

Real-Time Human Action Recognition Using DMMs-Based LBP and EOH Features

Mohammad Farhad Bulbul, Yunsheng Jiang, and Jinwen Ma^(✉)

Department of Information Science,
School of Mathematical Sciences and LMAM,
Peking University, Beijing 100871, China
jwma@math.pku.edu.cn

Abstract. This paper proposes a new feature extraction scheme for the real-time human action recognition from depth video sequences. First, three Depth Motion Maps (DMMs) are formed from the depth video. Then, on top of these DMMs, the Local Binary Patterns (LBPs) are calculated within overlapping blocks to capture the local texture information, and the Edge Oriented Histograms (EOHs) are computed within non-overlapping blocks to extract dense shape features. Finally, to increase the discriminatory power, the DMMs-based LBP and EOH features are fused in a systematic way to get the so-called **DLE features**. The proposed DLE features are then fed into the l_2 -regularized Collaborative Representation Classifier (l_2 -CRC) to learn the model of human action. Experimental results on the publicly available Microsoft Research Action3D dataset demonstrate that the proposed approach achieves the state-of-the-art recognition performance without compromising the processing speed for all the key steps, and thus shows the suitability for real-time implementation.

Keywords: Human action recognition · Depth Motion Maps · Local Binary patterns · Edge Oriented Histograms

1 Introduction

In the area of computer vision, human action recognition is a process of detecting and labeling the humans in the video, and it has drawn much attention to researchers due to the growing demands from vast applications, such as visual surveillance, video retrieval, health monitoring, fitness training, human-computer interaction and so on (*e.g.*, [1, 2]). In the past decade, research has mainly focused on learning and recognizing actions from image sequences captured by traditional RGB video cameras [3, 4]. However, these types of data source have some inherent limitations. For example, they are sensitive to color and illumination changes, occlusions, and background clutters.

With the recent release of the cost-effective depth sensors, such as Microsoft Kinect, numerous research works on human action recognition have been carried out based on the depth maps. An example of depth map sequence is shown in Fig. 1. Compared with conventional RGB cameras, the depth camera has many advantages. For instance, the outputs of depth cameras are insensitive to changes in lighting

conditions. Moreover, depth maps can provide 3D structural information for distinguishing human actions, which is difficult to characterize by using RGB video sequences. On the other hand, human skeleton information can be extracted from depth maps [5]. Specifically, 3D positions and rotation angles of the body joints can be estimated by using the Kinect Windows SDK [6].

In this paper, we present a computationally efficient and effective human action



Fig. 1. A depth map sequence for the *High throw* action

recognition method by utilizing Depth Motion Maps (DMMs) based Local Binary Patterns (LBPs) and Edge Oriented Histograms (EOHs) to the l_2 -regularized Collaborative Representation Classifier (l_2 -CRC). Specifically, first, for each depth video, all its video frames are projected onto three orthogonal Cartesian planes to generate the projected maps corresponding to three projection views (*front*, *side* and *top*). For each projection view, the accumulation of absolute differences between consecutive projected maps forms the corresponding DMMs (i.e., DMM_f , DMM_s and DMM_t) [7]. Next, from these three DMMs, three LBP feature vectors are calculated with overlapping blocks, and three EOH feature vectors are calculated with non-overlapping blocks. After that, these six feature vectors are concatenated through a sequential way to form the so-called **DLE features** (since it contains DMMs-based LBP and EOH features). Finally, the dimension of DLE feature vector is reduced by Principal Component Analysis (PCA), and it is fed into the l_2 -CRC to recognize human actions.

The rest of this paper is organized as follows. Section 2 reviews the related work. The whole approach (including the DLE features and l_2 -CRC) is presented in Sect. 3. The experimental results are demonstrated in Sect. 4. Finally, Sect. 5 contains a brief conclusion of this work.

2 Related Work

In this section, we review the recent related work for human action recognition from depth video sequences, including low-level features and high-level skeleton information.

