimensional velocity feature, 2-dimensional acceleration feature, 2-dimensional angular velocity feature and 2-dimensional angular acceleration feature. For the remaining 8 features (speed feature, acceleration feature, kinetic energy feature, kinetic energy change feature, potential energy feature, potential energy change feature, total energy feature and total energy change feature), the dimensionality is one. Therefore, for each joint, the total dimensionally of the kinetic feature is $2 \times 5 + 8 = 18$. There are 15 joints in total. Therefore, the final dimensionality of the kinetic feature for each frame is $18 \times 15 = 270$.

4.2. Correlation relational features

For each pair of joints, we propose a set of correlation relational features to describe their kinetic correlation relations.

4.2.1. Motion correlation relational features

To describe the correlation of the motion between each pair of joints, we compute the cosine similarity of the motion feature pair. The positions of the pair of joints $\{J_i, J_j\}$ at frame t are (x_i, y_i) and (x_j, y_j), their velocities are V_i and V_j, their accelerations are A_i and A_j, their angular velocities are AV_i and AV_j, and their angular accelerations are AA_i and AA_j, respectively. The **velocity correlation relation** (*VCR*) is:

$$VCR(i,j) = \frac{V_i \cdot V_j}{\parallel V_i \parallel \cdot \parallel V_j \parallel}. \quad (14)$$

The **acceleration correlation relation** (*ACR*) is:

$$ACR(i,j) = \frac{A_i \cdot A_j}{\parallel A_i \parallel \cdot \parallel A_j \parallel}. \quad (15)$$

The **angular velocity correlation relation** (*AVCR*) is:

$$AVCR(i,j) = \frac{AV_i \cdot AV_j}{\parallel AV_i \parallel \cdot \parallel AV_j \parallel}. \quad (16)$$

The **angular acceleration correlation relation** (*AACR*) is:

$$AACR(i,j) = \frac{AA_i \cdot AA_j}{\parallel AA_i \parallel \cdot \parallel AA_j \parallel}. \quad (17)$$

4.2.2. Energy correlation relational feature

The **energy flow** is designed to describe the correlation of the energy between each pair of joints. The energy flow describes the energy transformation from one joint to another. The velocity of joint J_i with respect to J_j at frame t is:

$$V(i,j) = V_i - V_j. \quad (18)$$

The relative displacement of J_i to J_j per unit time is:

$$S(i,j) = 1 \cdot V(i,j) = V(i,j). \quad (19)$$

The relative acceleration of J_i to J_j is:

$$A(i,j) = A_i - A_j. \quad (20)$$

The net force of J_i to J_j per unit mass is:

$$F(i,j) = 1 \cdot A(i,j) = A(i,j). \quad (21)$$

Thus, the energy flow (*EF*) from J_i to J_j is:

$$EF(i,j) = F(i,j)S(i,j) = A(i,j)V(i,j). \quad (22)$$

For a human skeleton with 15 joints, there are $C_{15}^2 = 105$ joint pairs. For each joint pair, we can extract 5-dimensional features (i.e., velocity correlation relation feature, acceleration correlation relation feature, angular velocity correlation relation feature, angular acceleration correlation relation feature and energy flow feature.). Therefore, the total dimensionality of the correlation relational features for each frame is $105 \times 5 = 525$.

4.3. Distance relational features

For each pair of joints, we propose a set of distance relational features to describe position relations in space.

4.3.1. Horizontal and vertical distance relations

The **horizontal and vertical distance relations** describe the relative positions in the horizontal and vertical directions. They are obtained by computing the differences between the x and y coordinates for each pair of joints, as shown in Fig. 2 (h). In detail, suppose that the positions of the pair of joints $\{J_i, J_j\}$ at frame t are (x_i, y_i) and (x_j, y_j); then, the horizontal distance relation (*HDR*) from J_i to J_j is:

$$HDR(i,j) = x_j - x_i. \quad (23)$$

The vertical distance relation (*VDR*) from J_i to J_j is:

$$VDR(i,j) = y_j - y_i. \quad (24)$$

4.3.2. Orientation sine and cosine distance relations

The **orientation sine and cosine distance relations** describe the horizontal and vertical positions with respect to the principal direction. They are obtained by first computing the orientation of the vector connecting each pair of joints relative to the principal direction and then calculating the sine and cosine values of this orientation, as shown in Fig. 2 (d). The orientation from J_i to J_j is:

$$ORT(i,j) = \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right). \quad (25)$$

