Dear Editor,

We revised the article following the suggestions of the reviewers and the associate editor. Attached is a detailed response to the reviewers’ comments.

Thank you for your consideration and best regards.

Alexandre Perez, Hedi Tabia, David Declercq and Alain Zanotti
Using the conflict in Dempster-Shafer evidence theory as a rejection criterion in classifier outputs combination for 3D human action recognition

Alexandre Perez$^{a,b,1}$, Hedi Tabia$^b$, David Declercq$^b$, Alain Zanotti$^a$

$^a$AUSY Expertise et Recherche, 6 Rue Troyon, 92310 SEVRES, France
$^b$ETIS/ENSEA, University of Cergy-Pontoise, CNRS, UMR 8051, France

Abstract

In this paper, we propose a comprehensive solution to 3D human action recognition including feature extraction, classification, and multiple classifier combination. We effectively present two feature extraction methods, four different types of well-known classifiers, and four multiple classifier combination strategies including a specially designed belief based method. In order to enhance the recognition accuracy, we propose a new rejection criterion based on the conflict from the information sources: the classifier outputs. We test our method on the MSRAction 3D dataset. Discarding examples using the conflict based criterion shows superior results than other combination approaches. Moreover this criterion allows choosing a tradeoff between the performance and rejection rate.

Keywords: human action recognition, classifier combination, Dempster-Shafer evidence theory, rejection, actionlet, HON4D

1. Introduction

Human action recognition from depth monocular sensor has become a very active area of research thanks to the appearance of low cost devices such as the Kinect. Numerous applications such as behavior monitoring, Human-Machine Interaction or sign language recognition have become popular applications of vision based action recognition systems.

State-of-the-art action recognition methods [1] methods for action recognition are usually based on a representation step followed by a machine learning classification step. There are two different approaches for the representation step, hand-crafted and learned representation. Hand-crafted features [2, 3, 4] which include holistic representation and local representation, aim to capture the essence of different action patterns by using expert prior knowledge. On the contrary, learned representations that are mainly based on CNNs can directly learn data representations from raw training samples and detect data-driven features for specific tasks without any prior knowledge [5, 6, 7, 8]. For the classification step, once the features have been extracted, the problem of action recognition is reduced to multi-class supervised learning, with as many classes as defined actions. The classifier performs two separate steps in order to predict the classes of unlabeled actions: training and testing. During training, labeled data is sent to the classifier and used to adapt a statistical decision procedure for distinguishing actions. These approaches aim to extract the most representative information from the data and find the classifier which provide the best generalization given the features. However, a such system is hard to design, especially when lot of actions are considered.

Even if a lot of work has been done in vision based machine learning, robust and real time action analysis is still a challenging task. Traditional approaches use a single feature descriptor and a single classification algorithm to estimate the class of a frame sequence. Challenges like a large number of classes and noisy inputs are still to be resolved.

The aim of classifier combination systems is to improve performance especially when data is not perfectly digitized, and contains distortions or errors, which is especially the case of 3D motion data. For instance, skeleton based approaches, which rely on 3D body joint locations extractor algorithms, can lead to misplaced joint location, especially when using acquisition from a single depth sensor. Other approaches are also subject to noise, especially stemming from self-occlusion. Moreover a feature/classifier couple performance may vary from a class to another. In other words, a feature/classifier couple can be efficient in discriminating some classes while another one could be better on other classes. To compensate for one feature/classifier couple weaknesses with another feature/classifier couple strength is the core idea behind combination.

In this paper, we provide an extensive solution to 3D action recognition including feature extraction, classification, multiple classifier combination and rejection. Comprehensive evaluations are conducted on the MSRAction 3D dataset. Two among the most effective proposed features in the literature are implemented and compared using different classifiers. We study the performance of four well-known classifiers: naïve Bayes, Support Vector Machine, decision tree and random forest classifiers. To further improve the accuracy of the system, four different combination approaches are proposed including a specially designed belief based method. A novel rejection strategy is also proposed to optimize the reliability of 3D human action recognition system. The rejection is based on the conflict gener-
ated during the belief based combination, in which, we suppose that a high level of conflict means an uncertain decision. In this way, we consider more interesting to decide that the recognition is uncertain rather than giving an erroneous estimated label.

The rest of the article is organized as follows. We present the underlying features extraction methods in Section 2 and describe the selected classifiers in Section 3. Section 4 introduces the proposed combination strategies. Experimental results are listed and discussed in Section 5. Finally, Section 6 concludes the paper.

1.1. Related work

Action recognition based on depth videos is a very active area. We focus our work on Oreifej et al.’s [9] and Wang et al.’s [10] methods which are introduced in section 2. Here, we review the most important related works.

