Pattern Recognition Letters 34 (2013) 20–24

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

# Multi-view action recognition using local similarity random forests and sensor fusion

Fan Zhu[a], Ling Shao[a,*], Mingxiu Lin[b]

[a] Department of Electronic and Electrical Engineering, The University of Sheffield, UK
[b] College of Information Science and Engineering, Northeastern University, China

ARTICLE INFO

*Article history:*
Available online 6 May 2012

*Keywords:*
Local similarity
Random forests
Sensor fusion
Voting strategy
IXMAS
Action recognition

ABSTRACT

This paper addresses the multi-view action recognition problem with a local segment similarity voting scheme, upon which we build a novel multi-sensor fusion method. The recently proposed random forests classifier is used to map the local segment features to their corresponding prediction histograms. We compare the results of our approach with those of the baseline Bag-of-Words (BoW) and the Naïve–Bayes Nearest Neighbor (NBNN) methods on the multi-view IXMAS dataset. Additionally, comparisons between our multi-camera fusion strategy and the normally used early feature concatenating strategy are also carried out using different camera views and different segment scales. It is proven that the proposed sensor fusion technique, coupled with the random forests classifier, is effective for multiple view human action recognition.

## 1. Introduction

Recent progresses on networking, image processing, and data storage have dramatically improved multi-media technologies in a variety of fields, including action recognition, video surveillance, robotics, human computer interaction and many others. Additionally, recent hardware developments also help to enhance the techniques to a much higher degree so that more advanced algorithms can be designed to deal with more challenging scenarios. A typical example is the recently released three-camera Microsoft Kinect sensor (Sung et al., 2011), which can capture both the RGB color and the depth information of a scene. For recognizing human activities in video, such a multi-spectrum ideology can be applied to elevate the recognition performance by increasing the number of input sensors. However, due to the unequal manifestation of different sensors, there is a high requirement for the recognition system with respect to sensor fusion strategies. Specifically, because of the unavoidable self-occlusions from each observation sensor, human activities would be observed with a high disparity from different sensors, which would consequently result in low recognition accuracy.

Many previous human action recognition works have considered challenging problems, such as illumination or background variations, occlusions and viewpoint changes (Souvenir and Babbs, 2008). Among them, data with viewpoint changes are very common and basically inevitable in real-world applications due to human or camera movements. The apparent deficiency of single-camera system prompts the advancement of recent approaches using multiple-cameras to deal with such viewpoint change problems. Action recognition results have been reported in (Weinland et al., 2010; Lv and Nevatia, 2007; Weinland, 2006) on the IXMAS dataset (Weinland et al., 2007), which includes human actions captured from five different viewpoints. However, most of the techniques only focus on the performance of each individual camera, but neither seeks for a fusion solution based on multi-sensors nor evaluates the overall performance using certain fusion approaches. Most of the activity recognition methods use spatial-temporal feature descriptors, e.g., 3DHOG (Klaser et al., 2008), and apply bag-of-words on these features so that a global representation can be generated for each action sequence to be labeled by a classifier, such as SVM or KNN. These approaches, however, can only produce impressive results under conditions that the datasets are relatively less challenging in either limited viewpoint changes or uncomplicated backgrounds. In addition, the performance of feature distance-based classifiers, such as KNN, would degrade when dealing with data that are composed of highly independent subsets, which can come from the multi-sensor data fusion phase.

In this paper, we propose a simple approach that employs local segments of binary silhouettes on the random forests classifier, and then apply a novel voting strategy to label the testing actions. The random forests was introduced in (Breiman, 2001), and it has the advantages over other learning algorithms in efficiency and effectiveness, and it can avoid the over-fitting problem by setting more decision trees. Although the action representation is simple and

[*] Corresponding author. Tel.: +44 1142225841.
*E-mail address:* ling.shao@sheffield.ac.uk (L. Shao).

0167-8655/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2012.04.016

not robust against viewpoint changes, we can still get impressive results due to the effectiveness of the random forests as a classifier and the voting strategy. We also extend our approach from single camera view to multi-camera fusion and evaluate the performance of different camera fusion scenarios on the IXMAS dataset. The remaining sections of this paper are structured as follows: Related work on multi-view action is summarized in Section 2. Section 3 introduces the segment division process and the local similarity random forests classifier. Section 4 presents the details of the voting strategy. Results of our proposed approach on both single camera view and multi-camera fusion are described and discussed in Section 5. And finally, conclusion is given in Section 6.

