Group Sparsity and Geometry Constrained Dictionary Learning for Action Recognition from Depth Maps

Jiajia Luo, Wei Wang, and Hairong Qi
The University of Tennessee, Knoxville
{jluo9, wwang34, hqi}@utk.edu

Abstract

Human action recognition based on the depth information provided by commodity depth sensors is an important yet challenging task. The noisy depth maps, different lengths of action sequences, and free styles in performing actions, may cause large intra-class variations. In this paper, a new framework based on sparse coding and temporal pyramid matching (TPM) is proposed for depth-based human action recognition. Especially, a discriminative class-specific dictionary learning algorithm is proposed for sparse coding. By adding the group sparsity and geometry constraints, features can be well reconstructed by the sub-dictionary belonging to the same class, and the geometry relationships among features are also kept in the calculated coefficients. The proposed approach is evaluated on two benchmark datasets captured by depth cameras. Experimental results show that the proposed algorithm repeatedly achieves superior performance to the state of the art algorithms. Moreover, the proposed dictionary learning method also outperforms classic dictionary learning approaches.

1. Introduction

Traditional human action recognition approaches focus on learning distinctive feature representations for actions from labelled videos and recognizing actions from unknown videos. However, it is a challenging task to label unknown RGB sequences due to the large intra-class variability and inter-class similarity of actions, cluttered background, possible camera movements and illumination changes.

Recently, the introduction of cost-effective depth cameras provides a new possibility to address difficult issues in traditional human action recognition. Compared to the monocular video sensors, depth cameras can provide 3D motion information so that the discrimination of actions can be enhanced and the influence of cluttered background and illumination variations can be mitigated. Especially, the work of Shotton *et al*. [16] provided an efficient human motion capturing technology to accurately estimate the 3D skeleton joint positions from a single depth image, which are more compact and discriminative than RGB or depth sequences. As shown in Figure 1, the action "drink" from the MSR DailyActivity3D dataset [19], can be well reflected from the extracted 3D joints by comparing the joints "head" and "hand" in the two frames. However, it is not that straightforward to tell the difference between the two frames from the depth maps or color images.

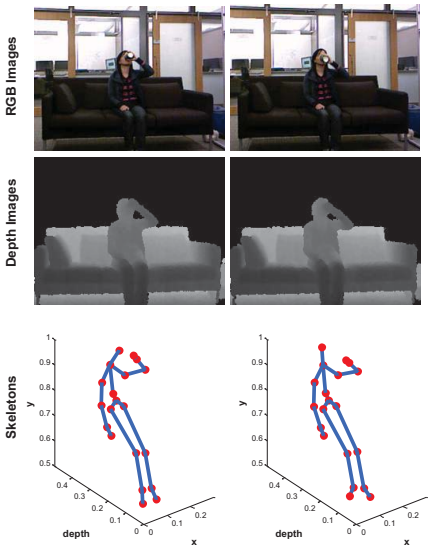

Figure 1. Sample images obtained by different cameras for the action "drink". The 3D joints are estimated by the method in [16].

Although with strong representation power, the estimated 3D joints also bring challenges to perform depth-data based action recognition. For example, the estimated 3D joint positions are sometimes unstable due to the noisy depth maps. In addition, the estimated 3D joint positions are frame-based, which require representation methods to be tolerant to the variations in speed and time of actions.

To extract robust features from estimated 3D joint positions, relative 3D joint features have been explored and

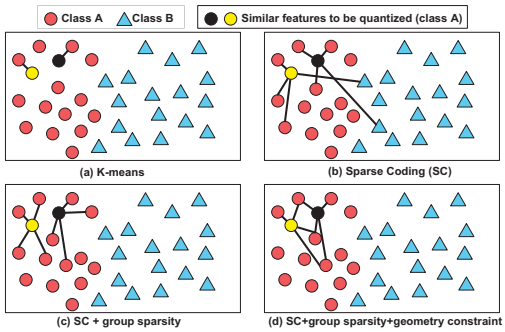

Figure 2. Illustration of different feature quantization strategies. (a) K-means. (b) Sparse coding. (c) Sparse coding with group sparsity constraint. (d) Proposed method (sparse coding with group sparsity and geometry constraint).

achieved satisfactory performance [19, 21, 24]. To represent depth sequences with different lengths, previous research mainly focused on temporal alignment of sequences [11, 14, 21] or frequencies evolution of extracted features [19] within a given period. However, the limited lengths of sequences, the noisy 3D joint positions, and the relatively small number of training samples may cause the overfitting problem and make the representation unstable.

In this paper, a new framework is proposed for depth-based human action recognition. Instead of modeling temporal evolution of features, our work emphasizes on the distributions of representative features within a given time period. To realize this representation, a new *Dictionary Learning* (DL) method is proposed and the *Temporal Pyramid Matching* (TPM) is used for keeping the temporal information. The proposed DL method aims to learn an overcomplete set of representative vectors (atoms) so that any input feature can be approximated by linear combination of these atoms. The coefficients for the linear combination are referred to as the "sparse codes".

