Fusing Multiple Features for Depth-Based Action Recognition

YU ZHU, WENBIN CHEN, and GUODONG GUO, West Virginia University

Human action recognition is a very active research topic in computer vision and pattern recognition. Recently, it has shown a great potential for human action recognition using the three-dimensional (3D) depth data captured by the emerging RGB-D sensors. Several features and/or algorithms have been proposed for depth-based action recognition. A question is raised: Can we find some complementary features and combine them to improve the accuracy significantly for depth-based action recognition? To address the question and have a better understanding of the problem, we study the fusion of different features for depth-based action recognition. Although data fusion has shown great success in other areas, it has not been well studied yet on 3D action recognition. Some issues need to be addressed, for example, whether the fusion is helpful or not for depth-based action recognition, and how to do the fusion properly. In this article, we study different fusion schemes comprehensively, using diverse features for action characterization in depth videos. Two different levels of fusion schemes are investigated, that is, feature level and decision level. Various methods are explored at each fusion level. Four different features are considered to characterize the depth action patterns from different aspects. The experiments are conducted on four challenging depth action databases, in order to evaluate and find the best fusion methods generally. Our experimental results show that the four different features investigated in the article can complement each other, and appropriate fusion methods can improve the recognition accuracies significantly over each individual feature. More importantly, our fusion-based action recognition outperforms the state-of-the-art approaches on these challenging databases.

Categories and Subject Descriptors: I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Motion*; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Depth cues*; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Sensor fusion*; I.4.9 [**Image Processing and Computer Vision**]: Applications; I.4.7 [**Image Processing and Computer Vision**]: Feature Measurement—*Feature representation*; I.5.2 [**Pattern Recognition**]: Design Methodology—*Feature evaluation and selection*

General Terms: Algorithms, Experimentation, Performance, Human Factors

Additional Key Words and Phrases: RGB-D sensor, depth maps, action recognition, spatiotemporal features, skeleton, 4D descriptor, data fusion, decision level, feature level, feature selection

ACM Reference Format:
Yu Zhu, Wenbin Chen, and Guodong Guo. 2015. Fusing multiple features for depth-based action recognition. ACM Trans. Intell. Syst. Technol. 6, 2, Article 18 (March 2015), 20 pages.
DOI: http://dx.doi.org/10.1145/2629483

1. INTRODUCTION

Human action recognition has been an active research topic for more than two decades. It has a wide range of applications in the real world, such as Human-Computer Interaction (HCI), video surveillance, and video retrieval and security [Poppe 2010]. Most of the work has focused on action recognition using the videos captured in the visible spectrum [Turaga et al. 2008; Aggarwal and Ryoo 2011; Weinland et al. 2011]. Very recently,

Authors' addresses: Y. Zhu, W. Chen, and G. Guo (corresponding author), Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26506; emails: {yzhu4, wnchen}@mix.wvu.edu, guodong.guo@mail.wvu.edu.

DOI: http://dx.doi.org/10.1145/2629483

with the emerging, low-cost RGB-D sensors (e.g., the Kinect), human action recognition in three-dimensional (3D) data has gained great attention in computer vision. Depth maps provide many advantages over traditional color images/videos. For example, first the depth maps provide the 3D structure and shape information, which makes several problems easier to deal with, such as segmentation and detection. Second, depth images/videos are insensitive to illumination changes. Third, a quite accurate estimation of 3D human skeleton joint positions can be obtained from the depth data [Shotton et al. 2011]. Therefore, using the Kinect sensor, three channels (RGB, depth, and skeleton joint positions) of data are provided, which not only bring great benefits for robotics and human-centered computing but also give a broader scope for action recognition as well [Chen et al. 2013].

1.1. Related Work on Depth-Based Action Recognition

Depth-based action recognition has been actively studied since 2010 [Li et al. 2010]. Several algorithms and/or features have been proposed in the literature.

There are several representative works for action recognition with 3D depth data. Li et al. [2010] proposed an action graph for depth action recognition. A bag of 3D points sampled on depth data is used to encode the action posture, and the action graph is used to model the dynamics of actions. Wang et al. [2012b] proposed to combine the skeleton feature and local occupation feature, then learned an actionlets ensemble model to represent actions. A multiple kernel learning method is used to combine the actionlets. Wang et al. [2012a] also proposed a semilocal feature called Random Occupancy Patterns (ROPs), which is extracted from four-dimensional (4D) volumes. Sparse coding is utilized to encode the features and the Support Vector Machine (SVM) is used for classification. Vieira et al. [2012] proposed the space-time occupancy patterns to represent depth sequences. Both space and time axes are divided into multiple segments. Occupancy feature is computed in each cell, and a nearest-neighbor classifier is applied for recognition. A different approach based on Motion History Images (MHIs) was proposed by Yang et al. [2012]. The main idea is to use accumulated depth maps and compute Histogram of Gradients (HOGs) features to represent human actions. More recently, Oreifej and Liu [2013] proposed a method called HON4D to describe the depth sequence as a histogram captured in the 4D space of time, depth, and spatial coordinates. A 600-cell polychoron is used to quantize and represent the features. They used the SVM classifier and showed a good performance for action recognition. In Xia and Aggarwal [2013], a modified spatiotemporal feature based on Cuboids is proposed to capture the action motion and eliminate the flip noise on the depth video. A feature selection scheme is applied to the proposed features and then the selected features are fed into the SVMs for classification.

On the other hand, by modeling the skeleton joints, human actions in a depth video can be represented by the sequence of human postures and can be fed into the learning-based algorithms, such as Hidden Markov Models (HMMs) and Dynamic Time Warping (DTW). In Sempena et al. [2011] and Reyes et al. [2011], similar ideas were proposed where feature vectors defined by skeleton joints are used to model the human action, and DTW is applied to the resulted feature vector for action recognition. Yang and Tian [2012, 2013] proposed another skeleton feature to model the human action posture frame by frame based on computing posture feature, motion feature, and the offset feature, according to the relative frames in the action video. In Wang et al. [2012b], pairwise skeleton joint positions were computed in each frame to shape the motion of the human body. Xia et al. [2012] proposed an alternative feature called HOJ3D based on the skeleton joints. A coordinate based on skeleton joints is constructed, and multiple 3D bins are used to extract histogram features, by counting the number of joints in each bin. A HMM is used for action classification. Similarly, Miranda et al. [2012] used

the pose descriptor in a torso-based coordinate system and the SVM classifier to learn key poses. A decision forest is then used to recognize the action classes.

There are also works using both the RGB videos and depth maps for action recognition [Sung et al. 2012; Ni et al. 2011; Zhao et al. 2012]. The HOG feature was used as the descriptor for both RGB and depth images in Sung et al. [2012]. The hand positions, body pose, and motion features were also extracted from skeleton joints. A two-layer maximum-entropy Markov model is trained for classification.

1.2. Related Work on Data Fusion

Data fusion has been studied extensively, and shown great performance in many areas, including multisensor systems [Hall and Llinas 1997], multimedia analysis [Atrey et al. 2010], human identification [Ben-Yacoub et al. 1999], face recognition [Chang et al. 2003], handwriting recognition [Xu et al. 1992], biometrics [Ross and Jain 2003], and so forth. Various methods have been proposed and investigated for data fusion.

In Atrey et al. [2010], a survey of multimodal fusion for multimedia analysis was conducted. A categorization of different fusion methods with a thorough review of literature was given. They argued that the linear weighted fusion and SVM fusion methods are more often used because of the efficiency of these methods. Kittler [1998] proposed the theoretical framework for combining different classifiers, and Alkoot and Kittler [1999] investigated various rule-based fusion methods and experimentally validated the performance. Later on, a more extensive study was conducted in Kuncheva [2002] on classifier fusion strategies. In Kuncheva et al. [2001], a decision template is proposed to represent different fusion methods. In Ross and Jain [2003], different levels of fusion methods were presented for biometrics applications.

The usefulness of data fusion in other areas motivated us to explore fusion-based approaches for depth-based action recognition, which has not been well studied yet, to the best of our knowledge. In this article, fusion methods are explored at two levels: feature level and decision level. For feature-level fusion, the random forests, joint mutual information, and conditional mutual info maximization approaches are studied. For decision-level fusion, the majority voting, naive-Bayes combination, rule-based fusion, SVM-based fusion, and multiagent system approaches are studied. These approaches are described in Section 3.

