PLOS ONE

RESEARCH ARTICLE

Learning Dictionaries of Sparse Codes of 3D Movements of Body Joints for Real-Time Human Activity Understanding

Jin Qi, Zhiyong Yang*

Brain and Behavior Discovery Institute, James and Jean Culver Vision Discovery Institute, Department of Ophthalmology, Georgia Regents University, Augusta, Georgia, 30912, United States of America

*zhyang@gru.edu

Abstract

Real-time human activity recognition is essential for human-robot interactions for assisted healthy independent living. Most previous work in this area is performed on traditional two-dimensional (2D) videos and both global and local methods have been used. Since 2D videos are sensitive to changes of lighting condition, view angle, and scale, researchers begun to explore applications of 3D information in human activity understanding in recently years. Unfortunately, features that work well on 2D videos usually don't perform well on 3D videos and there is no consensus on what 3D features should be used. Here we propose a model of human activity recognition based on 3D movements of body joints. Our method has three steps, learning dictionaries of sparse codes of 3D movements of joints, sparse coding, and classification. In the first step, space-time volumes of 3D movements of body joints are obtained via dense sampling and independent component analysis is then performed to construct a dictionary of sparse codes for each activity. In the second step, the space-time volumes are projected to the dictionaries and a set of sparse histograms of the projection coefficients are constructed as feature representations of the activities. Finally, the sparse histograms are used as inputs to a support vector machine to recognize human activities. We tested this model on three databases of human activities and found that it outperforms the state-of-the-art algorithms. Thus, this model can be used for real-time human activity recognition in many applications.

Citation: Qi J, Yang Z (2014) Learning Dictionaries of Sparse Codes of 3D Movements of Body Joints for Real-Time Human Activity Understanding. PLoS ONE 9(12): e114147. doi:10.1371/journal.pone.0114147

Editor: Marco Cristani, University of Verona, Italy

Received: June 11, 2014

Accepted: November 3, 2014

Published: December 4, 2014

Data Availability: The authors confirm that all data underlying the findings are fully available without restriction. All relevant data are within the paper and its Supporting Information files.

Funding: The authors have no funding or support to report.

Competing Interests: The authors have declared that no competing interests exist.

Introduction

A smart environment is a place where humans and objects (including mobile robots) can interact and communicate with each other in a human-like way [1]. It has a wide range of applications in home and office work, health care, assistive living, and industrial operations. Current pervasive computing technologies and low-cost digital imaging devices make feasible the development of smart environments. In smart environments, accurate, real-time human activity recognition is a paramount requirement since it allows to monitor individuals/patient's activities of daily living [2], such as taking medicine, dressing, cooking, eating, drinking, falling down, and feeling painful, to keep track of their functional health, and to timely intervene to improve their health [3–7]. Fig. 1 shows several human activities in the dataset CAD-60 [8], including "wearing contact lens","talking on the phone", "brushing teeth" and "writing on the white board".

Automated human activity understanding is a challenging problem due to the diversity and complexity of human behaviors [9]. Different people do the same activity in a multitude of ways; and even for a single person, he or she may do the same activity in different ways at different times. Most previous work in human activity understanding is performed on traditional 2D color images/videos and both global and local spatial-tempo features have been proposed (reviewed in [10–12]). Because it is difficult to deal with variations in 2D images/videos due to changes in lighting condition, view angle, and scale, researcher begun to explore applications of 3D information in human activity understanding [9]. In contrast to 2D images/videos, depth maps such as those acquired by the Microsoft Kinect system are related to object geometry and thus are independent of lighting conditions.

However, it is a difficult task to develop features to representation human activities based on 3D information. This is because depth images have much less textures than 2D images and are sensitive to occlusion [13]. Adopting recognition algorithms developed to work on 2D images and videos is not trivial either. For example, interest-point detectors such as Dollar [14] and STIP [15] perform badly on 3D videos. Currently, there are two approaches in using depth data for activity recognition, depth based and skeleton/joint based methods [9]. A recent study showed that relative joint positions carry significant information about activities [16], but these features are difficult to extract without human intervention. Thus, although several recognition algorithms that use manually selected joint-related features have been developed [8, 17–24], there is no consensus on what joint-related features should be extracted and how they should be used for activity recognition.

We propose a method that learns automatically sparse representations of human activities. Specifically, we treat 3D movements of joints as space-time volumes and densely sample the volumes along the time axis to obtain a set of sub-volumes. We then use the reconstructed independent component analysis (RICA) [25] to learn a dictionary of over-complete codes from the sub-volumes for each activity. In this learning procedure, the sub-volumes are represented by

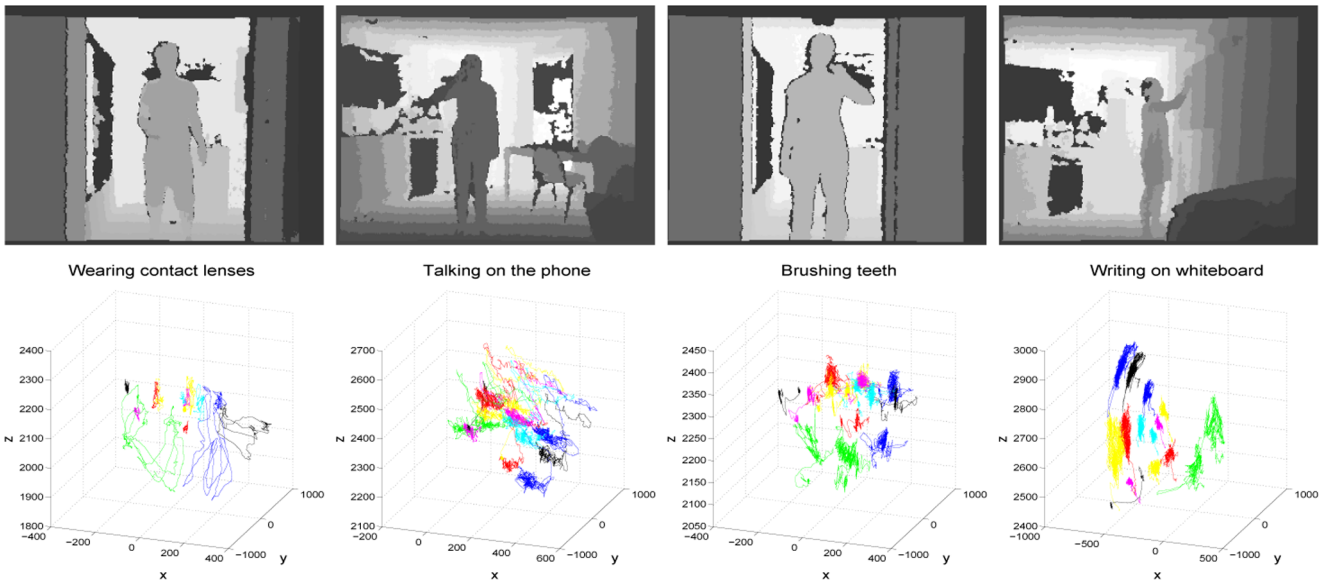

Figure 1. Four activities in the CAD-60 dataset. First row: depth images; Second row: joint trajectories.

doi:10.1371/journal.pone.0114147.g001

the learned codes in a sparse manner. From the coefficients of the sub-volumes projected to the sparse codes, we construct a sparse histogram for each activity. Finally, we concatenate the sparse histograms and use them as inputs to a multi-class support vector machine (SVM) to perform activity recognition.

We tested this model on three widely used databases of human activities and found that it outperforms the state-of-the-art algorithms. The contributions of this paper to joint-based activity recognition are:

- a general dictionary-based framework that automatically learns sparse, high-dimensional spatial-temporal features of 3D movements of joints,
- an efficient method that constructs sparse codes and histograms,
- a real-time system for human activity recognition that can be easily implemented,
- extensive evaluations on the proposed model and superior results on three datasets of human activities.

The paper is organized as follows. In Section 2, we briefly describe related work and how our model is different. In Section 3, we describe the procedures of data processing and learning dictionaries of codes of 3D movements of body joints. In Section 4, we propose a set of sparse histograms of the codes of human activities. In Section 5, we present an algorithm for activity recognition via a multi-class SVM with sparse histograms as input features. In Section 6, we report the recognition results of our model on three datasets of human activities and compare them to the state-of-the-art algorithms. In Section 7, we briefly summarize the main points of our model and address several aspects of the model that can be improved.

Methods

2.1 Related Work

We briefly describe related work below. For work on activity recognition based on 2D videos, we refer readers to several surveys [10–12].

Depth map-based approaches

Features automatically or manually extracted from depth images/videos have been proposed, including bag of points [26], Space-Time Occupancy Patterns (STOP) [27], Random Occupancy Pattern (ROP) [28], HOG from Depth Motion Maps (DMM-HOG) [24], Histogram of Oriented 4D Surface Normals (HON4D) [29], Pixel Response and Gradient Based Local Feature [30], Local Trajectory Gradients, and SIFT [31]. In [32], depth silhouettes are used as features and a hidden Markov Model (HMM) is used to model temporal dynamics of activities. Different from these methods, our algorithm is based on joints which are the best features for human activity recognition [16].