From the DL algorithm design perspective, recent trend is to develop "discriminative" dictionaries to solve classification problems. For example, Zhang and Li [25] proposed a discriminative K-SVD method by incorporating classification error into the objective function and learned the classifier together with the dictionary. Jiang *et al*. [6] further increased the discrimination by adding a label consistent term. Yang *et al*. [23] proposed to add the Fisher discrimination criterion into the dictionary learning. For these methods, labels of inputs should be known before training. However, this requirement cannot be satisfied in our problem. Since different actions contain shared local features, assigning labels to these local features would not be proper.

In this paper, we propose a discriminative DL algorithm for depth-based action recognition. Instead of simultaneously learning one overcomplete dictionary for all classes, we learn class-specific sub-dictionaries to increase the discrimination. In addition, the $l_{1,2}$**-mixed norm** and **geometry constraint** are added to the learning process to further increase the discriminative power. Existing class-specific dictionary learning methods [7, 15] are based on l_1 norm which may result in randomly distributed coefficients [4]. In this paper, we add the group sparsity regularizer [26], which is a combination of l_1- and l_2- norms to ensure features are well reconstructed by atoms from the same class. Moreover, the geometry relationship among local features are incorporated during the process of dictionary learning, so that features from the same class with high similarity will be forced to have similar coefficients.

The process that assigns each feature with coefficients according to a learned dictionary can be defined as "quantization", following the similar definition in the field of image classification. As shown in Figure 2, different quantization methods will generate different representations. Atoms from two classes are marked as "circles" (class A) and "triangles" (class B), respectively. We use two similar features to be quantized (both from class A) as an example to illustrate the coefficient distribution from various quantization methods. In k-means, features are assigned to the nearest atoms, which is sensitive to the variations of features. In the sparse coding with l_1 norm, features are assigned to atoms with lowest reconstruction error, but the distributions of selected atoms can be random and from different classes [4]. In the spare coding with group sparsity, features will choose atoms from the same group (class), but similar features may not choose the same atoms within the group. In our method, features from the same class will be forced to choose atoms within the same group, and the selections of atoms also relate to the similarity of features.

The main contributions of this paper are three-fold. First, a new discriminative dictionary learning algorithm is proposed to realize the quantization of depth features. Both the group sparsity and geometry constraints are incorporated to improve the discriminative power of the learned dictionary. Second, a new framework that based on sparse coding and temporal pyramid matching is proposed to solve the temporal alignment problem of depth features. Third, extensive experimental results have shown that both the proposed framework and the dictionary learning algorithm are effective for the task of action recognition based on depth maps.

2. Background of Sparse Coding

Given a dataset $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$, sparse coding is a process to solve the optimization problem as:

$$\min_{\mathbf{D},\mathbf{X}} \left\{ \sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2^2 + \lambda |\mathbf{x}_i|_1 \right\} \qquad (1)$$

where matrix $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_K]$ is the dictionary with K atoms and elements in matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ are coef-

ficients. Different from the K-means clustering that assigns every data with its nearest cluster center, sparse coding uses a linear combination of atoms in the dictionary $\mathbf{D}$ to reconstruct the data, and only a sparse number of atoms have nonzero coefficients.

To increase the discriminative power of dictionary, class-specific dictionary learning methods have been proposed that learn a sub-dictionary for each class [7, 15]. For example, Eq. 1 can be rewritten as:

$$\min_{\mathbf{D},\mathbf{X}} \sum_{i=1}^{C} \left\{ \|\mathbf{Y}_i - \mathbf{D}_i\mathbf{X}_i\|_F^2 + \lambda \sum_{j=1}^{N_i} |\mathbf{x}_j^i|_1 \right\} \qquad (2)$$

where $\mathbf{Y}_i = [\mathbf{y}_1^i, \ldots, \mathbf{y}_{N_i}^i]$ and $\mathbf{X}_i = [\mathbf{x}_1^i, \ldots, \mathbf{x}_{N_i}^i]$ are the dataset and coefficients for class i, respectively. Matrix $\mathbf{D}_i$ is the learned sub-dictionary for class i.

Since the sub-dictionaries are trained independently, it is possible that related atoms among those sub-dictionaries are generated. In this case, the sparse representation will be sensitive to the variations among features. Even though an incoherence promoting term $\sum_{i \neq j} \|\mathbf{D}_i^T \mathbf{D}_j\|_F^2$ can be added to the dictionary learning, correlated atoms still exist [15].

3. Proposed Method

The proposed depth-based human action recognition framework consists of three components, feature extraction from the 3D joint positions, feature representation using the discriminative DL and temporal pyramid matching, and classification. Our discussion below focuses on the construction of the discriminative dictionary which is the main contributor to the success of the proposed framework.

3.1. Feature Extraction

Given a depth image, 20 joints of the human body can be tracked by the skeleton tracker [16]. At frame t, the position of each joint i is uniquely defined by three coordinates $\mathbf{p}_i(t) = (x_i(t), y_i(t), z_i(t))$ and can be represented as a 3-element vector. The work of Wang *et al*. [19] showed that the pairwise relative positions result in more discriminative and intuitive features. However, enumerating all the joint pairs introduces some redundant and irrelevant information to the classification task [19].

