Structure-Preserving Binary Representations for RGB-D Action Recognition

Mengyang Yu, *Student Member, IEEE,* Li Liu and Ling Shao, *Senior Member, IEEE*

Abstract—In this paper, we propose a novel binary local representation for RGB-D video data fusion with a structure-preserving projection. Our contribution consists of two aspects. To acquire a general feature for the video data, we convert the problem to describing the gradient fields of RGB and depth information of video sequences. With the local fluxes of the gradient fields, which include the orientation and the magnitude of the neighborhood of each point, a new kind of continuous local descriptor called Local Flux Feature (LFF) is obtained. Then the LFFs from RGB and depth channels are fused into a Hamming space via the Structure Preserving Projection (SPP). Specifically, an orthogonal projection matrix is applied to preserve the pairwise structure with a shape constraint to avoid the collapse of data structure in the projected space. Furthermore, a bipartite graph structure of data is taken into consideration, which is regarded as a higher level connection between samples and classes than the pairwise structure of local features. The extensive experiments show not only the high efficiency of binary codes and the effectiveness of combining LFFs from RGB-D channels via SPP on various action recognition benchmarks of RGB-D data, but also the potential power of LFF for general action recognition.

Index Terms—RGB-D fusion, flux, binary, structure-preserving, dimensionality reduction, local feature

1 Introduction

RGB-D sensors such as Kinect receive increasing attention in the computer vision community [1]. They have been widely applied to many areas such as: human activity recognition [2], robot path planning [3], object detection [4], scene labeling [5], interactive gaming [6] and 3D mapping [7]. The combination of RGB and depth information enables enhanced capabilities of computer vision algorithms. It also provides an alternative way to learn features from video data for action recognition, especially through learning fused RGB-D representations.

To gain a more robust and accurate representation of samples, local feature descriptors such as: SIFT [8], HOG3D [9], HOG [10], HOF [11] and MBH [12] have been proposed and achieved notable success in classification and recognition. Based on these local features, the Bag-of-Words (BoW) model [13] and the Sparse Coding (SC) algorithm [14] have shown their effectiveness for both image classification and action recognition. During the last decade, extensive efforts have been put on the improvement of BoW and SC. However, in most situations, there are millions of local features with hundreds or even thousands of dimensions in vision-based tasks, which poses a severe restriction on the computational efficiency of similarity search in recognition algorithms. It is, therefore, highly desirable to find a compact and efficient but discriminative representation for local features.

The fast bitwise operations in Hamming space motivate us to propose a local binary representation for RGB-D video data. In this way, the similarity search is simply computing Hamming distances which are conducted by the XOR operation rather than computing Euclidean distances by the addition and multiplication in real numbers. Then the efficiency of classification and recognition algorithms will be significantly improved. Our proposed scheme is two-fold.

First, towards constructing a common representation applicable for both RGB and depth data, we view a video sequence in either RGB or depth as a scalar field in $\mathbb{R}^3$ with the frame coordinate (x, y) and the temporal axis t (for RGB data, we can use the three channels of red, green and blue to form three scalar fields in $\mathbb{R}^3$ separately. In the experiments, to alleviate the computational complexity, we only use the gray-scale information). To describe this scalar field, we compute the local flux of its gradient field and obtain a feature vector called Local Flux Feature (LFF) for each pixel. Generally speaking, the local flux $f_r(P)$ at point P is defined as the rate of the gradient field (flow) passing through a sphere surface with radius r centered at P. In other words, the local flux at point P captures the information of the orientation and the magnitude of the gradient field over a neighborhood of P, and $f_r(P)$, as a continuous function, represents an average quantity of the flow over this neighborhood. Many gradient-based features have been successfully applied to practical situations, since the gradient field represents the direction of the greatest change of a function. Theoretically, the Helmholtz theorem [15] in fluid mechanics states that we only

• *M. Yu, L. Liu and L. Shao are with the Department of Computer Science and Digital Technologies, Northumbria University, Newcastle upon Tyne, NE1 8ST, U.K.*
E-mail: m.y.yu@ieee.org, li2.liu@northumbria.ac.uk, ling.shao@ieee.org

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2491925, IEEE Transactions on Pattern Analysis and Machine Intelligence

need to know the divergence and curl of a twice continuously differentiable vector field to determine it. Given a C^2-smooth function $V(x, y, t) : \mathbb{R}^3 \to \mathbb{R}$, its gradient ∇V satisfies

$$\begin{aligned}\nabla \times \nabla V &= (\nabla_{ty} V - \nabla_{yt} V, \nabla_{xt} V - \nabla_{tx} V, \nabla_{yx} V - \nabla_{xy} V) \\ &= \mathbf{0},\end{aligned}$$

which means $\text{curl}(\nabla V) = \mathbf{0}$, showing that the divergence of ∇V provides the vital information for the gradient field. Fortunately, the divergence theorem converts computing the flux $f_r(P)$ through a closed sphere to computing the volume integral of the divergence inside the sphere. Obviously, computing $f_r(P)$ for every pixel is time-consuming and unnecessary. Thus we only calculate the local fluxes for the regions around the interest points or the points selected by dense sampling in RGB data and the corresponding pixels in depth data.

Second, we fuse the LFFs from RGB and depth channels of points into Hamming space. To make the above features more discriminative and meaningful in Hamming space, we propose a Structure Preserving Projection (SPP) method. Generally speaking, SPP preserves two levels of data structure. In terms of low-level features, we consider the relationship among local feature descriptors, i.e., their pairwise structure, which is maintained in the binary representation learning to embed high dimensional feature descriptors into a lower-dimensional structure-preserved Hamming space. In the learning phase, each pair of local features is given a weak label related to their Euclidean distance. Specifically, a *positive* pair is a pair of local features, if one feature of the pair is within the k nearest neighbors of the other; otherwise, it is a *negative* pair.

Considering the shape of the data distribution, the pairwise structure also includes the angles between each pair of local feature descriptors. Taking two negative pairs $(\mathbf{x}_1, \mathbf{x}_2)$ and $(\mathbf{x}_1, \mathbf{x}_3)$ as an example (since the majority of pairs are negative), they are encoded to the pairs which have large distances in the Hamming space. Nevertheless, an over-fitting condition is that pair $(\mathbf{x}_2, \mathbf{x}_3)$ is possibly mapped to the pair with a small distance as shown in Fig. 1. Therefore, preserving the angles can be regarded as a shape constraint for the structure of pairwise Euclidean distances. It ensures that the shape of data in the original space would not collapse in the Hamming space while pairwise distances are preserved.

Furthermore, in respect of high-level connection, we also want to establish links between samples and classes. The bipartite graph (a.k.a. bigraph) consisting of samples and classes, shows the relationship between samples and classes. To quantize the edges, we use the image-to-class (I2C) distance, which was first introduced in the naive Bayes nearest neighbor (NBN-

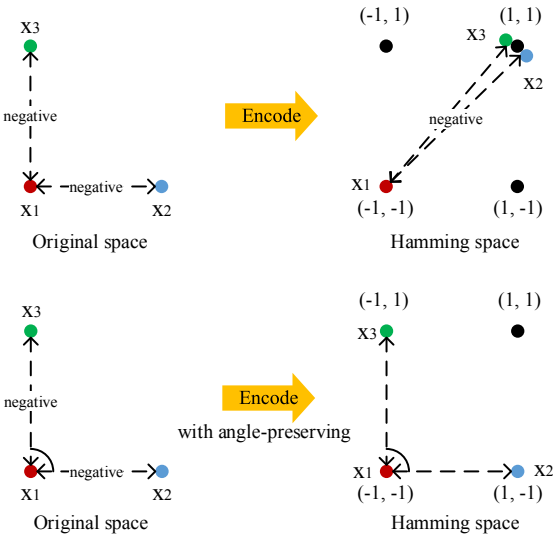


Fig. 1. Basic principle of the projection with angle-preserving in a two-dimensional example. The distances of two negative pairs $\|\mathbf{x}_1 - \mathbf{x}_2\|$ and $\|\mathbf{x}_1 - \mathbf{x}_3\|$ are expected to be maximized after the projection. The shape of $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ has collapsed in the Hamming space without angle-preserving, therefore, lost the discriminative ability.

N) classifier [16] and was also proven to be an optimal distance for classification in [16]. It represents the sum of all distances from the local features of an image to their corresponding nearest neighbors in each class. Although it was proposed for image classification, it can be applied to any kind of samples represented by local feature descriptors. I2C distances can effectively avoid the quantization error in the bag-of-features model. Our algorithm shows that the performance can be enhanced by combining the sample-to-class structure (bigraph regularization) and the pairwise geometrical structure. It is worthwhile to highlight several properties of the proposed scheme:

- LFF is a continuous feature descriptor without loss of orientations and magnitudes of the gradient field, which makes it more suitable for the discretization of the final binary representation since every discretization will bring the deviation into results.
- SPP simultaneously preserves two independent aspects of geometrical structure: Euclidean distances and angles, which could balance each other and avoid over-fitting.
- SPP considers two levels of the relationship of data structure based on local feature descriptors. Preserving the local structure and the global structure in the original feature space makes local feature descriptors more discriminative in the lower-dimensional space.
- Our scheme fuses RGB and depth information. The fused local feature descriptors have learned

the complementary nature of RGB and depth information.
- Our representation is linear and binary. This makes it extremely fast and useful for many practical applications.