Skeleton/Joint based approaches

It was observed in 1970's that a range of human activities can be recognized on the basis of 3D movements of body joints [33]. However, joint-based activity recognition drew research attention only recently due to the availability of low-cost Microsoft Kinect cameras that can acquire 3D videos of joint movements. Campbell and Bobick [17] proposed to compute action curves by projecting 3D joint trajectories on low-dimensional phase spaces and to classify actions based on action curves. This approach works only for simple activities. Lv et al. [18] proposed seven types of local features and used HMMs to describe the evolution of these features. In [19] a so-called Histogram of 3D Joint Location (HOJ3D) was designed to characterize the distribution of joints around the central joint (hip joint) and a HMM was developed to model temporal changes of the feature. In [20], SIFT features for objects and skeleton features for humans were developed and an MRF was used to model human activities. Sung et al. [8] computed HOG from RGBD data and position-angle features from joints and used a Maximum Entropy Markov Model (MEMM) to represent activities hierarchically. Wang et al. [34] designed Local Occupancy Pattern (LOP) which was computed from a set of 3D points around each joint. Finally, geometric relationships among joints were used in [23]. All these methods need manually designed features. In contrast, a set of dictionaries of sparse codes of human activities are obtained without manual interventions in the method we present here.

The work related to ours is the EigenJoints that describe positional differences between joints within or cross video frames and are used for action recognition via a Naive Bayes nearest neighbor classifier [24]. The EigenJoints are simple and easy to compute and so are the features of our model presented below. Our model is different in two ways. First, a set of dictionaries of codes of human activities are learned. Second, an approximate sparse coding is performed to obtain a set of sparse histograms for action recognition via a multi-class SVM.

2.2 Joint-Dictionary Learning

We propose to learn a set of dictionaries of sparse codes to represent the complex spatial-temporal relationships among body joints. For this purpose, we introduce some notations first.

2.2.1 Notations

The d-th video is denoted by V_d and the total number of frames in the d-th video V_d is N_f^d. The number of joints in each frame f is denoted by N and the 3-dimensional coordinate vector of each joint P in frame f is (x_i^f, y_i^f, z_i^f).

2.2.2 Sampling space-time-joint volume

For each frame f, we construct a matrix I_f by concatenating all the coordinates of N joints in frame f. Specifically, the i-th row of I_f is the coordinate vector (x_i^f, y_i^f, z_i^f) of the i th joint in frame f and the columns of I_f are the x,y,z coordinates of the joints. Therefore, the size of I_f is $N \times 3$. Each column C_i^f of the matrix I_f is subtracted by its mean $m(C_i^f)$. This operation makes I_f invariant to camera/human placements. Although this operation removes global body motion, it won't affect much the performance of the model developed here since the activities in the three tested datasets are indoor human daily activities that don't entail much global body motion. We then concatenate all the matrices $I_f(f = 1,2,\cdots,N_f^d)$ from the d-th video V_d to form a volume V_c^d as shown in Fig. 2. V_c^d is a matrix of a dimension of $N \times 3 \times N_f^d$. Mathematically, we have

$$I_f = \begin{bmatrix} x_1^f - m(C_1^f) & y_1^f - m(C_2^f) & z_1^f - m(C_3^f) \\ x_2^f - m(C_1^f) & y_2^f - m(C_2^f) & z_2^f - m(C_3^f) \\ x_3^f - m(C_1^f) & y_3^f - m(C_2^f) & z_3^f - m(C_3^f) \\ \vdots & \vdots & \vdots \\ x_{N-1}^f - m(C_1^f) & y_{N-1}^f - m(C_2^f) & z_{N-1}^f - m(C_3^f) \\ x_N^f - m(C_1^f) & y_N^f - m(C_2^f) & z_N^f - m(C_3^f) \end{bmatrix}, f = 1,2,\cdots,N_f^d \quad (1)$$

and

$$V_c^d(:,:,i) = I_i, i = 1,2,\cdots,N_f^d, \quad (2)$$

where

$$m(C_1^f) = \frac{1}{N}\sum_{i=1}^{N} x_i^f, m(C_2^f) = \frac{1}{N}\sum_{i=1}^{N} y_i^f, m(C_3^f) = \frac{1}{N}\sum_{i=1}^{N} z_i^f. \quad (3)$$

We densely sample V_c^d along the time dimension ("frame" axis in Fig. 2) to obtain $N_s(N_s = N_f^d)$ sub-volumes for each video. Thus, we take all possible sub-

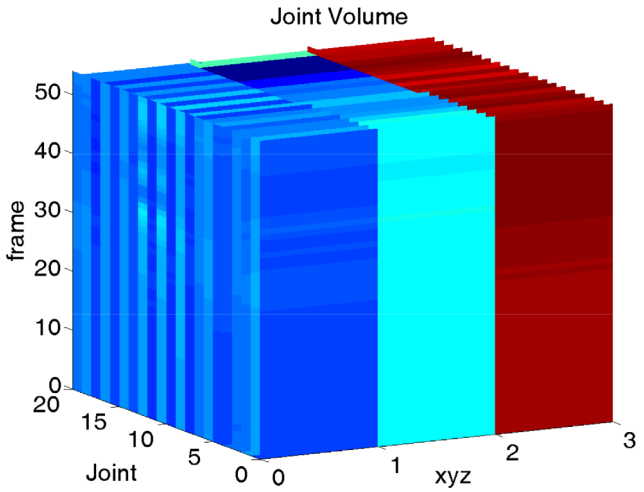

Figure 2. Joint volume. xyz axis: joint coordinates (x,y,z); joint axis: indices of joints; frame axis: index of video frames.

doi:10.1371/journal.pone.0114147.g002

volumes of V_c^d. One can use various methods to take sub-volumes at sampled points in the time dimension ($N_s \leq N_f^d$). Suppose that the sample sizes are $3, N, N_f^s$ along the "xyz" axis, the "joint" axis, and the "frame" axis, respectively. Each sample $S_i (i = 1, \cdots, N_s)$ is then a $N \times 3 \times N_f^s$ sub-volume sampled from V_c^d, which can be written as

$$S_i(:,:,j) = I_{t_j^i}, \; j = 1,2,\cdots,N_f^s, i = 1,2,\cdots,N_s, \tag{4}$$

where $I_{t_j^i}$ is the t_j^ith coordinate image.

The third dimension of sub-volume S_i can be permuted with the first dimension by a permutation operation

$$S_i^p = permute(S_i, [3,2,1]), \tag{5}$$

where vector $[3,2,1]$ indicates that the second dimension of S_i stays where it is but the first dimension is swapped with the third dimension. From equation (1), it can be seen that the same coordinate components of each joint form the columns of the permuted sub-volume S_i^p by the above permutation operation. As a result, either x, y, or z coordinate components of a joint in the sampled frames in sub-volume S_i form one column of the permuted sub-volume S_i^p. For example, the x coordinate components of the head joint in different frames in sub-volume S_i are one column of S_i^p. This is illustrated by the horizontal color bars in Fig. 3 since body joints in neighboring frames tend to have similar coordinates. To examine the sub-volumes, we form a new matrix S_v by by reordering the permuted sub-volumes S_i^p lexicographically.

$$S_v = \left[S_1^p(:) \quad S_2^p(:) \quad \cdots \quad S_{N_s-1}^p(:) \quad S_{N_s}^p(:) \right] \tag{6}$$

S_v represents the sub-volumes from one video with each column corresponding to one sub-volume. One S_v is shown in Fig. 3, where the "Sample index" axis indicates the indices of all the sub-volume samples and the "Coordinate index" axis is the row index of matrix S_v. As shown in Fig. 3, gradual changes between samples occur along the "Sample index" axis (corresponding to time axis). Thus, the configurational relationships among body joints update in the time domain, as they should in human activities.

2.2.3 Semantics of space-time-joint sub-volumes

The i-th sub-volume S_i described above contains several video frames (N_f^s frames) which may capture components of one or more activities. For big N_f^s, there are more frames in a sub-volume, which may capture an activity. For small N_f^s, there are few frames in a sub-volume, which may only capture a part of an activity. Two extreme cases are $N_f^s = 1$ and N_f^s is equal to the total number of the frames of the videos.

In following section, we propose to learn a set of dictionaries of codes that can be used to represent complex human activities. The words (i.e., codes) in the dictionaries should be components whose concatenations in the space and time domains constitute representations of human activities. Thus, N_f^s should be neither too small nor too big so that the sub-volumes are samples of components of human activities. Unfortunately, it is difficult to set a fixed value for N_f^s for all human activities, which may have components of a variety of spatial and temporal scales and may be captured by cameras of a range of imaging parameters. Therefore, we set the values of N_f^s via a learning procedure for the three datasets tested in this paper.