In the last few years, a lot of low-level features for recognizing human actions from depth video sequences have been introduced. In 2010, Li *et al.* [8] presented a framework to recognize human actions from sequences of depth maps. They employed an action graph to model the temporal dynamics of actions, and utilized a collection of 3D points to characterize postures. The loss of spatial context information between interest points and computational inefficiency were considered as limitations of this

approach. To improve recognition rates, Vieira *et al.* [9] introduced the Space Time Occupancy Patterns (STOP) feature descriptor. Furthermore, Wang *et al.* [10] considered 3-dimensional action sequences as 4-dimensional shapes and proposed Random Occupancy Pattern (ROP), and sparse coding was utilized to further improve the robustness of the proposed approach.

Following another way, Yang *et al.* [11] computed Depth Motion Maps (DMMs) based Histogram of Oriented Gradients (HOG) features and fed them into SVM classifier to recognize human actions. In 2013, Chen *et al.* [7] utilized the DMMs as feature descriptor and l_2 -CRC as classifier to recognize human actions. In that year, Oreifej and Liu [12] presented a new descriptor called histogram of oriented 4D surface normals (HON4D). To improve recognition accuracy, some researchers proposed methods based on features extracted from depth and RGB video sequences simultaneously. For instance, Luo *et al.* [13] extracted 3D joint features for each depth video, and utilized Centre-Symmetric Motion Local Ternary Pattern (CS-Mltp) to extract both the spatial and temporal features of the RGB sequences. Besides, some researchers are still working to improve the robustness as well as recognition rate of the action recognition methods. For example, binary range sample feature descriptor was proposed by Lu *et al.* [14]. In 2015, Chen *et al.* [15] proposed another DMMs method, where DMMs- based LBP features coupled with the Kernel-based Extreme Learning Machine (KELM) classifier was used to recognize human actions. Recently, inspired by the DMMs-based works in [7, 11, 15], Farhad *et al.* [16] proposed another method by designing an effective feature descriptor, called DMM-CT-HOG for short. More precisely, HOG was employed to DMMs-based contourlet sub-bands to compactly represent the body shape and movement information toward distinguishing actions.

There are also many skeleton based algorithms by utilizing high-level skeleton information extracted from depth maps. In 2012, Yang and Tian [17] proposed a human action recognition framework by using Eigen joints (position difference of points) and Naive-Bayes-Nearest-Neighbor (NBNN) classifier. In the same year, Xia *et al.* [18] proposed Histogram of 3D Joints Locations (HOJ3D) in depth maps. They applied Hidden Markov Model (HMM) to model the dynamics and action recognition. On the other hand, Wang *et al.* [19] utilized both skeleton and point cloud information and introduced an actionlet ensemble model to represent each action and capture the intra-class variance via occupancy information. In 2013, Luo *et al.* [20] proposed group sparsity and geometry-constraint dictionary learning (DL-GSGC) algorithm for recognizing human actions from skeleton data. Recently, body part based skeleton representation was introduced to characterize the 3D geometric relationships between different body parts by utilizing rotations and translation in [21]. In this approach, human actions were characterized as curves. In [22] the skeleton joint features were extracted from skeleton data and an evolutionary algorithm was utilized for feature selection. However, we have to point out that, though some of the skeleton-based methods show high recognition performance, they are not suitable in the case where skeleton information is not available.

3 Our Approach

In this section, DMMs and LBP are briefly reviewed, and our proposed DLE features (DMMs-based LBP and EOH features) are then introduced.

3.1 DMMs Construction

In the feature extraction stage, DMMs for each depth video sequence are first computed by using approaches demonstrated in [7]. Specifically, given a depth video sequence with K depth maps, each depth map is projected onto three orthogonal Cartesian planes (from front, side and top views) to get three projected maps. The accumulation of absolute differences between consecutive projected maps from front view constructs a DMM, which is referred to as DMM_f . Similarly, DMM_s and DMM_t can also be constructed for side and top views. The following equation is used to form each DMM:

$$DMM_v = \sum_{j=1}^{K-1} |map_v^{j+1} - map_v^j|, \quad (1)$$

where j is the frame index, $v \in \{f, s, t\}$ denotes the projection view, and map_v stands for the projected map. Figure 2 illustrates an example of DMMs for a High throw action video sequence.