In this paper, only one joint is selected as a reference joint, and its differences to all the other joints are used as features. Since the joint *Hip Center* has relatively small motions for most actions, it is used as a reference joint. Let the position for the *Hip Center* be $\mathbf{p}_1(t)$, the **3D joint feature** at frame t is defined as:

$$\mathbf{y}(t) = \{\mathbf{p}_i(t) - \mathbf{p}_1(t) | i = 2, \ldots, 20\} \qquad (3)$$

Note that both $\mathbf{p}_1$ and $\mathbf{p}_i$ are functions of time, and $\mathbf{y}(t)$ is a vector with 57 ($19 \times 3 = 57$) elements. For any depth sequence with T frames, there will be T joint features from $\mathbf{y}(1)$ to $\mathbf{y}(T)$.

Compared to the work of [19] using 20 joints as references by turns, our experimental result will show that only 1 joint used as reference is sufficient for the proposed framework to achieve state-of-the-art accuracies on benchmark datasets.

3.2. Group Sparsity and Geometry Constrained Dictionary Learning (DL-GSGC)

The process that generates a vector representation for any depth sequence with a specific number of extracted 3D joint features is referred to as "feature representation". Although the Bag-of-Words representation based on K-means clustering can serve the purpose, it discards all the temporal information and large vector quantization error can be introduced by assigning each 3D joint feature to its nearest "visual word". Recently, Yang *et al*. [22] showed that classification accuracies benefit from generalizing vector quantization to sparse coding. However, discrimination of the representation can be compromised due to the possible randomly distributed coefficients solved by sparse coding [4]. In this paper, a class specific dictionary learning method based on group sparsity and geometry constraint is proposed, referred to as **DL-GSGC**.

Group sparsity encourages the sparse coefficients in the same group to be zero or nonzero simultaneously [2, 4, 26]. Adding the group sparsity constraint to the class-specific dictionary learning has three advantages. First, the intra-class variations among features can be compressed since features from the same class tend to select atoms within the same group (sub-dictionary). Second, influence of correlated atoms from different sub-dictionaries can be compromised since their coefficients will tend to be zero or nonzero simultaneously. Third, possible randomness in coefficients distribution can be removed since coefficients have group-clustered sparse characteristics. In this paper, the **Elastic net regularizer** [26] is added as the group sparsity constraint since it has automatic group effect. The Elastic net regularizer is a combination of the l_1- and l_2 norms. Specifically, the l_1 penalty promotes sparsity, while the l_2 norm encourages the grouping effect [26].

Given a learned dictionary that consists of all the sub-dictionaries and an input feature from class i, it is ideal to use atoms from the i-th class to reconstruct it. In addition, similar features should have similar coefficients. Inspired by the work of Gao *et al*. [5], we propose to add geometry constraint to the class-specific dictionary learning process.

Let $\mathbf{Y} = [\mathbf{Y}_1, \ldots, \mathbf{Y}_C]$ be the dataset with N features for C classes, where $\mathbf{Y}_i \in \mathbb{R}^{f \times N_i}$ is the f-dimensional dataset from class i. **DL-GSGC** is designed to learn a discriminative dictionary $\mathbf{D} = [\mathbf{D}_1, \ldots, \mathbf{D}_C]$ with K atoms in total ($K = \sum_{i=1}^{C} K_i$), where $\mathbf{D}_i \in \mathbb{R}^{f \times K_i}$ is the class-

specified sub-dictionary associated with class i. The objective function of DL-GSGC is:

$$\min_{\mathbf{D},\mathbf{X}} \left\{ \begin{array}{l} \sum_{i=1}^{C} \{\|\mathbf{Y}_i - \mathbf{D}\mathbf{X}_i\|_F^2 + \|\mathbf{Y}_i - \mathbf{D}_{\in i}\mathbf{X}_i\|_F^2 + \\ \|\mathbf{D}_{\notin i}\mathbf{X}_i\|_F^2 + \lambda_1 \sum_{j=1}^{N_i} |\mathbf{x}_j^i|_1 + \lambda_2 \|\mathbf{X}_i\|_F^2\} \\ + \lambda_3 \sum_{i=1}^{J} \sum_{j=1}^{N} \|\alpha_i - \mathbf{x}_j\|_2^2 w_{ij} \end{array} \right\} \quad (4)$$

$$\text{subject to} \quad \|\mathbf{d}_k\|_2^2 = 1, \qquad \forall k = 1, 2, \ldots, K$$

where $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_C]$ represents the coefficients matrix and coefficients vector for the j-th feature in class i is $\mathbf{x}_j^i$. The value $\mathbf{D}_{\in i}$ is set to be $[\mathbf{0}, \ldots, \mathbf{D}_i, \ldots, \mathbf{0}]$ with K columns and the value $\mathbf{D}_{\notin i}$ is calculated as $\mathbf{D} - \mathbf{D}_{\in i}$. Term $\|\mathbf{Y}_i - \mathbf{D}\mathbf{X}_i\|_F^2$ represents the minimization of reconstruction error using dictionary $\mathbf{D}$. The terms $\|\mathbf{Y}_i - \mathbf{D}_{\in i}\mathbf{X}_i\|_F^2$ and $\|\mathbf{D}_{\notin i}\mathbf{X}_i\|_F^2$ are added to ensure that features from class i can be well reconstructed by atoms in the sub-dictionary $\mathbf{D}_i$ but not by other atoms belonging to different classes.