1.3. Our Approach

We study whether and how fusion-based approaches can help to improve the action recognition accuracies in depth videos. The underlying assumption is that there are complementary features that can be extracted in depth videos. For the purpose of fusion, the complementary features should be extracted and combined together appropriately, otherwise the overall accuracy might not be improved even with multiple features. In our preliminary work [Zhu et al. 2013], the spatiotemporal features and skeleton features were combined using the random forest method [Breiman 2001]. This fusion approach improves the accuracies of depth-based action recognition significantly. In this article, we will further explore our fusion-based idea, by combining more features with diversity, investigating and evaluating a variety of fusion methods comprehensively, and using more databases to validate the fusion methods for generality.

Several representative data fusion methods are explored for our problem. We compare different fusion methods and find the best ones to solve the specific problem of depth-based action recognition.

The major contributions of our work include the following: (1) Evaluation of different features on depth-based action recognition, using the same experimental setting. There are four different features chosen for the evaluation. Two of them capture local motions and were originally proposed for action recognition in RGB videos, the third one extracts

features according to the skeleton joints positions, and the last one extracts features from 3D surface normal distributions. (2) Exploration of two levels of fusion schemes, that is, the feature-level and decision-level fusions. Several methods are explored at each fusion level. (3) Validation of the capability of different fusion methods and finding the appropriate one for depth-based action recognition on four challenging databases.

The remaining of the article is organized as follows: In Section 2, we introduce four different features for depth-based action characterization. Conceptually these features represent the action patterns from different aspects. In Section 3, we describe different fusion methods belonging to two different fusion levels. The experiments are conducted in Section 4 with comparisons to the state-of-the-art methods. Finally, we draw conclusions.

2. FEATURE EXTRACTION AND DESCRIPTION ON DEPTH DATA

Feature extraction and representation is an important step for action recognition. To develop our fusion-based approach to depth-based action recognition, we use multiple, diverse representations to characterize the action patterns. In our preliminary work [Zhu et al. 2013], the spatiotemporal interest point features (STIPs) and skeleton features are extracted and fused for action recognition in 3D. Here we expand the preliminary work by integrating more features, and executing a systematic study of various fusion methods. The 4D descriptor (HON4D) can characterize the normal distributions of the 3D surfaces in performing an action [Oreifej and Liu 2013], and the space-time autocorrelation (STACOG) feature can represent statistical correlations of local derivatives [Kobayashi and Otsu 2012]. Totally there are four different features we have investigated to characterize depth action patterns from different aspects, and develop our fusion-based recognition framework. We introduce these features in the following.

2.1. Spatiotemporal Interest Point Features

STIP features capture the complex motion of human actions. These features are quite popular for action representation in color videos [Shabani et al. 2012], but not often in depth data. Here we adapt some STIP features to depth sequences. We attempted several combinations of the detectors and descriptors and find the best ones for depth action characterization. Because of the space limits, we briefly describe the STIP features that we used, and only the best one will be reported in each dataset experimentally (see Section 4).

The Harris3D detector [Laptev and Lindeberg 2004] computes the locations of the interest points based on a second-moment matrix of gradients with the convolution of spatiotemporal Gaussian kernel in the video sequence. It locates the spatiotemporal volumes where large variations of gradient exist along space and temporal directions. Specifically, a spatiotemporal second-moment matrix is computed from a video sequence f,

$$\mu = g(\cdot) \times \begin{pmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{pmatrix}, \tag{1}$$

where $g(\cdot)$ is a Gaussian weight function and L is the convolution of f with the spatiotemporal gradient. The interest point locations are determined by computing the local maxima of the response function $R = det(\mu) - k \cdot trace^3(\mu)$.

Another detector applied in this article is the Hessian detector [Willems et al. 2008]. The Hessian detector uses the response function $S = |det(H)|$ to measure the strength of each interest point, where H is the Hessian matrix.

Given the detected locations, various descriptors can be used to characterize the local motion patterns. The *HOG/HOF* descriptor was proposed in Laptev et al. [2008] to describe local human motion in RGB videos. It computes the HOG and Histogram of Optical Flow (HOF) in each local volume. Klaser et al. [2008] extends the HOG to HOG3D descriptor, which computes the 3D gradient and constructs a histogram as the feature vector. It computes the histogram of 3D gradient orientations. The integral videos can be precomputed to efficiently compute the gradients and combine both shape and motion information at the same time. The *extended SURF (ESURF)* descriptor [Willems et al. 2008] is an extension of the SURF [Bay et al. 2006] for action representation.

2.2. STACOGs

A method called the STACOG was proposed with the bag-of-frame-features computation in Kobayashi and Otsu [2012] to extract motion features from RGB action videos. It is computed with the frame-based STACOG features sampled densely along the time axis. In order to extract the feature, space-time gradient vector is calculated by taking derivatives (I_x, I_y, I_t) at each local space-time volume, around each space-time point. The gradients can be represented by the angles $\theta = \arctan(I_x, I_y)$ and $\phi = \arcsin(I_t/m)$, where $m = \sqrt{I_x^2 + I_y^2 + I_t^2}$ is the magnitude. Then a histogram is constructed by binning the gradients in a unit sphere. The histogram is defined as Space-Time Orientation Coding (STOC) vector. The autocorrelation functions can be computed for the space-time gradients:

$$F_0 = \Sigma_r m(r)h(r), \tag{2}$$

$$F_1(a_1) = \Sigma_r min[m(r), m(r + a_1)]h(r)h(r + a_1)^T, \tag{3}$$

where r is the reference point (x, y, t), h is the STOC vector, and a_1 is the displacement vector from the reference point, and F_0 and F_1 are the zero- and first-order autocorrelations. We adapt the STACOG features from RGB to depth data.

2.3. EigenJoints Feature

Human skeleton joints can be detected fast on depth data [Shotton et al. 2011]. The skeleton joint positions can be viewed as an alternative modality for action characterization. Features can be computed from skeleton joint positions to represent the action patterns, which are usually not available in color videos. Several features extracted from skeleton joints are proposed for depth-based action recognition such as Wang et al. [2012b], Xia et al. [2012], and Yang and Tian [2012]. We implemented these features and found that the method in Yang and Tian [2013] gives a better representation. Thus, we chose the histogram of the skeleton joints features to represent human actions.

Specifically, the features consist of three parts: (1) current posture: pairwise joint distances in current posture compared in the current frame; (2) motion: joint differences between current posture and the previous one; and (3) offset: joint differences between current posture and the original (in the first frame). Denote each 3D skeleton joint by $p_i = (x_i(t), y_i(t), z_i(t))$ at frame t. The number of skeleton joints in each frame is denoted as N. The feature vector can be computed by

$$f = [f_{current}\ f_{motion}\ f_{offset}], \tag{4}$$

$$f_{current} = \{p_i - p_j \mid i \neq j;\ i, j = 1..N\}, \tag{5}$$

$$f_{motion} = \{p_i(t) - p_i(t-1) \mid i = 1..N\}, \tag{6}$$

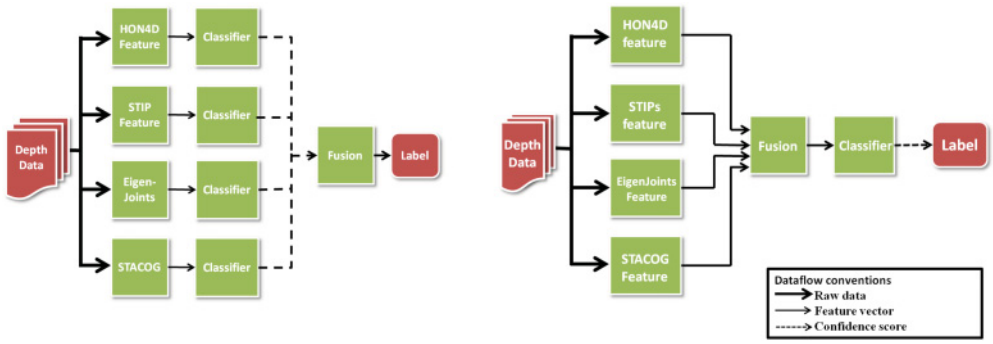

Fig. 1. Illustrate the schemes of the decision-level/late fusion (left) and feature-level/early fusion (right) in combining different features for 3D action recognition.

$$f_{offset} = \{p_i(t) - p_i(0) \mid i = 1..N\}, \tag{7}$$

where $p(0)$ denotes the original posture in each action sequence.