2.2.4 Joint-dictionary learning

We propose a method to learn a set of sparse codes that can be used to represent human activities. Sparse representation is useful for object recognition [25]. A number of algorithms have been proposed to learn sparse features, including restrict Boltzmann machines [35], spare auto-encoder [36], independent component analysis [37], sparse coding [38], and RICA [25]. Since RICA works well on approximately whitened data and is fast [25], we use RICA to learn a dictionary of codes from a set of sub-volumes $S_i, i = 1,2,\cdots,N_s$ for each activity. The learned dictionary is called "Joint-dictionary". To the best of our knowledge, this is the first work on feature learning from 3D movements of body joints.

For each activity $c, c = 1,2,\cdots,N_a$ (N_a is the number of activities), we obtain a dictionary W_c. Suppose N_c is the total number of sub-volume samples from activity c. Then the class-specific dictionary W_c can be obtained by solving the following optimization problem [25]

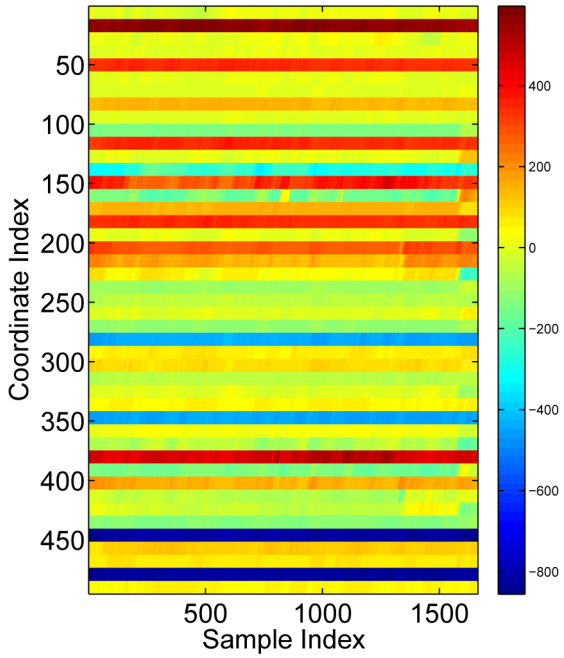

Figure 3. Coordinate samples from one video. Each column corresponds to one coordinate sub-volume sample. The "Sample index" axis indicates the indices of all sub-volume samples and the "Coordinate index" axis is the row index of matrix S_v.

doi:10.1371/journal.pone.0114147.g003

$$W_c = \arg\min_W \frac{\lambda}{N_c}\sum_{i=1}^{N_c} ||W^T W S_i^c(:) - S_i^c(:)||_2^2 + \sum_{i=1}^{N_c}\sum_{j=1}^{k} g(W_j S_i^c(:)), \quad (7)$$

where S_i^c is the ith sub-volume sample from activity c; $S_i^c(:)$ is a lexicographical operation on S_i^c to form a column vector; $g(\star)$ is a nonlinear convex function (e.g., smooth L_1 penalty function $g(\star) = \log\cosh(\star)$ [39] in this paper); and k, λ are the number of features (rows of W_c) and a balancing parameter, respectively.

The objective function in (7) is a smooth function. The optimization problem (7) can be easily solved by any unconstrained solvers (e.g., L-BFGS and CG [40]).

We propose to learn a class-specific dictionary W_c for each activity c and we pool all the learned class-specific dictionaries $W_c, c = 1,2,\cdots,N_a$ to form a code book W as follows

$$W = [\,{W_1}^t \quad {W_2}^t \quad \cdots \quad {W_{Na}}^t\,]. \quad (8)$$

The code book W contains $k \times N_a = 400 \times N_a$ words in total. Note that W is over-complete since the number of words is bigger than the size of sub-volumes.

Fig. 4 shows two dictionaries for "talking on the phone" and "writing on white board". Each dictionary contains 400 words. The words shown in Fig. 4 are used to represent 3D spatial-temporal sub-volumes and are different from conventional words (e.g., oriented bars) learned from 2D natural image patches [25]. These

words are the bases of segments of space-time concatenations of body joints by which any segment of an activity can be constructed linearly. Unfortunately, unlike independent components of natural scenes, which are like edge elements, the words obtained here are difficult to visualize.

2.3 Sparse Histograms

In this section, we propose an approximate sparse coding scheme and compile a set of sparse histograms. Any sample $\mathbf{x}$ can be sparsely represented by W as following

$$\arg\min_{\mathbf{s}} ||\mathbf{W}\mathbf{s}-\mathbf{x}||_2^2+\lambda||\mathbf{s}||_1, \tag{9}$$

where $\mathbf{s}$ is the sparse coefficients of sample $\mathbf{x}$ represented by dictionary $\mathbf{W}$. A number of algorithms have been proposed to solve the above problem of sparse representation [41].

Instead of solving the optimization problem (9) for each video, which is prohibitively time consuming, we propose to project any sample $\mathbf{x}$ onto $\mathbf{W}$ via

$$\mathbf{s}=\mathbf{W}^t\mathbf{x}, \tag{10}$$

where $\mathbf{s}$ is the coefficients of sample $\mathbf{x}$. The first N_s (400 in this paper) largest coefficients are kept and the rest coefficients of $\mathbf{s}$ are set to zero to make $\mathbf{s}$ sparse. Note that the dimension of $\mathbf{s}$ is $N_a \times 400$ (N_a is the number of activities). The number of the kept sparse coefficients (400 in this paper) seems to be big, but it is a lot smaller than the dimensionality of sub-volumes, which is $15 \times 3 \times 11 = 495$ for the CAD60 database, and the dimensionality of the entire video. In the Section 6 we show that N_a can be much smaller while good performance on activity recognition can be still achieved by our method.

The computation in equation (10) is very fast. Although this is an approximate sparse coding scheme, our results show that this approximation does not impair activity recognition (see Section 6).

We then obtain the histogram h of nonzero coefficients of samples of a video v by counting the number of occurrences of nonzero coefficients for each word in $\mathbf{W}$. Thus, the ith component of h is the number of occurrences of the ith word that appears in video v. Fig. 5 shows the histograms of "talking on the phone" and "writing on the white board" of the CAD-60 database. The two histograms are quite different upon a careful visual examination. We define the degree of sparsity of a histogram as the ratio of the number of non-zero bins to the bin size

$$Sparsity\ Degree = \frac{\#\ Number\ of\ Non\ -\ Zero\ Bins}{\#\ Total\ Number\ of\ Bins}. \tag{11}$$

The sparsity degrees of the two histograms in Fig. 5 are 10.375% and 13.104%, respectively. Thus, the histograms constructed this way are sparse.

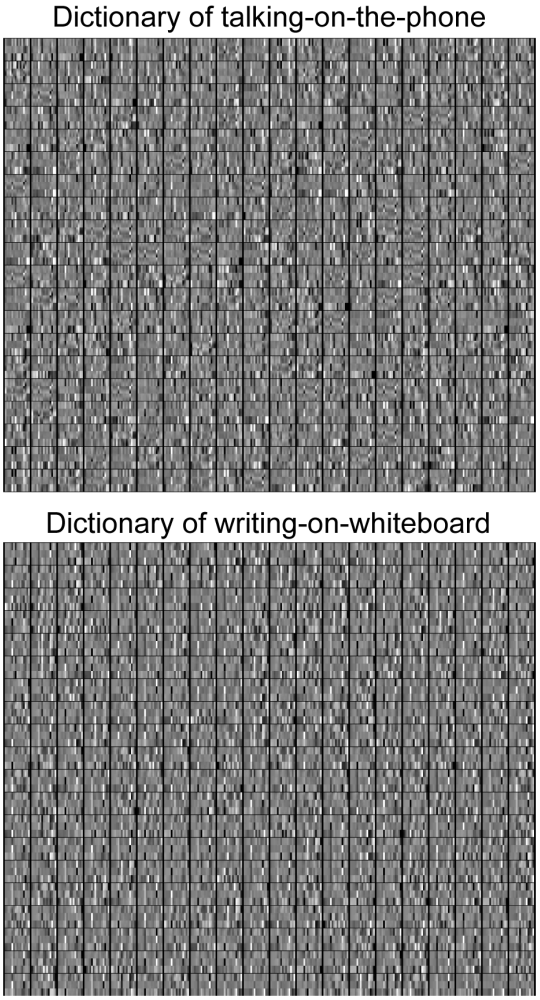

Figure 4. Two dictionaries for "talking on the phone" (top) and "writing on white board" (bottom). 400 words are shown in 400 squares.

doi:10.1371/journal.pone.0114147.g004

Note that the histogram bins in Fig. 5 have more or less the same height (about 0.3). This may be due to similar words in the dictionaries for the activities in the dataset. Since a dictionary is learned from each activity independently, it is likely that there are words that are shared by more than one activities. It is worthy to point out, though, that shared words do not impair the performance of our algorithm.