## 2. Related work

Action recognition algorithms based on multi-view cameras have recently received considerable attentions. Many approaches have been proposed and tested on the multi-view IXMAS dataset. Fig. 1 illustrates examples of actions and their associated silhouettes from this dataset. In (Shao et al., 2011), Shao et al. adopted body pose silhouettes as feature descriptors to build the Correlogram of Body Poses (CBP) global representation for each video sequence beyond its baseline Histogram of Body Poses (HBP) (Shao and Chen, 2010) representation, and achieved satisfying results on the IXMAS dataset. In (Junejo et al., 2008), Junejo et al. explored the self-similarities of action sequences overtime as a measurement to overcome view-changes. However, recognition in all these methods is done on individual cameras and the fusion of different camera views is neglected.

The idea of decision trees and its extension, decision forests, have been previously studied for both action localization and recognition, e.g., in (Reddy et al., 2009; Yo et al., 2011; Nebel et al., 2011). In (Lin et al., 2009), an action prototype tree is learned in both shape and motion spaces using the hierarchical *k*-means clustering. Then they take a prototype-prototype distance from the codebook as a measurement, upon which they calculate the joint likelihood with both action location and prototype. In (Mikolajczyk, 2008), in contrast to most previous recognition works that utilize small and flat codebooks, they apply a large number of features represented in many vocabulary trees instead. In addition to action recognition, their approach also accomplishes action localization simultaneously. An image-feature vocabulary using a novel quantization method with randomized trees as an alternative of k-means clustering was proposed in (Philbin et al., 2007). The advantage of choosing randomized trees for vocabulary generation has also been demonstrated in (Lin et al., 2009) that the randomized trees can dramatically outperform k-means in terms of both efficiency and accuracy, especially when dealing with large scale data. In contrast to (Shao et al., 2011; Shao and Chen, 2010; Lin et al., 2009), in which the random forests is employed as a vocabulary generating or indexing tool, Yao et al. (2010) applied the randomized trees to learn the 3D local video patches to acquire their corresponding votes in the 4D Hough-transformed space. The same as in (Shao and Chen, 2010), both action label and location are obtained from their voting framework.

Some existing techniques explored the fusion results from multi-cameras on the IXMAS dataset. Among them, a very common strategy for camera fusion is to concatenate the feature descriptors from different cameras, e.g., in (Wu et al., 2011; Yan et al., 2008; Cilla et al., 2010; Srivastava et al., 2009). Such a fusion method would benefit the recognition accuracy by describing the actions with more features. However, it will consequently lead to two new problems. Firstly, concatenating the feature descriptors will result in a much longer feature descriptor, which can be five times long for five cameras. Therefore, this will naturally increase the computational complexity for clustering and dimensionality reduction. The second shortcoming, which is more critical, is the potential likelihood that the fused recognition accuracy would deteriorate due to the unequal performance of each camera with respect to different actions. Correspondingly, the first problem can be solved with the randomized forests classifier and the second can be tackled by our new voting strategy.

## 3. Local segment representation and randomized tree training

### 3.1. Segment of 2D silhouettes

Silhouette extraction is a popular technique for action recognition. With the silhouette data, intra-class variations, such as background changes and clothing, that may affect the recognition performance are easily overcome. Obviously, the quality of the silhouettes is closely related to the recognition performance. Since silhouette extraction is not the focus of this paper and the 2D silhouette per frame for each action sequence is provided by the IXMAS dataset, we simply use those silhouettes to represent body poses without discussing how the silhouettes are extracted. A bounding box is placed around each silhouette and normalized to the size of $20 \times 30$, which is then converted into a 600 dimensional descriptor containing only binary values. As shown in Fig. 2, the two sequences at the right side of the graph illustrate two sets of 2D binary bounding boxes of camera 0 and camera 4.