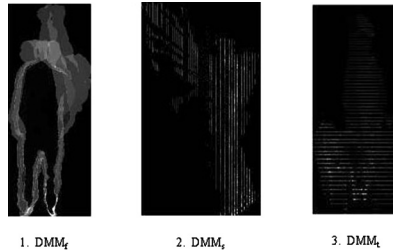


Fig. 2. DMMs for a *High throw* action video sequence

3.2 Overview of LBP

Local Binary Pattern (LBP) operator [23] is a powerful tool for texture description, and it has become very popular in various applications due to its discriminative power and computational simplicity. The original version of LBP operator works in 3×3 pixel blocks. All circumjacent pixels in the working block are thresholded by the center pixel and weighted by powers of 2 and then summed to label the center pixel. The LBP operator can also be extended to neighborhood with different sizes (see Fig. 3) [23]. To do this, consider a circular neighborhood denoted by (N, R) , where N is the number of sampling points and R represents the radius of the circle. These sampling points lie around the center pixel (x, y) and at coordinates $(x_i, y_i) = (x + R\cos(2\pi i/N), y - R\sin(2\pi i/N))$. If a sampling point does not fall at integer coordinates, then the

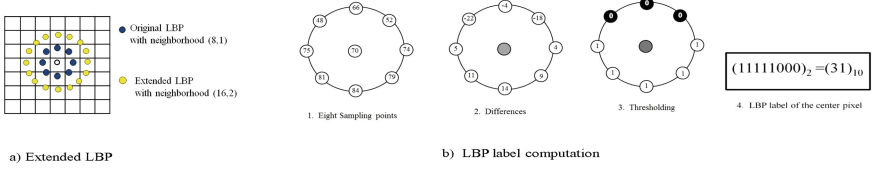


Fig. 3. Extended LBP and LBP label computation for the neighborhood (8, 1)

pixel value is bilinearly interpolated. Specifically, the LBP label for pixel (x, y) can be calculated as follows:

$$LBP_{N,R}(x, y) = \sum_{i=0}^{N-1} Th(f(x_i, y_i) - f(x, y)) \cdot 2^i, \quad (2)$$

where $Th(x) = 1$ if $(x \geq 0)$ and $Th(x) = 0$ if $x < 0$. An example of LBP label computation is shown in Fig. 3.

Another extension to the original operator is so called *uniform patterns* [23]. A local binary pattern is considered as uniform if the binary pattern has at most 2 bitwise transitions from 0 to 1 or vice versa when the bit pattern is in circular form. For example, the patterns 00000000 (0 transitions) and 01110000 (2 transitions) are uniform whereas the patterns 11001001 (4 transitions) and 01010011 (6 transitions) are not uniform pattern.

After calculating the LBP codes for all pixels in an image, an occurrence histogram is computed for the image or an image region to represent the texture information. Though the original LBP operator in Eq. 2 provides 2^N different binary patterns, only uniform patterns are sufficient to describe image texture [23]. Therefore, only uniform patterns are used to compute the occurrence histogram, *i.e.*, the histogram has an independent bin for each uniform pattern.

3.3 DLE Features Extraction

To construct DLE features, we use the computed DMMs, LBPs and EOHs with the following three stages.

In the first stage, LBP coded images corresponding to DMM_f , DMM_s and DMM_t are divided into 4×2 , 4×3 and 3×2 overlapped regions respectively [15]. For all the DMMs, histograms are computed for each block by setting 50 % overlap between two consecutive blocks. Finally, all histograms of all the blocks are merged to compute three feature vectors for the three DMMs. The LBP feature vectors for DMM_f , DMM_s and DMM_t are referred as LBP_f , LBP_s and LBP_t .

In the second stage, to compute EOHs [24], each DMM is first split into 4×4 non-overlapping rectangular regions. In each region, 4 directional edges (horizontal, vertical, 2 diagonals) and 1 non-directional edge are computed using Canny edge detector. Histograms are then computed for each block. The concatenation of all histograms for all regions yields a feature vector for the DMM. Thus, three feature vectors

for the three DMMs are generated for each action video sequence. An example of EOHs computation is shown in Fig. 4, where DMM_f of *Horizontal wave* action video sequence is set as an image. The EOH feature vectors for DMM_f , DMM_s and DMM_t are labeled as EOH_f , EOH_s and EOH_t .