2.4 Classification

We compile a sparse histogram for each activity and use it as a feature for recognition via a multi-class SVM. In this procedure, we train one SVM in a one-vs.-rest scheme for each activity; use the homogeneous kernel map expansion [42] with a "$\chi-$ square" kernel to expand the dimensionality of feature by 2 times;

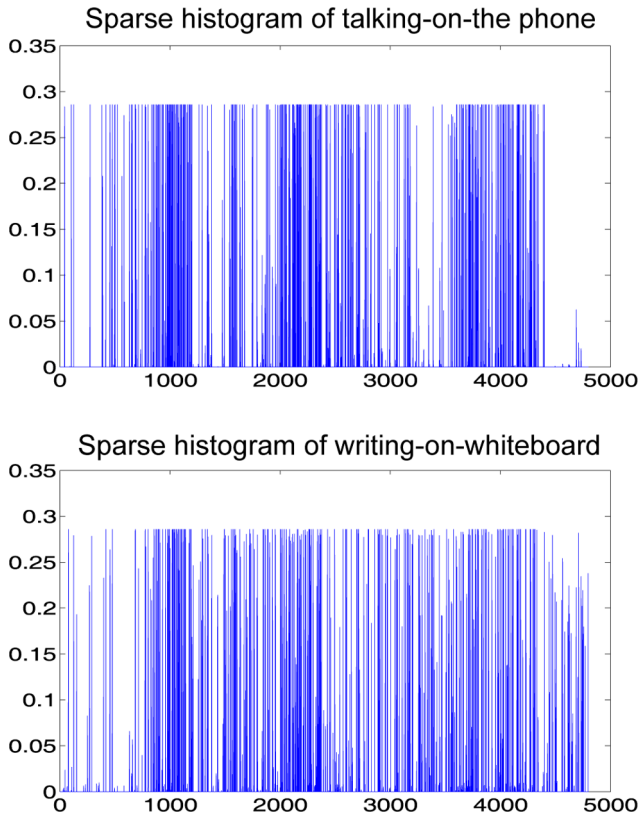

Figure 5. Sparse histograms of "talking on the phone" and "writing on the white board" in the CAD-60 dataset. The sparsity degrees are 10.375% and 13.104% respectively. Note that the sum of the histograms is 100.

doi:10.1371/journal.pone.0114147.g005

and implement the computing with the source codes of an open-source collection of vision algorithms called "VLFeat" (http://www.vlfeat.org/). The training and testing procedures are summarized in Fig. 6 and Fig. 7, respectively.

Results

We tested our algorithm on three publicly available datasets: the Cornell Activity Dataset-60 (CAD-60) [8], the MSR Action3D [43], and the MSR Daily Activity 3D [22]. Our results show that the model proposed here is better than the state-of-the-art methods.

3.1 CAD-60 dataset

The CAD-60 dataset is an RGBD dataset acquired with a Microsoft Kinect sensor at 30 Hz and has a resolution of 640×480 pixels [8] (Dataset S1). The 3D coordinates of 15 joints are the real-time outputs of the skeleton tracking algorithm of the sensor [44]. The dataset contains 14 human activities performed

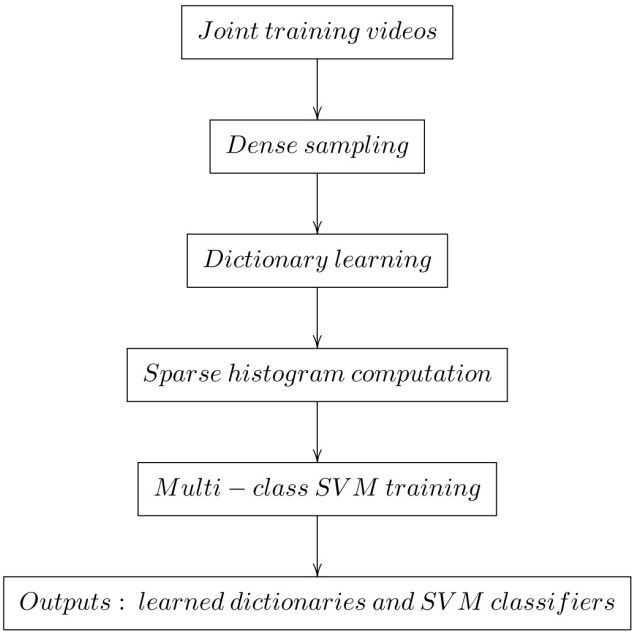

Figure 6. Flow chart for learning dictionaries and SVM classifiers.

doi:10.1371/journal.pone.0114147.g006

indoors by 4 subjects (two males and two females) for about 45 seconds. The total number of frames of each activity of each person is about one thousand. We follow the ''new person'' setting in [8] where data of 3 subjects were used for training and the remaining one subject for testing. To improve recognition performance, we mirrored the joints of the left-handed subject to make her activities similar to those of the other 3 right-handed subjects, which is a usually practice. Briefly, a plane P was first found by fitting four joints, left-arm, right-arm, left hip, and right hip. Then, a mirror plane P_m was computed under the constraints that P_m is perpendicular to P and passes through the middle point between the two arm joints and through the middle point between the two hip

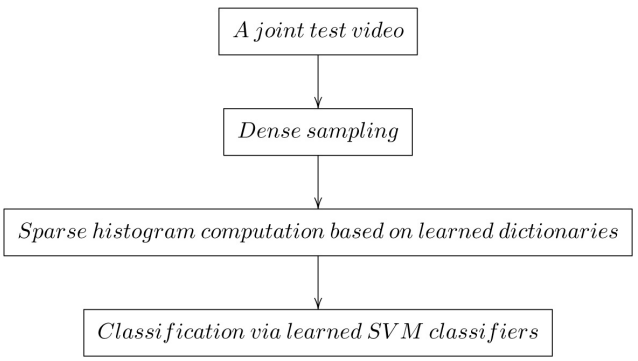

Figure 7. Flow chart for human activity recognition.

doi:10.1371/journal.pone.0114147.g007

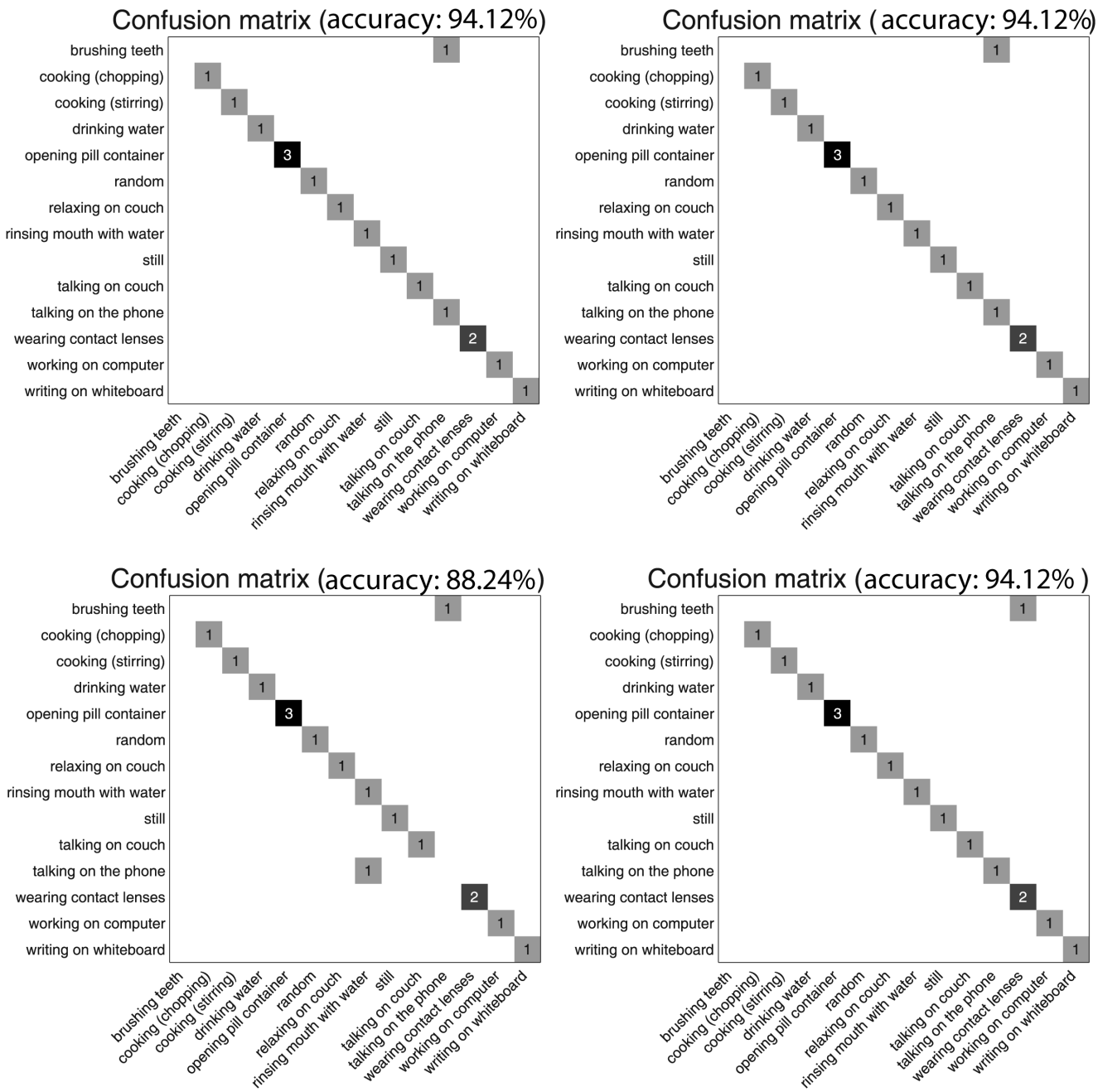

Figure 8. Four confusion matrices for four experimental settings.

doi:10.1371/journal.pone.0114147.g008

joints. Finally, all joints of the left-handed subject were mirrored with respect to P_m.