2 Related Work

Feature extraction from RGB video data has been well explored [17], [18], [19], [20]. Detectors such as Spatio-Temporal Interest Points (STIP) [21] and Dollar's [22] are usually used to locate interest points before feature extraction. Many video descriptors are extended from their counterparts in the image domain [8], [9], [23], [12], [24]. As 3D versions of SURF [25], SIFT [8] and HOF [11], 3D speeded up robust features (SURF3D) [26], 3D scale invariant feature transforms (3D-SIFT) [27] and 3D motion features [28], [29] have been proposed for action recognition respectively. The Histogram of Oriented Gradients (HOG) is widely used in the above schemes, which discretizes the gradient orientations. In our work, however, discretization only performs in the pixel computation. Fathi et al. [30] developed a method to extract mid-level motion features by using the low-level optical flow for action recognition. Recently, the dense trajectories [31] gained high accuracies in most action recognition datasets. However, this method suffers from extremely high computational complexity. More feature extraction methods for action recognition could be found in a survey provided by Poppe [32].

Compared to the conventional RGB cameras, the depth cameras are relatively new. The existing features are specifically extracted for the depth information, since characteristics such as color and texture on depth data are far less than on the RGB data. Motion History Image (MHI) [33] is a typical template matching method for the analysis of depth information and the applications of human motion recognition [34]. Using the depth information only, Shotton et al. [35] proposed a method for human body joints analysis which is the core component of the Kinect gaming system. Nevertheless, more feature extraction methods are for the fusion with RGB information. Based on HOG, Spinello and Arras [4] proposed a method called Histogram of Oriented Depths (HOD) for depth description and probabilistically combined HOD and HOG into a Combo-HOD to detect people in urban environments. Methods in [36] and [37] simply optimize all available information in their algorithms for object detection and recognition respectively. Similarly, Ni et al. [38] designed two color-depth fusion schemes for human activity recognition. Using the depth and skeleton information of actions, Wang et al. [2] proposed a new feature called Local Occupancy Pattern (LOP) and an actionlet ensemble model which indicates a structure of features. Recently, the HON4D descriptor [39] was proposed to build the histogram of the normal unit vectors from the depth channel for activity recognition.

Apart from feature extraction, there are also many approaches to analyze actions with a temporal model. A typical one is dynamic time warping (DTW) [40], which was proposed for speech processing first. Due to the time-sequential property, DTW was also widely used as a measurement method in human action recognition for both depth data [41] and body joints of skeletons [42].

The above works are specifically designed for either RGB or depth data. In our work, LFF is a general descriptor which is suitable for both RGB and depth data. Besides, by calculating the local flux of the continuous gradient vector field, there are no bins and histograms in the computation of LFF, which can avoid the quantization error in most histogram-based methods. The Gradient Vector Flow (GVF) [43] has been successfully used in active contour alignments by solving the PDEs for an energy minimization problem. Engel et al. [44] calculated the flux flow on the GVF and adopted it for pedestrian detection. Based on the 3D vector field, a rotation invariant descriptor called 3D-Div [45] was proposed for 3D object recognition by computing the divergence of the vector field. Nonetheless, the point-wise divergence in [45] cannot capture the neighborhood information of each point. In our work, we focus on the discriminative ability of the local flux and its advantage in RGB-D action recognition.

Preserving the intrinsic manifold/subspace structure is also involved in our algorithm to seek a more discriminative representation of local features. Manifold learning methods such as ISOMAP [46], Laplacian Eigenmap (LE) [47] and Locally Linear Embedding (LLE) [48], were designed to preserve the manifold structure of data in the original space. A unified review and other manifold learning algorithms can be seen in [49]. Normally, linear methods possess high efficiency. Locality Preserving Projection (LPP) [50] is the first linear projection preserving algorithm that preserves the high-dimensional local structure. Neighborhood Preserving Embedding (NPE) [51] also tries to preserve the local representation of data. Capturing the intrinsic geometrical structure of data, Sparse Concept Coding (SCC) [52], which is a matrix factorization method, provides a sparse representation of the image space. For pairwise structure preserving, a related work for fast vision applications [53] represents each image using a binary vector calculated via boosted coding. In contrast, few works have attempted angle preserving in dimensionality reduction. Caseiro et al. [54] applied rolling map to the classification problem. Although the angles measured by geodesics in the original manifold are equal to the ones in the mapped manifold, the algorithm is not linear.

However, these works mainly focused on the global representations rather than preserving both pairwise

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2491925, IEEE Transactions on Pattern Analysis and Machine Intelligence

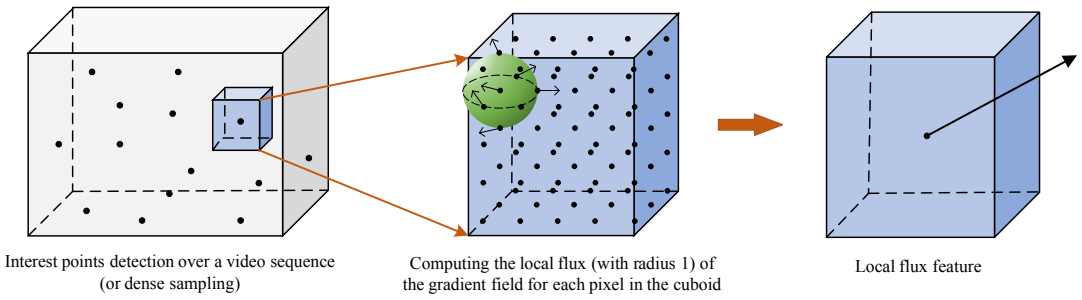


Fig. 2. Illustration of the computation of local fluxes in the gradient field. The output LFF is regarded as a foundation for learning binary codes.

structure of local feature descriptors and bipartite graph structure between samples and classes in the original space for designing efficient binary codes in Hamming space.

In the aspect of hash/binary code learning [55], one classical method is Locality-Sensitive Hashing (LSH) [56]. Another popular technique called Spectral Hashing (SpH) [57] was also proposed to preserve the locality information of data. Recently, a supervised method called Kernel-Based Supervised Hashing (KSH) [58] has shown good discriminative ability of binary codes and outperformed other supervised methods such as Linear Discriminant Analysis Hashing (LDAH) [59], Binary Reconstructive Embeddings (BRE) [60] and Minimal Loss Hashing (MLH) [61]. The above works mainly focus on preserving the pairwise distance, which is one part of SPP. To avoid overfitting as shown in Fig. 1, SPP also takes the pairwise angle into account. Towards local descriptors, Hamming Embedding (HE) [62] was proposed to map real-valued local features to binary codes. SPP contains a sample-to-class relationship [63] when each sample is represented by a set of local descriptors, since most visual tasks are sample-oriented. Experimental results show that these three terms, i.e., the pairwise distance, the pairwise angle and the sample-to-class relationship, all contribute to the outstanding performance of the proposed method.

3 Local Flux Feature

Local features extracted from local regions in an image or a video sequence are used to describe the local structure of a sample. Usually, local regions are the neighborhoods of points which are determined by using an interest point detector or by dense sampling of the image plane or video volume. And then, a feature vector is computed for each local region by characterizing its properties. In our algorithm, we compute the new Local Flux Features (LFFs) from the RGB-D video data and then combine the local feature $\mathbf{x}_{RGB}$ from RGB information with the local feature $\mathbf{x}_{Depth}$ from depth information to obtain a concatenated feature vector $X \in \mathbb{R}^D$.

3.1 Flux Computation

The concept of flux has been studied deeply in applied physics, especially in fluid mechanics and electromagnetic theory. The flux of a vector field over a simply-connected closed district (a sphere in this paper) is defined as the quantity of this vector field passing through the district. This quantity includes the information of the orientation and the magnitude of the vector field over the district. It is used for a description of the vector field. To describe a video sequence which is regarded as a scalar field, we consider its gradient field and compute the local flux of the gradient field.

Given a video sequence $V(x, y, t)$ in either RGB[1] or depth, it can be seen as a function $V : \mathbb{R}^3 \to \mathbb{R}$. We assume V is a C^2-smooth function, i.e., $V \in C^2(\Omega)$, where Ω is the district of the video sequence, usually an $L \times W \times H$ cuboid. In fact, in discrete condition, derivative computation can be regarded as an approximation by a convolution operation of matrices. Then for scalar field $V(x, y, t)$, we consider its gradient field $\nabla V(x, y, t) = (\nabla_x V, \nabla_y V, \nabla_t V)$. To describe the gradient field ∇V, we assign an $l \times w \times h$ cuboid centered at each candidate point (interest points or dense samples) and compute the local flux of every pixel (or lattice point if we regard the coordinates of a pixel as integers) in the cuboid. To be specific, denote $B_P(r) = \{(x', y', t') | (x'-x)^2 + (y'-y)^2 + (t'-t)^2 \leq r^2\}$ as the sphere with the center $P = (x, y, t)$ and radius r, the local flux at the point P over the sphere $\partial B_P(r)$ is calculated as

$$f_r(P) = \oint_{\partial B_P(r)} \nabla V \cdot \mathrm{d}\mathbf{S}, \tag{1}$$

where $\mathrm{d}\mathbf{S}$ represents the directed area unit of the boundary surface $\partial B_P(r)$. However, computing on