The orientation with respect to the principal direction is:

$$RORT(i,j) = ORT(i,j) - ORT(neck, belly). \quad (26)$$

The orientation sine distance relation (*OSR*) from J_i to J_j is:

$$OSR(i,j) = \sin(RORT(i,j)). \quad (27)$$

The orientation cosine distance relation (*OCR*) from J_i to J_j is:

$$OCR(i,j) = \cos(RORT(i,j)). \quad (28)$$

4.3.3. Eigenvector direction distance relation

To find the most discriminate direction to recognize actions, using the whole dataset, we apply PCA to the two-dimensional positions of each joint to reduce the dimensionality to one. This is also the projected position of the joint with respect to the direction with the max eigenvalue. Notice that this process is conducted separately for each joint. In other words, we repeat the PCA 15 times in total. Thus, we obtain new positions of each joint in the one-dimensional coordinate system. Then, distances are calculated between each pair of joints to form the **eigenvector direction distance relation**. In detail, suppose that the positions of the pair of joints $\{J_i, J_j\}$ at frame t are z_i and z_j in each eigenvector direction. Then, the eigenvector direction distance relation (*EVDR*) is:

$$EVDR(i, j) = z_j - z_i. \tag{29}$$

4.3.4. Link distance relation

Since the human skeleton is a tree structure, a path from one joint to another with no repetition exists and is unique. The reason behind this conclusion is that, treating the head as the root node and the others as the child nodes, a path from the ancestor joint to the descendant joint exists and is unique. By calculating the lowest common ancestor joint, we can obtain the **link distance relation** by summing the distances between each adjacent joint from the two descendant joints to the ancestor joint, as shown in Fig. 2(i). In detail, suppose that the lowest common ancestor of J_i and J_j is J_a and that the father of J_k is $J_{f(k)}$. Thus, the distance between J_i and J_j is:

$$DST(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{30}$$

The link distance relation (*LDR*) from J_i to J_j is:

$$LDR(i, j) = \sum_{k=i}^{a} DST(k, f(k)) + \sum_{k=j}^{a} DST(k, f(k)). \tag{31}$$

4.3.5. Distance relational trajectory features

To add temporal information, we use the method introduced in Section 3 to calculate the trajectories of the distance relational features. The parameters of the trajectory features are the same as those used for HLPF [1]. The trajectory length is 2, and the step size is 3. Furthermore, we add the letter 'T' to the name of a feature to represent the corresponding trajectory feature in later sections.

For a human skeleton with 15 joints, we can extract $C_{15}^2 = 105$-dimensional features for the horizontal distance relation feature, vertical distance relation feature, orientation sine distance relation feature, orientation cosine distance relation feature, eigenvector direction distance relation feature and link distance relation feature. Similarly, we can extract $C_{15}^2 = 210$-dimensional features for the horizontal distance relation trajectory feature, vertical distance relation trajectory feature, orientation sine distance relation trajectory feature, orientation cosine distance relation trajectory feature, eigenvector direction distance relation trajectory feature and link distance relation trajectory feature. Therefore, the total dimensionality of the distance relational features is $105 \times 6 + 210 \times 6 = 1890$.

4.4. Geometric relational features

For each triplet of joints, we propose a set of geometric relational features to describe the spatial position relations.

4.4.1. Joint vector inner product

From the perspective of the joint vector inner product space, we construct the **joint vector inner product** by computing the inner product of a pair of vectors, as shown in Fig. 2(e). At frame t, let (x_i, y_i), (x_j, y_j), and (x_k, y_k) denote the positions of the joints $\{J_i, J_j$, and $J_k\}$ in a triplet of joints, respectively. The joint vector inner product (*JVIP*) from J_i to J_j and J_k is calculated as:

$$JVIP(i, j, k) = \overrightarrow{J_iJ_j} \cdot \overrightarrow{J_iJ_k} = (x_j - x_i)(x_k - x_i) + (y_j - y_i)(y_k - y_i). \tag{32}$$

4.4.2. Joint vector cosine similarity

To describe the correlation of the joint vectors in each joint triplet, we calculate the cosine similarity to construct the **joint vector cosine similarity**, as shown in Fig. 2(e). The joint vector cosine similarity (*JVCS*) from J_i to J_j and J_k is calculated as:

$$JVCS(i, j, k) = \frac{JVIP(i, j, k)}{DST(i, j)DST(i, k)}. \tag{33}$$

4.4.3. Joint triangle area perimeter rate

We design the **joint triangle area perimeter rate** using the ratio of the area to the perimeter of the triangle spanned by the triplet, as shown in Fig. 2(e). Let $DST(i, j) = D_{i,j}$; then, the perimeter of triangle $J_iJ_jJ_k$ is:

$$PER(i, j, k) = D_{i,j} + D_{i,k} + D_{j,k}. \tag{34}$$

Let $C = \frac{PER(i,j,k)}{2}$; then, the area of the triangle $J_iJ_jJ_k$ is:

$$AR(i, j, k) = \sqrt{C(C - D_{i,j})(C - D_{i,k})(C - D_{j,k})} \tag{35}$$

The joint triangle area perimeter rate (*APR*) of the triplet $\{J_i, J_j, J_k\}$ is:

$$APR(i, j, k) = \frac{AR(i, j, k)}{PER(i, j, k)}. \tag{36}$$

4.4.4. Geometric relational trajectory features

To add temporal information, we use the method introduced in Section 3 to calculate the trajectories of the geometric relational features.

For a human skeleton with 15 joints, we can extract a $C_{15}^3 \times 3 = 1365$-dimensional feature for the joint vector inner product feature and joint vector cosine similarity feature. We can extract a $C_{15}^3 = 455$-dimensional feature for the joint triangle area perimeter rate feature and extract a $C_{15}^3 \times 3 \times 2 = 2730$-dimensional feature for the joint vector inner product trajectory feature and joint vector cosine similarity trajectory feature. We can extract $C_{15}^3 \times 2 = 910$-dimensional feature for the joint triangle area perimeter rate trajectory feature. The total dimensionality of the geometric relational features is $1365 \times 2 + 455 + 2730 \times 2 + 910 = 9555$.