Fig. 8 shows 4 confusion matrices for four cases where three subjects are chosen for training and the remaining subject for training. We compare our results to 9 algorithms in terms of average accuracy, precision, and recall in Table 1. The

Table 1. Performance of our model and other methods on the CAD-60 dataset.

Algorithm	Accuracy (%)	Precision (%)	Recall(%)
Sung et al., AAAI PAIR 2011, ICRA 2012 [8].		67.9	55.5
Koppula, Gupta, Saxena, IJRR 2012 [20]		80.8	71.4
Zhang, Tian, NWPJ 2012 [46]		86	84
Ni, Moulin, Yan, ECCV 2012 [47]	Accur: 65.32		
Yang, Tian, JVCIR 2013 [48]		71.9	66.6
Piyathilaka, Kodagoda, ICIEA 2013 [49]		70*	78*
Ni et al., Cybernetics 2013 [50]		75.9	69.5
Gupta, Chia, Rajan, MM 2013 [51]		78.1	75.4
Wang et al., PAMI 2013 [52]	Accur: 74.70		
Ours	**91.17**	**89.11**	**89.28**

doi:10.1371/journal.pone.0114147.t001

results of other algorithms are from the website http://pr.cs.cornell.edu/humanactivities/results.php that reports results on the dataset. As shown in Table 1, our algorithm is the best in terms of accuracy, precision, and recall on this dataset. Since some authors reported the performance of their algorithms in terms of only part of the above metrics, there are blank cells in Table 1.

3.2 MSR Action3D dataset

The MSR Action3D dataset contains 20 activities acquired from 10 subjects, each of whom performed each activity 2 or 3 times. The resolution is 320×240 pixels and the frame rate is 16 Hz. The dataset provides the 3D movement data of 20 joints per person. We used 557 videos out of the 567 videos in the dataset since 10 videos have missing joints or erroneous joints [22] (Dataset S2).

To allow fair comparison, we followed the same setting as [22]: subjects Nos. 1, 3, 5, 7, and 9 as the training set and subjects Nos. 2, 4, 6, 8, and 10 as the testing set. The 20 actions are divided into three subsets, AS1, AS2, and AS3 according to the experimental setting in [22, 43], which are listed in Table 2. AS1 and AS2

Table 2. Subsets of actions, AS1, AS2, and AS3 in the MSR Action 3D dataset.

Action Set 1(AS1)	Action Set 2 (AS2)	Action Set 3(AS3)
Horizontal arm wave	High arm wave	High throw
Hammer	Hand catch	Forward kick
Forward punch	Draw x	Side kick
High throw	Draw tick	Jogging
Hand clap	Draw circle	Tennis swing
Bend Two	hand wave	Tennis serve
Tennis serve	Forward kick	Golf swing
Pickup & throw	Side boxing	Pickup & throw

doi:10.1371/journal.pone.0114147.t002

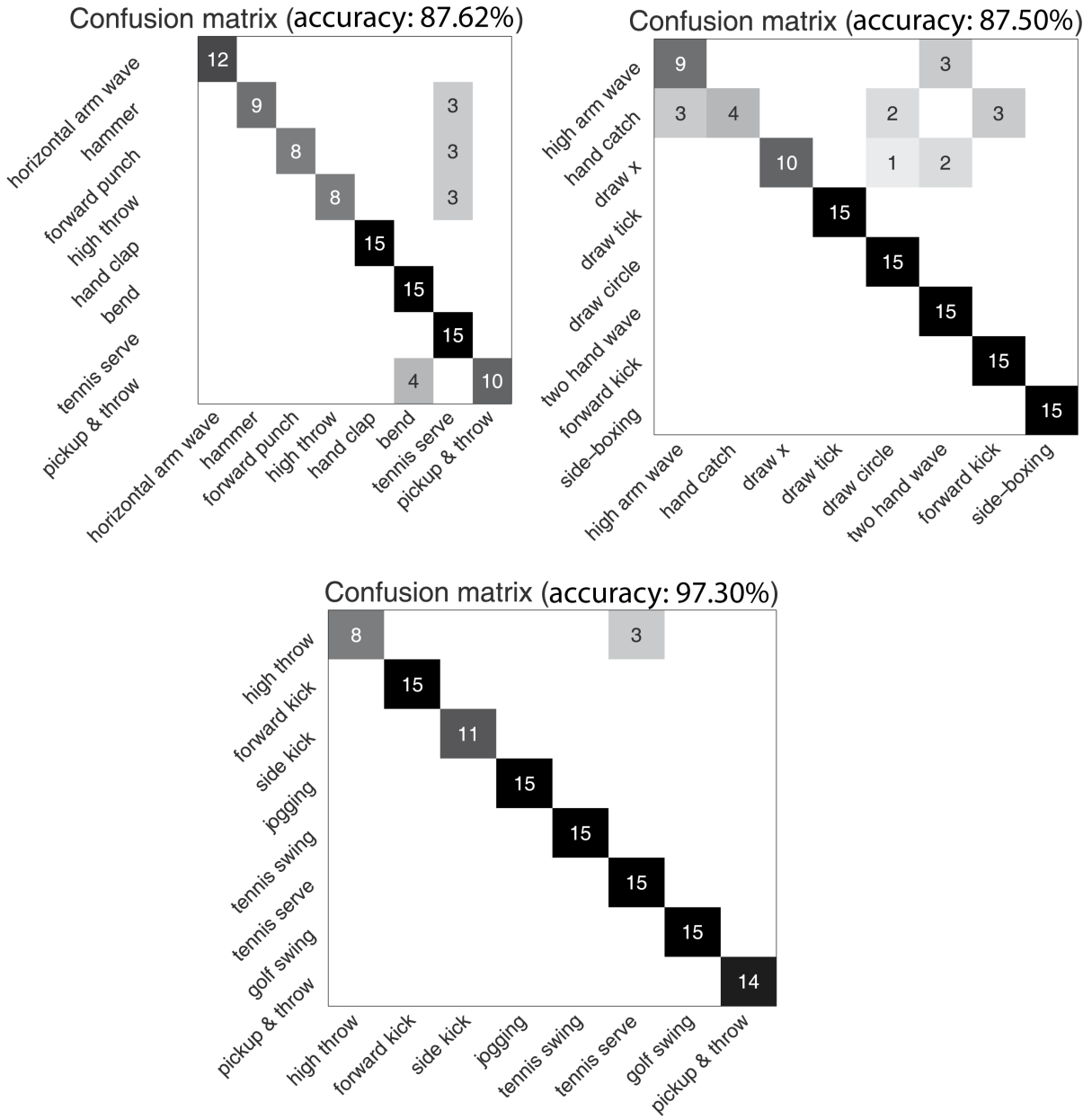

Figure 9. Three confusion matrices for AS1, AS2, AS3 in the MSR Action 3D dataset.

doi:10.1371/journal.pone.0114147.g009

contain similar actions and AS3 contains complex actions composed of simpler ones.

The accuracy of our algorithm on AS1, AS2 and AS3 is 87.62%, 87.5% and 97.3%, respectively. The average accuracy on the dataset is 90.81%. The three confusion matrices for AS1, AS2, and AS3 are shown in Fig. 9. Thus, our algorithm performs better on AS3 than AS1 and AS2.

Table 3. Performance of our model and other methods on the MSR Action 3D dataset.

Method	Accuracy (%)
HON4D + Ddiscb [29]	88.89
HON4D [29]	85.85
Jiang et al. [22]	88.20
Jiang et al. [34]	86.50
Yang et al. [24]	85.52
Dollar + BOW[14]	72.40
STIP + BOW [15]	69.57
Vieira et al. [27]	78.20
Klaser et al. [53]	81.43
Ours	**90.81**

doi:10.1371/journal.pone.0114147.t003

Table 3 compares the performance of our model to other 9 methods. The accuracies of methods are from a recent paper [29]. The performance (90.80%) of our model is the best.