1. In fact, we only need the gray-scale information in our algorithm.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2491925, IEEE Transactions on Pattern Analysis and Machine Intelligence

the lattice points on the boundary $\partial B_P(r)$ is difficult and inaccurate. According to the divergence theorem, we have

$$\oint_{\partial B_P(r)} \nabla V \cdot \mathrm{d}\mathbf{S} = \int_{B_P(r)} \nabla \cdot \nabla V \ \mathrm{d}B_P(r), \quad (2)$$

i.e., we only need to compute for the points inside the sphere $B_P(r)$. Note that in the light of the Helmholtz theorem [15] in fluid mechanics, we only need to know the divergence and the curl of a twice continuously differentiable vector field to determine it. Hence, the fact that $curl(\nabla V) = \nabla \times \nabla V = \mathbf{0}$ implies that the divergence of ∇V provides the vital information, which is captured by the local flux $f_r(P)$. For realistic computation, we adopt the numerical approximation for the discrete condition of pixels:

$$f_r(P) = \int_{B_P(r)} \Delta V \ \mathrm{d}B_P(r) \approx \sum_{Q \in B_P(r) \cap \mathbb{Z}^3} \Delta V(Q), \quad (3)$$

where Δ is the Laplace operator. Suppose there are $D/2$ pixels in an $l \times w \times h$ cuboid, then we compute $D/2$ local fluxes in a specific order[2] and obtain an LFF vector $\mathbf{x} = (x_1, \cdots, x_{D/2}) \in \mathbb{R}^{D/2}$. Fig. 2 illustrates the outline of the computation of local fluxes. Having computed the LFF $\mathbf{x}_{RGB}$ from the RGB channel and $\mathbf{x}_{Depth}$ in the corresponding point from the depth channel, we concatenate their normalizations and obtain the new feature

$$X = \left[\frac{\mathbf{x}_{RGB}}{\|\mathbf{x}_{RGB}\|}, \frac{\mathbf{x}_{Depth}}{\|\mathbf{x}_{Depth}\|} \right]^T \in \mathbb{R}^D. \quad (4)$$

The combined LFF is regarded as the basic feature for the later learning of binary codes in our algorithm.

4 Structure Preserving Projection

In this section, we introduce our Structure Preserving Projection (SPP) algorithm. SPP simultaneously preserves the local structure and the integrated shape of local features. In addition, SPP also considers a higher level relationship among local features, i.e., the bipartite graph consisting of samples and classes. SPP aims to seek a specific matrix $\Theta \in \mathbb{R}^{D \times d}$ $(d < D)$ to construct a binary function

$$H(X) = \mathrm{sgn}(\Theta^T X), \quad (5)$$

such that their discriminative ability for action recognition is improved. For computational convenience, we choose $\{-1, +1\}$ rather than $\{0, 1\}$ to represent binary codes in our algorithm.

2. In the experiments, we obtain the LFF by listing the corresponding local flux values in the following pixel order: $(1,1,1), \cdots, (l,1,1), (1,2,1), \cdots, (l,2,1), \cdots, (l,w,1), \cdots, (l,w,h)$. In fact, the order has no effect on the final recognition results. The only requirement is the consistency of order in a vision task.

4.1 Pairwise Structure Preserving

We denote the set composed of all local features by $\mathcal{F} = \{X_1, \cdots, X_N\}$, where N is the number of local features in training data. As mentioned above, we aim to seek the binary representations with discriminative ability in the lower-dimensional space. We are concerned about the relationship between every two local features in the high-dimensional space, which should also be retained in the lower-dimensional space.

4.1.1 Pairwise Label

First, we assign a weak label for each pair of local features. With the pairwise labels, acquiring the class information of each local feature is unnecessary. Besides, similar local features with small Euclidean distances may appear in samples from many different classes. Motivated by the binary property of $H(X)$, we employ the pairwise label $\{-1, +1\}$ to represent the relationship between two local features based on the pairwise distance between them. Thus we have the pairwise label

$$\ell_{ij} = \begin{cases} +1, & X_i \in \mathrm{N}_k(X_j) \text{ or } X_j \in \mathrm{N}_k(X_i) \\ -1, & \text{otherwise} \end{cases},$$

where $\mathrm{N}_k(X)$ is the set of k nearest neighbors of X. To maintain the local structure, we make the product of each component in $H(X_i)$ and $H(X_j)$ consistent with their pairwise label ℓ_{ij}, i.e., $H(X_i)_m \cdot H(X_j)_m = \ell_{ij}$, $\forall m$. We denote $\mathcal{P} = \{(i,j) | X_i, X_j \in \mathcal{F}\}$. Therefore, we need to minimize the following function

$$\begin{aligned} &\sum_{(i,j)\in\mathcal{P}} \sum_{m=1}^{D} (\ell_{ij} - H(X_i)_m H(X_j)_m)^2 \\ &= \sum_{(i,j)\in\mathcal{P}} \sum_{m=1}^{D} (2 - 2\ell_{ij} H(X_i)_m H(X_j)_m) \\ &= \sum_{(i,j)\in\mathcal{P}} \left(2D - 2\ell_{ij} \sum_{m=1}^{D} H(X_i)_m H(X_j)_m \right) \\ &= \sum_{(i,j)\in\mathcal{P}} 2D - 2\ell_{ij} \langle H(X_i), H(X_j) \rangle. \end{aligned} \quad (6)$$

Then equivalently, we only need to maximize

$$\sum_{(i,j)\in\mathcal{P}} \ell_{ij} \langle H(X_i), H(X_j) \rangle. \quad (7)$$

The above function reaches its maximum value when $\ell_{ij}\mathrm{sgn}(\Theta^T X_i)$ and $\mathrm{sgn}(\Theta^T X_j)$ are similarly sorted due to the rearrangement inequality [64]. In other words, if $\ell_{ij} = 1$, X_i and X_j are then similarly encoded and vice versa.

Considering the effect of noise, we additionally assign a pairwise weight W_{ij}^P to the local feature pair (i,j) to avoid the disturbance:

$$W_{ij}^P = \exp\left(-l_{ij} \|X_i - X_j\|^2\right). \quad (8)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2491925, IEEE Transactions on Pattern Analysis and Machine Intelligence

Then the objective function for pairwise labels becomes

$$\sum_{(i,j)\in\mathcal{P}} W_{ij}^{P}\ell_{ij}\langle H(X_i), H(X_j)\rangle. \tag{9}$$

4.1.2 Pairwise Angle

In addition to the distance factor, we are also concerned about the shape of the entire set of local features, which is regarded as a constraint for preserving the pairwise Euclidean distances. The shape constraint firms the data structure in the projected space and avoids some certain errors caused by the pairwise labels. We denote the angle between two local features X_i and X_j by θ_{ij}. Note that angle θ_{ij} is with the vertex at coordinate origin. Thus, the local features should be *centralized* before the further learning process. Orthogonal transformation ($d = D$ and $\Theta^T\Theta = \Theta\Theta^T = I$) preserves the lengths of local features and the angles between them since we have $\langle\Theta^T X_i, \Theta^T X_j\rangle = X_i^T\Theta\Theta^T X_j = X_i^T X_j = \langle X_i, X_j\rangle$, $\forall i, j$. When $d < D$, however, this property does not hold in orthogonal projection. We hope the angle $\widehat{\theta}_{ij}$ in the projected space[3] is (approximately) equal to θ_{ij}. Note that the distances are irrelevant with the angles, i.e., the pair of local features with a long distance can have a small angle and the pair with a short distance may have a large angle. Thus it is desirable to retain the angles of all pairs. We define our optimization problem for angle preserving in the low dimensional space:

$$\arg\max_{\Theta} \sum_{(i,j)\in\mathcal{P}} \langle X_i, X_j\rangle \cdot \langle\Theta^T X_i, \Theta^T X_j\rangle. \tag{10}$$

Although it is the optimization for preserving the inner product, the following proposition shows that the optimal Θ^* preserves the pairwise angles.

Proposition 1: Suppose Θ^* is the optimal solution of the optimization problem (10), then for any $1 \le i, j \le N$, the projection Θ^* preserves the angle between the local features X_i and X_j.

Proof: According to the Cauchy-Schwarz inequality, we have

$$\begin{aligned}&\sum_{(i,j)\in\mathcal{P}} \langle X_i, X_j\rangle \cdot \langle\Theta^T X_i, \Theta^T X_j\rangle \\ &\le \left(\sum_{(i,j)\in\mathcal{P}} \langle X_i, X_j\rangle^2\right)^{\frac{1}{2}} \left(\sum_{(i,j)\in\mathcal{P}} \langle\Theta^T X_i, \Theta^T X_j\rangle^2\right)^{\frac{1}{2}},\end{aligned}$$

and the equality holds if and only if $\langle X_i, X_j\rangle$ ($(i,j) \in \mathcal{P}$) and $\langle\Theta^T X_i, \Theta^T X_j\rangle$ ($(i,j) \in \mathcal{P}$) are collinear. We can first set a norm constraint $\sum_{(i,j)\in\mathcal{P}}\langle\Theta^T X_i, \Theta^T X_j\rangle^2 = 1$ for Θ. Then the objective function in Eq. (10) is smaller than a constant. If Θ^* is the optimal solution of the optimization problem (10), the left-hand-side of the above inequality reaches its maximum value at Θ^*. Then there exists a constant $\lambda \in \mathbb{R}$ such that

$$\frac{\langle(\Theta^*)^T X_i, (\Theta^*)^T X_j\rangle}{\langle X_i, X_j\rangle} = \lambda, \ \forall (i,j) \in \mathcal{P}.$$

Since for $i = j$, we have $\|(\Theta^*)^T X_i\| = \lambda\|X_i\|$, then $\lambda > 0$. Therefore, for the projected angle $\widehat{\theta}_{ij}$, it satisfies

$$\begin{aligned}\cos\widehat{\theta}_{ij} &= \frac{\langle(\Theta^*)^T X_i, (\Theta^*)^T X_j\rangle}{\|(\Theta^*)^T X_i\|\|(\Theta^*)^T X_j\|} \\ &= \frac{\langle(\Theta^*)^T X_i, (\Theta^*)^T X_j\rangle}{\sqrt{\langle(\Theta^*)^T X_i, (\Theta^*)^T X_i\rangle}\sqrt{\langle(\Theta^*)^T X_j, (\Theta^*)^T X_j\rangle}} \\ &= \frac{\lambda\langle X_i, X_j\rangle}{\sqrt{\lambda\langle X_i, X_i\rangle}\sqrt{\lambda\langle X_j, X_j\rangle}} \\ &= \frac{\langle X_i, X_j\rangle}{\sqrt{\langle X_i, X_i\rangle}\sqrt{\langle X_j, X_j\rangle}} = \frac{\langle X_i, X_j\rangle}{\|X_i\|\|X_j\|} \\ &= \cos\theta_{ij},\end{aligned}$$

which implies that the projection matrix Θ^* is an angle-preserving projection. □