4.5. Joints kinetic and relational features

We combine all of these features together to produce the JKRF. For a human skeleton with 15 joints, the total dimensionality of the JKRF is 12240, including 270-dimensional kinetic features, 525-dimensional correlation relational features, 1890-dimensional distance relational features and 9555-dimensional geometric relational features.

5. Experiments

In this section, we introduce the settings used in our experiments and show the experimental results of our novel features.

5.1. Experimental settings

We conduct experiments on the JHMDB [1], sub-JHMDB [1] and Penn Action datasets [10].

The **JHMDB** contains 21 human actions: *brush hair, catch, clap, climb stairs, golf, jump, kick ball, pick, pour, pull-up, push, run, shoot*

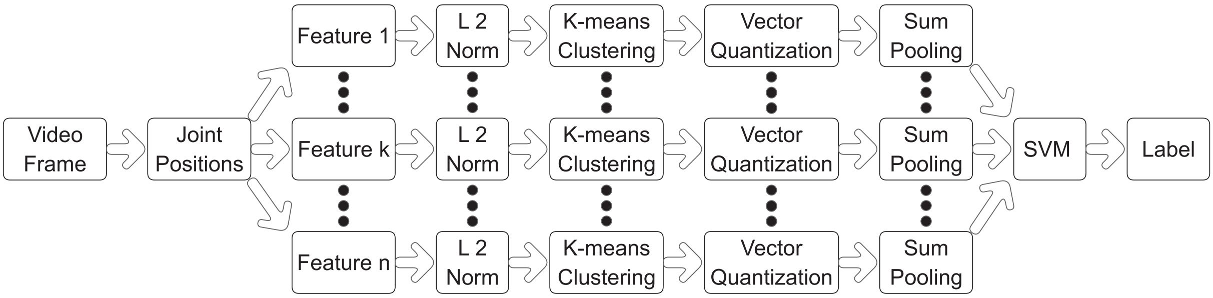


Fig. 3. Framework of bag of features using JKRF.

ball, shoot bow, shoot gun, sit, stand, swing baseball, throw, walk and *wave*. Video clips are restricted to the duration of the action. There are 36 – 55 clips per action class, with each clip containing 15 – 40 frames of size 320 × 240. 15 body joints and the scale of the person are annotated in each frame. Consequently, there are 928 clips with 31838 annotated frames in total. 15 joints, including *shoulders, elbows, wrists, hips, knees, ankles, neck, face* and *belly*, are all annotated manually, no matter whether the joints are inside the frame. There are three training and testing splits for the JHMDB and the sub-JHMDB, with 70% of clips for training and 30% for testing. The performance reported here is the average of these three splits.

The **sub-JHMDB** is a subset of the JHMDB and consists of 316 clips distributed over 12 actions: *catch, climb stairs, golf, jump, kick ball, pick, pull-up, push, run, shoot ball, swing baseball* and *walk*. The main difference between the sub-JHMDB and the JHMDB is that the human body is fully visible in the sub-JHMDB.

The **Penn Action** dataset contains 15 human actions: *baseball pitch, baseball swing, bench press, bowl, clean and jerk, golf swing, jump rope, jumping jacks, pull-up, push-up, sit-up, squat, strum guitar, tennis forehand* and *tennis serve*. There are 82 – 231 clips per action class, with each clip containing 18 – 663 frames of size 640 × 480. 13 body joints are annotated in each frame. As a result, there are 2326 clips, in which not all the body joints are visible, with 163841 annotated frames in total. Compared with the JHMDB, the scale of the person and the positions of the *neck* and the *belly* are not annotated. Since our algorithm needs such information, we use the midpoint of the *left shoulder* and the *right shoulder* as the position of the *neck* and use the midpoint of the *left hip* and the *right hip* as the position of the *belly*. In addition, we use $\frac{1}{100}Euclid\_distance(pos\_img(neck), pos\_img(belly))$ as the scale of the person. There is one training and testing split, with 50% of clips for training and 50% for testing, provided by [10] for the Penn Action dataset.

The performance is evaluated by computing the average accuracy over all classes for these three datasets.

After all the joints are extracted, we use the same method used in constructing the HLPF [1] to normalize the joint positions with respect to the scale of the person:

$$pos\_world\_x(joint) = \left(\frac{pos\_img\_x(joint)}{frame\_width} - 0.5\right) \times \frac{frame\_width}{frame\_height \times scale}. \quad (37)$$

$$pos\_world\_y(joint) = \left(\frac{pos\_img\_y(joint)}{frame\_height} - 0.5\right) \times \frac{1}{scale}. \quad (38)$$