3.3 MSR Daily Activity 3D dataset

The MSR Daily Activity 3D dataset contains 16 activities each of which was performed twice by 10 subjects [22] (Dataset S3). The dataset contains 320 videos in each of 3 channels, RGB, depth, and joint. There are 20 body joints recorded whose positions are quite noisy due to two poses: "sitting on sofa" and "standing close to sofa".

The experimental setting is the same as in [22] which split the dataset into 3 subsets, AS1, AS2, and AS3 as listed in Table 4. We followed the same setting as [22]: subjects Nos. 1,3,5,7, and 9 as the training set and subjects Nos. 2,4,6,8,and 10 as the testing set. The accuracy of our algorithm on AS1, AS2 and AS3 is 71.67%, 81.25%, and 85.00%, respectively and the average accuracy is 79.31%. The confusion matrices are shown in Fig. 10. Our algorithm performs better on AS3 than AS1 and AS2.

Table 5 lists the results of our model and several other methods. The results of other methods are from a recent paper [22]. The accuracy of our model is 79.31% which is lower than the best result (85.75%). However, only joint information is

Table 4. Subsets of actions, AS1, AS2, and AS3 in the MSRDaily Activity 3D dataset.

Action Set 1(AS1)	Action Set 2 (AS2)	Action Set 3(AS3)
eat	drink	use laptop
read book	call cellphone	cheer up
write on a paper	use vacuum cleaner	play guitar
use laptop	sit still	stand up
toss paper	play game	sit down
walk	lie down on sofa	

doi:10.1371/journal.pone.0114147.t004

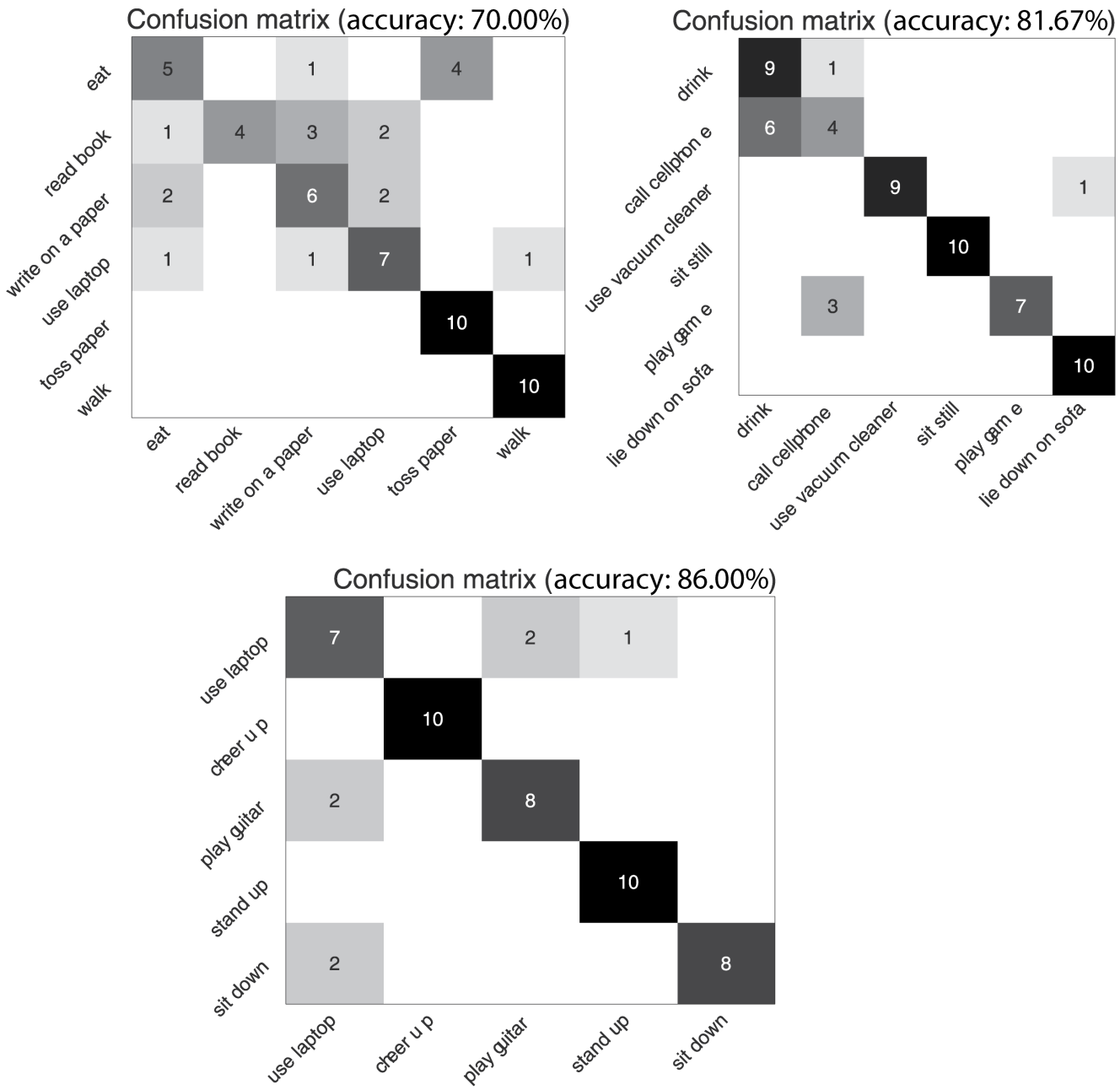

Figure 10. Three confusion matrices for AS1, AS2, AS3 in the MSR Daily Activity 3D dataset.

doi:10.1371/journal.pone.0114147.g010

used in our model while both joint and depth information is used to obtain the best result [22]. Compared to other models that use only joint information, our model is the best, outperforming the best earlier result which is 68%.

Table 5. Performance of our model and other methods on the MSR Daily Activity 3D dataset.

Method	Accuracy(%)
Dynamic Temporal Warping [54]	54
Only LOP features[22]	42.5
Only Joint Position features [22]	68
SVM on Fourier Temporal Pyramid Features [22]	78
Actionlet Ensemble [22]	85.75
Ours	**79**

doi:10.1371/journal.pone.0114147.t005

3.4 Comparison with a baseline method

We have evaluated the performance of our method on three public datasets. Our method has four steps: generating samples, learning dictionaries, constructing sparse histograms, and classifying via SVMs. In this section, we replace the RICA-based dictionary learning in our method with the k-means clustering. We cluster samples with the k-means algorithm and take the clusters as words in the dictionaries. We call this method as a baseline method. The results of this baseline method and our original method on the three datasets are shown in Table 6. Both methods perform well, with our original method being slightly better. Thus, the joint dictionaries and sparse histograms in both methods are responsible for the good performance.

3.5 Parameter setting and time performance

There are seven parameters in our model. They are N_f^s, the sampling size along the z-direction; N_w, the number of words in each class-specific dictionary; λ, the balancing parameter in Eq. 7; N_s, the number of the largest coefficients; N_t, the factor by which the dimensionality of feature vector is expanded; γ, the parameter of the $\chi-$ square kernel; and λ_{svm}, the balancing parameter of the SVM. These parameters are probably independent of each other since they are for different phrases of our algorithm, sampling, dictionary learning, sparse histogram, and SVM training.

Of the seven parameters, the sampling size N_f^s, the number of words N_w, and the number of the largest coefficients N_s are new in our algorithm while other parameters appeared in other published studies [25, 42]. Therefore, we explore how to choose the values of these three parameters while setting other parameters to the values recommended by other researchers [25, 42]. We run our algorithm with different parameter values on the CAD60 dataset. Fig. 11 shows the average

Table 6. Performance of our model and the baseline method on the three databases.

	CAD-60	MSRAction3D	MSRDaily Activity3D
Our method	91.17	90.81	79
Baseline method	89.71	88.34	77.03

doi:10.1371/journal.pone.0114147.t006

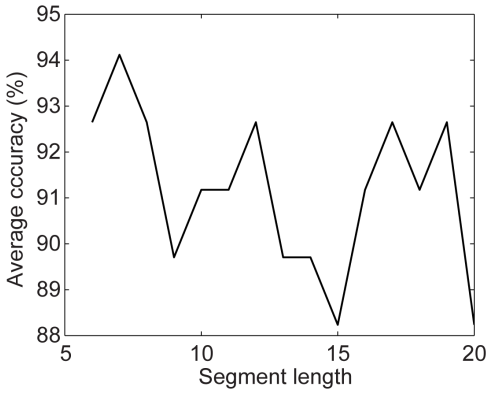

Figure 11. Average accuracy as a function of the number of frames of sub-volumes on the CAD-60 dataset.

doi:10.1371/journal.pone.0114147.g011

accuracy as a function of the sampling size f_s when $N_w = 400$ and $N_s = 400$; Fig. 12 shows the average accuracy as a function of the number of words N_w when $N_f^s = 11$ and $N_s = 400$; and Fig. 13 shows the average accuracy as a function of the number of the largest coefficients N_s when $N_f^s = 11$ and $N_s = 400$. These good results on action recognition obtained under a wide range of parameter settings show that our method is not sensitive to parameter values. Therefore, setting the parameters in our algorithm for good recognition performance is not challenging.