In order to consider the temporal order of poses, we use temporally densely sampled segments, which have overlaps with neighboring segments. Each segment is set to be the size of $20 \times 30 \times T$, where $T$ refers to the segment's length in frame number. With this setting, each segment has the full spatial size of the input

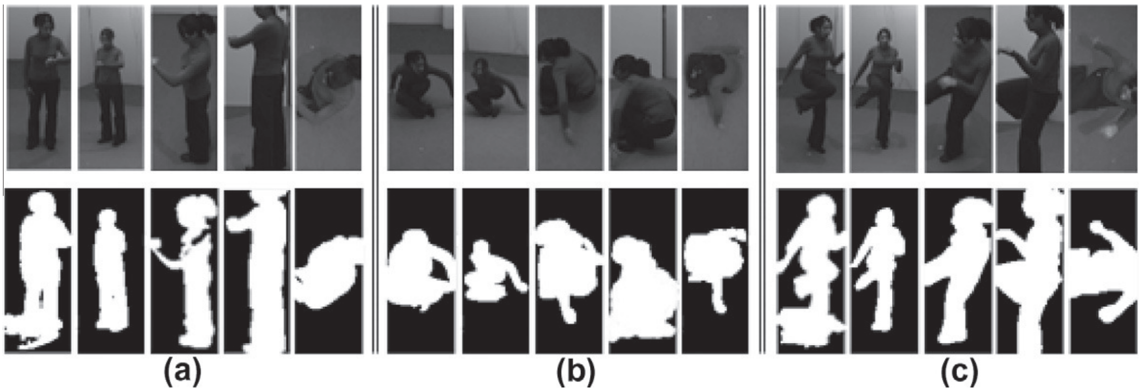


**Fig. 1.** Body poses from (a) check watch, (b) sit down, and (c) kick. Each action is performed by the same person Amel and captured from cameras 0–4.

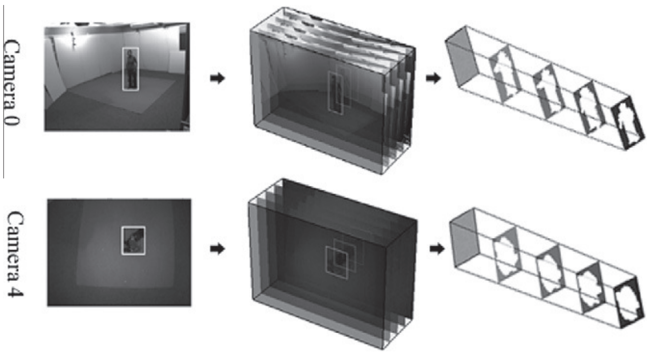


**Fig. 2.** Silhouettes extracted from different camera views.

silhouette sequence and only varies temporally. A segment is then represented by further concatenating each row of its 2D silhouettes, which results in a 600 × $T$ dimensional segment descriptor. It is claimed in (Cao et al., 2012) that the very densely sampled segments can help boost the recognition performance, thus we place the segments as densely as possible in the temporal axis. Specifically, the overlap between consecutive segments is $T-1$ frame, i.e., the step for the sliding segment is one frame. Assuming the total frame number of a silhouettes sequence is $N$, $N-T+1$ segments can be generated from a video sequence.

### 3.2. Randomized tree training

The training process is constructed according to the standard random forests structure in (Breiman, 2001). The local segments from the training sets are trained with the random forests classifier, which is assembled by a set of randomized decision trees. In each decision tree, $M$ segment features are randomly selected from the training sets and placed at a root node, which is mapped to a set of termination leaf nodes through the interior binary splitting joints. At each interior joint, $f$ variables are randomly selected out of the $F$ feature dimension and the decision threshold $t$ is correspondingly chosen in the range $\{t|\min_t f(v_i) \leqslant t \leqslant \max_i f(v_i)\}$. The splitting function is defined as:

$$f_{l,r}(v_i) = \begin{cases} 1, & \text{if } \{i \in I_n | f(v_i) > t\} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

After training, assuming $K$ leaf nodes are generated in a decision tree, each local segment $m \in M$ must fall into a leaf node $k \in K$. As illustrated in Fig. 3, the class label at a leaf node $k$ $p_c^k$, refers to the proportion of segments within each action class that reaches this leaf node after training, i.e., $\sum_k p_c^k = 1$.