Then, we use the normalized joint positions to extract features. L_2 normalization is used to normalize relational features because of their high dimensionality and large range of variation. A codebook is generated separately for each feature using *K*-means. We set the number of visual words per feature to $K = 20$ for the JHMDB and $K = 30$ for the sub-JHMDB and the Penn Action datasets. Frame features are assigned to their closest codeword to obtain the code coefficients; then, sum pooling operations are used to generate histograms to represent each video. For classification, we use the SVM with an RBF-χ^2 kernel [32], which is a multichannel classifier. It uses all kinds of histograms as inputs in parallel. In detail, for each feature *f*, a distance matrix D_f is computed. It contains the χ^2-distance between the histograms (h_i^f, h_j^f) of all video pairs (v_i, v_j). Using μ_f to denote the mean of the distance matrix D_f and *n* to denote the number of features, the kernel matrix is as follows:

$$K(v_i, v_j) = \exp\left(-\frac{1}{n}\sum_f \frac{D_f(h_i^f, h_j^f)}{\mu^f}\right). \quad (39)$$

The framework of our algorithm is shown in Fig. 3. Here, $n = 36$ because there are 36 kinds of sub-features in JKRF.

5.2. Performances of JKRF and HLPF for various codebook (K-means) sizes

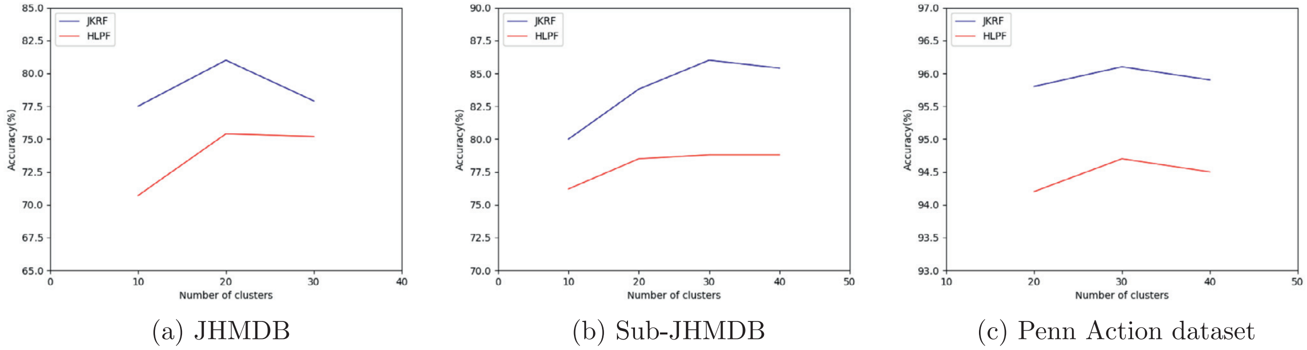
We test the performances of JKRF and HLPF for various codebook (K-means) sizes. As shown in Fig. 4, the variation trends of the performances of JKRF and HLPF are the same, and the JKRF perform better than the HLPF consistently for all codebook sizes on all datasets. In addition, the codebook size has little influence on the performance of JKRF. We choose the codebook size with the best performance to perform our experiments. Thus, we empirically set $K = 20$ for the JHMDB and $K = 30$ for the sub-JHMDB and the Penn Action dataset in the following experiments.

5.3. Comparison of the results of JKRF and HLPF for different classifiers

We compare the performances of JKRF and HLPF for different classifiers, including *K*-nearest neighbor, gradient boosting trees, softmax regression, random forest, SVM with a linear kernel, SVM with an RBF kernel, and SVM with an RBF-χ^2 kernel. For all classifiers except SVM with an RBF-χ^2 kernel, we concatenate histograms of different features into a long vector to be used as inputs. As shown in Fig. 5, for all classifiers on all datasets, except *K*-nearest neighbor on the Penn Action dataset, JKRF perform better than HLPF. For JKRF and HLPF on all datasets, the SVM with an RBF-χ^2 kernel is the best classifier. Therefore, we use the SVM with an RBF-χ^2 kernel as the classifier.

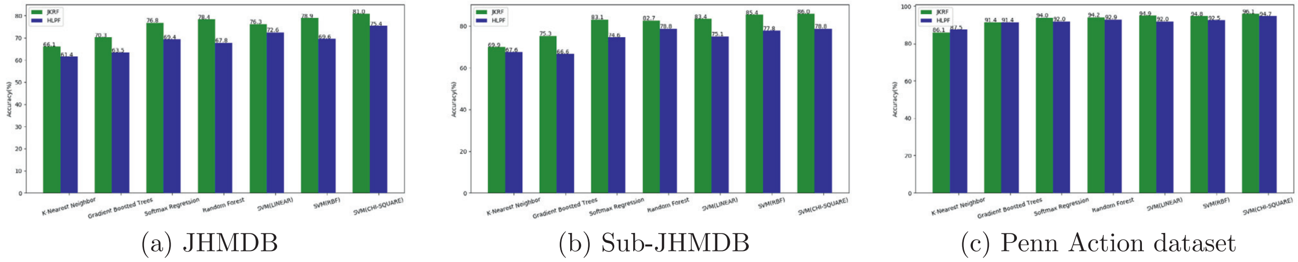
5.4. Performances of each JKRF feature

Fig. 6 compares the performances of the kinetic features (*KF*), correlation relational features (*CRF*), distance relational features (*DRF*) and geometric relational features (*GRF*) for both the single features and their combinations using the JHMDB, sub-JHMDB and Penn Action datasets.