The values of the parameters for all the experiments are listed in Table 7. For simplicity, we set the parameter values the same for the three databases except N_f^s, the sampling size along the z-direction, which may depend on the speed of the activities and the frame rate of the videos. As shown in Tables 1, 3, 5, and 6 and

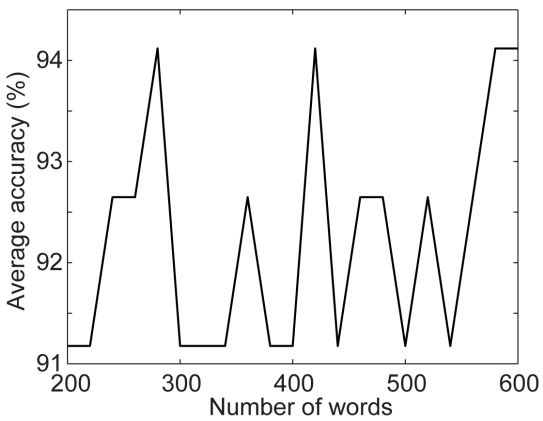

Figure 12. Average accuracy as a function of the number of words in the dictionaries on the CAD-60 dataset.

doi:10.1371/journal.pone.0114147.g012

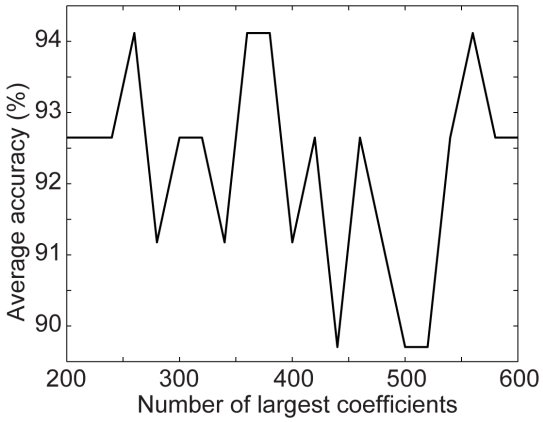

Figure 13. Average accuracy as a function of the number of largest coefficients kept in the sparse histograms on the CAD-60 dataset.

doi:10.1371/journal.pone.0114147.g013

Figs. 8–13, there are a range of parameter values in our method that lead to very good performance, which may be further improved by finely tuned parameter values.

The proposed algorithm was implemented in Matlab without any optimization in programming. We evaluated the time performance of our method using Intel(R) Core(TM)2 Duo CPU E8600@3.33 GHz with 64 bit Windows 7 professional SP1 OS. Only one core (2 cores available) was used based on single thread programming. We report 4 measures, i.e., the average training time (ATT), the average testing time per video (ATTPV), the average number of training videos (ANTV), the average number of test videos (ANOTV), and the average number of training classes (ANTC) on the three datasets in Table 8.

As shown in the table, our method took 0.50, 0.03, and 0.10 seconds/per video to classify the activities of the CAD-60 dataset, the MSR Action3D dataset, and the MSR Daily Activity 3D dataset respectively. The training time was 513.43 seconds, 73.02 seconds, and 125.60 seconds on the CAD-60 dataset, the MSR Action3D dataset, and the MSR Daily Activity 3D dataset respectively. This time performance can be improved significantly by optimized C++ codes running on much faster CPUs. Therefore, our model is a real-time method that can be used in smart environments and deployed in robots for human-robot collaborations.

Table 7. Values of the parameters of our method.

Database	N_f^s	N_w	λ	N_s	N_t	γ	λ_{svm}
CAD-60	11	400	0.5	400	3	0.01	0.01
MSRAction3D	13	400	0.5	400	3	0.01	0.01
MSRDailyActivity3D	21	400	0.5	400	3	0.01	0.01

doi:10.1371/journal.pone.0114147.t007

Table 8. Time performance of our method evaluated on the databases CAD-60, MSR Action3D and MSR Daily Activity 3D. **ATT**: Average Training Time (seconds per setting); **ATTPV**: Average Testing Time per Video (seconds per video); **ANTV**: Average Number of Training Videos per Setting; **ANOTV**: Average Number of Testing Videos per Setting; **ANTC**: Average Number of Training Classes per Setting.

Database	ATT	ATTPV	ANTV	ANOTV	ANTC
CAD-60	513.43	**0.50**	51	17	14
MSRAction3D	73.02	**0.03**	114	109	8
MSRDaily Activity3D	125.60	**0.10**	60	60	6

doi:10.1371/journal.pone.0114147.t008

Discussion

In this paper we proposed a real-time algorithm that makes use of joint information to recognize human activities. In the first step of the algorithm, videos of 3D movements of body joints are sampled to obtain a set of spatial-temporal 3D volumes, which entail the complex spatial-temporal relationships of joints of human activities at a data size that is much smaller than that of a RGBD volume. Second, RICA is performed on the spatial-temporal 3D volumes to obtain a set of dictionaries of codes that form a sparse representation of human activities. An approximate spare coding scheme is then used to compile a set of spare histograms as features for activity recognition. Finally, a multi-class SVM is used to perform activity recognition. We performed extensive tests on this algorithm on three widely used datasets of human activities. Our results show that this algorithm produces so far the best recognition accuracy on these datasets.

Our algorithm automatically learns discriminative features for activity recognition and is very fast and easy to implement. Since joint information can be obtained by low-cost cameras such as the Microsoft Kinect systems, our algorithm can be used in smart environments and deployed in robots for human-robot collaborations. This model can be improved by the rich information in depth images. To include this information, we will extend the model presented here and our recent model of activity recognition based on multi-scale activity structures [45].

Supporting Information

Dataset S1. CAD-60 dataset.
doi:10.1371/journal.pone.0114147.s001 (RAR)

Dataset S2. MSR Action 3D dataset.
doi:10.1371/journal.pone.0114147.s002 (RAR)

Dataset S3. MSR Daily Activity 3D dataset.
doi:10.1371/journal.pone.0114147.s003 (RAR)

Acknowledgments

We thank Drs. He Cui and Suxibing Liu for helpful comments.

Author Contributions

Conceived and designed the experiments: ZY JQ. Performed the experiments: JQ ZY. Analyzed the data: JQ ZY. Contributed reagents/materials/analysis tools: JQ ZY. Wrote the paper: ZY JQ.

References

1. **Cook DJ, Das SK** (2005) Smart Environments: Technologies, Protocols, and Applications. John Wiley & Sons, Inc., 1–10 pp.
2. **Reisberg B, Finkel S, Overall J, Schmidt-Gollas N, Kanowski S, et al.** (2001) The alzheimer's disease activities of daily living international scale. International Psychogeriatrics 13: 163–181.
3. **Farias ST, Mungas D, Reed BR, Harvey D, Cahn-Weiner D, et al.** (2006) MCI is associated with deficits in everyday functioning. Alzheimer Dis Assoc Disord 20: 217–223.
4. **Schmitter-Edgecombe M, Woo E, Greeley DR** (2009) Characterizing multiple memory deficits and their relation to everyday functioning in individuals with mild cognitive impairment. Neuropsy-chology 23: 168–177.
5. **Wadley VG, Okonkwo O, Crowe M, Ross-Meadows LA** (2008) Mild cognitive impairment and everyday function: evidence of reduced speed in performing instrumental activities of daily living. Am J Geriatr Psychiatry 16: 416–424.
6. **Das B, Cook DJ, Schmitter-Edgecombe M, Seelye AM** (2012) Puck: An automated prompting system for smart environments: Toward achieving automated prompting–challenges involved. Personal Ubiquitous Comput 16: 859–873.
7. **Kaushik P, Intille SS, Larson K** (2008) User-adaptive reminders for home-based medical tasks. A case study. Methods Inf Med 47: 203–207.
8. **Sung J, Ponce C, Selman B, Saxena A** (2012) Unstructured human activity detection from rgbd images. In: Robotics and Automation (ICRA), 2012 IEEE International Conference on. pp. 842–849.
9. **Ye M, Zhang Q, Liang W, Zhu J, Yang R, et al.** (2013) A survey on human motion analysis from depth data. In: Grzegorzek M, Theobalt C, Koch R, Kolb A, editors, Time-of-Flight and Depth Imaging. Springer, volume 8200 of *Lecture Notes in Computer Science*, pp. 149–187.
10. **Poppe R** (2010) A survey on vision-based human action recognition. Image and Vision Computing 28: 976–990.
11. **Aggarwal J, Ryoo M** (2011) Human activity analysis: A review. ACM Comput Surv 43: 16: 1–16: 43.
12. **Moeslund TB, Hilton A, Krüger V** (2006) A survey of advances in vision-based human motion capture and analysis. Comput Vis Image Underst 104: 90–126.
13. **Camplani M, Salgado L** (2012) Efficient spatio-temporal hole filling strategy for kinect depth maps. In: Proc. SPIE Three-Dimensional Image Processing (3DIP) and Applications II. volume 8290, pp. 82900E-82900E-10.
14. **Dollár P, Rabaud V, Cottrell G, Belongie S** (2005) Behavior recognition via sparse spatio-temporal features. In: IEEE InternationalWorkshop on Performance Evaluation of Tracking and Surveillance (PETS). Beijing, China, pp. 65–72.
15. **Laptev I** (2005) On space-time interest points. Int J Comput Vision 64: 107–123.
16. **Jhuang H, Gall J, Zuffi S, Schmid C, Black MJ** (2013) Towards understanding action recognition. In: IEEE International Conference on Computer Vision (ICCV). pp. 3192–3199.
17. **Campbell LW, Bobick AF** (1995) Recognition of human body motion using phase space constraints. In: Proceedings of the Fifth International Conference on Computer Vision. Washington, DC, USA: IEEE Computer Society, ICCV '95, pp. 624–630.
18. **Lv F, Nevatia R** (2006) Recognition and segmentation of 3-d human action using hmm and multiclass adaboost. In: Proceedings of the 9th European Conference on Computer Vision - Volume Part IV. Berlin, Heidelberg: Springer-Verlag, ECCV'06, pp. 359–372.