2.4. Histogram of Oriented 4D Normals (HON4D)

The depth data can be represented as a surface in 4D space with a set of points (x, y, t, z), where z is the depth value of the point. Then the normal to the surface can be computed by

$$n = \nabla S = \left(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, \frac{\partial z}{\partial t}, -1\right)^T. \tag{8}$$

The surface normals over all voxels in the depth sequence can be used for action representation [Oreifej and Liu 2013]. A 600-cell polychoron in 4D space is used to quantize the 4D normals to derive the feature. The HON4D will be combined with other features together to develop our fusion-based approach.

3. FUSION METHODS

Data fusion has gained much attention in recent years. It can be accomplished at different levels [Ross and Govindarajan 2005], for example, early fusion (sensor, feature levels), where fusion is conducted before matching, and late fusion (rank, score, decision levels), where fusion is executed after matching. We present several fusion methods at both the decision and feature levels to solve our problem of depth-based action recognition (see Figure 1 for an illustration).

3.1. Feature-Level Fusion

According to Ross and Jain [2003], feature-level fusion is usually conducted through feature normalization and feature selection or transformation because of the relationship between different feature sets and the curse of dimensionality [Donoho et al. 2000]. The objective of feature-level fusion is to combine different feature sets to generate a new feature vector. For feature selection, we adopt two representative approaches from Brown et al. [2012]. Totally, we explore three methods for feature-level fusion to deal with the problem of depth-based action recognition.

3.1.1. Random Forests. Random Forests (RFs) [Breiman 2001] are usually considered as a classifier (or regressor) using tree predictors in which each tree splits the data depends on the randomly selected features. The RFs can be considered as a fusion method where the fusion is done through randomly selecting and combining different features at each tree node. There are many nice properties of the RFs method:

(1) robustness to noise, (2) efficiency for classification, and (3) improvement of accuracy by growing multiple trees and voting for the most possible class. Here, we use the RFs for fusion of distinct features and action classification jointly.

Let the feature vector be $v \in \mathbb{R}^N$, where the number of the features for each sample is N. A number $n < N$ is specified at each node of the tree, where n features are randomly selected to determine the split of that node. The randomly selected n features are used in the tree node.

The best split is determined by the information gain using these features. Several decision trees are growing to generate a forest, and each tree grows until it reaches the maximum tree depth max_{dep}, or the tree node receives the given number of minimum samples min_{node}. In the leaf nodes, the probabilistic distribution for each class is computed. In this way, the feature fusion is executed randomly and naturally in the tree building process.

In recognition phase, each new observation x goes down to one of the leaf nodes in each tree, denoted as $l(t, x)$, which contains the distribution P_n of all classes. The RFs classifier chooses the class label that gets the most votes over all the trees:

$$\hat{c} = \arg\max_j \frac{1}{T} \sum_{t=1}^{T} p^j_{l(t,x)}, \tag{9}$$

where $\hat{c}$ is the predicted class label, T is the total number of trees, and $l(t, x)$ is the leaf node of tree t where the test sample x falls into. $p^j_{l(t,x)}$ is the posterior probabilities for class j at leaf node $l(t, x)$, $p^j_n = \frac{|S_j|}{|S|}$, where $|S|$ is the total number of samples in this leaf node and $|S_j|$ is the number of samples of class j in S.

3.1.2. Joint Mutual Information. Joint Mutual Information (JMI) was proposed to select a discriminative feature subset from the feature pool [Yang and Moody 1999]. In our case, different features are normalized and concatenated to construct the feature pool. We investigate if the feature selection by JMI can fuse different features for our action recognition in 3D.

The mutual information between X and Y can be defined [Yang and Moody 1999] by

$$I(X;Y) = H(X) - H(X|Y) = \Sigma_{x\in X}\Sigma_{y\in Y}\, p(xy) \log \frac{p(xy)}{p(x)p(y)}, \tag{10}$$

where $H(X)$ denotes the entropy of the random variable:

$$H(X) = -\Sigma_{x\in X} p(x) \log p(x), \tag{11}$$

and the conditioned form of entropy $H(X|Y)$ can be written as

$$H(X|Y) = -\Sigma_{y\in Y}\, p(y) \Sigma_{x\in X} p(x \mid y) \log p(x \mid y). \tag{12}$$

As a feature selection method, the JMI can be viewed as using a criterion J to measure how useful a feature or feature subset is when used by a classifier. This criterion is defined as

$$J_{jmi}(\mathbf{v}_k) = \Sigma_{\mathbf{v}_k\in S} I(\mathbf{v}_k \mathbf{v}_j; Y), \tag{13}$$

where S is the previously selected feature set, Y is class label, and $\mathbf{v}_k$ is the kth feature in the feature vector $\mathbf{v}$.

The JMI pairs the candidate features in $\mathbf{v}_k$ with each newly selected feature to increase the complementary information between features [Brown et al. 2012].

3.1.3. Conditional Mutual Info Maximization. Conditional Mutual Info Maximization (CMIM) is an alternative feature selection method [Fleuret 2004]. Different from JMI, the CMIM method adds a new feature only if the optimal value based on a criterion is larger than using the features already selected, such that the information has not been brought by any already selected features. This criterion is given by

$$J_{cmim}(\mathbf{v}_k) = min_{\mathbf{v}_k \in S}\left[I(\mathbf{v}_k; Y \mid \mathbf{v}_j)\right], \tag{14}$$

which can be equally written as

$$J_{cmim}(\mathbf{v}_k) = I(\mathbf{v}_k; Y) - max_{\mathbf{v}_k \in S}[I(\mathbf{v}_k; \mathbf{v}_j) - I(\mathbf{v}_k; \mathbf{v}_j \mid Y)], \tag{15}$$

where S is the previously selected feature set, Y is class label, and $\mathbf{v}_k$ is the kth feature in the feature vector $\mathbf{v}$.

Using the feature selection procedures, different feature sets can be fused together to feed into a classifier for action recognition.

3.2. Decision-Level Fusion

Different from feature-level fusion, the decision-level fusion or late fusion deals with the fusion process on the decision level, where classifier outputs are combined to make the final decision.

Let $\mathbf{v} \in \Re^n$ be a feature vector extracted from an input pattern, and let $\{\omega_1, \omega_2, \ldots, \omega_c\}$ be the class labels of c classes. For a *classifier D,* the output of D given the input pattern $\mathbf{v}$ can have two representations: $D(\mathbf{v}) = [d_1(\mathbf{v}), d_2(\mathbf{v}), \ldots d_c(\mathbf{v})]$, where $d_i(\mathbf{v}) \in [0, 1],\ i = 1..c$, is an estimate of the posterior probability $P(\omega_i|\mathbf{v})$ offered by classifier D. The other is $D(\mathbf{v}) = \omega_i,\ i \in \{1 \ldots c\}$, where ω_i is the class label given by classifier D.

When there exist totally L classifiers, denoted by $\{D_1, \ldots, D_L\}$, the representations for multiclassifiers can be given [Kuncheva et al. 2001] by

(1) Decision Profile:

$$DP(\mathbf{v}) = \begin{bmatrix} d_{1,1}(\mathbf{v}) & \cdots & d_{1,j}(\mathbf{v}) & \cdots & d_{1,c}(\mathbf{v}) \\ \cdots & & \cdots & & \cdots \\ d_{i,1}(\mathbf{v}) & \cdots & d_{i,j}(\mathbf{v}) & \cdots & d_{i,c}(\mathbf{v}) \\ \cdots & & \cdots & & \cdots \\ d_{L,1}(\mathbf{v}) & \cdots & d_{L,j}(\mathbf{v}) & \cdots & d_{L,c}(\mathbf{v}) \end{bmatrix}, \tag{16}$$

where $d_{i,j}(\mathbf{v})$ denotes the estimate of posterior probability of class j made by classifier D_i, $i \in [1,\ L]$, $j \in [1,\ c]$.

(2) Decision Vector:

$$DV(\mathbf{v}) = \left[\omega_1^c\ \cdots\ \omega_i^c\ \cdots\ \omega_L^c\right], \tag{17}$$

where ω_i^c is the class label given by classifier D_i. Given the preceding representations, we will present specific fusion methods for decision-level fusion as follows.

Popular methods for decision-level fusion include (weighted) majority voting, naive-Bayes combination, weighted sum, minimum, maximum, median, product, SVM-based fusion, and multiagent system [Kuncheva et al. 2001; Ross and Jain 2003; Kittler 1998; Atrey et al. 2010; Da C. A. and Fairhurst 2009].