(a) JHMDB (b) Sub-JHMDB (c) Penn Action dataset

Fig. 4. Performances of JKRF and HLPF for various codebook(K-means) sizes on the JHMDB, sub-JHMDB and Penn Action datasets.



(a) JHMDB (b) Sub-JHMDB (c) Penn Action dataset

Fig. 5. Comparison of the results of JKRF and HLPF for different classifiers on the JHMDB, sub-JHMDB and Penn Action datasets.

For the JHMDB and sub-JHMDB, we can observe that the original features consistently outperform the trajectory features with respect to the *DRF* and *GRF* by a significant margin ($HDR > HDRT$, $VDR > VDRT$, $OCR > OCRT$, $OSR > OSRT$, $EVDR > EVDRT$, $LDR > LDRT$, $JVIP > JVIPT$, $JVCS > JVCST$ and $APR > APRT$). However, for the Penn Action dataset, we obtain the opposite conclusion. This result occurs because the shot is movable in the JHMDB and sub-JHMDB. Thus, some errors are introduced in the construction of the trajectory features. The original features are more effective in this situation. For the Penn Action dataset, the shot is fixed. The trajectory features carry more discriminative information.

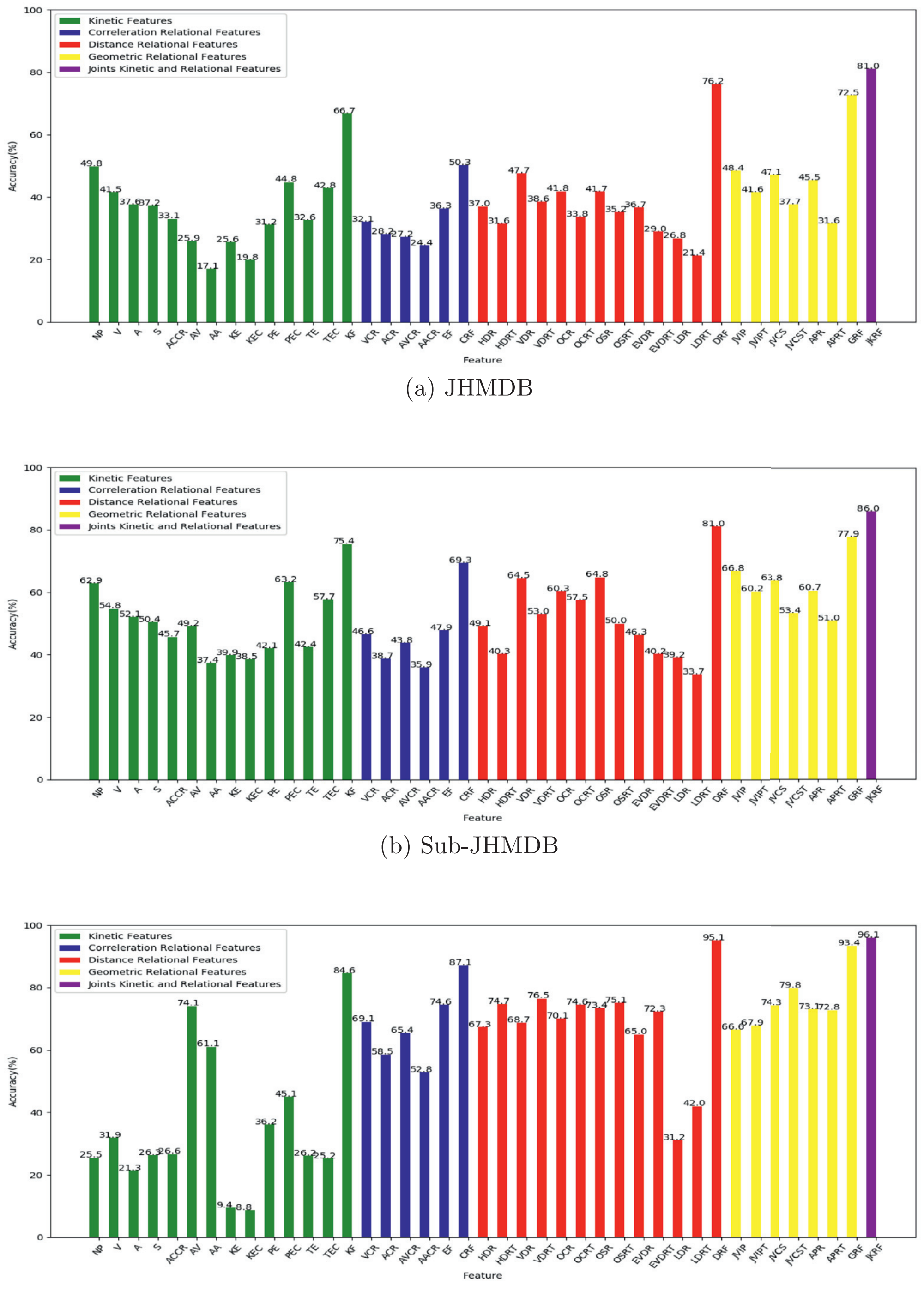
Moreover, since *V*, *A* and *AA* can be considered as the trajectories of *NP*, *V* and *AV* with a trajectory $\text{length} = 1$, respectively, and correlation relational features describe the correlation relations of kinetic features, we can draw the same conclusions for *KF* and *CRF*. For the JHMDB and sub-JHMDB, Fig. 6 shows that $NP > V$, $V > A$, $AV > AA$, $VCR > ACR$ and $AVCR > AACR$. For the Penn Action dataset, we can draw the same conclusions for *KE* and *CRF*, except that $V > NP$. In other words, the original features are more effective than features which describe trajectory information for *KF* and *CRF* on all datasets. This result shows that an unfixed shot influences only the performances of *DRF* and *GRF*. Regarding energy features, e.g., *KE*, *KEC*, *PE*, *PEC*, *TE* and *TEC*, the energy change features can also be considered as the trajectory features of the corresponding energy features. However, only the performances of *KE* and *KEC* result in the energy feature performing better. A possible explanation for this conclusion is that the main direction of the action of a human is along the vertical direction and that the vertical position change of the human body is more important. We can also see that the performances of *KE* and *KEC* for the Penn Action dataset are less than 10%. This result may occur because the information of the square of the first-order difference is not discriminative enough for the shot-fixed dataset.