Xia et al. [11] proposed the Histograms of 3D Joints (HO3JD) feature based on normalized skeleton joints location. Every joint location is used in a probabilistic voting in spherical bins. Classification using Hidden Markov Models shows promising results. Xia et al. [12] proposed a filtering method to extract spatio-temporal interest points (STIPs) with noise removal and the depth cuboid similarity feature (DCSF) which describes the depth cuboid around the STIPs. Yang et al. [13] proposed a holistic approach: the depth motion map feature sums up the depth video in three images from which Histogram of Oriented Gradient features [14] are extracted. The main drawback in this method is the lack of temporal information, leading to poor results when the movements order is discriminant, e.g. when trying to discriminate a hand waving from left to right and from right to left. In Vieira et al.’s article [15] another holistic approach has been proposed: each sequence is divided into a spatiotemporal grid, then a global occupancy pattern feature, which represents the pixel distribution, is computed. This method allows intra-action variations which is usually important for human action recognition. They use an action graph based system to learn a statistical model for each action class and use a state machine to segment long depth sequences using a neutral pose classifier.

While a robust feature could be designed to handle specific action recognition task, it is extremely difficult for a visual-based approach to deal with all kind of action especially when data is not perfectly digitized, and contains distortions or errors which is the case of 3D motion. Each single classifier has different sensitivity to different changes in the action appearance. A combination scheme involving different classifiers, which integrates various information sources, is likely to improve the overall system performance. The classifier combination can be implemented at two levels, feature level [16, 17, 18] and decision level [19, 20, 21]. In our approach we use the decision level combination which is more appropriate when the component classifiers use different types of features.

Classifier outputs combination aims to take advantage of each classifier, for a given feature, when it has good performance. In other words, the goal is to select the correct output which gives the correct class for an observation from each classifier. Several approaches have been considered in the literature. To combine those classifiers efficiently, vote combination approaches [22, 19] aim to find a linear combination of classifier outputs to improve performance. Lam and Suen [23] showed that majority voting for classifier combination provides better performance than all sources taken separately, assuming statistical independence of the information sources. Kuncheva et al. [24] studied the influence of the dependence between the sources, that are the classifier outputs, on the performance of the vote combination.

Iosifidis et al. [25] proposed a Bayesian fusion approach for recognizing action from different cameras. In their method, action and viewing angle classification are achieved independently for all cameras. The Bayesian framework was exploited in order to provide the optimal combination of the action classification results, coming from all available cameras.

The Dempster-Shafer evidence theory allows to assign a belief function to each classifier output, which is more informative than a simple weight. Xu et al. [26] used different combination strategies for pedestrian detection and succeeded in improving detection rate. Combinations based on Dempster-Shafer theory showed better results than probability voting. Quost et al. [27] proposed to use a pairwise approach for multi-class classification using Dempster-Shafer theory by interpreting the output of each pairwise classifier with a conditional belief function. Quost et al. [28] proposed two approaches to deal with non-independent classifiers. They learn an optimal combination scheme. These approaches show their limits when confronted with highly redundant information.

The descriptors in these methods are based on different data, including skeleton and point clouds. These variety of descriptor leads to different classifiers with their own strengths and weaknesses for action recognition. In this context, combining these algorithms is justified. Moreover, the resulting conflict, in the meaning of Dempster-Shafer theory, from the combination of classifier outputs was never exploited, which justifies this work.

1.2. Overview of the proposed recognition system

Figure 1 overviews the approach proposed in this article. We provide a complete solution to 3D action recognition including feature extraction, classification, multiple classifiers combination and rejection. Two of the most effective feature extraction methods are implemented and compared using four different classifiers namely the Naïve Bayes, the SVM, the decision tree and the random forests classifiers. A multiple classifier combination system is also introduced. In this approach, besides the traditional combination schemes (vote, linear and Bayesian combination), a novel combination strategy is designed using the Dempster-Shafer evidence theory. The theory offers the possibility to derive a measure of contradiction between the classifier decisions to be fused. We exploit this possibility in order to define rejection criteria. This allows boosting the accuracy on the accepted data, which is desired in situations where a miss-classification is very expensive or must not happen.
2. Selected features

We present here the different features considered in our system.

2.1. Histogram of Oriented 4D Normals

The Histogram of Oriented 4D Normals (HON4D) has been proposed by Oreifej and Liu [9] for depth map description. The authors describe the depth video sequence using a histogram capturing the distribution of the orientation of the surface normal \( n \) in the 4D volume of time, depth and spatial coordinates.

\[
n = \left( \frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, \frac{\partial z}{\partial t}, -1 \right)^T
\]

As the depth sequence represents a depth function of space and time, they proposed to capture the observed changing structure using a histogram of oriented 4D surface normals. To construct HON4D, the 4D space is initially quantized using the vertices of a regular polychoron. Afterwards, the quantization is refined using a novel discriminative density measure such that additional projectors are induced in the directions where the 4D normals are denser and more discriminative.

2.2. Actionlet ensemble

Wang et al. [10] proposed new features suitable for depth based data and proposed also a new model called the actionlet ensemble. This model is meant to represent a wide variety of actions and is based on both skeleton and point cloud information. The designed features aim to be more robust when skeleton information alone is not sufficient to handle intra-class variation, e.g. when skeleton joints tracking goes wrong due to skeleton occlusion (it could be self-occlusion as well as occlusion with an object being manipulated).