The group sparsity constraint is represented as $\lambda_1|\mathbf{x}_j^i|_1 + \lambda_2\|\mathbf{x}_j^i\|_2^2$, and the geometry constraint is represented as $\lambda_3 \sum_{i=1}^{J} \sum_{j=1}^{N} \|\alpha_i - \mathbf{x}_j\|_2^2 w_{ij}$. In the geometry constraint, elements in vector α_i are calculated coefficients for "template" feature $\mathbf{y}_i$. Here, templates are small sets of features randomly selected from all classes. In total, there are J templates used for similarity measure. Especially, coefficients α_i for the template $\mathbf{y}_i$ belonging to class m can be calculated by Eqs. 5 and 6:

$$\beta = \min_{\beta} \|\mathbf{y}_i - \mathbf{D}_m\beta\|_2^2 + \lambda_1|\beta|_1 + \lambda_2\|\beta\|_2^2 \quad (5)$$

$$\alpha_i = [\underbrace{\mathbf{0}}_{K_1}, \ldots, \underbrace{\mathbf{0}}_{K_{m-1}}, \beta, \underbrace{\mathbf{0}}_{K_{m+1}}, \ldots, \underbrace{\mathbf{0}}_{K_C}] \quad (6)$$

In α_i, only coefficients corresponding to the atoms from the same class m are nonzero. The weight w_{ij} between the query feature $\mathbf{y}_j$ and template feature $\mathbf{y}_i$ is defined as:

$$w_{ij} = exp(-\|\mathbf{y}_i - \mathbf{y}_j)\|_2^2/\sigma) \quad (7)$$

3.2.1 Optimization Step - Coefficients

The optimization problem in Eq. 4 can be iteratively solved by optimizing over $\mathbf{D}$ or $\mathbf{X}$ while fixing the other. After fixing the dictionary $\mathbf{D}$, the coefficients vector $\mathbf{x}_j^i$ can be calculated by solving the following convex problem (details are provided in the supplementary material):

$$\min_{\mathbf{x}_j^i} \left\{ \left\|\mathbf{s}_j^i - \mathbf{D}_i'\mathbf{x}_j^i\right\|_2^2 + \lambda_1|\mathbf{x}_j^i|_1 + \lambda_3 L(\mathbf{x}_j^i) \right\} \quad (8)$$

where

$$\mathbf{s}_j^i = [\mathbf{y}_j^i; \mathbf{y}_j^i; \underbrace{0; \ldots; 0}_{f+K}] \quad (9)$$

$$\mathbf{D}_i' = [\mathbf{D}; \mathbf{D}_{\in i}; \mathbf{D}_{\notin i}; \sqrt{\lambda_2}\mathbf{I}] \quad (10)$$

$$L(\mathbf{x}_j^i) = \sum_{m=1}^{A_i} \|\alpha_m - \mathbf{x}_j^i\|_2^2 w_{mj} \quad (11)$$

where $\mathbf{I} \in \mathbb{R}^{K \times K}$ is an identity matrix. Note that w_{mj} represents the weight between feature $\mathbf{y}_j^i$ and template $\mathbf{y}_m$ calculated by Eq. 7. To remove the influence of shared features among classes, we use templates belonging to the same class as the input feature for similarity measure at this stage. According to Eqs. 6 and 7, we know that term $L(\mathbf{x}_j^i)$ encourages the calculated coefficients to have zeros at atoms not from the same class as the input feature. In total, there are A_i templates used to calculate the unknown coefficient $\mathbf{x}_j^i$.

Since the analytical solution can be calculated for Eq. 8 if the sign of each element in $\mathbf{x}_j^i$ is known, the *feature-sign search method* [9] can be used to obtain the coefficients. However, the augmented matrix $\mathbf{D}_i'$ needs to be normalized before using the feature-sign search method. Let $\overline{\mathbf{D}_i'}$ be the l_2 column-wise normalized version of $\mathbf{D}_i'$. By simple derivations, we know that $\mathbf{D}_i' = (\sqrt{2+\lambda_2})\overline{\mathbf{D}_i'}$. Therefore, Eq. 8 can be rewritten as:

$$\min_{\overline{\mathbf{x}}_j^i} \left\{ \begin{array}{l} \left\|\mathbf{s}_j^i - \overline{\mathbf{D}_i'}\overline{\mathbf{x}}_j^i\right\|_2^2 + \frac{\lambda_1}{\sqrt{2+\lambda_2}}|\overline{\mathbf{x}}_j^i|_1 + \\ \frac{\lambda_3}{2+\lambda_2} \sum_{m=1}^{A_i} \|\sqrt{2+\lambda_2}\alpha_m - \overline{\mathbf{x}}_j^i\|_2^2 w_{mj} \end{array} \right\} \quad (12)$$

where $\overline{\mathbf{x}}_j^i = \sqrt{2+\lambda_2}\mathbf{x}_j^i$. Therefore, the feature-sign search method can be applied to Eq. 12 to obtain $\overline{\mathbf{x}}_j^i$, and the coefficients for input feature $\mathbf{y}_j^i$ should be $\frac{1}{\sqrt{2+\lambda_2}}\overline{\mathbf{x}}_j^i$. The detailed derivations can be found in the supplementary material.