The common conclusion we can draw from these results is that for all kinds of features, the combinational features, e.g., *KF*, *CRF*, *DRF* and *GRF*, have significantly improved performances compared to that of a single feature. In detail, the performance of *KF* is 10.5% – 16.9% better than that of the best single feature among the kinetic features on these datasets. The performance of *CRF* is 12.5% – 21.4% better than that of the best single feature among the correlation relational features. The performance of *DRF* is 16.2% – 28.5% better than that of the best single feature among the distance relational features. The performance of *GRF* is 11.1% – 24.1% better than that of the best single feature among the geometric relational features. In addition, the combination of all these features (JKRF) achieves the best performance. Therefore, we conclude that all the features should be used jointly.

5.5. Comparison to HLPF

Since all the features in HLPF [1] and JKRF are computed from a single joint, pairs of joints and triplets of joints, respectively, a comparison of each feature type make sense. In detail, for each single joint, we compare the combination of the normalized positions feature, the Cartesian trajectory feature and the radial trajectory feature in HLPF with the kinetic features in JKRF. For each pair of joints, we compare the combination of the distance relations feature, the orientation relations features and their trajectory features in HLPF with the combination of the correlation relational features and the distance relational features in JKRF. For each triplet of joints, we compare the combination of the angle relations feature and its trajectory feature in HLPF with the geometric relational features in JKRF. For HLPF, we directly use the publicly available code to compute features.

From Tables 1–3, we can conclude that for each method, except single joint features for the Penn Action dataset, our features consistently outperform the corresponding features in HLPF by a significant margin. This result occurs because we have mined the motion information more sufficiently and our features describe the relations of joints more comprehensively.



(a) JHMDB

(b) Sub-JHMDB

(c) Penn Action dataset

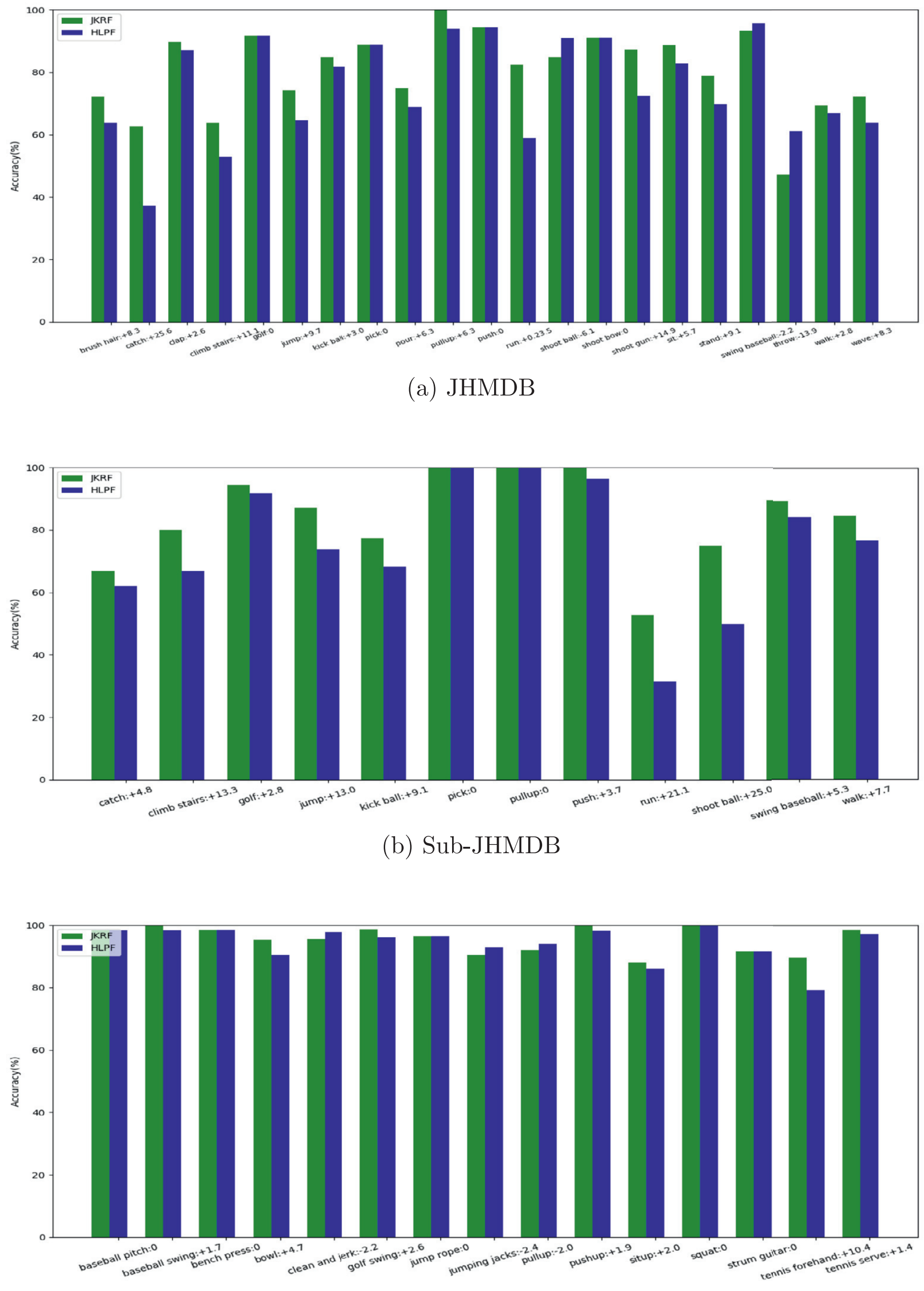
Fig. 6. Performance of each single feature and each combinational feature in JKRF for the JHMDB, sub-JHMDB, and Penn Action datasets.