3.2.1. Majority Voting. Majority voting is one of the most common approaches for decision-level fusion [Kittler 1998]. The idea is to assign the final class label by "voting" over the different classifiers, and select the one that the majority classifiers agree on. For each *classifier D_i in L classifiers,* the output of D_i given an input pattern $\mathbf{v}$ is a predicted class label ω_i^c, and the final class label is assigned according to which class label is the majority in the decision vector $DV(\mathbf{v}) = [\omega_1^c\ \cdots\ \omega_i^c\ \cdots\ \omega_L^c]$. If more than one label occurs, the class label will be randomly selected from those labels.

It is reasonable to assign different weights to the decisions made by different classifiers, when the performance of these classifiers is quite different. Larger weights can be assigned to the decisions made by more accurate classifiers. So the discriminant function for class ω_k can be rewritten as

$$g_k = \Sigma_{i=1}^{L} w_i s_i^k, \tag{18}$$

where w_i is the weight of classifier D_i, and s_i^k is an indicator function defined as

$$s_i^k = \begin{cases} 1, & \text{if the classifier } D_i \text{ outputs class label } \omega_k \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

The weights for each classifier can be learned in a validation set.

3.2.2. Naive-Bayes Combination. The naive-Bayes fusion method relies on transforming decision labels into probabilities, under the assumption that different classifiers are mutually independent in the multiclassifier system [Xu et al. 1992]. The first step is to construct Confusion Matrix CM_i for each classifier D_i. Each element on the kth row and sth column denotes the number of patterns of the training dataset of which the true label is ω_k but is assigned to class ω_s by D_i. The next step is to construct the Label Matrix LM_j for each classifier, where each element is defined by

$$lm_{k,s}^i = \hat{P}(\omega_k \mid D_i(\mathbf{v}) = \omega_s) = \frac{cm_{k,s}^i}{cm_{\cdot,s}^i}, \tag{20}$$

where $cm_{k,s}^i$ denotes the element on the kth row and sth column of CM_i, and $cm_{\cdot,s}^i$ denotes the sum of the sth column of CM_i.

For each pattern $\mathbf{v}$, classifier D_j outputs a class label; the estimated probability of the class label ω_i is computed by

$$\theta_i(\mathbf{v}) = \Pi_{j=1}^{L} P(i \mid D_j(\mathbf{v}) = s_j) = \Pi_{j=1}^{L} lm_{i,s_j}^j. \tag{21}$$

We found that the preceding multiplication in Xu et al. [1992] cannot work well for our problem. Replacing it with summation can result in much better results:

$$\theta_i(\mathbf{v}) = \Sigma_{j=1}^{L} lm_{i,s_j}^j. \tag{22}$$

3.2.3. Sum, Minimum, Maximum, Median, and Product Rules. These fusion methods can be categorized as rule-based methods [Kittler 1998; Kuncheva 2002]. These basic rules are defined to combine multiple classifiers and can generally perform well if the quality of temporal alignment between different modalities is good [Atrey et al. 2010].

Denote θ the predicted class label, and $P(\omega_j \mid \mathbf{d}_i)$ the *posteriori* probability of θ assigned as class ω_j by the measurement vector $\mathbf{d}_i$ from the ith classifier. We have

(i) Sum rule: Assign $\theta \to \omega_j$ if

$$(1-L)P(\omega_j) + \Sigma_{i=1}^{L} P(\omega_j \mid \mathbf{d}_i) = \max_{j=1}^{c} \left[(1-L)P(\omega_j) + \Sigma_{i=1}^{L} P(\omega_j \mid \mathbf{d}_i) \right]. \tag{23}$$

(ii) Maximum rule: Assign $\theta \to \omega_j$ if

$$\max_{i=1}^{L} P(\theta = \omega_j \mid \mathbf{d}_i) = \max_{j=1}^{c} \max_{i=1}^{L} P(\theta = \omega_j \mid \mathbf{d}_i). \tag{24}$$

(iii) Minimum rule: Assign $\theta \to \omega_j$ if

$$\min_{i=1}^{L} P(\theta = \omega_j \mid \mathbf{d}_i) = \max_{j=1}^{c} \min_{i=1}^{L} P(\theta = \omega_j \mid \mathbf{d}_i). \tag{25}$$

(iv) Product rule: Assign $\theta \rightarrow \omega_j$ if

$$P^{-(L-1)}(\omega_j)\Pi_{i=1}^{L} P(\theta = \omega_j \mid \mathbf{d}_i) = \max_{j=1}^{c} P^{-(L-1)}(\omega_j)\Pi_{i=1}^{L} P(\theta = \omega_j \mid \mathbf{d}_i). \tag{26}$$

(v) Median rule: Assign $\theta \rightarrow \omega_j$ if

$$\max_{i=1}^{L} P(\theta = \omega_j \mid \mathbf{d}_i) = \max_{j=1}^{c} median_{i=1}^{L} P(\theta = \omega_j \mid \mathbf{d}_i). \tag{27}$$

3.2.4. SVM-Based Fusion. SVM-based fusion is a decision-level fusion method that combines multiple individual SVM classifiers by a new SVM classifier using their confidence scores. Basically, given the training samples $\mathbf{x}_i \in \Re^n$ and the class labels $y_i \in \{-1, 1\}$, the SVM [Vapnik 1998] has the following optimization problem:

$$\min_{W,b,} \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C\Sigma_{i=1}^{l}\xi_i,\ s.t.\ y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0,\ i = 1, \ldots, l. \tag{28}$$

The SVM-based fusion is based on a two-layer structure, where the input to the higher-layer SVM is the confidence scores given by individual lower-layer SVM classifiers [Atrey et al. 2010]. Each lower-layer SVM classifier uses one feature set and outputs the confidence scores instead of class labels. The confidence scores of all individual classifiers are combined into a new feature vector, and then fed into the higher-layer SVM for final classification.

3.2.5. Multiagent System. A Multiagent System (MAS) was proposed to solve the multiclassifier classification problem [Da C. A. and Fairhurst 2009]. An auction-based negotiation is used for classification. The idea is that all agents/classifiers are considered to be the buyers who are trying to reach an agreement in relation to an input pattern. Specifically, the confidence scores of each agent/classifier D_i given an input pattern $\mathbf{v}$ is computed in the first step, denoted as $D_i(\mathbf{v}) = [d_{i,1}(\mathbf{v}), d_{i,2}(\mathbf{v}), \ldots, d_{i,c}(\mathbf{v})]$, and the class with maximum confidence of each agent is selected as the chosen class for that agent, the maximum confidence value of each agent is denoted as $[d_{1,m_1}, d_{2,m_2}, \ldots, d_{L,m_L}]$. Given L agents, the cost for the agents is defined as a vector $\mathbf{c^j} = \{c_1^j, c_2^j, \ldots, c_L^j\}$. The cost for the ith agent is computed by

$$c_i^j = \begin{cases} d_{j,m_j} - d_{j,i}, & i \neq j \\ d_{j,i} - \Sigma_{k \neq i}^{c} d_{i,k}, & i = j. \end{cases} \tag{29}$$

The agent with the highest cost $\arg_j \max_{j=1}^{L} c_j^j$ is considered the loser. Then the confidence of the chosen class of all agents is changed according to the difference between the current confidence and their responding cost: $d_{i,j} = d_{i,j} - c_j^j$. After the confidence values of each agent have been updated, the agent can decide whether or not to keep the chosen class, according to the current confidence values. When an agent loses twice in succession, it is then discarded from the negotiation. The remaining agents continue this process, until only one agent remains in the auction.