3.2.2 Optimization Step - Dictionary

Fixing the coefficients, atoms in the dictionary can be updated. In this paper, the sub-dictionaries are updated class by class. In other words, while updating the sub-dictionary $\mathbf{D}_i$, all the other sub-dictionaries will be fixed. Terms that are independent of the current sub-dictionary can then be omitted from optimization, and the objective function when updating the sub-dictionary $\mathbf{D}_i$ can be given as:

$$\min_{\mathbf{D}_i} \{\|\mathbf{Y}_i - \mathbf{D}\mathbf{X}_i\|_F^2 + \|\mathbf{Y}_i - \mathbf{D}_{\in i}\mathbf{X}_i\|_F^2\} \quad (13)$$

To solve Eq. 13, atoms in the sub-dictionary $\mathbf{D}_i$ are updated one by one. Let $\mathbf{d}_k^i$ be the k-th atom in the sub-dictionary $\mathbf{D}_i$. When updating atom $\mathbf{d}_k^i$, all the rest atoms in $\mathbf{D}$ are fixed, and the first derivative of Eq. 13 over $\mathbf{d}_k^i$ can be represented as:

$$\nabla(f(\mathbf{d}_k^i)) = (-4\mathbf{Y}_i + 2\mathbf{M}\mathbf{X}_i + 4\mathbf{d}_k^i\mathbf{x}_{i(k)})\mathbf{x}_{i(k)}^T \quad (14)$$

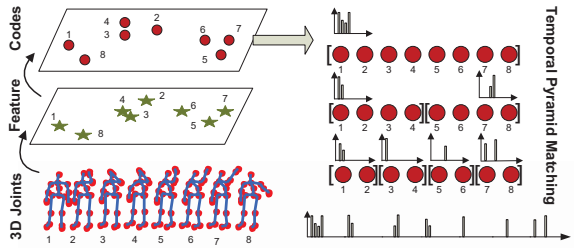

Figure 3. Temporal pyramid matching based on sparse coding.

where $\mathbf{x}_{i(k)}$ is the r-th row ($r = \sum_{j=1}^{i-1} K_j + k$) in matrix $\mathbf{X}_i \in \mathbb{R}^{K \times N_i}$, and it is corresponding to the coefficients contributed by the atom $\mathbf{d}_k^i$. Matrix $\mathbf{M}$ is of the same size as $\mathbf{D}$ and is equal to $\mathbf{M}_1 + \mathbf{M}_2$. Here $\mathbf{M}_1$ is the matrix after replacing the r-th column in $\mathbf{D}$ with zeros, and $\mathbf{M}_2$ is the matrix after replacing the r-th column with zeros in $\mathbf{D}_{\in i}$. The updated atom $\mathbf{d}_k^i$ can be calculated by setting Eq. 14 to zero, which is:

$$\mathbf{d}_k^i = (\mathbf{Y}_i - 0.5\mathbf{M}\mathbf{X}_i)\,\mathbf{x}_{i(k)}^T / \|\mathbf{x}_{i(k)}\|_2^2 \qquad (15)$$

3.3. Representation and Classification

After constructing the discriminative dictionary $\mathbf{D}$, the coefficients for a given feature $\mathbf{y}$ can be calculated by solving the following optimization problem:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda_1 |\mathbf{x}|_1 + \lambda_2 \|\mathbf{x}\|_2^2 + \lambda_3 \sum_{i=1}^{J} \|\alpha_i - \mathbf{x}\|_2^2 w_i \qquad (16)$$

Similar to the derivation in Sec. 3.2.1, the feature-sign search method [9] can be used to obtain the coefficients.

To keep the temporal information during the feature representation, a temporal pyramid matching (TPM) based on a pooling function $\mathbf{z} = \mathcal{F}(\mathbf{X})$ is used to yield the histogram representation for every depth sequence. In this paper, the max pooling is selected as many literature work did [20, 22]. TPM divides the video sequence into several segments along the temporal direction. Histograms generated from segments by max pooling are concatenated to form the representation, as shown in Figure 3. In this paper, the depth sequence is divided into 3 levels with each containing 1, 2 and 4 segments, respectively.

To speed up the process of training and testing, a linear SVM classifier [22] is used on the calculated histogram.

4. Experiments

Two benchmark datasets, *MSR-Action3D dataset* [10] and *MSR DailyActivity3D dataset* [19], are used for evaluation purpose. For both datasets, we compare the performance from two aspects, the effectiveness of the proposed framework (i.e., DL-GSGC+TPM) as compared to state-of-the-art approaches and the effectiveness of the proposed dictionary learning algorithm (i.e., DL-GSGC) as compared to state-of-the-art DL methods. In addition, since the second dataset also contains the RGB video sequence, we further compare the performance between using the RGB sequence and the depth map sequence. In all experiments, the proposed approaches constantly outperform the state-of-the-art.

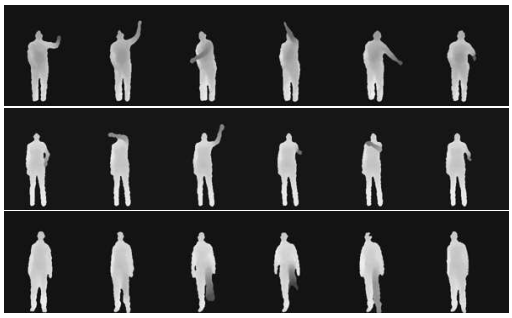

Figure 4. Sample frames from the MSR Action3D dataset. From top to bottom, frames are respectively from actions: Draw X, Draw Circle, and Forward Kick.