Table 1
Classification accuracy (%) comparison of different features between HLPF and JKRF for the JHMDB.

Methods	HLPF	JKRF	Gain
Single joint features	61.8	66.7	+4.9
Pairwise joints features	67.8	77.0	+9.2
Triplet joints features	57.9	72.5	+14.6
Combinational features	75.4	**81.0**	+5.6

Table 2
Classification accuracy (%) comparison of different features between HLPF and JKRF for the sub-JHMDB.

Methods	HLPF	JKRF	Gain
Single joint features	70.0	75.3	+5.3
Pairwise joints features	74.1	82.3	+8.2
Triplet joints features	73.9	77.9	+6.0
Combinational features	78.8	**86.0**	+7.2



(a) JHMDB

(b) Sub-JHMDB

(c) Penn Action dataset

Fig. 7. Class accuracy of each action on the JHMDB, sub-JHMDB and Penn Action dataset for JKRF and HLPF. Numbers correspond to the accuracy difference between JKRF and HLPF (positive numbers indicate that JKRF perform better).

Table 3
Classification accuracy (%) comparison of different features between HLPF and JKRF for the Penn Action dataset.

Methods	HLPF	JKRF	Gain
Single joint features	89.1	84.6	-4.5
Pairwise joints features	90.5	96.1	+5.6
Triplet joints features	90.7	93.4	+2.7
Combinational features	94.7	**96.1**	+1.4

In addition, a quantitative comparison per class is presented in Fig. 7. It can be concluded that JKRF achieve large improvements over HLPF for actions that are difficult to distinguish, such as *run*, *walk*, *kick ball*, *jump* and *climb stairs*.

5.6. Comparison to state-of-the-art methods

HLPF [1], improved dense trajectory features [2] encoded using Fisher vectors [25] (iDT-FV), P-CNN [16], spatio-temporal features

Table 4
Comparison with state-of-the-art methods for the JHMDB, sub-JHMDB and Penn Action datasets.

Method	JHMDB	sub-JHMDB	Penn Action
STIP [3]	–	–	82.9
DT [8]	56.6	46.6	94.5
Seg DT [13]	58.6	–	95.0
iDT [2]-FV [25]	65.9	–	–
HLPF [1]	76.0	75.1	94.7
P-CNN [16]	74.6	72.5	–
JKRF (ours)	**81.0**	**86.0**	**96.1**

[3] (STIP), dense trajectory [8] (DT) and segmentation dense trajectory [13] (Seg DT) are state-of-the-art methods for action recognition. The comparison of these methods is shown in Table 4. For all methods, we use the results reported in their respective works. The figure demonstrates that JKRF improve upon the state-of-the-art methods by 6% for the JHMDB, 10.9% for the sub-JHMDB and 1.1% for the Penn Action dataset. These results reveal that the joints kinetic and relational features derived from normalized joints positions are the best features for action recognition.

6. Conclusions

In this paper, we propose a set of joints kinetic and relational features (JKRF). The kinetic features describe the motion information of each joint. The correlation relational features and distance relational features describe the kinetic correlations and the distance relations between pairs of joints. The geometric relational features describe the position relations among triplets of joints. JKRF perform significantly better than the state-of-the-art on three benchmark datasets.