4. EXPERIMENTS

In this section, we conduct experiments on four challenging depth-based action databases, using four different features and various fusion methods. First, we transformed the depth data into gray level depth videos and projected all the skeleton joint positions into image coordinates. After this preprocessing, feature extraction is conducted on each database. Every action sequence is represented by four different feature vectors, that is, the STIP, STACOG, EigenJoints, and HON4D, respectively. For feature quantization, the K-means clustering method is used to derive histograms for

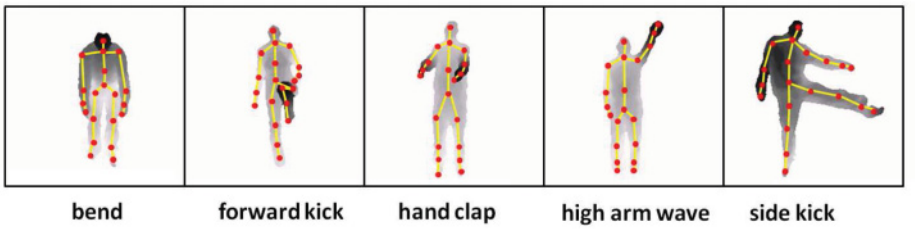

Fig. 2. Some examples (with skeleton joints shown) in the MSRAction-3D dataset.

each feature in each action video. Before investigating the comprehensive fusion-based framework, we analyze the performance of the individual feature for action recognition. Note that the same training and test sets are used for both individual features and various fusion methods. The SVM is used as the supervised classifier for each individual feature vector. After evaluating the individual features, various fusion methods are investigated. On feature-level fusion, different feature vectors are normalized first before fusion. RFs can both select features and execute the classification task. For other feature-level fusion methods, the SVM is used as the classifier. On decision-level fusion, the SVM classifiers are trained for each feature independently, from which multiple decisions are made for each test pattern. The confidence scores or the intermediate decision class labels are transmitted into the fusion engine for fusion with different methods.

In the following, we introduce the four databases first, followed by some experimental settings, and then the experimental results. We also provide some analysis and discussions about the experimental results.

4.1. Databases

Four challenging 3D action databases are used in our experiments to evaluate the performance of different fusion approaches. In brief, these four databases capture various human actions/activities under different circumstances (viewpoints, locations, backgrounds, etc.) with different considerations (# of actions, # of human subjects, different scenarios, etc.) and complexity. In additon, the performed actions are quite different in these databases. More details are given in the following.

The MSRAction3D dataset [Li et al. 2010] captures 20 human actions using a depth camera similar to the Kinect sensor. In total, 10 subjects were asked to perform 20 action classes three times each. Each video clip is of resolution 640×480 at 15fps. We used all of the 557 video clips, along with the skeleton joint locations provided by Li et al. [2010]. In our experiment, we follow the same settings of "cross-subjects" as in Li et al. [2010]. The whole dataset was divided into three subsets; half of the subjects are used for training while the other half are used for testing. The final accuracy on this dataset is the average of the accuracies over the three subsets. See Figure 2 for some example images in this dataset.

The MSRDailyActivity3D dataset [Wang et al. 2012b] was collected with human daily activities by the Kinect. In total, there are 16 activities in this dataset: *drink, eat, read book, call cellphone, write on a paper, use laptop, use vacuum cleaner, cheer up, sit still, toss paper, play game, lay down on sofa, walk, play guitar, stand up, and sit down*. Each subject performed an activity in two scenarios, one "sitting on sofa" and the other "standing.' The number of activity videos is 320. Three types of data (i.e., the RGB, depth, and skeleton joint positions) are provided in this dataset. The specific subject IDs which are used in training and testing are listed in Table I. See Figure 3 for examples of depth images in this dataset.

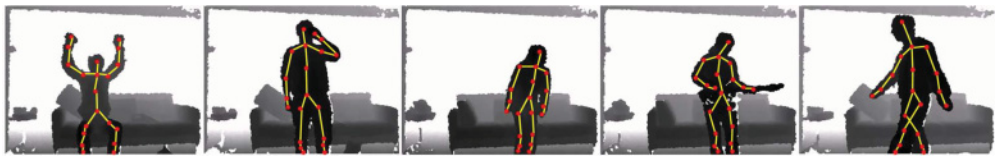

Fig. 3. Some examples (with skeleton joints shown) from the MSRActivity3D dataset. The actions (from left to right) are cheer up, drink, stand up, play guitar, and walk.

Fig. 4. Some example images (with skeleton joints shown) in the UTKinect-Action dataset. The actions (from left to right) are carry, clap hands, pick up, push, and wave.

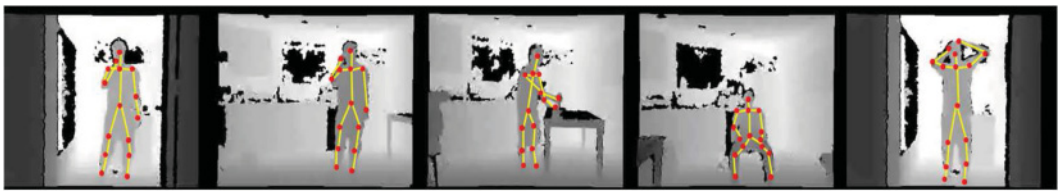

Fig. 5. Some example images (with skeleton joints shown) from the CAD-60 dataset. The actions (from left to right) are brush teeth, talk on phone, cook, relax on couch, and wear contact lens.

The UTKinect-Action dataset [Xia et al. 2012] contains 10 different action classes performed by 10 subjects, collected by a stationary Kinect sensor. The 10 action classes are *walk, sit down, stand up, pick up, carry, throw, push, pull, wave hands, clap hands.* Depth sequences are provided with resolution 320×240, and skeleton joint locations are also provided in this dataset. In our experiments, we used the cross-subjects scheme with half of the subjects for training while the remaining are for testing (see Table I), which is different from the leave-one-out scheme used in Xia et al. [2012] where more subjects were used for training. Some example images of this dataset are shown in Figure 4.

The Cornell Activity Dataset-60 (CAD-60) [Sung et al. 2012] has 60 RGB-D sequences collected by the Kinect; each video is of length about 45s. In this dataset, four different subjects performed 12 different activities in five locations. The five locations are *office, kitchen, bedroom, bathroom, and living room*. To reduce the computational complexity, we first subsample each video to the length about 500 frames. Then we follow the same procedure of "new person" as in Sung et al. [2012] for training and testing (see Table I). See Figure 5 for some example images of this dataset.

4.2. Experimental Settings

We follow the same experimental settings as our conference paper [Zhu et al. 2013]. Specifically, for the STIP feature extraction, Harris3D detectors with HOG/HOF descriptors are used for the MSRAction3D dataset. The Harris3D detector and HOG3D descriptor are used for the UTKinect-Action dataset. On the CAD-60 dataset, the Hessian detector and ESURF descriptor are adopted. On the MSRDailyActivity3D dataset, the Hessian detector and HOG3D descriptor are used to extract local features. These are the best STIP features in each dataset, based on a systematic evaluation. Because

Table I. Subject IDs Used for Training and Testing, Respectively, in Each Database

Dataset	Training Subject IDs	Testing Subject IDs
MSRAction3D	2, 3, 5, 7, 9	1, 4, 6, 8, 10
MSRDailyActivity3D	1, 6, 8, 9, 10	2, 3, 4, 5, 7
UTKinect-Action3D	1, 2, 3, 4, 8	5, 6, 7, 9, 10
CAD-60	1, 3, 4	2

of the space limit, we do not present the detailed evaluations here. The K-means clustering method is applied to quantize the STIP features into histograms. Empirically, we set $K = 100$ to get the clusters or keywords. For the skeleton joints feature, the bag-of-words scheme is used for quantization. In order to get the STACOG feature, we follow the settings in Kobayashi and Otsu [2012], adopting a hemisphere for coding the gradients. Four orientation bins along the longitude are arranged on each of five layers along the latitude, and one bin is located at pole; totally there are $B = 21$ bins. We restrict $N \in \{0, 1\}$, where zeroth-order F_0 and the first-order feature F_1 are considered. The dimensionality of STACOG features is $d = B + 13B^2 = 5{,}754$. The Linear Discriminant Analysis (LDA) is performed for dimension reduction. For the HON4D feature, each video sequence is divided into $5 \times 4 \times 3$ spatiotemporal cells ($5 \times 4 \times 2$ cells for the UTKinect-Action dataset because this dataset has typically shorter video clips) and a separate HON4D feature is obtained for each cell. The final descriptor is a concatenation of the HON4Ds obtained from all the cells. We used the code provided by the authors [Oreifej and Liu 2013] for the HON4D feature extraction on all databases, except the MSRDailyActivity3D dataset. Some kinds of local HON4D features were provided by the authors in Oreifej and Liu [2013], which may be useful to deal with the changes of subjects' locations and temporal motions for the actions in the MSRDailyActivity3D dataset. Because the dimensionality of the HON4D feature is much higher than the other three features, we use PCA to reduce the HON4D feature dimension to 100 in our experiments. For feature normalization, we employ the Gaussian normalization scheme. For the classifiers used in the experiment, the SVMs with χ^2 kernel are used. For the RFs the number of trees can be selected from $[1, 500]$, and the number of features used in each split can be selected from $[3, 60]$. The related parameters were adjusted in a tuning set, which is about 20% of the training examples in each training dataset.