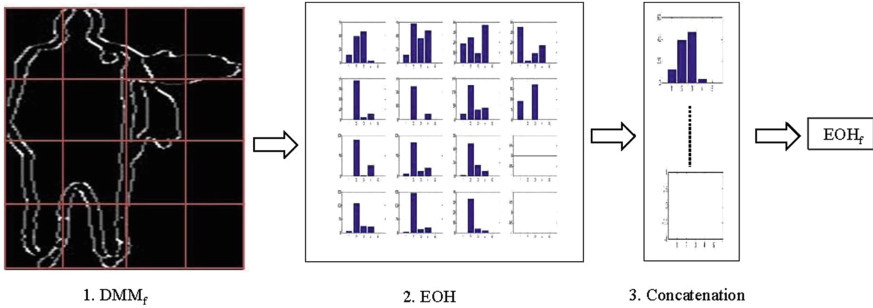


Fig. 4. EOH computation for DMM_f of *Horizontal wave* action sequence

In the third stage, the computed EOH_f and LBP_f are fused to get a feature vector that describes texture and edge features compactly for DMM_f and this feature vector is mentioned as $Feat_f$. In the same way, $Feat_s$ and $Feat_t$ can be calculated and their concatenation with $Feat_f$ provides a feature vector for the relevant action video sequence. Figure 5 illustrates our feature extraction approach for the *High throw* action video sequence.

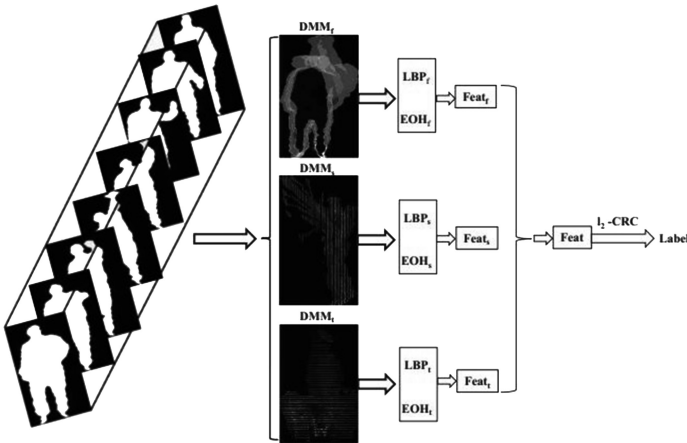


Fig. 5. Architecture of the proposed action recognition method

3.4 L_2 -CRC

Motivated by the success of l_2 -CRC in human action recognition [7, 16], we use this classifier to model human actions. For a better explanation of the l_2 -CRC, let us

consider a data set with C classes. By arranging the training samples column wisely, we can obtain an over-complete dictionary $B = [B_1, B_2, \dots, B_C] = [b_1, b_2, \dots, b_n \in R^{d \times n}$, where d is the dimensionality of samples, n is the total number of training samples, $\mathbf{B}_j \in R^{d \times m_j}$, ($j = 1, 2, \dots, C$) is the subset of the training samples belonging to the j^{th} class and $\mathbf{b}_i \in R^d$ ($i = 1, 2, \dots, n$) is the single training sample.

Let us express any unknown sample $\mathbf{V} \in R^d$ using matrix \mathbf{B} as follows:

$$v = \mathbf{B}\gamma, \tag{3}$$

Here γ is a $n \times 1$ vector associated with coefficients corresponding to the training samples. Practically, one cannot solve Eq. 3 directly as it's typically underdetermined [25]. Usually we obtain the solution by solving the following norm minimization problem:

$$\hat{\gamma} = \arg \min_{\gamma} \{ \| \mathbf{v} - \mathbf{B}\gamma \|_2^2 + \alpha \| \mathbf{M}\gamma \|_2^2 \}, \tag{4}$$

where \mathbf{M} stands for the Tikhonov regularization matrix [26] and α represents the regularization parameter. The term involved with \mathbf{M} permits the imposition of prior knowledge of the solution by utilizing the approach described in [27–29], where the training samples that are highly dissimilar from a test sample are provided less weight than the training samples that are highly similar. Specifically, the form of the matrix $\mathbf{M} \in R^{d \times n}$ is considered as follows:

$$\mathbf{M} = \begin{pmatrix} \| \mathbf{v} - \mathbf{b}_1 \|_2 & 0 & \dots & 0 \\ 0 & \| \mathbf{v} - \mathbf{b}_n \|_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \| \mathbf{v} - \mathbf{b}_n \|_2 \end{pmatrix} \tag{5}$$