4.2 Bigraph Regularization

Not only the pairwise structure of local features, but also the connection between samples and classes, which is regarded as a higher level relationship among local features, is considered in our algorithm. We use the image-to-class (I2C) distance to measure the bipartite graph (a.k.a. bigraph) that consists of video samples and classes. Although the I2C distance was first introduced to measure the distances between images and classes, it can also be applied to all kinds of samples that are represented by local features. Our goal is to preserve the I2C distances in the lower-dimensional space. Given the set of local features of a sample $\mathcal{X}_i = \{X_{i1}, \cdots, X_{im_i}\}$, which contains all local features of sample i, the distance between sample i and class c is defined as

$$I_{\mathcal{X}_i}^{c} = \sum_{X\in\mathcal{X}_i} \|X - \mathrm{NN}^c(X)\|^2, \tag{11}$$

where $\mathrm{NN}^c(X)$ is the nearest neighbor (NN) of the local feature X in class c and $\|\cdot\|$ is the L_2-norm.

However, the complexity of NN-search linearly depends on the number of local features, which renders the nearest neighbor search in such a large-scale space of local features of each class will still cost much time. Hence, we first implement a K-means clustering algorithm for each class. In other words, we first find K centroids for each set $\bigcup_{C(\mathcal{X}_i)=c}\mathcal{X}_i$, $c = 1, \cdots, C$, where C is the number of classes and $C(\cdot) \in \{1, \cdots, C\}$ is the label information function that represents the class label of the input. In this way, the searching range of nearest neighbors is reduced to the set of cluster centers, which has a much smaller size than the original space, i.e., for $c = 1, \cdots, C$, we set

$$\mathrm{NN}^c(X) \in \text{Centroids } \{S_1, \cdots, S_K\} \text{ of } \bigcup_{C(\mathcal{X}_i)=c} \mathcal{X}_i.$$

3. Since Hamming space is a discrete space, we first consider the angles in the linear subspace before taking the sign function.

Having obtained I2C distances, we build a bigraph $G = (V_1, V_2, E)$, where V_1 and V_2 are the node sets of samples and classes respectively. G is a complete and weighted bigraph. For each edge in E connecting sample i and class c, it has the weight W_{ic}^D determined by the I2C distance, named the I2C similarity. By heat kernel, we define the I2C similarity as follows:

$$W_{ic}^{I2C} = \exp\left(-(I_{\mathcal{X}_i}^c)^2/\sigma\right),\ i = 1, \cdots, n,\ c = 1, \cdots, C, \tag{12}$$

where σ is the Gaussian smoothing parameter and n is the number of training samples. Correspondingly, we have the I2C distance in the objective Hamming space:

$$\widehat{I}_{\mathcal{X}_i}^c = \sum_{X \in \mathcal{X}_i} \|\mathrm{H}(X) - \mathrm{NN}^c(\mathrm{H}(X))\|^2. \tag{13}$$

With the above defined I2C similarity W_{ic}^{I2C} and the projected I2C distance $\widehat{I}_{\mathcal{X}_i}^c$, we can define the following optimization problem to quantize the bigraph regularization, i.e., I2C structure in the low dimensional space:

$$\arg\min_{\Theta} \sum_{i=1}^{n} \sum_{c=1}^{C} \widehat{I}_{\mathcal{X}_i}^c \cdot W_{ic}^{I2C}. \tag{14}$$

By minimizing the above equation, the sample which has a small I2C distance to class c in the high dimensional space is still close to class c in the low dimensional space. According to the rearrangement inequality [64], the above objective function reaches its minimum value if and only if $\{\widehat{I}_{\mathcal{X}_i}^c\}$ and $\{I_{\mathcal{X}_i}^c\}$ are similarly sorted, which means the projected I2C distances preserve the bigraph structure in the high dimensional space.

4.3 Objective Function and Optimization

In addition, to make the projected space more compact, we set the orthogonality constraint on the projection matrix, i.e., $\Theta^T\Theta = I$. Combining the objective functions for the pairwise structure and the bigraph regularizer, we obtain our final optimization problem for SPP:

$$\begin{aligned}\arg\max_{\Theta^T\Theta=I} & \sum_{(i,j)\in\mathcal{P}} W_{ij}^P \ell_{ij} \langle H(X_i), H(X_j)\rangle \\ & + \sum_{(i,j)\in\mathcal{P}} \langle X_i, X_j\rangle \cdot \langle \Theta^T X_i, \Theta^T X_j\rangle \\ & - \beta \sum_{i=1}^{n}\sum_{c=1}^{C} \widehat{I}_{\mathcal{X}_i}^c \cdot W_{ic}^{I2C},\end{aligned} \tag{15}$$

where β is the regularization parameter.

Optimization: Considering the discreteness of the binary function, we first use approximation $\mathrm{sgn}(x) \approx x$ to relax the objective function in the optimization problem (15) into a real-valued space. Then the objective function of the pairwise label part (see Eq. (9)) becomes

$$\begin{aligned}& \sum_{(i,j)\in\mathcal{P}} W_{ij}^P \ell_{ij} \langle H(X_i), H(X_j)\rangle \\ &= \sum_{(i,j)\in\mathcal{P}} W_{ij}^P \ell_{ij} \langle \mathrm{sgn}(\Theta^T X_i), \mathrm{sgn}(\Theta^T X_j)\rangle \\ &\approx \sum_{(i,j)\in\mathcal{P}} W_{ij}^P \ell_{ij} \langle \Theta^T X_i, \Theta^T X_j\rangle \\ &= \sum_{(i,j)\in\mathcal{P}} W_{ij}^P \ell_{ij}\, \mathrm{tr}(\Theta^T X_i (\Theta^T X_j)^T) \\ &= \sum_{(i,j)\in\mathcal{P}} W_{ij}^P \ell_{ij}\, \mathrm{tr}(\Theta^T X_i X_j^T \Theta).\end{aligned}$$

And for I2C distances, we denote $\mathrm{NN}^c(X) = X^c$. Note that after applying projection matrix Θ, the nearest neighbors may change. However, for the large-scale local feature space, we approximately adopt the sum of the distances from $\Theta^T X$ to the projected nearest neighbor $\Theta^T X^c$. Then the projected I2C distance (see Eq. (13)) after applying matrix Θ becomes

$$\begin{aligned}\widehat{I}_{\mathcal{X}_i}^c &\approx \sum_{X\in\mathcal{X}_i} \|\Theta^T X - \Theta^T X^c\|^2 \\ &= \sum_{X\in\mathcal{X}_i} \|\Theta^T (X - X^c)\|^2 \\ &= \sum_{k=1}^{m_i} \mathrm{tr}(\Theta^T (X_{ik} - X_{ik}^c)(\Theta^T (X_{ik} - X_{ik}^c))^T) \\ &= \sum_{k=1}^{m_i} \mathrm{tr}(\Theta^T (X_{ik} - X_{ik}^c)(X_{ik} - X_{ik}^c)^T \Theta) \\ &=: \sum_{k=1}^{m_i} \mathrm{tr}(\Theta^T \Delta X_{ik}^c (\Delta X_{ik}^c)^T \Theta),\end{aligned}$$

where $\Delta X_{ik}^c = X_{ik} - X_{ik}^c$, $k = 1, \cdots, m_i$. Thus, by simple algebraic derivation, the optimization problem (15) is reduced to

$$\arg\max_{\Theta^T\Theta=I} \mathrm{tr}(\Theta^T M \Theta), \tag{16}$$

where

$$\begin{aligned}M = & \sum_{(i,j)\in\mathcal{P}} (W_{ij}^P \ell_{ij} + \langle X_i, X_j\rangle) X_i X_j^T \\ & - \beta \sum_{i=1}^{n}\sum_{c=1}^{C}\sum_{j=1}^{m_i} W_{ic}^{I2C} \Delta X_{ij} \Delta X_{ij}^T.\end{aligned} \tag{17}$$

Notice that $W_{ij}^P \ell_{ij} + \langle X_i, X_j\rangle = W_{ji}^P \ell_{ji} + \langle X_j, X_i\rangle$, $\forall i, j$, then we have

$$\begin{aligned}M = & \sum_{1\le i<j\le N} (W_{ij}^P \ell_{ij} + \langle X_i, X_j\rangle)(X_i X_j^T + X_j X_i^T) \\ & + \sum_{i=1}^{N} (W_{ii}^P \ell_{ii} + \langle X_i, X_i\rangle) X_i X_i^T \\ & - \beta \sum_{i=1}^{n}\sum_{c=1}^{C}\sum_{j=1}^{m_i} W_{ic}^{I2C} \Delta X_{ij} \Delta X_{ij}^T.\end{aligned}$$

Thus M is a real-valued symmetric matrix. It is clear that the solution to the optimization problem (16) is the eigenvectors corresponding to the largest d eigenvalues of M. We summarize our algorithm in the following Algorithm 1.