19. **Xia L, Chen CC, Aggarwal JK** (2012) View invariant human action recognition using histograms of 3d joints. In: CVPR Workshops. pp. 20–27.

20. **Koppula HS, Gupta R, Saxena A** (2013) Learning human activities and object affordances from rgb-d videos. Int J Rob Res 32: 951–970.

21. **Sung J, Ponce C, Selman B, Saxena A** (2011) Human activity detection from rgbd images. In: Plan, Activity, and Intent Recognition. AAAI, volume WS-11-16 of *AAAI Workshops*, pp. 47–55.

22. **Wang J, Liu Z, Wu Y, Yuan J** (2012) Mining actionlet ensemble for action recognition with depth cameras. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 1290–1297.

23. **Yao A, Gall J, Gool L** (2012) Coupled action recognition and pose estimation from multiple views. Int J Comput Vision 100: 16–37.

24. **Yang X, Tian Y** (2012) Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on. pp. 14–19.

25. **Le QV, Karpenko A, Ngiam J, Ng AY** (2011) Ica with reconstruction cost for efficient overcomplete feature learning. In: Shawe-taylor J, Zemel R, Bartlett P, Pereira F, Weinberger K, editors, Advances in Neural Information Processing Systems 24. pp. 1017–1025.

26. **Li W, Zhang Z, Liu Z** (2010) Action recognition based on a bag of 3d points. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on. pp. 9–14.

27. **Vieira A, Nascimento E, Oliveira G, Liu Z, Campos M** (2012) Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences. In: Alvarez L, Mejail M, Gomez L, Jacobo J, editors, Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Springer Berlin Heidelberg, volume 7441 of *Lecture Notes in Computer Science*. pp. 252–259.

28. **Wang J, Liu Z, Chorowski J, Chen Z, Wu Y** (2012) Robust 3d action recognition with random occupancy patterns. In: Fitzgibbon A, Lazebnik S, Perona P, Sato Y, Schmid C, editors, Computer Vision ECCV 2012, Springer Berlin Heidelberg, Lecture Notes in Computer Science. pp. 872–885.

29. **Oreifej O, Liu Z** (2013) Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. 2013 IEEE Conference on Computer Vision and Pattern Recognition 0: 716–723.

30. **Zhang H, Parker L** (2011) 4-dimensional local spatio-temporal features for human activity recognition. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. pp. 2044–2049.

31. **Lei J, Ren X, Fox D** (2012) Fine-grained kitchen activity recognition using rgb-d. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing. New York, NY, USA: ACM, UbiComp '12, pp. 208–211.

32. **Jalal A, Uddin M, Kim J, Kim TS** (2011) Daily human activity recognition using depth silhouettes and r transformation for smart home. In: Abdulrazak B, Giroux S, Bouchard B, Pigot H, Mokhtari M, editors, Toward Useful Services for Elderly and People with Disabilities, Springer Berlin Heidelberg, volume 6719 of *Lecture Notes in Computer Science*. pp. 25–32.

33. **Johansson G** (1975) Visual motion perception. Scientific American 232: 76–88.

34. **Wang J, Liu Z, Chorowski J, Chen Z, Wu Y** (2012) Robust 3d action recognition with random occupancy patterns. In: Fitzgibbon A, Lazebnik S, Perona P, Sato Y, Schmid C, editors, Computer Vision ECCV 2012, Springer Berlin Heidelberg, Lecture Notes in Computer Science. pp. 872–885.

35. **Hinton GE, Osindero S, Teh YW** (2006) A fast learning algorithm for deep belief nets. Neural Comput 18: 1527–1554.

36. **Bengio Y, Lamblin P, Popovici D, Larochelle H** (2007) Greedy layer-wise training of deep networks. In: Schölkopf B, Platt J, Hoffman T, editors, Advances in Neural Information Processing Systems 19, Cambridge, MA: MIT Press. pp. 153–160.

37. **Hyvärinen A, Karhunen J, Oja E** (2001) Independent Component Analysis. John Wiley and Sons, Inc.

38. **Olshausen B, Field D** (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381: 607–609.

39. **Hyvarinen A** (2009) Natural image statistics a probabilistic approach to early computational vision. London: Springer-Verlag.

40. **Le QV, Ngiam J, Coates A, Lahiri A, Prochnow B, et al.** On optimization methods for deep learning. In: Getoor L, Scheffer T, editors, ICML. Omnipress, pp. 265–272.

41. **Wright J, Ma Y, Mairal J, Sapiro G, Huang T, et al.** (2010) Sparse representation for computer vision and pattern recognition. Proceedings of the IEEE 98: 1031–1044.

42. **Vedaldi A, Zisserman A** (2012) Efficient additive kernels via explicit feature maps. Pattern Analysis and Machine Intelligence, IEEE Transactions on 34: 480–492.

43. **Li W, Zhang Z, Liu Z** (2010) Action recognition based on a bag of 3d points. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on. pp. 9–14.

44. **Shotton J, Fitzgibbon A, Cook M, Sharp T, Finocchio M, et al.** (2013) Real-time human pose recognition in parts from single depth images. In: Cipolla R, Battiato S, Farinella GM, editors, Machine Learning for Computer Vision, Springer Berlin Heidelberg, volume 411 of *Studies in Computational Intelligence*. pp. 119–135.

45. **Zhu X, Li M, Li X, Yang Z, Tsien J** (2012) Robust action recognition using multi-scale spatial-temporal concatenations of local features as natural action structures. PLOS ONE 7: doi:10.1371/journal.pone.0046686.

46. **Zhang C, Tian Y** (2012) Rgb-d camera-based daily living activity recognition. Journal of Computer Vision and Image Processing 2.

47. **Ni B, Moulin P, Yan S** (2012) Order-preserving sparse coding for sequence classification. In: Fitzgibbon A, Lazebnik S, Perona P, Sato Y, Schmid C, editors, Computer Vision ECCV 2012, Springer Berlin Heidelberg, Lecture Notes in Computer Science. pp. 173–187.

48. **Yang X, Tian Y** (2014) Effective 3d action recognition using eigenjoints. J Vis Comun Image Represent 25: 2–11.

49. **Piyathilaka L, Kodagoda S** (2013) Gaussian mixture based hmm for human daily activity recognition using 3d skeleton features. In: Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on. pp. 567–572.

50. **Ni B, Pei Y, Moulin P, Yan S** (2013) Multilevel depth and image fusion for human activity detection. Cybernetics, IEEE Transactions on 43: 1383–1394.

51. **Gupta R, Chia AYS, Rajan D** (2013) Human activities recognition using depth images. In: Proceedings of the 21st ACM International Conference on Multimedia. New York, NY, USA: ACM, MM '13, pp. 283–292.

52. **Wang J, Liu Z, Wu Y, Yuan J** (2013) Learning actionlet ensemble for 3d human action recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on 36: 914–927.

53. **Kläser A, Marszal ek M, Schmid C** (2008) A spatio-temporal descriptor based on 3d-gradients. In: British Machine Vision Conference. pp. 995–1004.

54. **Müller M, Röder T** (2006) Motion templates for automatic classification and retrieval of motion capture data. In: Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, SCA '06, pp. 137–146.