Two features have been proposed; the pairwise relative joint positions and the Local Occupancy Pattern (LOP). The authors introduced the relative joint positions \( p_i \) as a feature since it is more representative than the position themselves. \( p_i \) is computed for each joint \( i \). The LOP feature is a histogram of occupied bins, in spatial cuboid neighborhood centered around one single joint, at a given frame. In order to compute the feature \( o_{xyz} \) of a bin \( b_{xyz} \). The authors count the voxels that falls in this bin. The bin is then normalized thanks to a sigmoid \( \delta \).

\[
o_{xyz} = \delta\left( \sum_{q \in b_{xyz}} I_q \right)
\]

The LOP feature \( o_i \) is defined as the concatenation of all \( o_{xyz} \) for the joint \( i \). LOP feature allows describing the “depth appearance” around joints. It also allows describing interactions between the subject and the objects used to perform the action. Both features are concatenated as \( g_i = (p_i, o_i) \). Each element \( g_i \) of \( g \) depends on the time since it is computed for each frame.

Therefore the authors apply a Short Fourier Transform to obtain the Fourier Temporal Pyramid features \( G_i \) at every joint. The Fourier Temporal Pyramids represent the temporal structure of the actions and are robust to both temporal misalignment and data noise.

Actionlets are thus defined as a conjunctive structure on the Fourier Temporal Pyramid features. However, for a given action, Fourier Temporal Pyramid features aren’t discriminative around every joint. The authors propose a mining algorithm to find the discriminative actionlets. Defining the confidence and the ambiguity of an actionlet, they iteratively add larger (actionlet with a larger cardinality) actionlet to the actionlet pool until no discriminative actionlet is added, i.e. new actionlets haven’t enough confidence. Actionlets that are too ambiguous are removed from this pool. This process is performed for every action.

3. Selected classifiers

After the feature description step, we present in this section the supervised classifiers used in our method to learn the actions. Here, we propose the use of the four different classifiers namely the naïve Bayes, Support Vector Machines (SVM), decision tree and random forest classifiers.

3.1. Naïve Bayes classifier

The Naïve Bayes Classifier is a probabilistic classifier based on the Bayesian theorem. Given a depth video to be classified, represented by a feature vector \( x = (x_1, \ldots, x_P) \) with dimension \( P \), the classifier assigns to this video probabilities \( p(C_k|x_1, \ldots, x_P) \), for each \( C_k, k \in \{1, \ldots, K\} \) possible action outcomes of models in the dataset. Since \( x \) can be a high-dimensional feature vector, we use Bayes’ theorem to make the model more tractable. The conditional probability can be decomposed as \( p(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)} \). Under the ‘naïve’ conditional independence assumptions, each bin \( x_i \) is conditionally independent of every other feature \( x_j \) for \( j \neq i \).

\[
p(x_i|C_k, x_j, x_k, \ldots, x_j) = p(x_i|C_k), \forall i \neq j, k, \ldots, l
\]

This means that under the above independence assumptions, the conditional distribution over the video class \( C \) is:

\[
p(C_k|x_1, \ldots, x_P) = \frac{1}{Z} p(C_k) \prod_{i=1}^{P} p(x_i|C_k)
\]

where the evidence \( Z = p(x) \) is a scaling factor dependent only on \( x_1, \ldots, x_p \), that is a constant.

In our paper, naïve Bayes classifiers have no kernel estimator.
3.2. SVM classifier

The SVMs were introduced by Cortes and Vapnik [29] as supervised learning machines with control ability for regression and binary classification problems. In the case of classification, a SVM constructs an optimal separating hyperplane in a high-dimensional feature space. This hyperplane has the largest distance (also called margin) to the nearest training data point of any action class i.e. the support vectors. The optimal separating hyperplane output is given by:

\[ f(x) = \text{sign}(x^Tw + b) \]

with the input feature vector \( x \) and where the bias \( b \) and the row vector of weights \( w \) are trained by maximizing the margin \( \frac{1}{\|w\|} \), under the constraint that the training feature vectors are well classified and outside the margin.

Even in a finite dimensional space of features, it often happens that the sets to discriminate are not linearly separable in that space. The kernel trick allows us to map the original finite-dimensional space into a much higher-dimensional space, presumably making the separation easier in that space. This SVM modification can be implemented with a wide range of kernels.

In this paper, we use a polynomial kernel of degree 3.

3.3. Decision tree classifier

Decision trees fall in the category of supervised learning methods, used for classification or regression. In this paper, we use the C4.5 [30] algorithm to generate decision trees.

C4.5 builds decision trees using the information entropy. The training data is a set \( T \) of labeled P-dimensional samples \( x = (x_1, x_2, \ldots, x_P) \) belonging to the class \( C_i \), \( i \in \{1, \ldots, K\} \).