To measure the training quality of each leaf node, i.e., the proportion of local segments from sequences of a same action falling into the same leaf node, the information gain is defined at each split node:

$$\Delta E = -\frac{|I_1|}{|I_n|}E(I_1) - \frac{|I_r|}{|I_n|}E(I_r) \tag{2}$$

where $\Delta E$ refers to the information gain, $E(\,)$ denotes entropy, $I_l$, $I_r$ and $I_n$ indicate the left splitting features, the right splitting features and the total input features at the splitting node respectively. This equation is given with respect to the splitting function at each splitting node. Since the $f$ variables are randomly chosen from the feature vector and the decision threshold $t$ cannot be predefined to maximize the information gain, a set of combinations of '$f$'s and '$t$'s are recursively tried to boost training quality.

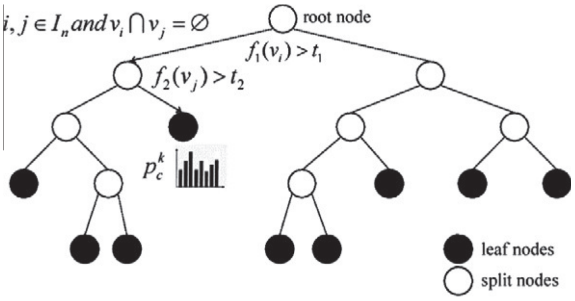


**Fig. 3.** Decision tree growing.

### 3.3. Learning the forests

The training set is equally divided into a number of subsets, which are then dispatched to different decision trees. Normally, to boost the general performance, the subsets are set to have overlaps with each other. Assuming the total training feature number is $N$ and there are $N'$ decision trees within the random forests classifier, the features that are dispatched to each decision tree are more than the number $N/N'$. In the testing phase, each segment feature is pushed to the root node of each decision tree in the random forests classifier, and is forwarded to a terminating leaf node eventually. The path between a root node and a terminating leaf node consists of a set of split nodes, where each split node contains a binary splitting function. When the segment feature drops into a terminating leaf node, a histogram $p_n$, which refers to the proportion of segments per class label that fall into this leaf node during training phase, is the soft voting result at the decision tree $n \in N'$. Finally, the prediction histogram of the whole forests is obtained by summing up the voting histograms from all the decision trees:

$$P_f = \sum_{n=1}^{N'} P_n \tag{3}$$

## 4. Multi-camera voting strategy

Because of the high feature disparity in both training and testing phases, prediction histograms from different segments would be significantly varied. Specifically, for some histograms, a class label can be quite discriminative with a high proportion of votes gained from the random forests, while others may yield relatively flat voting histograms. Such unequal property would potentially reflect in reduced recognition performance. To overcome the inequality problem and restrain the influence of those ambivalent prediction histograms, we design a voting strategy, which assigns a weight for each feature prediction.

Suppose a testing video sequence is represented by $I = \{I_1, I_2, \ldots, I_T\}$, where $T$ is the total segment number within the testing video sequence and $I_t$ denotes the local feature of segment $t \in T$. Let $Q_c(x)$ be the event that the testing video sequence $x$ belongs to class label $c \in C$ and $P_j(t)$ be the prediction histogram of the segment $t \in [1{:}T]$. If the class label $c' \in C$, i.e., the bin $c$ of a segment's prediction histogram, gets the most votes at the local segment $t$, the probabilities for all the class labels of the testing video sequence $x$ can be computed as:

$$p\left(Q_c(x)|\sum_{t=1}^{T} P_f(t)\right) = \sum_{t=1}^{T} p(Q_c(x)|V_t^{\max} = c', P_f(t))p(V_t^{\max} = c'|P_f(t)) \tag{4}$$

where $V_t^{\max}$ refers to the bin with maximum voting value on the histogram. In other word, this strategy is weighting on all the bins of a local segment prediction histogram, where the weight is obtained by the proportion of the maximum bin value of the histogram to the total number of votes from all the decision trees. Finally, the