According to [30] the coefficient vector $\hat{\gamma}$ is calculated as follows:

$$\hat{\gamma} = (\mathbf{B}^T \mathbf{B} + \alpha \mathbf{M}^T \mathbf{M})^{-1} \mathbf{B}^T \mathbf{V}. \tag{6}$$

By using the class labels of all the training samples, $\hat{\gamma}$ can be partitioned into C subsets $\hat{\gamma} = [\hat{\gamma}_1; \hat{\gamma}_2; \dots; \hat{\gamma}_C]$ with $\hat{\gamma}_j$ ($j = 1, 2, 3, \dots, C$). After partitioning $\hat{\gamma}$ the class label of the unknown sample \mathbf{v} is then calculated as follows:

$$class(\mathbf{v}) = \arg \min_{j \in \{1, 2, \dots, C\}} \{ e_j \}, \tag{7}$$

where $e_j = \| \mathbf{v} - \mathbf{B}_j \hat{\gamma}_j \|_2$.

4 Experimental Results

In this section, we first evaluate our proposed method on publicly available MSR-Action3D dataset and then compare our recognition results with other methods.

4.1 MSR-Action3D Dataset and Setup

MSR-Action3D dataset [8] contains 20 actions, where each action is performed by 10 different subjects 2 or 3 times facing the RGB-D camera. The list of 20 actions is: *high wave, horizontal wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, and pickup throw*. This dataset is a challenging dataset as it contains many actions with similar appearance. In order to have a fair evaluation of the proposed method, we follow the same experimental settings as described in [8]. Specifically, we divide the 20 actions into three action subsets (AS1, AS2 and AS3), which are shown in Table 1. For each action subset, five subjects (1, 3, 5, 7 and 9) are used for training and the rest for testing. Usually, this type of experimental setup is known as **cross subject test**.

Table 1. Three subsets of the MSR-Action3D dataset

Label	Action set 1 (AS1)	Label	Action set 2 (AS2)	Label	Action set 3 (AS3)
2	Horizontal wave	1	High wave	6	High throw
3	Hammer	4	Hand catch	14	Forward kick
5	Forward punch	7	Draw x	15	Side kick
6	High throw	8	Draw tick	16	Jogging
10	Hand clap	9	Draw circle	17	Tennis swing
13	Bend	11	Two hand wave	18	Tennis serve
18	Tennis serve	14	Forward kick	19	Golf swing
20	Pickup throw	12	Side boxing	20	Pickup throw

In all the experiments, for each action video sequence, the first/last four frames are removed due to two reasons. Firstly, at the beginning/end of an action video sequence, the subjects are mostly at stand-static position with a small body movement, which is not necessary for the motion characteristics of the involved action. Secondly, in our approach of computing *DMMs*, small movements at the beginning/end result in a stand-still body shape with large pixel values along the *DMM* edges, which leads to a large amount of recognition error.

To find an appropriate value for the parameter N (number of sampling points) and R (radius) in LBP computation, we carry out experiments on different values of (N, R) . Specifically, for each value of R in $\{1, 2, \dots, 6\}$, we select four values $\{4, 6, 8, 10\}$ for N . We observe the promising result at pair $(4, 1)$. Moreover, the computational complexity of the uniform LBP features depends on the number of sampling points (*i.e.*, N), because the dimensionality of the LBP histogram feature based on uniform patterns is $N(N - 1) + 3$. [23] Since the pair $(4, 1)$ makes low computational complexity and high recognition accuracy, we set $N = 4$ and $R = 1$ for the entire experiment. Besides, in the *l2-CRC*, the key parameter α is set as $\alpha = 0.0001$ according to the five-fold cross-validation. In order to improve the computational efficiency in the classification step, Principle Component analysis (PCA) is utilized to reduce the dimension of the *DLE* feature vector. The PCA transform matrix is calculated using the training feature

set and then applied to the test feature set. The principle components that account for 99 % of the total variation are retained.