At each node of the tree, we choose the attribute of the data that most effectively splits its set of samples, according to a splitting criterion. This criterion is defined as the normalized information gain (difference in entropy), i.e., for set of samples \( S \):

\[ \text{Gain}(x, T) = \text{Info}(T) - \sum_{i=1}^{k} p_i \cdot \log(p_i) \] (5)

where

\[ \text{Info}(T) = -\sum_{i=1}^{k} \frac{|T_i|}{|T|} \text{Info}(T_i) \] (6)

\[ \text{Info}(x, T) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \text{Info}(T_i) \] (7)

\( T_i \) represents the subset of \( T \) induced by the value of \( x_i \) and \( p_i = \frac{|T_i|}{|T|} \). The attribute with the highest normalized information gain is chosen to take the decision.

3.4. Random forests classifier

Random forests have been introduced by Leo Breiman [31]. They operate by constructing a multiplicity of decision trees from different learning sets and outputting the class that is the mode of the classes of the individual trees.

The training algorithm for random forests uses the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set \( T \) of feature vectors \( x \), with the set \( L \) of the corresponding class labels \( C_i \), \( i \in \{1, \ldots, K\} \), bagging repeatedly selects a random sample with replacement of the training set and fits trees to these samples. For \( b = \{1, \ldots, B\} \) sample, with replacement, \( n \) training examples from \( T \), \( L \); call these \( T_b, L_b \). Train a decision or regression tree \( f_b \) on \( T_b, L_b \). After training, the recognition of gesture samples \( T' \) can be made by taking the majority vote in the case of decision trees. This bootstrapping procedure leads to better classification performance since it decreases the variance of the decision function, without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in the training set, the average of several trees could be less sensitive, as long as the trees are not correlated. Simply training many trees on a single training set would give strongly correlated trees (or even the same tree repeatedly, if the training algorithm is deterministic); bootstrap sampling is a way of making the trees almost uncorrelated by showing them different training sets. The number of samples, \( B \), is a free parameter. Typically, a few hundred to several thousand trees are used, depending on the size and nature of the training set. An optimized number of trees \( B \) can be found using cross-validation, or by observing the out-of-bag error: the mean prediction error on each training sample \( T_i \), using only the trees that did not have \( T_i \) in their bootstrap sample. The training and test error tend to level off after a certain amount of trees have been built.

In this paper random forests are made of 100 trees.

4. Combination Strategy

In this section, a multiple classifier combination system is introduced. The aim of this systems is to improve performance especially when data is not perfectly digitized, and contains distortions or errors, which is especially the case of 3D motion data. Therefore, a combination of different classifiers which can integrate complementary information should lead to improved classification accuracy. The classifier combination can be implemented at two levels, feature level and decision level. In our approach we use the decision level combination which is more appropriate when the component classifiers use different types of features. Here, each classifier output, termed information source, is the input of our fusion system.

We define \( S = \{S_1, \ldots, S_N\} \) as the set of all information sources. Let \( S_j(x) = i \) denotes the source of information \( S_j \) voting for the class \( C_i \) given the observation \( x \). The indicator function of a given information source is defined as:

\[ M_j(x) = \begin{cases} 1 & \text{if } S_j(x) = i, \\ 0 & \text{otherwise} \end{cases} \] (8)

4.1. Basic combination strategies

4.1.1. Vote

The vote combination is a simple combination method. In this model each information source has the same importance and the class whose label gets the most votes is considered as the most likely class. This method shows good classification results if we consider that all information sources have comparable performance.
The classifier output combination is given by:

\[ M_k^x(x) = \sum_{j=1}^{N} \beta_{jk} M_k^x(x), \forall k \]  

(9)

One of the issues of this combination strategy is to differentiate between classes when two or more get the largest amount of votes. One way to deal with this issue is to add another class \( C_0 \) which takes account of the conflict between sources. We use the majority vote rule:

\[ E(x) = \begin{cases} 
  k & \text{if } \max_i M_k^x(x), \\
  0 & \text{otherwise}
\end{cases} 
\]

(10)

with \( k \) corresponding to class \( C_k \).

### 4.1.2. Linear combination

Voting methods are based solely on the output label computed by each classifier. No expertise or accuracy is considered. In these methods the decision of each information source is treated as one vote, but what happens if one of the classifiers is much more accurate than any other? The idea of a linear combination is to give a weight \( \beta_{jk} \) to each information source \( S_j \). Thus, by using this combination, we are likely to consider the information sources differently depending on how accurate they have been in the past. In our study, \( \beta_{jk} \) represents the confidence given to the classifier decision. The linear classifier output combination is defined as:

\[ M_k^x(x) = \sum_{j=1}^{m} \beta_{jk} M_k^x(x), \]  

(11)

It is possible to use the prior knowledge of the classifier performance on each class to determine \( \beta_{jk} \). Our approach is based on the knowledge of the spatial distribution of the features in the Hilbert features space. The idea is that the more a class (i.e., an action) has a large intra-class variance, the more its samples are likely to be far from the centroid of this class. In other words, if a sample is equidistant from two centroids, it is more likely to belong to the class with the higher intra-class variance. Thus we define the belief \( \beta_{jk} \) as:

\[ \beta_{jk}(x) = \frac{1}{\sigma_k \sqrt{2\pi}} \exp \left( -\frac{d_k(x)^2}{2\sigma_k^2} \right) \]  

(12)

where \( \sigma_k \) is the variance of the class \( C_k \) and \( d_k(x) \) is the euclidean distance between the feature \( x \) and the centroid of the class \( C_k \).