4.3. Gaussian Normalization

After the feature representation, the data range of different features might be very different; a direct fusion of such features might not perform well. The Gaussian normalization is used to map different features into a comparable range. Suppose there are M video sequences in the database; the four types of features can form an $M \times N$ feature matrix $F = f_{ij}$, where f_{ij} is the jth feature component in feature vector $f_{i,\cdot}$; each feature vector is of N dimensions. Our goal is to normalize the entries in each column $f_{\cdot,j}$ to the same range so as to ensure that each individual feature component receives equal weight in determining the similarity between two vectors. We compute the mean μ_j and standard deviation σ_j of the sequence and then normalize the original sequence into a normal distribution $N \sim (0, 1)$ as follows:

$$f'_{ij} = \frac{f_{ij} - \mu_j}{\sigma_j}, \tag{30}$$

then the probability of a feature component value in the range of $[-1, 1]$ is approximately 99%. An additional shift will guarantee that 99% of feature values are within

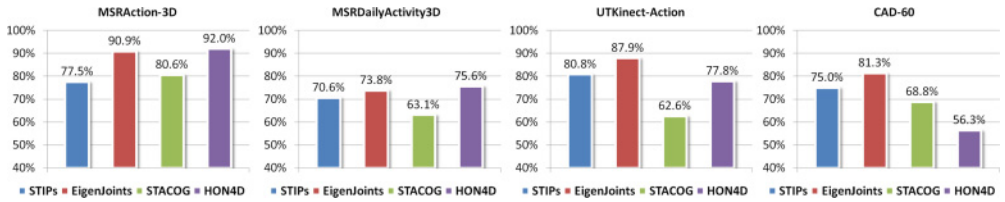

Fig. 6. An evaluation of the individual features on four databases: MSRAction3D, MSRDailyAcitivity3D, Kinect-Action, and CAD-60. The same training and test data are used for each feature to have a fair comparison.

[0,1]:

$$\tilde{f}_{ij} = \frac{f'_{ij} + 1}{2}. \tag{31}$$

After this shift, we can consider that all of the feature component values are within the range of [0,1]. Therefore, this normalization process ensures the same range of the feature components when different types of feature are used.

4.4. Experimental Results

We present the experimental results of individual features first, and then the fusion results based on different fusion methods.

4.4.1. Results of Individual Features. We first investigate the individual features on the depth action datasets. The bag-of-words approach is used for histogram construction and the SVM is used as the classifier. In order to explore the capability of different features, we use the bag-of-feature approach for the STACOG feature, other than the bag-of-frame as in Kobayashi and Otsu [2012]; for the HON4D feature, we adopted the uniform settings, without using the skeleton information for local nonuniform quantization as in Oreifej and Liu [2013]. Different from Yang and Tian [2012] where the naive-Bayes nearest-neighbor classifier was used, we extract the skeletons and then construct the histogram features for the SVM classifier.

The experimental results on four databases using four different features are shown in Figure 6. The HON4D feature performs the best on the MSRAction3D (accuracy: 92.0%) and MSRDailyActivity3D (accuracy: 75.6%) databases, while on the other two databases, its accuracies are lower than some other features. On the other hand, the EigenJoints feature achieves the best results on the UTKinect-Action (accuracy: 87.9%) and CAD-60 (accuracy: 81.3%) databases. This feature performs the second best in the other two databases. It can also be observed that the STIP feature and the STACOG feature exhibit comparable performance, although they are not the best on these four databases. This fair comparison of different features has not been carried out in previous research. Our evaluation tells that no single feature can perform the best in all databases. This is also one of the reasons why we are interested in studying the fusion-based approach for depth-based action recognition.

After evaluating the individual features on the depth action databases, we conduct experiments applying various fusion methods, and show the performance using the same data (training and test sets) on the four databases.

4.4.2. Fusion Results on MSRAction-3D Dataset. The experimental results on MSRAction-3D dataset using various fusion methods are shown in Table II. The highest accuracy of 98.2% is achieved by the sum rule based fusion method, which is significantly better than any single features. For example, the accuracy of the best single feature HON4D is 92.0%. We can also observe that in the feature-level fusion scheme, the RFs fusion

Table II. The Recognition Accuracies of Individual Features and Various Fusion Methods on Four Datasets The Decision-Level Fusion Methods Include the MAS, MAJ, SVM, SUM, MIN, MAX, MED, and PRODUCT, and the feature-level fusion methods include the RFs, JMI, and CMIM (see text for the meaning of each fusion method).

Method		Accuracy			
		MSRAction3D	UTKinect	CAD-60	MSRActivity
Single Feature	STIPs	77.5%	80.8%	75.0%	70.6%
	EigenJoints	90.9%	87.9%	81.3%	73.8%
	STACOG	80.6%	62.6%	68.8%	63.1%
	HON4D	92.0%	77.8%	56.3%	75.6%
Decision-Level Fusion	MAS	93.3%	83.8%	68.8%	59.4%
	MAJ	96.3%	**92.9%**	**87.5%**	**88.1%**
	SVM	97.3%	86.9%	81.3%	79.4%
	SUM	**98.2%**	91.9%	68.8%	85.6%
	MIN	90.6%	61.6%	50.0%	58.8%
	MAX	95.2%	88.9%	68.8%	72.5%
	MED	96.3%	90.9%	62.5%	70.0%
	PRODUCT	96.4%	86.9%	68.8%	72.5%
Feature-Level Fusion	RFs	**97.3%**	**92.9%**	**87.5%**	**88.8%**
	JMI	94.2%	85.9%	81.3%	69.4%
	CMIM	94.6%	85.9%	81.3%	70.0%

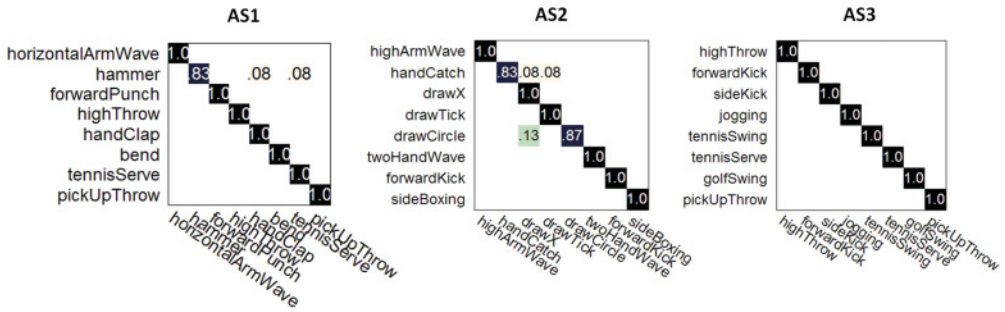

Fig. 7. The confusion matrix of the SUM rule-based fusion on the MSRAction3D dataset.

approach achieves the accuracy of 97.3%, close to the best result obtained by the sum rule based decision-level fusion method. The confusion matrix is shown in Figure 7, where most of the actions can be separated well.

4.4.3. Fusion Results on UTKinect-Action Dataset. The results of different fusion methods with the four individual features on the UTKinect-Action dataset are shown in the second column of Table II. From the results, we can see that the decision-level fusion gets an accuracy 92.9% by applying the majority voting method. Comparable accuracies are achieved by the sum rule and median rule based fusion methods, which are 91.9% and 90.9%, respectively. On the other hand, RFs exhibit a better accuracy (92.9%) than the other feature-level fusion methods. The confusion matrix of the majority voting method is shown in Figure 8, which shows that the actions "carry" and "throw" impact the overall accuracy because several samples are incorrectly classified as other (similar) actions, for example, the action "carry" is actually a walking subject carrying an object, which confuses the system to classify it as walking. The ambiguity may also happen between actions "throw" and "push" in this dataset, thus classifying such actions is still challenging.

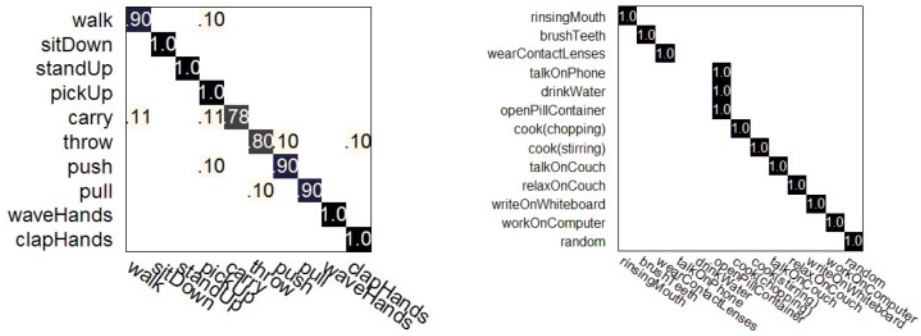

Fig. 8. The confusion matrix of the majority voting fusion method on the UTKinect-Action (left) and CAD-60 dataset (right).