Algorithm 1 Structure Preserving Projection for Local Flux Feature

Input: Training video sequences $V_1, \cdots, V_n$ in gray-scale and $V_1', \cdots, V_n'$ in depth, the radius r for the sphere $B_P(r)$, the parameter k for pairwise structure preserving, the number of centroids K in K-means, the label information function $C(\cdot) \in \{1, \cdots, C\}$, the regularization parameter β and the objective dimension d.

Output: The projection matrix Θ.

1: Detect interest points (or densely sample) $\{P_1, \cdots, P_{m_i}\}$ from the i-th training video V_i, $i = 1, \cdots, n$;
2: Compute two LFFs for each point in gray-scale and depth respectively by Eq. (3) and combine them by Eq. (4) to obtain the local feature set of the i-th training video $\mathcal{X}_i = \{X_{i1}, \cdots, X_{im_i}\}$ and the whole local feature set $\mathcal{F} = \bigcup \mathcal{X}_i = \{X_1, \cdots, X_N\}$;
3: Centralize $X_i \leftarrow \frac{1}{N}\sum_{j=1}^{N} X_j, \forall i$;
4: Construct local feature pairing set $\mathcal{P} = \{(i,j) | X_i, X_j \in \mathcal{F}\}$ and their corresponding pairwise labels $\ell_{ij} = \{-1, +1\}$, where $\ell_{ij} = +1$ if $X_i \in \mathrm{N}_k(X_j)$ or $X_j \in \mathrm{N}_k(X_i)$, and $\ell_{ij} = -1$ otherwise;
5: Employ the K-means clustering algorithm on the set of local features of each class $\bigcup_{C(\mathcal{X}_i)=c} \mathcal{X}_i$, $c = 1, \cdots, C$;
6: Compute pairwise weight W_{ij}^P and I2C similarity W_{ic}^{I2C} by Eqs. (8) and (12);
7: Compute the matrix M by Eq. (17);
8: **return** the eigenvectors corresponding to the largest d eigenvalues of M.

4.4 Complexity Analysis

In this section, we provide a time complexity analysis of our algorithm. During the training phase, our algorithm mainly consists of three parts. The first part is the computation of LFFs. The derivative computation is actually the convolution of matrices which at most needs $O(3DL_m \log L_m)$ time [65], where $L_m = \max\{L, W, H\}$. The second part is the computation of pairwise structure preserving. The k-NN algorithm in the construction of pairwise labels and the computation of pairwise angles cost $O(kN^2)$ and $O(N^2)$ time respectively. The last part is the construction of the I2C similarity matrix $\left(W_{ic}^{I2C}\right)$. The time complexity of this part is $O(nCKDN)$. In total, the time complexity of the training phase is at most $O(3DL_m \log L_m) + O((k+1)N^2) + O(nCKDN)$.

In the test phase, binary codes can significantly reduce the runtime of the recognition algorithm since the distance computation in Hamming space is simply based on the XOR operation. Denote τ_m and τ_{XOR} as the time of one multiplication and one XOR operation respectively. Then the computational complexity of NBNN in the original space is $O(N_{train}N_{test}D)\tau_m$, where N_{train} and N_{test} are the numbers of local features in training and test sets respectively. With the binary local features, the time complexity is reduced to $O(N_{train}N_{test}d)\tau_{XOR}$. In general, we have $d \ll D$ and $\tau_{XOR} \ll \tau_m$. Thereby, when N_{train} and N_{test} are in the magnitude of millions or even greater, the hashing algorithm's effect is self-evident. We will list the run-time in the following section.

5 Experiments and Results

In this section, we systematically evaluate our proposed method on three different RGB-D benchmarks: the SKIG hand gesture dataset [66], the MSRDailyActivity3D dataset [2] and the CAD-60 activity dataset [67]. Fig. 3 shows some example frames of these three datasets. Details of the datasets are introduced in the following subsection.

5.1 Datasets and Settings

The **SKIG** dataset has 2160 hand gesture sequences (1080 RGB sequences and 1080 depth sequences) collected from 6 subjects. All these sequences are synchronously captured with a Kinect sensor (including a RGB camera and a depth camera). This dataset collects 10 categories of hand gestures in total: *circle (clockwise)*, *triangle (anti-clockwise)*, *up-down*, *right-left*, *wave*, *"Z"*, *cross*, *comehere*, *turnaround* and *pat*. In the collection process, all these ten categories are performed with three hand postures: fist, index and flat. To increase the diversity, the sequences are recorded under 3 different backgrounds (i.e., wooden board, white plain paper and paper with characters) and 2 illumination conditions (i.e., strong light and poor light). Consequently, for each subject, there are $10(categories) \times 3(poses) \times 3(backgrounds) \times 2(illumination) \times 2(RGB\ and\ depth) = 360$ gesture sequences. The training size for each category is varied as one of $\{10, 20, 35, 45, 60, 70\}$ and the rest of the sequences are used for testing.

The **MSRDailyActivity3D** dataset is a human activity dataset captured with the RGB channel and the depth channel using the Kinect sensor. The total sequence number is 640 (i.e., 320 sequences for each channel) with 16 activities: *drink*, *eat*, *read book*, *call cellphone*, *write on a paper*, *use laptop*, *use vacuum cleaner*, *cheer up*, *sit still*, *toss paper*, *play game*, *lie down on sofa*, *walk*, *play guitar*, *stand up*, *sit down*. There are 10 subjects in the dataset and each subject performs each activity twice, once in standing position, and once in sitting position. The training size for each subject is chosen as one of $\{5, 10, 15, 20, 25\}$ and the rest is used for testing.

The **Cornell Activity** dataset (CAD-60) contains 60 RGB-depth sequences acted by four subjects and captured with a Kinect camera. The actions in this dataset are categorized into five different environments: office, kitchen, bedroom, bathroom, and living room. Three or four common activities were identified for each environment, giving a total of twelve unique

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2491925, IEEE Transactions on Pattern Analysis and Machine Intelligence

Fig. 3. Example frames of the three RGB-D datasets we used in the experiments. From top to bottom: SKIG, MSRDailyActivity3D and CAD-60.

TABLE 1
Performance comparison (%) of NBNN with the LFFs computed on detected points with different radii. The training sizes are 70, 25 and 4 in each class for SKIG, MSRDailyActivity3D and CAD-60, respectively. All the code lengths are 96-bit.

	$r=1$	$r=2$	$r=3$	$r=4$	$r=5$	$r=6$	$r=7$	$r=8$	$r=9$	$r=10$
SKIG	88.5	90.3	92.4	**93.7**	93.1	92.5	91.6	91.2	90.7	88.4
MSRDailyActivity3D	85.7	86.2	88.7	**89.8**	88.9	88.1	87.6	87.5	86.7	85.2
CAD-60	93.2	94.1	94.9	95.2	**95.7**	94.8	94.1	93.5	92.2	90.8

actions: rinsing mouth, brushing teeth, wearing contact lens, drinking water, opening pill container, cooking (chopping), cooking (stirring), talking on couch, relaxing on couch, talking on the phone, writing on whiteboard, working on computer. The training size for each action is assigned as one of $\{1, 2, 3, 4\}$ and remaining sequences are adopted for testing.

All the training samples are selected randomly from every class in each dataset and all the procedures are repeated five times. We report the averages as the final results.

For the experimental settings, we fix the size of the cuboid $l \times w \times h$ in the computation of LFF as $7 \times 7 \times 9$. We set $r = 4, 4, 5$ in each dataset respectively due to the comparison results with different radii r in Table 1. If the radius r is too small, the LFF degenerates to the second order derivative, and if r is too big, LFFs are almost the same for adjacent pixels, which tends to be less discriminative. We always set $k = 15$ for the pairwise data structure. Actually, we utilize the training data as the cross-validation set in SPP. The parameter K of the K-means is selected from one of $\{100, 200, \cdots, 1000\}$ with the step of 100 , which yields the best performance by 10-fold cross-validation. The optimal parameter β is selected from $\{0.1, 0.2, \cdots, 1.0\}$ with the step of 0.1 by 10-fold cross-validation on the cross-validation set, as well. In particular, the nested cross-validation strategy is applied to these two parameters, i.e., K and β. We always first fix the value of K as one of $\{100, 200, \cdots, 1000\}$ and

TABLE 2
Performance comparison (%) of different variants of LFF+SPP to prove the effectiveness of the improvement on RGB-D fusion. All the code lengths are 96-bit. The bold numbers represent the best performance for each dataset.

Methods \ Datasets	SKIG	MSRDaily Activity3D	CAD-60
LFF+SPP[1]	85.1	82.4	90.4
LFF+SPP[2]	89.6	83.1	93.5
LFF+SPP[3]	91.2	85.8	94.2
LFF+SPP	**93.7**	**89.8**	**95.7**

(SPP[1] is the original SPP without the bigraph regularization; SPP[2] denotes the original SPP without the pairwise label preserving term; SPP[3] represents the original SPP without the pairwise angle preserving term.)

select the best parameter β from $\{0.1, 0.2, \cdots, 1.0\}$, and then assign another value to K and select the best parameter β from $\{0.1, 0.2, \cdots, 1.0\}$ again. In this way, the optimal pair of parameters K and β can be obtained under the nested cross-validation strategy.

Since the acceleration of NBNN is quite conspicuous using the Hamming distance instead of the L_2-norm in the NN-search and NBNN classifier always outperforms the BoW model, we mainly use NBNN to evaluate our recognition precision.

5.2 Compared Results

First of all, we illustrate the effectiveness of all the three terms used in SPP, i.e., the pairwise label preserving term, the pairwise angle preserving term and the bigraph regularization. We remove one of them and keep the other two terms, and optimize the problem in (15). The results are listed in Table 2, from which we can observe that the bigraph regularization contributes the most to the accuracies.