### 4.1.3. Bayesian combination

The Bayesian combination aims to compute the probabilities \( P(\hat{C}_j | \hat{C}_1, \ldots, \hat{C}_N) \), where \( \hat{C}_j \) is the predicted action label from the information source \( S_j \). These probabilities can be estimated using the Bayes formula

\[ P(C_i | \hat{C}_1, \ldots, \hat{C}_N) = \frac{P(C_i | \hat{C}_1, \ldots, \hat{C}_N) P(C_i)}{P(\hat{C}_1, \ldots, \hat{C}_N)} \]  

(13)

Where the different probabilities can be estimated during the training step. However, because of the huge number of training samples needed for the estimation, a statistical independence of the conditional sources given a decision is generally assumed [32] (this can be checked by various criteria such as the calculation of the confusion matrices). We obtain:

\[ P(C_i | \hat{C}_1, \ldots, \hat{C}_N) = \frac{\prod_{j=1}^{N} P(C_i | \hat{C}_j) P(C_i)}{\prod_{j=1}^{N} P(\hat{C}_j)} \]  

(14)

### 4.2. Belief based combination

#### 4.2.1. Basic concepts in the theory of belief functions

The theory of belief functions, also known as the Dempster-Shafer theory, is a tool for representing and combining information sources. It aims to model imperfect knowledge, especially, when sources of information are not entirely reliable.

The belief function theory starts by assuming a universe of discourse, or frame of discernment (FOD), consisting of a finite set of mutually exclusive atomic hypotheses. Let \( C = \{C_1, \ldots, C_K\} \) denotes this FOD. We define the basic belief assignment (bba) with a mass function \( m : 2^C \rightarrow [0, 1] \) such as \( \sum_{A \subseteq C} m(A) = 1, m(\emptyset) = 0 \) the belief allocated to any subset \( A \) of \( C \).

Each source of information is assigned a bba. In a first step we combine information sources. The second step is the decision making one. We consider that at least one of our information sources, is reliable. We can use the disjunctive combination \( \oplus \) defined as:

\[ m_1 \oplus m_2(A) = \sum_{B \subseteq E} m_1(B)m_2(E), \forall A \subseteq C \]  

(15)

Once the information fusion is performed, the system can synthesize one’s final decision regarding the class membership of the action to be recognized. To take this decision we use the maximum plausibility (optimist choice) where the plausibility function \( P(A) \) represents the maximum share of belief which could support the hypothesis \( A \).

\[ P(A) = \sum_{A \cap B = \emptyset} m(B), \forall A \subseteq C \]  

(16)

Since the output of each source \( S_j \) for an observation \( x \) is a single class \( C_i \), we define the bba of the source \( S_j \) as:

\[ \begin{cases} 
  m(C_i) = \beta_{jk}, & i \mid S_j(x) = i \\
  m(C) = 1 - m(C_i)
\end{cases} \]  

(17)

with \( \beta_{jk} \) as defined in equation (12).

#### 4.2.2. Discount operation

Our sources of information may not be reliable and may lead to conflict and poor decision making. Taking into account the reliability of a classifier output in the final decision process would clearly improve the classification performance. For this end, we introduce a discount rate \( \alpha \in [0, 1] \) such as \( (1 - \alpha) \) represents the credibility of a source. The discount operation is defined as:

\[ m_1(\alpha) = m_1 \oplus m_2(\alpha) \]  

(18)

(15)
\[
\begin{align*}
\begin{array}{l}
\delta m(A) = (1 - \alpha)m(A), \ \forall A \subseteq C \\
\gamma m(C) = (1 - \alpha)m(C) + \alpha
\end{array}
\end{align*}
\] (18)

The challenging task is to determine the discount rates for our sources. Moreover we want to consider the reliability of a source for a given class to recognize. Indeed we want to take into account that a classifier can perform well on some classes and have poor performance on other classes.

We propose to use the training set to estimate discount rate \(\alpha\). A 10-fold cross-validation on the training set allows us to evaluate mean performance \(p_{e,c}\) of a classifier \(S_j\) on a class \(C_i\). Since we only deal with simple bbas where focal elements are the ignorance (i.e. \(C\)) and a singleton hypothesis, we can define \(\alpha\) as the mean performance of the considered classifier on the class corresponding to this singleton \(\alpha = 1 - p_{e,c}\). This approach requires having a sufficient amount of training examples.

### 4.2.3. Using conflict as a confidence measure

The belief function theory offers the possibility to derive a measure of contradiction between the classifier decisions to be fused. This measure can be managed to boost the accuracy on the accepted data, which is desired in situations where a miss classification is very expensive or must not happen. In this section, we propose to use the conflict measure to determine if the decision quality.