Acknowledgments

This work is supported by the 973 project 2015CB351803, NSFC No. 61572451 and No. 61390514, Youth Innovation Promotion Association CAS CX2100060016, and Fok Ying Tung Education Foundation WF2100060004.

References

[1] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, M.J. Black, Towards understanding action recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3192–3199.

[2] H. Wang, C. Schmid, Action recognition with improved trajectories, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3551–3558.

[3] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, in: BMVC 2009-British Machine Vision Conference, BMVA Press, 2009. 124–1

[4] X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of visual words and fusion methods for action recognition: comprehensive study and good practice, Comput. Vision Image Understand. 150 (2016) 109–125.

[5] Y. Yang, D. Ramanan, Articulated pose estimation with flexible mixtures-of-parts, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 1385–1392.

[6] B. Xiaohan Nie, C. Xiong, S.-C. Zhu, Joint action recognition and pose estimation from video, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1293–1301.

[7] V.K. Singh, R. Nevatia, Action recognition in cluttered dynamic scenes using pose-specific part models, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 113–120.

[8] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Dense trajectories and motion boundary descriptors for action recognition, Int. J. Comput. Vision 103 (1) (2013) 60–79.

[9] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

[10] W. Zhang, M. Zhu, K.G. Derpanis, From actemes to action: a strongly-supervised representation for detailed action understanding, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2248–2255.

[11] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, IEEE Trans. Image Process. 24 (12) (2015) 5659–5670.

[12] C. Hong, J. Yu, D. Tao, M. Wang, Image based 3d human pose recovery by multi-view locality sensitive sparse retrieval, IEEE Trans. Industr. Electr. 62 (6) (2015) 3742–3751.

[13] J. Lu, J.J. Corso, et al., Human action segmentation with hierarchical supervoxel consistency, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3762–3771.

[14] A. Yao, J. Gall, L. Van Gool, Coupled action recognition and pose estimation from multiple views, Int. J. Comput. Vision 100 (1) (2012) 16–37.

[15] B. Xiaohan Nie, C. Xiong, S.-C. Zhu, Joint action recognition and pose estimation from video, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1293–1301.

[16] G. Chéron, I. Laptev, C. Schmid, P-cnn: Pose-based cnn features for action recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3218–3226.

[17] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: European conference on computer vision, Springer, 2010, pp. 143–156.

[18] I. Laptev, On space-time interest points, Int. J. Comput. Vision 64 (2–3) (2005) 107–123.

[19] P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on, IEEE, 2005, pp. 65–72.

[20] G. Willems, T. Tuytelaars, L. Van Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, in: European conference on computer vision, Springer, 2008, pp. 650–663.

[21] A. Klaser, M. Marszałek, C. Schmid, A spatio-temporal descriptor based on 3d–gradients, in: BMVC 2008-19th British Machine Vision Conference, British Machine Vision Association, 2008. 275–1

[22] G. Willems, T. Tuytelaars, L. Van Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, in: European conference on computer vision, Springer, 2008, pp. 650–663.

[23] Y. Luo, Y. Wen, D. Tao, J. Gui, C. Xu, Large margin multi-modal multi-task feature extraction for image classification, IEEE Trans. Image Process. 25 (1) (2016) 414–427.

[24] Y. Luo, D. Tao, K. Ramamohanarao, C. Xu, Y. Wen, Tensor canonical correlation analysis for multi-view dimension reduction, IEEE Trans. Knowl. Data Eng. 27 (11) (2015) 3111–3124.

[25] D. Oneata, J. Verbeek, C. Schmid, Action and event recognition with fisher vectors on a compact feature set, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1817–1824.

[26] X. Peng, C. Zou, Y. Qiao, Q. Peng, Action recognition with stacked fisher vectors, in: European Conference on Computer Vision, Springer, 2014, pp. 581–595.

[27] M. Tan, B. Wang, Z. Wu, J. Wang, G. Pan, Weakly supervised metric learning for traffic sign recognition in a lidar-equipped vehicle, IEEE Trans. Intell. Transp. Syst. 17 (5) (2016) 1415–1427.

[28] J. Yu, X. Yang, F. Gao, D. Tao, Deep multimodal distance metric learning using click constraints for image ranking, IEEE Trans. Cybern. (2016) 1–11.

[29] J. Yu, Y. Rui, D. Tao, Click prediction for web image reranking using multimodal sparse coding, IEEE Trans. Image Process. 23 (5) (2014) 2019–2032.

[30] J. Yu, Y. Rui, Y.Y. Tang, D. Tao, High-order distance-based multiview stochastic learning in image classification, IEEE Trans. Cybern. 44 (12) (2014) 2431–2442.

[31] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local svm approach, in: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, 3, IEEE, 2004, pp. 32–36.

[32] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: a comprehensive study, in: Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on, IEEE, 2006. 13–13

[33] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[34] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[36] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European Conference on Computer Vision, Springer, 2014, pp. 818–833.

[37] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in Neural Information Processing Systems, 2014, pp. 568–576.

[38] S. Ji, W. Xu, M. Yang, K. Yu, 3d convolutional neural networks for human action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 221–231.

[39] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.

[40] L. Wang, Y. Qiao, X. Tang, Action recognition with trajectory-pooled deep-convolutional descriptors, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4305–4314.