Next, for all three datasets, we apply three different schemes to achieve RGB-D video classification: (1) Detected interest points[4] + LFF + SPP; (2) Dense sampling[5] + LFF + SPP; (3) Detected interest points + LFF + SPP + Bag-of-Words. For (1) and (2), we adopt NBNN as the classifier and the linear SVM is applied for the third scheme for classification. The codebook lengths of BoW for each dataset are chosen as one of $\{500, 1000, 1500, 2000\}$ and the best results are reported.

For each scheme, we apply SPP on LFFs from RGB and depth information. According to all the possible combinations, we evaluate four different kind of local binary codes on three datasets: LFF(RGB-D)+SPP denotes our full algorithm; LFF(RGB)+SPP only uses RGB information to compute LFFs and then apply SPP; LFF(D)+SPP only uses depth information to compute LFFs and then apply SPP; LFF+SPP(RGB-D) concatenates LFF(RGB)+SPP and LFF(D)+SPP.

From Figs. 4–6, we can observe that the performance of our full algorithm is consistently higher than that of other versions on the three datasets. And dense sampling generally outperforms interest points detection due to the large amount of local feature descriptors. Another observation is that LFF(RGB-D)+SPP always outperforms LFF+SPP(RGB-D), since the former outputs the fused binary representation with the consideration of the structures of RGB-D features. In contrast, LFF+SPP(RGB-D) outputs binary codes separately for RGB and depth features, therefore, loses the connection between RGB and depth features.

In Fig. 7, we also compare the performance of our algorithm with different code lengths by using different point selection methods, i.e., interest points detection (Dollar's detector and STIP) and dense sampling, on the three datasets. It is noticeable that, on the CAD-60 dataset, the accuracy of dense sampling is slightly lower than that of interest points detection because the noise of the background has a negative effect on the dense sampling when the code length increases. In this situation, the detection method is more effective than dense sampling.

Finally, Fig. 8 shows the average runtime comparison. Our learned binary codes show a significant advantage compared to the original LFF consisting of real numbers since NBNN largely depends on NN-search. All the experiments are conducted using Matlab 2013a on a server configured with a 12-core processor and 128G of RAM running the Linux OS.

5.3 Comparison with Other Methods

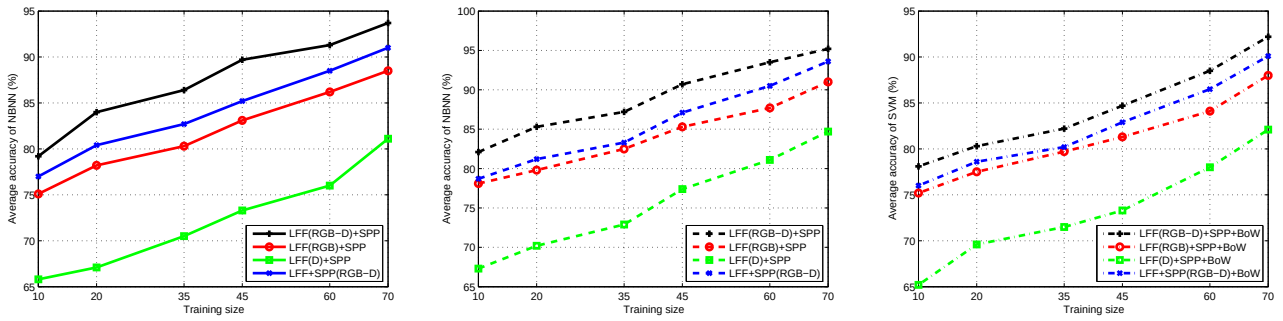
In Table 3, we first compare the proposed LFF descriptor with state-of-the-art video descriptors (i.e., HOG, HOF, MBH, HON4D and HOG3D) for RGB-D action recognition. All the methods are computed on the interest points from the RGB channel detected by Dollar's detector and the corresponding points from the depth channel. As we can see, LFF outperforms HOG, HOF, MBH and HOG3D in the RGB and depth channels and the RGB-D concatenation scheme. Although HON4D, as a descriptor specifically designed for depth sequences, achieves better performance in the depth channel, it can only be extracted from depth data and the recognition accuracies are relatively low. In contrast, our LFF is considered to be a general feature descriptor for both RGB and depth data and LFF in the RGB-D concatenation scheme reaches the highest accuracy in the experiment of feature comparison.

Since SPP is a projection for learning binary codes, we can also compare our SPP algorithm with other

4. Dollar's interest points detector [22] is used in our experiments. We only detect the interest points on the RGB data and find the corresponding locations on the depth video as the detected points for depth data.

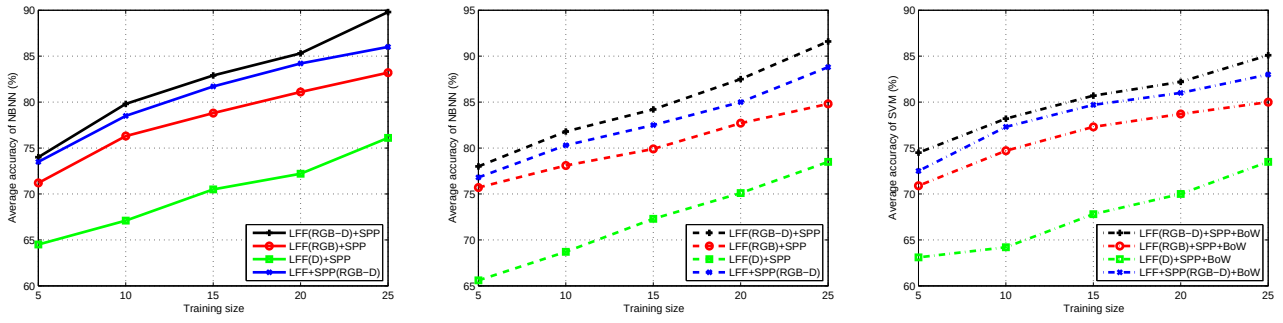
5. We set the distance between adjacent pixels as 5.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2491925, IEEE Transactions on Pattern Analysis and Machine Intelligence



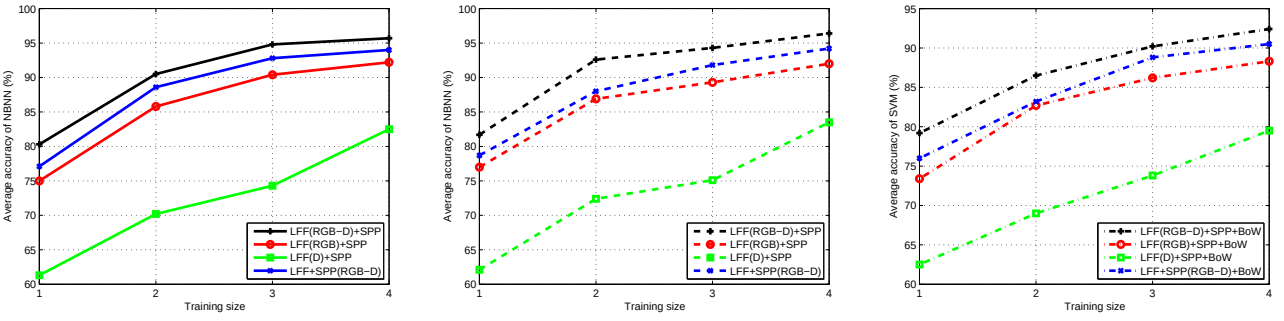
(a) Dollar's detector (b) Dense sampling (c) SVM with the BoW model

Fig. 4. Performance comparison with different training sizes in each category and different versions of LFFs on the SKIG dataset at 96-bit.



(a) Dollar's detector (b) Dense sampling (c) SVM with the BoW model

Fig. 5. Performance comparison with different training sizes for each subject and different versions of LFFs on the MSRDailyActivity3D dataset at 96-bit.

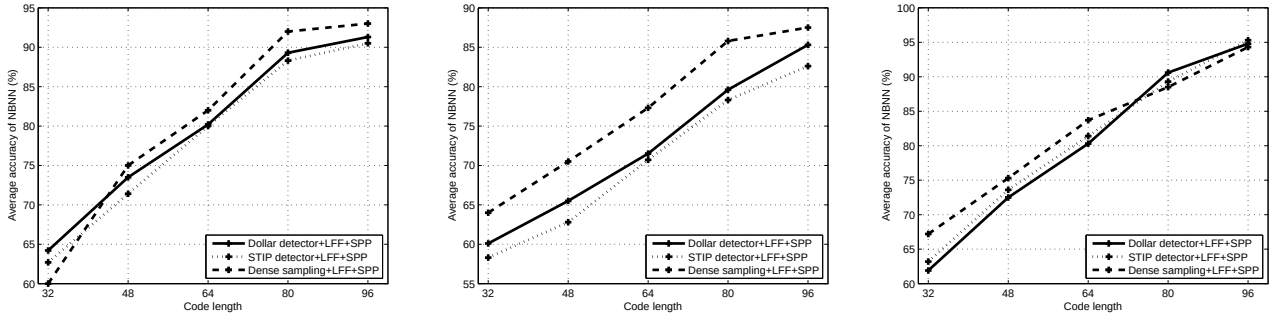


(a) Dollar's detector (b) Dense sampling (c) SVM with the BoW model

Fig. 6. Performance comparison with different training sizes in each action and different versions of LFFs on the CAD-60 dataset at 96-bit.