4.4.4. Fusion Results on CAD-60 Dataset. On the CAD-60 dataset, the decision-level and feature-level fusions achieve the same best accuracy of 87.5% based on the MAJ rule and RFs, respectively. This accuracy is much higher than each of the individual features. From Table II, one can observe that most of the decision-level fusion methods perform poorly, some even lower than the individual features. The feature-level fusion methods (except the RFs) have the same accuracy of 81.3% as the best individual feature, that is, the EigenJoints feature. These results clearly show that some fusion methods cannot work well, depending on the input data and the specific fusion methods. That is why we need to investigate the different fusion methods carefully, in order to find the workable methods for the specific problem and special data.

4.4.5. Fusion Results on MSRDailyActivity3D Dataset. Results of different fusion methods on the MSRDailyActivity3D dataset are shown in the last column of Table II. The RF method as a feature-level fusion scheme achieves the highest accuracy of 88.8%, higher than any other fusion methods in this dataset. Among the decision-level fusion methods, the majority voting has an accuracy of 88.1%, very close to the RF method. The recognition accuracies of these two fusion methods are higher than each of the individual features, as shown in the top rows in Table II.

4.5. Comparison with the State-of-the-Art Methods

We further compare our fusion-based approach with the state-of-the-art methods for depth-based action recognition on the four challenging datasets. In all our experiments, the cross-subjects action recognition is conducted because it is more appropriate in practical applications. We list all the published results on the four databases, to the best of our knowledge. In particular, on the MSRAction3D dataset, half of the subjects are used for training and the remaining half for testing. Table III shows the reported results in the literature on the MSRAction3D dataset. We can see that the sum rule-based fusion can achieve an accuracy of 98.2% and the RFs feature-level fusion can get an accuracy of 97.3%; both are much higher than all of the previous reported results. On the UTKinect-Action dataset, the results are shown in Table IV, where the majority voting and RF methods have the same accuracy of 92.9%, which is also higher than all the state-of-the-art methods on this dataset. For the CAD-60 dataset, the same "new person" setting is used as in previous approaches in our experiment. The precision/recall were computed as the performance measure to have a direct comparison with the previous methods. The results are shown in Table V. One can see that both the majority voting and RF methods can get the same recall value; however, the majority voting has a higher precision value than the RFs. Finally, we compare our results with the state-of-the-art on the MSRDailyActivity3D dataset. From Table VI, one can see

Table III. Comparison of the Recognition Accuracies between Our Fusion-Based Approaches and All State-of-the-Art Methods on MSRAction3D Dataset

Method	Accuracy
High Dimensional Convolutional Network [Wang et al. 2012a]	72.5%
Action Graph [Li et al. 2010]	74.7%
HOJ3D [Xia et al. 2012]	79.0%
Key Pose Learning [Miranda et al. 2012]	80.3%
EigenJoints [Yang and Tian 2013]	82.3%
STOP [Vieira et al. 2012]	84.8%
ROP [Wang et al. 2012a]	86.2%
Actionlet [Wang et al. 2012b]	88.2%
HON4D [Oreifej and Liu 2013]	88.9%
DSTIP+DCSF [Xia and Aggarwal 2013]	89.3%
Part-set [Wang et al. 2013]	90.2%
Depth Motion Maps [Yang et al. 2012]	91.6%
DS-SRC [Theodorakopoulos et al. 2013]	93.6%
JAS (Cosine)+MaxMin+HOG^2 [Ohn-Bar and Trivedi 2013]	94.8%
STIP+Joint+RFs [Zhu et al. 2013] (Our Preliminary)	94.3%
Decision-Level Fusion (SUM Rule)	**98.2%**
Feature-Level Fusion (Random Forests)	**97.3%**

Table IV. Comparison of the Recognition Accuracies between Our Fusion-Based Approaches and All State-of-the-Art Methods on the UTKinect-Action Dataset

Method	Accuracy
Posture Word [Xia and Aggarwal 2013]	79.57%
DSTIP+DCSF [Xia and Aggarwal 2013]	85.8%
HOJ3D [Xia et al. 2012]	90.9%
DS-SRC [Theodorakopoulos et al. 2013]	91.0%
STIP+Joint+RFs [Zhu et al. 2013] (Our Preliminary)	91.9%
Decision-Level Fusion (Majority Voting)	**92.9%**
Feature-Level Fusion (Random Forests)	**92.9%**

Note: We used a lesser number of training examples, while the leave-one-out setting was used in Xia and Aggarwal [2013].

Table V. Performance Comparison of Our Fusion-Based Approaches with the State-of-the-Art Methods on the CAD-60 Dataset

Method	Precision/Recall
Sung et al. [2012]	67.9%/55.5%
Yang and Tian [2012]	71.9%/66.6%
Koppula et al. [2013]	80.8%/71.4%
Zhu et al. [2013] (Our Preliminary)	93.2%/84.6%
Decision Level Fusion (Majority Voting)	**96.4%/84.6%**
Feature Level Fusion (Random Forests)	**90.9%/84.6%**

that the RFs has the highest accuracy of 88.8%, which outperforms the DCSF+Joint approach in Xia and Aggarwal [2013]. The majority voting can get an accuracy of 88.1%, which is lower than the 88.2% reported in Xia and Aggarwal [2013]; however, four actions were eliminated in their experiments, while we used all 16 actions.

Through the comparisons with the state-of-the-art methods, our fusion-based approaches perform the best on all four challenging datasets. The appropriate fusion methods have been found based on our exploration, at both the decision and feature

Table VI. Performance Comparison between Our Fusion-Based Approaches and the State-of-the-Art Methods on MSRDailyActivity3D Dataset

Method	Accuracy
NBNN+parts+time [Seidenari et al. 2013]	70.0%
Local HON4D [Oreifej and Liu 2013]	80.0%
DCSF [Xia and Aggarwal 2013]	83.6%
RGGP+Fusion [Liu and Shao 2013]	85.6%
Actionlet [Wang et al. 2012b]	85.8%
DCSF+Joint [Xia and Aggarwal 2013]	88.2%
Decision-Level Fusion (Majority Voting)	**88.1%**
Feature-Level Fusion (Random Forests)	**88.8%**

Note: All the actions are used in our experiment, while in [Xia and Aggarwal 2013] four actions (with less motion) were removed from the dataset in their experiment.

levels, while some other fusion methods cannot work well for our problem. The comprehensive results demonstrate that *proper fusions* of different features are important that can significantly improve the action recognition performance in depth videos.

5. CONCLUSIONS

We have presented a comprehensive study of fusing diverse features for depth-based action recognition. Both the decision-level and feature-level fusion schemes have been explored with different methods at each fusion level. A number of experiments have been conduced on four depth databases. Experimentally, we have shown that the four different features that we investigated can be complementary to each other, characterizing the depth actions from different aspects. Given the diverse features, different fusion methods perform quite differently in action recognition. Based on a systematic evaluation, the appropriate fusion methods have been found to significantly improve the recognition accuracies over each individual feature. We have also shown that our fusion-based action recognition in depth videos can outperform the state-of-the-art methods on all four challenging databases.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their help to improve the article.

REFERENCES

J. K. Aggarwal and Michael S. Ryoo. 2011. Human activity analysis: A review. *ACM Comput. Surv. (CSUR)* 43, 3 (2011), 16.

F. M. Alkoot and J. Kittler. 1999. Experimental evaluation of expert fusion strategies. *Pattern Recogn. Lett.* 20, 11 (1999), 1361–1369.

P. K. Atrey, M. A. Hossain, A. El S., and M. S. Kankanhalli. 2010. Multimodal fusion for multimedia analysis: A survey. *Multimedia Syst.* 16, 6 (2010), 345–379.

H. Bay, T. Tuytelaars, and G. Luc Van. 2006. Surf: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision*. Springer, 404–417.

S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz. 1999. Fusion of face and speech data for person identity verification. *IEEE Trans. Neural Networks* 10, 5 (1999), 1065–1074.