4.1. Parameters Setting

For the DL-GSGC dictionary learning, there are three parameters: λ_1, λ_2 and λ_3 that corresponding to group sparsity and geometry constraints, respectively. According to our observation, the performance is best when $\lambda_1 = 0.1 \sim 0.2$, $\lambda_2 = 0.01 \sim 0.02$ and $\lambda_3 = 0.1 \sim 0.2$. Initial sub-dictionaries are obtained by solving $\|\mathbf{Y}_i - \mathbf{D}_i\mathbf{X}_i\|_F^2 + \lambda_1 \sum_{j=1}^{N_i} |\mathbf{x}_j^i|_1 + \lambda_2 \|\mathbf{X}_i\|_F^2$ using online dictionary learning [12] and the number of atoms is set to be 15 for each sub-dictionary. For geometry constraint, 1500 features are used to build the templates. Note that all these features are collected from a subset of training samples, and cover all the classes. Compared to the total number of training features, the number of templates is relatively small.

4.2. MSR Action3D Dataset

The MSR-Action3D dataset [10] contains 567 depth map sequences. There are 20 actions performed by 10 subjects. For each action, the same subject performs it three times. The size of the depth map is 640×480. Figure 4 shows the depth sequences of three actions: *draw x*, *draw circle*, and *forward kick*, performed by different subjects. For all experiments on this dataset, the 1500 templates used for geometry constraint are collected from two training subjects.

4.2.1 Compared with State-of-the-art Algorithms

We first evaluate the proposed algorithm (**DL-GSGC + TPM**) in terms of recognition rate and compare it with the state-of-the-art algorithms that have been applied on the MSR Action3D dataset. For fair comparison, all results are

Method	Accuracy
Recurrent Neural Network [13]	42.5%
Dynamic Temporal Warping [14]	54.0%
Hidden Markov Model [11]	63.0%
Bag of 3D Points [10]	74.7%
Histogram of 3D Joints [21]	78.97%
Eigenjoints [24]	82.3%
STOP Feature [17]	84.8%
Random Occupy Pattern [18]	86.2%
Actionlet Ensemble [19]	88.2%
DL-GSGC+TPM	**96.7%**
DL-GSGC+TPM($\lambda_2 = 0$)	**95.2%**
DL-GSGC+TPM($\lambda_3 = 0$)	**94.2%**

Table 1. Evaluation of algorithms on the cross subject test for the MSRAction3D dataset.

obtained using the same experimental setting: 5 subjects are used for training and the rest 5 subjects are used for testing. In other words, it is a cross-subject test. Since subjects are free to choose their own styles to perform actions, there are large variations among training and testing features.

Table 1 shows the experimental results by various algorithms. Our proposed method achieves the highest recognition accuracy as 96.7%, and accuracies reduced to 95.2% and 94.2% if only one constraint is kept. Note that the work of [19] required a feature selection process on 3D joint features and a multiple kernal learning process based on the SVM classifier to achieve the accuracy of 88.2%, whereas our algorithm use simple 3D joint feature as described in Sec. 3.1, combined with the proposed feature representation and a simple linear SVM classifier. Therefore, the proposed dictionary learning method and framework is effective for the task of depth-based human action recognition.

Figure 5 shows the confusion matrix of the proposed method. Actions of high similarity get relative low accuracies. For example, action *Draw Tick* tends to be confused with *Draw X*.

4.2.2 Comparison with Sparse Coding Algorithms

To evaluate the performance of the proposed **DL-GSGC**, classic DL methods are used for comparison. These methods include K-SVD [1], sparse coding used for image classification based on spatial pyramid matching (ScSPM) [22], and the dictionary learning with structured incoherence (DLSI) [15]. In addition, for all the evaluated DL methods, the feature-sign search method is used for coefficients calculation, the TPM and max pooling are used to obtain the vector representation, and the linear SVM classifier is used for classification. We refer to the corresponding algorithms as K-SVD, ScTPM and DLSI for simplicity.

Comparisons are conducted on three subsets from the MSR Action3D dataset, as described in [10]. For each subset, 8 actions are included. All the subsets(AS1, AS2 and AS3) are deliberately constructed such that similar movements are included within the group while A3 further contains complex actions with large and complicated body movements. On each subset, three tests are performed by choosing different training and testing samples. Since each subject will perform the same action 3 times, Test1 and Test2 choose $1/3$ and $2/3$ samples for training respectively. Test3 uses the cross subjects setting, which is the same as described in Sec. 4.2.1. Compared with Test1 and Test2, Test3 is more challenging since the variations are larger between training and testing samples.