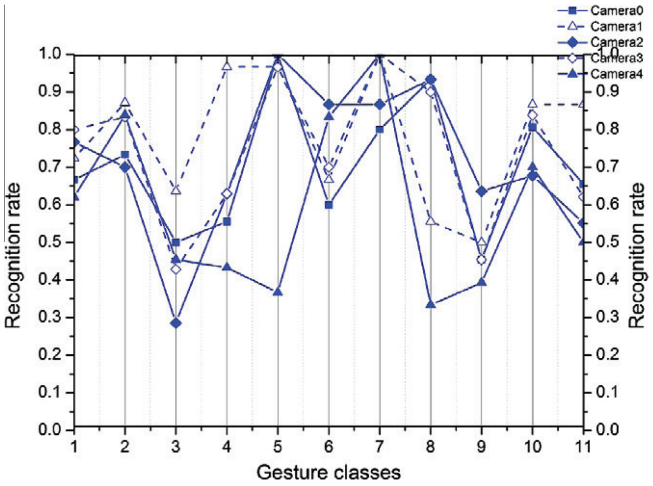


**Fig. 4.** Individual camera performance of all the five cameras.

class label satisfying the following maximization judgment is chosen:

$$\arg\max_{c}\left(p\left(Q_c(x)|\sum_{t=1}^{T}P_f(t)\right)\right) \tag{5}$$

The multi-camera fusion strategy is designed by further assigning a weight onto the prediction histogram of each camera view. Similar to the local voting strategy, the weight is computed by the proportion of the segments that have the same maximum voting class label in their prediction histograms to the total number of segments within the video sequence, where the segments with this class label are more than those with other class labels. Such a weighting strategy is based on the fact that cameras from different observation views would have fluctuating performance against different specific actions, as shown in Fig. 4. The weight can be described as:

$$W^{v} = \frac{N^{v}_{c=c''}}{T} \tag{6}$$

where $W^v$ represents the weight for each camera view $v = 0, 2, \ldots, 4$, $T$ is the total segment number within the testing video sequence, $N^v_c$ refers to the segments number with class label $c \in C$ at camera view $v$ and $c'' \in C$ denotes the class label that contains most segments among all the class labels. Since each camera has the same number of segments for the same testing video sequence and the same number of decision trees to classify each local segment, the multi-camera fusion result is obtained by accumulating each camera's weighted prediction histogram without further normalization:

$$\arg\min_{c}\left(W^{v}\otimes p\left(Q^{v}_{c}(x)|\sum_{t=1}^{T}P^{v}_{f}(t)\right)\right) \tag{7}$$

## 5. Experimental results

### 5.1. Dataset

We test and compare our method with other techniques on the IXMAS dataset (Weinland, 2006). This dataset includes 11 actions: check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, kick and pick up, each of which action is performed 3 times by 10 different actors (5 males and 5 females). Five camera views are available for each action in the dataset, which makes it possible for us to demonstrate the performance of both the single view technique and our multi-camera fusion strategy.

### 5.2. Evaluation

For evaluation, we choose the leave-one-out cross-validation method, i.e., in each iteration action sequences performed by one out of ten subjects are selected as the testing set and the remaining sequences as training set, and the final recognition rate is the average of the ten iterations. To optimize the performance, we vary the segment length $t$ from 8 to 18, and the best result is achieved when $T = 18$. In the case of $T = 18$, a segment feature has the dimension $d = 20 \times 30 \times 18 = 10800$. Then we reduce the dimensionality of such a binary representation using PCA and set the reduced dimension to be $k = 30$. For the random forests classifier, we set the number of decision trees to be 600 and the number of predictors sampled at each splitting node to be equal to the square of the feature dimension.

To demonstrate the effectiveness of our method, we first compare the results of our method with those of the baseline BoW (Nowak et al., 2006) and the NBNN (Boiman et al., 2008) methods, which both employ the same segment representation as in the proposed algorithm for fair comparison. Fig. 5 depicts recognition results of these three techniques when individual camera views are used. For most views, the results of the BoW method and the proposed method are analogous, while both significantly outperform the NBNN method. Table 1 shows results on different combination of camera views with different methods. The highest classification accuracy we achieve is 88%, which outperforms most of the methods in comparison. The analogous performance between the proposed method and the BoW method of each single view (shown in Fig. 5) also proves the effectiveness of the proposed camera fusion strategy as it outperforms the BoW concatenation fusion method by almost 10%. As shown in Table 1, the AFMKL method achieves the best results. Note that the learning process of the AFMKL method is much more complicated than our method (where we only take the concatenations of binary silhouettes in different frame segments as inputs of the random forests classifier) that the performance of the AFMKL method for each single camera view is consequently much better, i.e., 5% better in average for Cameras0-2. However, for the fusion performance comparison, the result of our method is only 0.2% lower than the AFMKL method, which, therefore, proves the effectiveness of our fusion strategy.