L. Breiman. 2001. Random forests. *Machine Learning* 45, 1 (2001), 5–32.

G. Brown, A. Pocock, M. J. Zhao, and M. Luján. 2012. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *J. Mach. Learn. Res.* 13 (2012), 27–66.

K. Chang, K. Bowyer, and P. Flynn. 2003. Face recognition using 2D and 3D facial data. In *Proceedings of the ACM Workshop on Multimodal User Authentication*. 25–32.

L. L. Chen, H. Wei, and J. Ferryman. 2013. A survey of human motion analysis using depth imagery. *Pattern Recogn. Lett.* 34, 15 (2013), 1995–2006.

M. C. Da C. A. and M. Fairhurst. 2009. Analyzing the benefits of a novel multiagent approach in a multimodal biometrics identification task. *IEEE Syst. J.* 3, 4 (2009), 410–417.

D. L. Donoho and others. 2000. High-dimensional data analysis: The curses and blessings of dimensionality. In *Proceedings of the AMS Math Challenges Lecture.* 1–32.

F. Fleuret. 2004. Fast binary feature selection with conditional mutual information. *J. Mach. Learn. Res.* 5 (2004), 1531–1555.

D. L. Hall and J. Llinas. 1997. An introduction to multisensor data fusion. *Proc. IEEE* 85, 1 (1997), 6–23.

J. Kittler. 1998. Combining classifiers: A theoretical framework. *Pattern Anal. Appl.* 1, 1 (1998), 18–27.

A. Klaser, M. Marszałek, C. Schmid, and others. 2008. A spatio-temporal descriptor based on 3D-gradients. In *Proceedings of the British Machine Vision Conference*.

T. Kobayashi and N. Otsu. 2012. Motion recognition using local auto-correlation of space–time gradients. *Pattern Recog. Lett.* 33, 9 (2012), 1188–1195.

H. S. Koppula, R. Gupta, and A. Saxena. 2013. Learning human activities and object affordances from RGB-D videos. *Int. J. Rob. Res.* 32, 8 (2013), 951–970.

L. I. Kuncheva. 2002. A theoretical study on six classifier fusion strategies. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 2 (2002), 281–286.

L. I. Kuncheva, J. C. Bezdek, and R. PW Duin. 2001. Decision templates for multiple classifier fusion: An experimental comparison. *Pattern Recog.* 34, 2 (2001), 299–314.

I. Laptev and T. Lindeberg. 2004. Velocity adaptation of space-time interest points. In *Proceedings of the 17th International Conference on Pattern Recognition*, Vol. 1. 52–56.

I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. 2008. Learning realistic human actions from movies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1–8.

W. Q. Li, Z. Y. Zhang, and Z. C. Liu. 2010. Action recognition based on a bag of 3D points. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Workshops*. 9–14.

L. Liu and L. Shao. 2013. Learning discriminative representations from RGB-D video data. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*. AAAI Press, 1493–1500.

L. Miranda, T. Vieira, D. Martinez, T. Lewiner, A. W. Vieira, and M. F. M. Campos. 2012. Real-time gesture recognition from depth data through key poses learning and decision forests. In *Proceedings of the 25th SIBGRAPI Conference on Graphics, Patterns and Images*. 268–275.

B. B. Ni, G. Wang, and P. Moulin. 2011. RGBD-HuDaAct: A color-depth video database for human daily activity recognition. In *Proceedings of the IEEE Computer Vision Workshops (ICCV'11)*. 1147–1153.

E. Ohn-Bar and M. M. Trivedi. 2013. Joint angles similarities and HOG2 for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 465–470.

O. Oreifej and Z. C. Liu. 2013. HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 716–723.

R. Poppe. 2010. A survey on vision-based human action recognition. *Image Vision Comput.* 28, 6 (2010), 976–990.

M. Reyes, G. Domínguez, and S. Escalera. 2011. Featureweighting in dynamic timewarping for gesture recognition in depth data. In *Proceedings of the IEEE Computer Vision Workshops*. 1182–1188.

A. Ross and A. K. Jain. 2003. Information fusion in biometrics. *Pattern Recog. Lett.* 24, 13 (2003), 2115–2125.

A. A Ross and R. Govindarajan. 2005. Feature level fusion of hand and face biometrics. In *Defense and Security*. International Society for Optics and Photonics, 196–204.

L. Seidenari, V. Varano, S. Berretti, P. Pala, and B. Alberto Del. 2013. Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In *Proceedings of CVPR International Workshop on Human Activity Understanding from 3D Data (HAU3D'13)*. 479–485.

S. Sempena, N. U. Maulidevi, and P. R. Aryan. 2011. Human action recognition using dynamic time warping. In *Proceedings of the International Conference on Electrical Engineering and Informatics (ICEEI)*. 1–5.

A. H. Shabani, D. A. Clausi, and J. S. Zelek. 2012. Evaluation of local spatio-temporal salient feature detectors for human action recognition. In *Proceedings of the 9th IEEE Conference on Computer and Robot Vision*. 468–475.

J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. 2011. Real-time human pose recognition in parts from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1297–1304.

J. Sung, C. Ponce, B. Selman, and A. Saxena. 2012. Unstructured human activity detection from RGBD images. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 842–849.

I. Theodorakopoulos, D. Kastaniotis, G. Economou, and S. Fotopoulos. 2014. Pose-based human action recognition via sparse representation in dissimilarity space. *J. Vis. Commun. Image Rep.* 25, 1 (Jan. 2014), 12–23.

P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. 2008. Machine recognition of human activities: A survey. *IEEE Trans. Circuits Syst. Video Technol.* 18, 11 (2008), 1473–1488.

V. N. Vapnik. 1998. *Statistical Learning Theory*. Wiley, New York.

A. Vieira, E. Nascimento, G. Oliveira, Z. C. Liu, and M. Campos. 2012. Stop: Space-time occupancy patterns for 3D action recognition from depth map sequences. *Prog. Pattern Recog., Image Anal., Comput. Vis., Appl.* (2012), 252–259.

C. Y. Wang, Y. Z. Wang, and A. L. Yuille. 2013. An approach to pose-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 915–922.

J. Wang, Z. C. Liu, J. Chorowski, Z. Y. Chen, and Y. Wu. 2012a. Robust 3D action recognition with random occupancy patterns. In *Proceedings of the European Conference on Computer Vision*. Springer, 872–885.

J. Wang, Z. C. Liu, Y. Wu, and J. S. Yuan. 2012b. Mining actionlet ensemble for action recognition with depth cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1290–1297.

D. Weinland, R. Ronfard, and E. Boyer. 2011. A survey of vision-based methods for action representation, segmentation and recognition. *Comput. Vision Image Understanding* 115, 2 (2011), 224–241.

G. Willems, T. Tuytelaars, and Luc Van G. 2008. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proceedings of the European Conference on Computer Vision*. 650–663.

L. Xia and J. K. Aggarwal. 2013. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2834–2841.

L. Xia, C. C. Chen, and J. K. Aggarwal. 2012. View invariant human action recognition using histograms of 3D joints. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Workshops*. 20–27.

L. Xu, A. Krzyzak, and C. Y. Suen. 1992. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Trans. Sys., Man Cybern.* 22, 3 (1992), 418–435.

H. Yang and J. Moody. 1999. Feature selection based on joint mutual information. In *Proceedings of the International ICSC Symposium on Advances in Intelligent Data Analysis*. 22–25.

X. D. Yang and Y. L. Tian. 2014. Effective 3D action recognition using eigenjoints. *J. Visual Commun. Image Represent.* 25, 1 (2014), 2–11.

X. D. Yang and Y. L. Tian. 2012. Eigenjoints-based action recognition using naive-Bayes-nearest-neighbor. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Workshops (CVPRW'12)*. 14–19.

X. D. Yang, C. Y. Zhang, and Y. L. Tian. 2012. Recognizing actions using depth motion maps-based histograms of oriented gradients. In *Proceedings of the 20th ACM International Conference on Multimedia*. 1057–1060.

Y. Zhao, Z. C. Liu, L. Yang, and H. Cheng. 2012. Combing RGB and depth map features for human activity recognition. In *Proceedings of the 2012 Asia-Pacific Signal Information Processing Association Annual Summit and Conference (APSIPA ASC'12)*. 1–4.

Y. Zhu, W. B. Chen, and G. D. Guo. 2013. Fusing spatiotemporal features and joints for 3D action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'13)*. 486–491.

Received July 2013; revised December 2013; accepted March 2014