Indeed, the presence of a mass on the empty set is an alarm to indicate that there is confusion between two or more decisions. Thus, the belief \(m(\emptyset)\) is a measure for the conflict or disagreement between all the sources that have been combined. Also any belief mass assigned to \(C\), the whole frame of discernment, represents the belief that the assignment of masses to other focal elements may be based on evidence that is not legitimate. Hence, \(m(C) > 0\) constitutes the doubt into the correctness of the bbas, and can be used to reject a sample to be classified. Again, setting a threshold value for the conflict and rejecting samples that induced a higher conflict allows boosting the rate of correct classification. In fact, in a decision process it could be more interesting to decide that the recognition is uncertain rather than giving an erroneous estimated label.

The conflict \(\Phi\) can be defined as the mean of the basic belief assignments on the empty set and \(C\) after the combination. We call it the global conflict. However this definition of the conflict leads to poor results where the conflict is not a good measure of the decision quality.

Several notions of conflict can be found: [33], [34]. We focus on the following definitions:

**Global conflict.** The conflict can be defined as the mean between the bba of the empty set and the FOD, i.e., the contradiction and the ignorance in the combination.

\[
\Phi = \frac{1}{2}(m(\emptyset) + m(C))
\] (19)

**Total conflict.** [35] introduced a distance based conflict. Rather than defining the conflict as measure of the resultant mass, they define a distance between source opinions, i.e., their bbas:

\[
d(m_1, m_2) = \sqrt{\frac{1}{2}(m_1 - m_2)^T D(m_1 - m_2)}
\] (20)

where

\[
D(A, B) = \begin{cases} 
1 & \text{if } A = B = \emptyset, \\
\left| \frac{A}{|A|} \right| & \forall A, B \in 2^C. 
\end{cases}
\] (21)

Assuming that the more two bbas are far from each other and the more they are in conflict, the conflict between one source of information \(S_j\) in the set of sources \(S = \{S_1, \ldots, S_N\}\) and the others can be defined as:

\[
\phi_j(j, V) = \frac{1}{N-1} \sum_{c=1, c \neq j}^N d(m_j, m_c)
\] (22)

We define the conflict as the mean of those conflicts as:

\[
\Phi = \frac{1}{N} \sum_{j=1}^N \phi_j(j, V)
\] (23)

**Intrinsic conflict.** Florea and Bosse (2009) introduced the intrinsic conflict which represent the conflict in a single bba.

\[
\phi_j(m_j) = \sum_{A, B \in C} m_j(A) m_j(B) |A \cup B| - |A \cap B| \quad |A \cup B|
\] (24)

The mean intrinsic conflict of every bbas can be used as a conflict measure.

\[
\Phi = \frac{1}{J} \sum_{j=1}^J \phi_j(m_j)
\] (25)

Thanks to this approach we are able to take into account the given belief to each source opinion rather than defining the conflict as a measure on the final bba.

### 5. Experiments and results

In this section, we evaluate the performance of the feature extraction, classification, multiple classifier combination and the rejection methods on the MSRAAction3D dataset [36]. This dataset is made of 567 depth map sequences where 10 subjects perform 20 actions, each subject performing each action 2 or 3 times. The actions are: high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, sideboxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick up and throw. The resolution of each depth map is 640x480 pixels and the frame rate is 15 fps. These actions are performed without the actors holding any object in their hands. Additionally, the background is clear from any noise. However the classification is still challenging due to the similarity between some actions.

For our method evaluation, we use two different experimental protocols: (1) the cross-subject test setting [36, 10] in which,
the samples of half of the subjects are used as training data, and the rest of the samples are used as test data, and (2) the 10-fold cross-validation setting where subsampling is done randomly.

5.1. Feature extraction performance

The performance of the two earlier presented feature extraction methods (section 2) are evaluated using various classifiers. Table 1 presents the performance of these methods in term of recognition rate. Actionlet feature slightly outperforms the HON4D with the SVM, the Decision Tree and the Naive Bayes classifiers, yet it gives lower performance with Random Forest classifier. From Table 1, Actionlet and HON4D behave in the same way in both setting. Notice that the accuracies in the cross subject test are relatively low. This is due to the small number of subjects and also the significant variations of the same action performed by different subjects. As stated in [36], during the data collection, subjects were free to choose the style of the actions. For instance, some subjects chose to perform the hand clap without stretching their arms whereas others did. It is expected that more subjects are required in order to improve the cross subject performance.

5.2. Classifier performance

From Table 1, one can notice that the Actionlet feature has better discriminative power than the HON4D feature. Thus using the Actionlet feature, the performance of the four classifiers SVM, Random Forest, Decision Tree and Naive Bayes can be evaluated. SVM is the most efficient classifier. It performs the highest classification accuracy on both 10-cross validation (90.71%) and cross-subject setting (82.41%). The performance of Random Forest classifier (81.09% and 72.40% in both 10-cross and cross-subject settings) is lower than SVM. Despite its simplicity the Naive Bayes classifier achieves better recognition rate on both validation settings than the Decision Tree classifier.

5.3. Combination strategy performance

We evaluated the set of all possible classifier combinations with at least two classifiers. Table 2 presents a few test examples to illustrate our evaluation methodology. In Table 2 ‘AS’ denotes the couple Actionlet-SVM, ‘AF’ stands for Actionlet-RandomForest, ‘AT’ stands for Actionlet-DecisionTree, ‘AB’ stands for Actionlet-NaiveBayes, ‘HS’ stands for HON4D-SVM, ‘HF’ stands for HON4D-RandomForest, ‘HT’ stands for HON4D-DecisionTree and ‘HB’ stands for HON4D-NaiveBayes.