Table 2 shows the comparison between the early concatenation fusion method and the fusion strategy we propose. Our approach outperforms the early concatenation fusion method inmost scenarios. For Cameras 0&1 comparison, the reason that the early concatenation fusion method is slightly better than our method might be that the individual performance of Camera0 and Camera1 over different action classes has similar distribution, which conse-

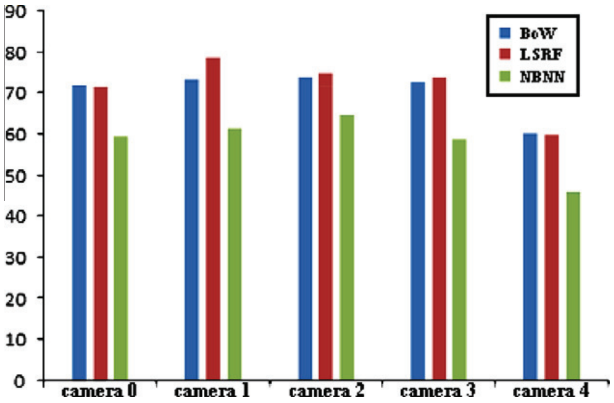


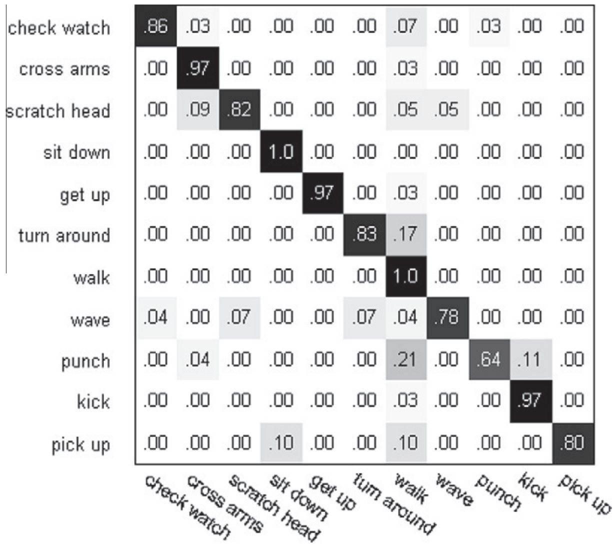
**Fig. 5.** Comparison between our method and the BoW and the NBNN methods on each camera view.

**Table 1**
Classification accuracies (%) of different methods for both single and multiple camera views on the IXMAS dataset.

| Method | Camera 0 | Camera 1 | Camera 2 | Cameras 0 and 2 | Cameras 0–2 | Cameras 0–4 |
|---|---|---|---|---|---|---|
| Proposed method | 71.5 | 78.7 | 73.9 | 85.7 | 86.6 | **88** |
| BoW (Nowak et al., 2006) | 71.6 | 72.3 | 72.7 | 74.2 | 79.1 | 78.7 |
| NBNN (Boiman et al., 2008) | 59.5 | 61.3 | 64.8 | 62.4 | 63.1 | 61.1 |
| AFMKL (Wu et al., 2011) | 81.9 | 80.1 | 77.1 | 86.6 | 87.7 | **88.2** |
| GMKL (Varma and Babu, 2009) | 76.4 | 74.5 | 73.6 | 76.2 | 81.3 | 81.3 |
| Liu and Shah (2008) | 76.7 | 73.3 | 72.1 | - | - | 82.8 |

**Table 2**
Classification accuracy (%) comparison between early concatenation fusion strategy and our fusion strategy under different scenarios.