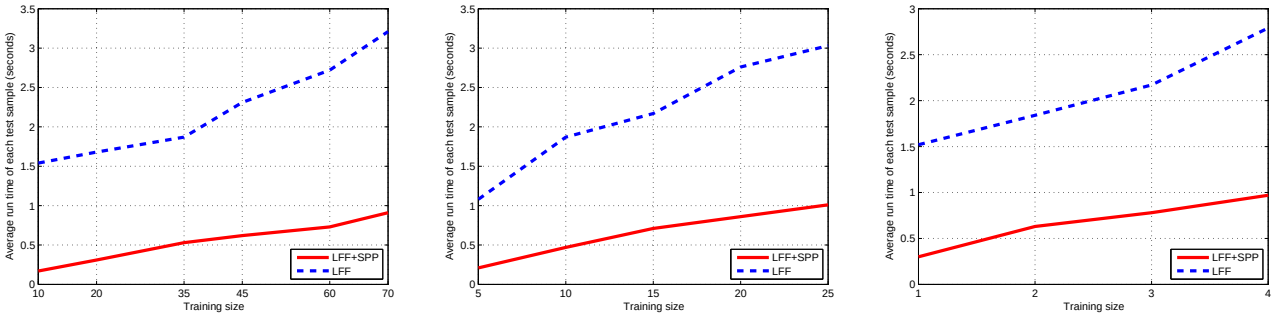
hashing methods. In our experiments, we compare the proposed method against seven general hashing algorithms including KSH [58], BRE [60], MLH [61], LSH [56], SpH [57], AGH [68], PCAH [69], BSSC [53] and RBM [70]. All the above methods are computed on the same extracted LFFs for a unified standard. All the compared methods are then evaluated on five different lengths of codes (32, 48, 64, 80, 96) and their results at 96-bit, which appear to be the best, are reported. Under the same experimental setting, all the parameters used in the compared methods have been strictly chosen according to their original papers. We list the compared results in Table 3 where RGB channel and depth channel represent only employing the methods in RGB and depth respectively, RGB-D fusion is the procedure of our algorithm and RGB-D cat is the concatenation of the features gained in RGB channel and depth channel. The results of the above mentioned other hashing methods in RGB-D fusion are not consistently higher than that in RGB-D concatenation, since not all of them preserve data structure. The training sizes are 70, 25 and 4

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2491925, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) SKIG (training size 60) (b) MSRDailyActivity3D (training size 20) (c) CAD-60 (training size 3)

Fig. 7. Performance comparison of NBNN with different point selection methods on three datasets.



(a) SKIG (b) MSRDailyActivity3D (c) CAD-60

Fig. 8. Average runtime of one test sample of NBNN by using 96-bit binary codes after SPP and the original 882-dimensional LFF with different training sizes.

TABLE 3

Performance comparison (%) of our algorithm and other coding methods on three datasets. In the RGB-D fusion scheme, we first concatenate features in RGB and depth, then apply hashing methods. In the RGB-D concatenation (Cat) scheme, we first apply hashing methods to features in RGB and depth separately, then concatenate them. The bold numbers represent the best performance for each dataset.

Methods	SKIG				MSRDailyActivity3D				CAD-60			
	RGB Channel	Depth Channel	RGB-D Cat	RGB-D Fusion	RGB Channel	Depth Channel	RGB-D Cat	RGB-D Fusion	RGB Channel	Depth Channel	RGB-D Cat	RGB-D Fusion
HOG	81.4	72.7	82.9	-	76.4	62.3	79.2	-	78.4	60.3	79.6	-
HOF	79.0	71.2	80.6	-	75.6	62.2	78.9	-	77.0	58.5	77.8	-
MBH	82.1	74.7	83.2	-	76.7	63.1	80.1	-	79.5	61.2	81.8	-
HON4D	-	80.1	-	-	-	78.4	-	-	-	69.2	-	-
HOG3D	81.8	73.4	83.1	-	77.2	62.4	79.5	-	78.5	60.4	80.5	-
LFF	84.0	76.2	85.4	-	80.6	72.8	81.6	-	81.0	63.6	83.2	-
Action ensemble*	-	-	-	-	-	87.6	-	-	-	91.8	-	-
HOG3D+IFV	86.9	79.8	89.7	92.1	83.1	75.1	85.6	89.5	91.0	80.8	92.4	94.8
LFF+IFV	88.7	80.5	91.5	93.2	84.8	76.0	87.4	**91.1**	91.4	82.0	93.4	95.1
HOG3D+SPP	86.3	78.6	88.2	91.4	84.3	71.5	85.2	87.4	88.1	67.4	92.1	93.0
LFF+SPP	88.5	81.1	91.0	**93.7**	83.2	76.1	86.0	89.8	92.2	82.5	94.0	**95.7**
LFF+KSH	81.7	67.9	82.4	80.1	80.1	72.1	82.5	81.0	76.0	52.5	77.2	76.8
LFF+BRE	79.8	63.4	80.2	80.8	78.1	68.1	81.3	79.8	75.5	56.7	76.0	76.1
LFF+MLH	77.5	63.8	78.4	78.8	74.2	69.3	75.0	76.2	75.3	48.6	75.8	74.7
LFF+LSH	69.4	54.2	71.4	68.2	60.5	41.1	62.3	58.4	61.4	30.7	62.5	60.2
LFF+SpH	77.5	68.1	78.5	79.0	76.2	60.7	78.3	78.2	70.7	50.4	71.3	73.1
LFF+AGH	74.2	70.5	77.4	78.3	77.5	63.2	78.4	79.5	73.6	48.2	74.7	74.0
LFF+PCAH	68.0	60.2	68.3	60.4	61.3	48.7	63.0	62.1	65.3	41.0	67.9	60.1
LFF+BSSC	77.8	55.4	80.3	81.3	76.9	65.3	76.7	78.0	74.2	48.2	76.8	77.2
LFF+RBM	78.5	67.6	79.5	79.7	77.2	60.0	78.3	78.5	77.4	58.3	79.7	78.8

∗ The action ensemble method adopted the depth and skeleton information with real-valued features. The skeleton information is only available in MSRDailyActivity3D and CAD-60.

All the results (except action ensemble, LFF+IFV and HOG3D+IFV) are calculated by the NBNN classifier. The linear SVM is applied to LFF+IFV and HOG3D+IFV.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2015.2491925, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 4
t-Test on performance improvements.

Methods / Datasets	LFF+SPP vs. LFF+KSH	LFF+SPP vs. LFF+BRE	LFF+SPP vs. LFF+MLH	LFF+SPP vs. LFF+SpH	LFF+SPP vs. LFF+LSH	LFF+SPP vs. HOG3D+SPP
SKIG	9.97×10^{-13}	4.31×10^{-12}	1.52×10^{-14}	3.09×10^{-12}	1.45×10^{-14}	2.49×10^{-7}
MSRDailyActivity3D	3.98×10^{-12}	9.72×10^{-12}	3.26×10^{-13}	2.27×10^{-12}	5.78×10^{-16}	1.52×10^{-6}
CAD-60	3.57×10^{-13}	8.58×10^{-15}	3.55×10^{-15}	8.88×10^{-14}	1.46×10^{-17}	2.34×10^{-6}

TABLE 5
Recognition accuracy (%) of LFF and dense trajectory features on the UCF YouTube and HMDB51 datasets.

Feature	UCF YouTube	HMDB51
Trajectory	67.5	28.0
HOG	72.6	27.9
HOF	70.0	31.5
MBH	80.6	43.2
Trajectory/HOG/HOF/MBH combined	84.1	46.6
LFF (r = 1)	79.6	41.5
LFF (r = 3)	84.3	45.8
LFF (r = 5)	**85.2**	**46.9**
LFF (r = 7)	84.7	46.0
LFF (r = 9)	83.2	45.5

The LFF features are extracted along the same trajectories in the video sequences as the dense trajectory features.

for datasets SKIG, MSRDailyActivity3D and CAD-60, respectively. Table 3 also reports the recognition accuracies of LFF and HOG3D using the improved Fisher vector (IFV) [71], for which 200 Gaussians are used in the GMM. The results show two phenomena: 1. LFF as a continuous feature outperforms other discrete histogram based features; 2. SPP outperforms other hashing methods.

5.4 Statistical Significance Test

To show the statistical significance of improvements, we conduct a t-test on the MAP improvements. In testing the null hypothesis that the population mean is equal to a specified value μ_0, the statistic

$$t = \frac{\overline{x} - \mu_0}{s/\sqrt{m}}$$

is used, where $\overline{x}$ is the sample mean, s is the sample standard deviation of the sample and m is the sample size. Then the degree of freedom used in the test is $m - 1$. We set $m = 10$ and code length $d = 96$ for this experiment. Table 4 lists the one-tail results of the t-test, which shows that the improvements are statistically significant.

5.5 Results on RGB Video dataset

To further illustrate the effectiveness of LFF, in this experiment, we compare the RGB version of LFF with the state-of-the-art feature: dense trajectory features on the **UCF YouTube** [72] and **HMDB51** [73] datasets for action recognition. The UCF YouTube dataset contains 1168 video sequences collected from 11 action categories. Most of them are sports activities, which are drawn from existing YouTube videos; therefore, the dataset contains large variations and approximates a real-world database. For this dataset, we deliberately use the full-sized sequences without any bounding boxes as the input to evaluate our method's robustness against complex and noisy backgrounds. We use the Leave-One-Out setup, i.e., testing on each original sequence while training on all the other sequences. The HMDB51 dataset contains 6849 realistic action sequences collected from a variety of movies and online videos. Specifically, it has 51 action classes and each has at least 101 positive samples. We adopt the official setting of [73] with three train/test splits. Each split has 70 training and 30 testing clips for each class. Table 5 illustrates that our proposed LFF ($r = 5$) can achieve competitive results with dense trajectory feature (DTF) which produces the state-of-the-art performance on recent publications [31], [74]. Note that for fair comparison of feature descriptors, all the compared features are extracted around the same points, i.e., the points on the trajectories.