Table 2: Summary of different features and classifiers used

<table>
<thead>
<tr>
<th>Test</th>
<th>AS</th>
<th>AF</th>
<th>AT</th>
<th>AB</th>
<th>HS</th>
<th>HF</th>
<th>HT</th>
<th>HB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>used</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>used</td>
<td>x</td>
<td>x</td>
<td>used</td>
</tr>
<tr>
<td>2</td>
<td>used</td>
<td>used</td>
<td>used</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>3</td>
<td>x</td>
<td>used</td>
<td>used</td>
<td>x</td>
<td>x</td>
<td>used</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>4</td>
<td>used</td>
<td>used</td>
<td>used</td>
<td>x</td>
<td>used</td>
<td>x</td>
<td>used</td>
<td>x</td>
</tr>
<tr>
<td>5</td>
<td>used</td>
<td>used</td>
<td>used</td>
<td>used</td>
<td>used</td>
<td>used</td>
<td>used</td>
<td>used</td>
</tr>
</tbody>
</table>

To speed up calculations in the fusion process based on belief functions, we have reduced $C$ to the elements that get at least one vote without influencing on the outcome.

Figure 2 presents the performance of the four proposed combination methods: simple vote combination, linear combination, Bayesian combination and belief based combination. In this figure, we present the performance of every possible source combination (with at least two sources). The performance of each classifier alone are summarized in table 1. It clearly appears, from Figure 2, that the simple vote combination is outperformed by the linear and the belief based approaches. These results are due to the lack of weighting for each classifier vote based on a confidence measure. The linear combination outperforms in all tests the vote combination approach which also proves that the choice of the weights, based on the feature distribution for each class is effective. Moreover, as expected it appears that the vote combination is sensitive to the number of sources. Indeed, the less sources vote, the more each vote counts and when sources are not reliable this leads to very poor results. An even number of sources also amplifies this phenomenon because when sources are not reliable, votes tends to be in conflict, leading to no majority so the conflict class $C_0$ will be selected. Using weights allows the linear combination to be more robust to erroneous votes and to outperform in some cases the best classifier alone.

Figure 2 shows that the Bayesian combination gives poor performance compared to the other combination strategies. It may be noted that the probabilities $P(C_j|C)$ for the Bayesian combination were estimated on small set, i.e. a dozen examples for each action in the cross-subject test. With a larger and more representative learning set, one can expect that the Bayesian fusion can achieve performance at least equivalent to the vote combination. We can observe that the proposed belief function based fusion approach always outperforms the Bayesian, the weighted linear and the voting fusion approaches, especially for the classifiers with lower classification performances. The gap between the accuracies of the single best classifier and the belief function based fusion presumably shows the potential of this method.

Using only the best feature extraction method, the performance of the four classifier output combination is reported in Table 3. The linear method of combining individual classifiers with trained weights achieves higher accuracies than the vote combination rules. Thanks to the a priori weights computed on the learning set, the linear combination can be successful even when a majority of classifiers fail to find the right action by giving more credit to the best classifiers. Same behavior can be observed for the Bayesian combination as in Figure 2. The belief function based combination gives the best accuracy on both 10-cross and cross-subject validation protocols. Interestingly, although for the majority of the combination tests there is an improvement, some cases of the fusion schemes did not improve very much on the single-best classifier rate. This is probably due to dependencies between the classifiers. If we used a large number of features and built the classifiers on disjoint subsets the chance to obtain good improvement over the single best classifier would have been higher.
Table 1: Test accuracy (%) of feature extraction methods using different classification models

<table>
<thead>
<tr>
<th>Features</th>
<th>SVM</th>
<th>Random Forest</th>
<th>Decision Tree</th>
<th>Naive Bayes</th>
<th>SVM</th>
<th>Random Forest</th>
<th>Decision Tree</th>
<th>Naive Bayes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actionlet</td>
<td>90.7152</td>
<td>81.0986</td>
<td>57.4614</td>
<td>59.3153</td>
<td>82.4161</td>
<td>72.4095</td>
<td>50.4799</td>
<td>55.4239</td>
</tr>
<tr>
<td>HON4D</td>
<td>86.8129</td>
<td>81.9493</td>
<td>52.6397</td>
<td>34.5985</td>
<td>79.0652</td>
<td>73.8268</td>
<td>46.6499</td>
<td>33.8395</td>
</tr>
</tbody>
</table>

Figure 2: Performance with all classifier combinations. White squares mean the corresponding source is used and black squares it isn’t used.

Table 3: Test accuracies (%) of classifier combination methods.