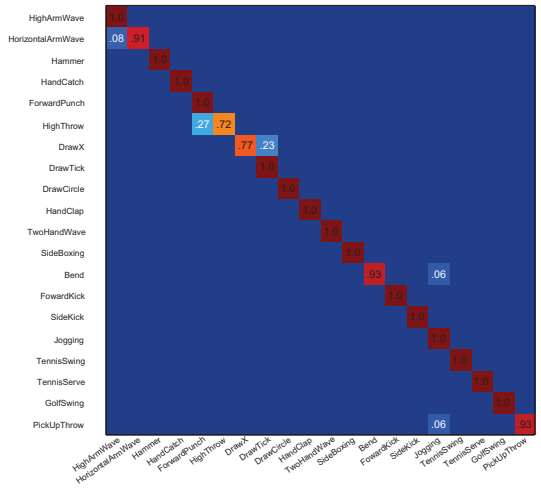

Figure 5. Confusion matrix for MSR Action3D dataset.

Table 2 shows the results on the three subsets. Note that the overall accuracies based on all actions (20 actions) are also provided for each test. It shows that the performance of **DL-GSGC** is superior to other sparse coding algorithms in terms of accuracies on all tests. In addition, class-specific dictionary learning methods, such as **DL-GSGC** and DLSI, perform better than methods learning a whole dictionary simultaneously for all classes (*e.g.*, K-SVD and ScTPM). Moreover, the proposed framework (*i.e.*, sparse coding + TPM), is effective for action recognition, since accuracies when using different sparse coding methods outperform the literature work in both Tables 1 and 2. Especially, our method outperforms other algorithms in Table 1 based on 3D joint features by $15\% \sim 17\%$ on test 3.

4.3. MSR DailyActivity3D Dataset

The MSR DailyActivity3D dataset contains 16 daily activities captured by a Kinect device. There are 10 subjects in this dataset, and each subject performs the same action twice, once in standing position, and once in sitting position. In total, there are 320 samples with both depth maps and RGB sequences available. Figure 6 shows the sample frames for the activities: *drink*, *write* and *stand up*, from

Method (%)	Test 1				Test 2				Cross Subjects Test			
	AS1	AS2	AS3	Overall	AS1	AS2	AS3	Overall	AS1	AS2	AS3	Overall
[10]	89.5	89.0	96.3	91.6	93.4	92.9	96.3	94.2	72.9	71.9	79.2	74.7
[21]	98.5	96.7	93.5	96.2	98.6	97.9	94.9	97.2	87.9	85.5	63.5	79.0
[24]	94.7	95.4	97.3	95.8	97.3	98.7	97.3	97.8	74.5	76.1	96.4	82.3
K-SVD	98.8	95.6	98.8	97.8	100	98.0	100	98.9	92.4	91.9	95.5	92.0
ScTPM	98.8	95.6	98.8	97.3	100	98.0	100	98.9	96.2	92.9	96.4	92.7
DLSI	97.4	98.1	99.4	97.6	98.8	97.2	100	97.9	96.6	93.7	96.4	93.2
DL-GSGC	**100**	**98.7**	**100**	**98.9**	**100**	**98.7**	**100**	**98.9**	**97.2**	**95.5**	**99.1**	**96.7**

Table 2. Performance evaluation of sparse coding based algorithms on three subsets.

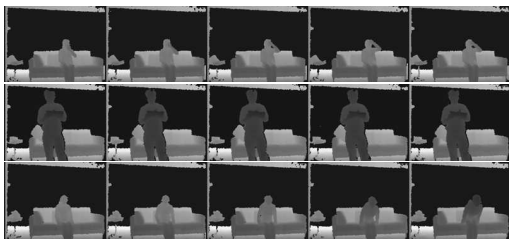

Figure 6. Sample frames for the MSR DailyActivity3D dataset. From top to bottom, frames are from actions: drink, write, and stand up.

top to bottom. As shown in Figure 6, some activities in this dataset contain small body movements, such as *drink* and *write*. In addition, the same activity performed in different positions have large variations in the estimated 3D joint positions. Therefore, this dataset is more challenging than the MSR Action3D dataset. Experiments performed on this dataset is based on cross subjects test. In other words, 5 subjects are used for training, and the rest 5 subjects are used for testing. The number of templates is also 1500 which are collected from 2 training subjects. Table 3 shows the experimental results by using various algorithms.

4.3.1 Comparison with State-of-the-art Algorithms

We first compare the performance of **DL-GSGC** with literature work that have been conducted on this dataset. As shown in Table 3, the proposed method outperforms the state-of-the-art work [19] by 10% and the geometry constraint is more effective for performance improvement. In addition, other DL methods are incorporated in our framework for comparison, referred to as K-SVD, ScTPM and DLSI. Experimental results show that the performance of **DL-GSGC** is superior to other DL methods by $4\% \sim 5\%$. In addition, class-specific dictionary learning methods, *e.g.*, DL-GSGC and DLSI, are better for classification task than K-SVD and ScTPM. Moreover, the proposed framework outperforms the state-of-the-art work [19] by $5\% \sim 10\%$ when different DL methods are used. Considering the large intra-class variations and noisy 3D joint positions in this dataset, the proposed framework is quite robust.