4.2 Comparison with Other Methods

We compare the performance of our proposed method with several other competitive methods that were conducted on MSR-Action3D dataset with the same experimental setup. The comparison of the average recognition accuracy is shown in Table 2. It can be seen that, the recognition result of our method outperforms all the other methods listed in the table. Overall, our recognition result indicates that the fusion of DMMs-based LBP and EOH features obtains higher discriminatory power. On the other hand, the block-based dense features generated by LBPs and EOHs provides effective texture and edge information. Figure 6 shows the confusion matrices for the three action subsets separately. The confusion matrices state that actions with high similarities get relatively low accuracies. For example, action *draw x* is confused with *draw tick* due to the similarities of their *DMMs* and therefore action *draw x* is classified with low accuracy (see confusion matrix for the action subset AS2). The assigned label for each action (see Table 1) is used as axes label in the matrices to understand the classification accuracy and error of the corresponding action.

Table 2. Comparison of average recognition accuracies (%) corresponding to the three action subsets for the *cross subject test*

Method	Average accuracy (%)
Li <i>et al.</i> [8]	74.7
Yang <i>et al.</i> [11]	91.6
Chen <i>et al.</i> [7]	90.5
Farhad <i>et al.</i> [16]	92.3
Xia <i>et al.</i> [18]	79.0
Charaoui <i>et al.</i> [22]	93.2
Vemulapalli <i>et al.</i> [21]	92.5
Chen <i>et al.</i> [15]	94.9
Luo <i>et al.</i> [13]	93.8
Ours	95.8

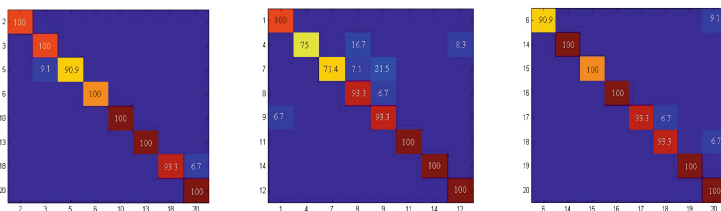


Fig. 6. Confusion matrices for the subset AS1, AS2 and AS3 (from left to right)

Table 3. Processing times (*mean* \pm *STD*) associated with different components of our framework

Components	Average processing time (ms)
DMMs calculation	$2.8 \pm 0.34/\text{frame}$
EOH feature extraction	$16.20 \pm 6.40/\text{action sample}$
LBP feature extraction	$12.0 \pm 0.15/\text{action sample}$
Dimensionality reduction (PCA)	$0.13 \pm 0.05/\text{action sample}$
l_2 -CRC	$0.41 \pm 0.20/\text{action sample}$

4.3 Computational Time

There are five main components in our human action recognition framework: DMMs calculation, EOHs computation, and LBPs extraction, dimensionality reduction (PCA) and action recognition (l_2 -CRC). Table 3 shows the average processing times of the five components for each depth video sequence with 33 frames (in our experiments each video sequence has 33 frames on average). Our code is written with MATLAB and the processing time is obtained on a PC with 3.20 GHz Intel Core i5-3470 CPU. Noted that, according to the processing time, the proposed method can process 30 frames per second, and thus it is compatible for the real-time operation.

5 Conclusion

This paper has presented a computationally efficient and effective human action recognition method for the depth video sequences by applying the fused version of DMMs-based LBP and EOH features to the l_2 -CRC classifier. Experimental results on the public domain datasets have revealed that our method provides higher action recognition accuracy compared to the existing methods and allows to recognize actions in real-time.

Acknowledgment. This work was supported by the Natural Science Foundation of China for Grant 61171138

References

1. Chen, C., Kehtarnavaz, N., Jafari, R.: A medication adherence monitoring system for pill bottles based on a wearable inertial sensor. In: Proceedings of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. pp. 4983–4986 (2014)
2. Chen, C., Jafari, R., Kehtarnavaz, N.: Improving human action recognition using fusion of depth camera and inertial sensors. *IEEE Trans. Hum.-Mach. Syst.* **45**(1), 51–61 (2015)
3. Laptev, I.: On space-time interest points. *Int. J. Comput. Vision* **64**(2/3), 107–123 (2005)
4. Niebles, J., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. *Int. J. Comput. Vision* **79**(3), 299–318 (2008)

5. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1297–1304 (2011)
6. Microsoft: kinect for windows. <http://www.microsoft.com/en-us/kinectforwindows/>
7. Chen, C., Liu, K., Kehtarnavaz, N.: Real-time human action recognition based on depth motion maps. *J. Real-Time Image Process.* (2013)
8. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d Points. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 9–14 (2010)
9. Vieira, A., Nascimento, E., Oliveira, G., Liu, Z., Campos, M.: Space-time occupancy patterns for 3d action recognition from depth map sequences. In: Proceedings of Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, pp. 252–259 (2012)
10. Wang, J., Liu, Z., Chorowski, J., Chen, Z., Wu, Y.: Robust 3d action recognition with random occupancy patterns. In: Proceedings of European Conference on Computer Vision, pp. 872–885 (2012)
11. Yang, X., Zhang, C., Tian, Y.: Recognizing actions using depth motion maps-based histograms of oriented gradients. In: Proceedings of ACM International Conference on Multimedia, pp. 1057–1060 (2012)
12. Oreifej, O., Liu, Z.: Hon4d: histogram of oriented 4d normals for activity recognition from depth sequences. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 716–723 (2013)
13. Luo, J., Wang, W., Qi, H.: Spatio-temporal feature extraction and representation for RGB-D human action recognition. *Pattern Recogn. Lett.* **50**, 139–148 (2014)
14. Lu, C., Jia, J., Tang, C.K.: Range-sample depth feature for action recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 772–779 (2014)
15. Chen, C., Jafari, R., Kehtarnavaz, N.: Action recognition from depth sequences using depth motion maps-based local binary patterns. In: Proceedings of IEEE Winter Conference on Applications of Computer Vision, pp. 1092–1099 (2015)
16. Farhad, M., Jiang, Y., Ma, J.: Human action recognition based on DMMs, HOGs and Contourlet transform. In: Proceedings of IEEE International Conference on Multimedia Big Data, pp. 389–394 (2015)
17. Yang, X., Tian, Y.: Eigen joints-based action recognition using Naive-Bayes-Nearest-Neighbor. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 14–19 (2012)
18. Xia, L., Chen, C., Aggarwal, J.: View invariant human action recognition using histograms of 3d joints. In: Proceedings of Workshop on Human Activity Understanding from 3D Data, pp. 20–27 (2012)
19. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1290–1297 (2012)
20. Luo, J., Wang, W., Qi, H.: Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In: Proceedings of the 14th IEEE International Conference on Computer Vision, pp. 1809–1816 (2013)
21. Vemulapalli, R., Arrate, F., Chellappa, R.: Human action recognition by representing 3d human skeletons as points in a lie group. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 588–595 (2014)

22. Chaaaraoui, A.A., Padilla-Lopez, J.R., Climent-Perez, P., Florez-Revuelta, F.: Evolutionary joint selection to improve human action recognition with RGB-D devices. *Expert Syst. Appl.* **41**(3), 786–794 (2014)
23. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture Classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7), 971–987 (2002)
24. <http://clickdamage.com/sourcecode/code/edgeOrientationHistogram.m>
25. Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T., Yan, S.: Sparse representation for computer vision and pattern recognition. *Proc. IEEE* **98**(6), 1031–1044 (2010)
26. Tikhonov, A., Arsenin, V.: Solutions of ill-posed problems. *Math. Comput.* **32**(144), 1320–1322 (1978)
27. Chen, C., Tramel, E.W., Fowler, J.E.: Compressed-sensing recovery of images and video using multi-hypothesis predictions. In: *Proceedings of the 45th Asilomar Conference on signals, Systems, and Computers*, pp. 1193–1198 (2011)
28. Chen, C., Li, W., Tramel, E.W., Fowler, J.E.: Reconstruction of hyperspectral imagery from random projections using multi-hypothesis prediction. *IEEE Trans. Geosci. Remote Sens.* **52**(1), 365–374 (2014)
29. Chen, C., Fowler, J.E.: Single-image Super-resolution Using Multi-hypothesis Prediction. In: *Proceedings of the 46th Asilomar Conference on Signals, Systems, and Computers*, pp. 608–612 (2012)
30. Golub, G., Hansen, P.C., O’Leary, D.: Tikhonov-regularization and total least squares. *SIAM J. Matrix Anal. Appl.* **21**(1), 185–194 (1999)