6 Conclusion

The basic goal of this paper is to obtain a fused local binary representation for RGB-D action recognition. To achieve this goal, we first introduced a continuous local descriptor called Local Flux Feature (LFF) based on the gradient field of video data, which is more suitable for the discretization of binary codes than histogram based local descriptors. After acquiring LFFs from RGB and depth channels, we applied the Structure Preserving Projection (SPP) to learn discriminative local binary representations. SPP preserves the characteristics in two levels including pairwise structure of local features and the relationship between video samples and classes at the same time without the collapse of data structure. The systematical experiments have shown not only the high efficiency of the proposed local binary representations, but also its superior performance than other local features and other hashing methods in terms of recognition accuracy on three RGB-D datasets.

Acknowledgements

This work was supported in part by Northumbria University, in part by National Natural Science Foundation of China under Grant 61528106, and in part by the Newton International Exchanges Scheme.

REFERENCES

[1] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.

[2] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1290–1297.

[3] M. Paton and J. Kosecka, "Adaptive rgb-d localization," in *Conference on Computer and Robot Vision*, 2012, pp. 24–31.

[4] L. Spinello and K. O. Arras, "People detection in rgb-d data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 3838–3843.

[5] X. Ren, L. Bo, and D. Fox, "Rgb-(d) scene labeling: Features and algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2759–2766.

[6] L. Cruz, D. Lucio, and L. Velho, "Kinect and rgbd images: Challenges and applications," in *SIBGRAPI Conference on Graphics, Patterns and Images - Tutorials*, 2012, pp. 36–49.

[7] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[9] A. Kläser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *Proceedings of the British Machine Vision Conference*, 2008, pp. 1–10.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

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

[12] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *European Conference on Computer Vision*, 2006, pp. 428–441.

[13] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision*, 2003, pp. 1470–1477.

[14] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Neural Information Processing Systems*, 2006, pp. 801–808.

[15] G. Arfken and H. Weber, *Mathematical Methods for Physicists*. Elsevier, 2005.

[16] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[17] L. Liu, L. Shao, X. Li, and K. Lu, "Learning spatio-temporal representations for action recognition: A genetic programming approach," *IEEE Transaction Cybernetics*, 2015.

[18] L. Shao, X. Zhen, D. Tao, and X. Li, "Spatio-temporal laplacian pyramid coding for action recognition," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 817–827, 2014.

[19] F. Zhu and L. Shao, "Weakly-supervised cross-domain dictionary learning for visual recognition," *International Journal of Computer Vision*, vol. 109, no. 1-2, pp. 42–59, 2014.

[20] L. Liu, L. Shao, and P. Rockett, "Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition," *Pattern Recognition*, vol. 46, no. 7, pp. 1810–1818, 2013.

[21] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

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

[23] W. T. Freeman and M. Roth, "Orientation histograms for hand gesture recognition," in *International Workshop on Automatic Face and Gesture Recognition*, vol. 12, 1995, pp. 296–301.

[24] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[25] H. Bay, T. Tuytelaars, and L. J. V. Gool, "Surf: Speeded up robust features," in *European Conference on Computer Vision*, 2006, pp. 404–417.

[26] H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[27] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *ACM International Conference on Multimedia*, 2007, pp. 357–360.

[28] S. Hadfield, K. Lebeda, and R. Bowden, "Natural action recognition using invariant 3d motion encoding," in *European Conference on Computer Vision*, 2014, pp. 758–771.

[29] M. B. Holte, T. B. Moeslund, N. Nikolaidis, and I. Pitas, "3d human action recognition for multi-view camera systems," in *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 2011, pp. 342–349.

[30] A. Fathi and G. Mori, "Action recognition by learning mid-level motion features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[31] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, 2013.

[32] R. Poppe, "A survey on vision-based human action recognition," *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[33] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

[34] M. A. R. Ahad, J. K. Tan, H. Kim, and S. Ishikawa, "Motion history image: its variants and applications," *Machine Vision and Applications*, vol. 23, no. 2, pp. 255–281, 2012.

[35] J. Shotton, A. W. 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1297–1304.

[36] L. Spinello and K. O. Arras, "Leveraging rgb-d data: Adaptive fusion and domain adaptation for object detection," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 4469–4474.

[37] K. Lai, L. Bo, X. Ren, and D. Fox, "Sparse distance learning for object recognition combining rgb and depth information," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 4007–4013.

[38] B. Ni, G. Wang, and P. Moulin, "Rgbd-hudaact: A color-depth video database for human daily activity recognition," in *IEEE International Conference on Computer Vision Workshops*, 2011, pp. 1147–1153.

[39] O. Oreifej and Z. Liu, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 716–723.

[40] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *KDD workshop*, vol. 10, no. 16, 1994, pp. 359–370.

[41] S. Sempena, N. U. Maulidevi, and P. R. Aryan, "Human action recognition using dynamic time warping," in *International Conference on Electrical Engineering and Informatics*, 2011, pp. 1–5.

[42] M. Müller and T. Röder, "Motion templates for automatic classification and retrieval of motion capture data," in *ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2006, pp. 137–146.

[43] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.

[44] D. Engel and C. Curio, "Scale-invariant medial features based on gradient vector flow fields," in *International Conference on Pattern Recognition*, 2008, pp. 1–4.

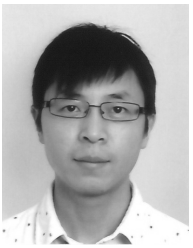
[45] S. A. A. Shah, M. Bennamoun, F. Boussaid, and A. A. El-Sallam, "A novel local surface description for automatic 3d object recognition in low resolution cluttered scenes," in *IEEE International Conference on Computer Vision Workshops*, 2013, pp. 638–643.

[46] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[47] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Neural Information Processing Systems*, 2001, pp. 585–591.
[48] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
[49] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet, "Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering," in *Neural Information Processing Systems*, 2003.
[50] X. He and P. Niyogi, "Locality preserving projections," in *Neural Information Processing Systems*, 2003.
[51] X. He, D. Cai, S. Yan, and H. Zhang, "Neighborhood preserving embedding," in *IEEE International Conference on Computer Vision*, 2005, pp. 1208–1213.
[52] D. Cai, H. Bao, and X. He, "Sparse concept coding for visual analysis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2905–2910.
[53] G. Shakhnarovich, "Learning task-specific similarity," Ph.D. dissertation, Massachusetts Institute of Technology, 2005.
[54] R. Caseiro, P. Martins, J. F. Henriques, F. S. Leite, and J. Batista, "Rolling riemannian manifolds to solve the multi-class classification problem," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 41–48.
[55] L. Liu, M. Yu, and L. Shao, "Multiview alignment hashing for efficient image search," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 956–966, 2015.
[56] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *International Conference on Very Large Data Bases*, 1999, pp. 518–529.
[57] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Neural Information Processing Systems*, 2008, pp. 1753–1760.
[58] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2074–2081.
[59] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua, "Ldahash: Improved matching with smaller descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 66–78, 2012.
[60] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *Advances in Neural Information Processing Systems*, 2009, pp. 1042–1050.
[61] M. Norouzi and D. J. Fleet, "Minimal loss hashing for compact binary codes," in *International Conference on Machine Learning*, 2011, pp. 353–360.
[62] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *European Conference on Computer Vision*, 2008, pp. 304–317.
[63] L. Liu, M. Yu, and L. Shao, "Local feature binary coding for approximate nearest neighbor search," in *British Machine Vision Conference*, 2015.
[64] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities*. Cambridge university press, 1952.
[65] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.
[66] L. Liu and L. Shao, "Learning discriminative representations from rgb-d video data," in *International Joint Conference on Artificial Intelligence*, 2013.
[67] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from rgbd images," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 842–849.
[68] W. Liu, J. Wang, S. Kumar, and S.-F. Chang, "Hashing with graphs," in *International Conference on Machine Learning*, 2011, pp. 1–8.
[69] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 12, pp. 2393–2406, 2012.
[70] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[71] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *European Conference on Computer Vision*, 2010, pp. 143–156.
[72] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1996–2003.
[73] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: A large video database for human motion recognition," in *IEEE International Conference on Computer Vision*, 2011.
[74] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 3169–3176.

Mengyang Yu (S'14) received the B.S. and M.S. degrees from the School of Mathematical Sciences, Peking University, Beijing, China, in 2010 and 2013 respectively. He is currently working toward the Ph.D. degree in the Department of Computer Science and Digital Technologies at Northumbria University, Newcastle upon Tyne, U.K. His research interests include computer vision, machine learning and data mining.

Li Liu received the B.Eng. degree in electronic information engineering from Xi'an Jiaotong University, Xi'an, China, in 2011, and the Ph.D. degree in the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, U.K., in 2014. Currently, he is a research fellow in the Department of Computer Science and Digital Technologies at Northumbria University. His research interests include computer vision, machine learning and data mining.



Ling Shao (M'09–SM'10) is a Professor with the Department of Computer Science and Digital Technologies at Northumbria University, Newcastle upon Tyne, U.K. Previously, he was a Senior Lecturer (2009-2014) with the Department of Electronic and Electrical Engineering at the University of Sheffield and a Senior Scientist (2005-2009) with Philips Research, The Netherlands. His research interests include Computer Vision, Image/Video Processing and Machine Learning. He is an associate editor of IEEE Transactions on Image Processing, IEEE Transactions on Cybernetics and several other journals. He is a Fellow of the British Computer Society and the Institution of Engineering and Technology.