<table>
<thead>
<tr>
<th>Combination strategy</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10 cross-validation</td>
</tr>
<tr>
<td>Vote</td>
<td>88.54</td>
</tr>
<tr>
<td>Linear</td>
<td>90.53</td>
</tr>
<tr>
<td>Bayesian</td>
<td>68.68</td>
</tr>
<tr>
<td>Belief function</td>
<td>90.72</td>
</tr>
</tbody>
</table>

5.4. Influence of the discount

In this section, we study the effect of discounting information sources when using the belief based combination. From results shown in Figure 3, the discount operation improves the performance of the belief based combination. The improvement increases when we reject examples $s_i$ for which the conflict is above the threshold $t$. One can see an average performance improvement for the test subsets $S_t = \{ x \in \Phi(x) \leq t \}$. Taking into account the estimated performance of the classifier (i.e. performance on the learning set), thanks to the discount operation, allows accuracy enhancement as shown in figure 3 (test 1 in table 2).

5.5. Influence of the conflict on decision performance

Thanks to the conflict management approach, the belief function based combination accuracy is improved. In Figure 2 with a conflict threshold of 0.3, our belief based approach outperforms all other combination strategies, including the performance of the best classifier alone. With this threshold we reject 63.9% of examples. With a rejection rate of 28.7%, this approach still outperforms the best classifier alone in 87.5% of tests. Finally, the belief function based combination is more robust to non reliable sources than other approaches and is able to keep good performance even when many classifiers are not reliable. This behavior can be explained from the double weighting of each vote, based on the reliability of the classifier for each class and the spatial distribution of each class in the feature space. These criteria are computed on the training set so there is no prior knowledge needed. However a representative training set is even more important with this approach since our weights depend on it.

In Figure 4, we study the performance of the method with respect to the conflict nature. Figure 4 shows that the use of the intrinsic conflict outperforms the use of the total conflict, which itself outperforms the global conflict. The main issue with the global conflict is that, the more sources there are the higher is the conflict.
The system can be extended to any feature and classifier to improve performance. In a multi-class problem, allowing to combine efficiently features and classifiers that are efficient only on a specific class can lead to enhanced performance.

The limit of this model is the performance of the classification depending on the conflict which appears not to be a monotonic function.

Table 4 shows a comparative evaluation of our method and five other methods. The comparison is performed on the cross-subject test setting. The recognition accuracy of the dynamic temporal warping is only 54%, because some of actions in the dataset are very similar to each other, and there are typical large temporal misalignment in the dataset. The accuracy of recurrent neural network is 42.5%. The accuracy of Hidden Markov Model is 63%. The accuracy of the Histograms of 3D Joints based method is 78.97%. The proposed method using the best feature extraction method and the best classifier (Actionlet and SVM) achieves an accuracy of 82.42%. The same classifier achieves 79.07% using the HON4D feature. Using the discounted belief function combination strategy, our method achieves an accuracy of 85.96% which outperforms state-of-the-art methods. Table 4 also shows the benefits of using the conflict management approach. This later improves significantly the belief based combination accuracy with a rejection rate equals to 14.8%. Figure 5 presents the average accuracy achieved when using the intrinsic conflict. The performance increases in nearly linear from 0 to 50% rejection rate.

6. Conclusion

In this paper, we effectively compared the effect of recently emerged techniques used in action recognition, including two efficient feature extraction methods, four well-known classification methods, four classifier combination strategies. We also proposed a classifier combination approach based on the belief function theory. We tested our system on the MSRAction3D dataset. The results showed the interest of the belief function based approach, which achieves higher accuracy compared to other combination strategies. The experimental results also reveal the advantage of the discount operation, where the credibility of a source is defined according to its performance on the learning set. This paper described also a rejection approach based on the conflict from bbas combination. The definition of the conflict is not trivial and we studied the influence of different conflict definitions. Using the Dempster-Shafer theory in classifiers combination, with reject, allows this system to outperform other combination approaches as well as the best classifier alone. The proposed approach is robust to unreliable sources and a small number of sources than other combination approaches. The reject approach is useful in a human-computer interaction context since it allows to choose a tradeoff between the erroneous decision rate and the number of times the user has to perform an action again.

The system can be extended to any feature and classifier to improve performance. In a multi-class problem, allowing to combine efficiently features and classifiers that are efficient only on a specific class can lead to enhanced performance.

The limit of this model is the performance of the classification depending on the conflict which appears not to be a monotonic function.

Table 4: Comparison with state-of-the-art methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recurrent Neural Network [37]</td>
<td>42.50</td>
</tr>
<tr>
<td>Dynamic Temporal Warping [38]</td>
<td>54</td>
</tr>
<tr>
<td>Hidden Markov Model [39]</td>
<td>63</td>
</tr>
<tr>
<td>Action Graph on Bag of 3D Points [36]</td>
<td>74.70</td>
</tr>
<tr>
<td>HON4D + SVM</td>
<td>79.07</td>
</tr>
<tr>
<td>Actionlet + SVM</td>
<td>82.42</td>
</tr>
<tr>
<td>Belief function</td>
<td>81.46</td>
</tr>
<tr>
<td>Belief function + discount</td>
<td>85.96</td>
</tr>
<tr>
<td>Belief function + conflict management</td>
<td>90.34</td>
</tr>
</tbody>
</table>

References