Method	Accuracy
Cuboid+HoG*	53.13%
Harris3D+HOG/HOF*	56.25%
Dynamic Temporal Wrapping [14]	54%
3D Joints Fourier [19]	68%
Actionlet Ensemble [19]	85.75%
K-SVD	90.6%
ScTPM	90.6%
DLSI	91.3%
DL-GSGC	**95.0%**
DL-GSGC ($\lambda_2 = 0$)	**93.8%**
DL-GSGC ($\lambda_3 = 0$)	**92.5%**

Table 3. Performance evaluation of the proposed algorithm with eight algorithms. Algorithms marked with (*) are applied on RGB videos and all rest algorithms are applied on depth sequences.

4.3.2 Comparison with RGB Features

Since both depth and RGB videos are available in this dataset, we also compare the performance of RGB features with that of depth features. For traditional human action recognition problem, spatio-temporal interest points based methods have been heavily explored. Two important steps are spatio-temporal interest point detection and local feature description. As for feature representation, Bag-of-Words representation based on K-means clustering is widely used. In this paper, we follow the same steps to perform action recognition from RGB videos. To be specific, the classic Cuboid [3] and Harris3D [8] detectors are used for feature detection, and the HOG/HOF descriptors are used for description. The Bag-of-Words representation is used for feature representation.

Table 3 provides the recognition rates by using different feature detectors and descriptors on RGB video sequences. Compared with the performance of depth features, recognition rates on RGB sequences are lower. We argue the main reason to be that this dataset contains many actions with high similarity but small body movements, *e.g.*, *Drink, Eat,*

Write, Readbook. In this case, the 3D joint features containing depth information are more reliable than RGB features. In addition, the K-mean clustering method will cause larger quantization error than sparse coding algorithms. Therefore, depth information is important for the task of action recognition, and the sparse coding based representation is better for quantization.

5. Conclusion

This paper presented a new framework to perform human action recognition on depth sequences. To better represent the 3D joint features, a new discriminative dictionary learning algorithm (**DL-GSGC**) that incorporated both group sparsity and geometry constraints was proposed. In addition, the temporal pyramid matching method was applied on each depth sequence to keep the temporal information in the representation. Experimental results showed that the proposed framework is effective that outperformed the state-of-the-art algorithms on two benchmark datasets. Moreover, the performance of DL-GSGC is superior to classic sparse coding methods. Although the DL-GSGC is proposed for dictionary learning in the task of depth-based action recognition, it is applicable to other classification problems, such as image classification and face recognition.

6. Acknowledgement

This work was supported in part by National Science Foundation under Grant NSF CNS-1017156.

References

[1] M. Aharon, M. Elad, and A. M. Bruckstein. K-svd: An algorithm for designing overcomplete dicitonaries for sparse representation. *IEEE Trans. Signal Processing*, 54:4311–4322, 2006. 6

[2] H. Bondell and B. Reich. Simultaneous regression shrinkage, variable selection and supervised clustering of predictors with oscar. *Biometrics*, 64(1):115–123, 2008. 3

[3] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. *VS-PETS*, 2005. 7

[4] Y. Fang, R. Wang, and B. Dai. Graph-oriented learning via automatic group sparsity for data analysis. *IEEE 12th International Conference on Data Mining*, 2012. 2, 3

[5] S. Gao, I. Tshang, L. Chia, and P. Zhao. Local features are not lonely - laplacian sparse coding for image classification. *CVPR*, 2010. 3

[6] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. *CVPR*, 2011. 2

[7] S. Kong and D. Wang. A dictionary learning approach for classification: separating the particularity and the commonality. *ECCV*, 2012. 2, 3

[8] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. *CVPR*, 2008. 7

[9] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse code algorithms. *NIPS*, 2007. 4, 5

[10] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. *In Human communicative behavior analysis workshop (in conjunction with CVPR)*, 2010. 5, 6, 7

[11] F. Lv and R. Nevatia. Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. *ECCV*, pages 359–372, 2006. 2, 6

[12] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. *ICML*, 2009. 5

[13] J. Martens and I. Sutskever. Learning recurrent neural networks with hessian-free optimzation, 2011. 6

[14] M. Muller and T. Roder. Motion templates for automatic classification and retrieval of motion capture data. *In proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on compute animation*, pages 137–146, 2006. 2, 6, 7

[15] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. *CVPR*, 2010. 2, 3, 6

[16] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth cameras. *CVPR*, 2011. 1, 3

[17] A. W. Vieira, E. R. Nascimento, G. Oliveira, Z. Liu, and M. Campos. Stop: space-time occupancy patterns for 3d action recognition from depth map sequences. *17th Iberoamerican congress on pattern recognition*. 6

[18] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3d action recognition with random occupancy patterns. *ECCV*, 2012. 6

[19] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. *CVPR*, 2012. 1, 2, 3, 5, 6, 7

[20] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. *CVPR*, 2010. 5

[21] L. Xia, C. C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3d joints. *CVPR Workshop*, 2012. 2, 6, 7

[22] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. *CVPR*, 2009. 3, 5, 6

[23] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. *ICCV*, 2011. 2

[24] X. Yang and Y. Tian. Eigenjoints-based action recognition using naive bayes nearest neighbor. *CVPR 2012 HAU3D Workshop*, 2012. 2, 6, 7

[25] Q. Zhang and B. Li. Discriminative k-svd for dicitonary learning in face recognition. *CVPR*, 2010. 2

[26] H. Zou and H. Hastie. Regression and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67(2):301–320, 2005. 2, 3