| Segment temporal scale | Early concatenation fusion | | Our method | |
|---|---|---|---|---|
| | Cameras 0&1 | Cameras 0–4 | Cameras 0 and 1 | Cameras 0–4 |
| 8 Frames | 72.9 | 77 | **74** | **80.7** |
| 10 Frames | 75.3 | 78.5 | **78.3** | **83.9** |
| 18 Frames | **80.5** | 85.4 | 80.1 | **88** |



| | check watch | cross arms | scratch head | sit down | get up | turn around | walk | wave | punch | kick | pick up |
|---|---|---|---|---|---|---|---|---|---|---|---|
| check watch | .86 | .03 | .00 | .00 | .00 | .00 | .07 | .00 | .03 | .00 | .00 |
| cross arms | .00 | .97 | .00 | .00 | .00 | .00 | .03 | .00 | .00 | .00 | .00 |
| scratch head | .00 | .09 | .82 | .00 | .00 | .00 | .05 | .05 | .00 | .00 | .00 |
| sit down | .00 | .00 | .00 | 1.0 | .00 | .00 | .00 | .00 | .00 | .00 | .00 |
| get up | .00 | .00 | .00 | .00 | .97 | .00 | .03 | .00 | .00 | .00 | .00 |
| turn around | .00 | .00 | .00 | .00 | .00 | .83 | .17 | .00 | .00 | .00 | .00 |
| walk | .00 | .00 | .00 | .00 | .00 | .00 | 1.0 | .00 | .00 | .00 | .00 |
| wave | .04 | .00 | .07 | .00 | .00 | .07 | .04 | .78 | .00 | .00 | .00 |
| punch | .00 | .04 | .00 | .00 | .00 | .00 | .21 | .00 | .64 | .11 | .00 |
| kick | .00 | .00 | .00 | .00 | .00 | .00 | .03 | .00 | .00 | .97 | .00 |
| pick up | .00 | .00 | .00 | .10 | .00 | .00 | .10 | .00 | .00 | .00 | .80 |

**Fig. 6.** The confusion matrix of Cameras 0–4.

quently leads to mediocre performance of our fusion strategy. The confusion matrix for 11 actions, when fusing five camera views, is shown in Fig. 6. The reason why the action ‘‘punch’’ has the lowest recognition rate may be that most ‘‘punch’’ actions are performed by the actors’ arms which move in front of the actors’ upper bodies so that the variations cannot be reflected on the 2D silhouettes. And the reason why the ‘‘turn around’’ actions are always miss-classified with the ‘‘walk’’ actions may be due to the high similarity between these two actions.

## 6. Conclusion

In this paper, we propose a novel method for action recognition based on the random forests and a multi-sensor fusion strategy. Since the focus is on classification and multi-sensor fusion, we directly use the silhouettes available in the IXMAS dataset to represent local segments. Our multi-sensor fusion strategy is built to overcome the unequal classification capabilities that would happen due to the high disparity in different observation views. To achieve this, we weight on each camera prediction histogram, inside which each voting segment is first weighted with respect to classification results of all decision trees in the entire random forests. We demonstrate both the multi-sensor fusion results and the single-sensor results using different segment scales, and compare them with the baseline BoW and the NBNN methods. Extensive experimental results show that the proposed algorithm outperforms the above two methods and our 5 camera fused result is comparable with state-of-the-art solutions even though we only use a primitive feature representation.

## References

Boiman, O., Shechtman, E., Irani, M., 2008. In defense of nearest-neighbor based image classification. In: CVPR.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Cao, X., Ning, B., Yan, P., Li, X., 2012. Selecting key poses on manifold for pairwise action recognition. IEEE Trans. Ind. Inform. 8 (1), 168–177.

Cilla, R., Patricio, M.A., Berlanga, A., Molina, J.M., 2010. Fusion of single view soft k-NN classifiers for multicamera human action recognition. In: Corchado, E., Grana Romay, M., Manhaes Savio, A. (Eds.), HAIS, 2010.

Junejo, N., Dexter, E., Laptev, I., Perez, P., 2008. Cross-view action recognition from temporal self-similarities. In: ECCV.

Klaser, A., Marszalek, M., Schmid, C., 2008. A spatial-temporal descriptor based on 3D-gradients. In: BMVC.

Lin, Z., Jiang, Z., Davis, L.S., 2009. Recognizing actions by shape-motion prototype trees. In: ICCV.

Liu, J., Shah, M., 2008. Learning human actions via information maximization. In: CVPR.

Lv, F., Nevatia, R., 2007. Single view human action recognition using key pose matching and viterbi path searching. In: CVPR.

Mikolajczyk, K., 2008. Action recognition with motion-appearance vocabulary forest. In: CVPR.

Nebel, J.C., Lewandowski, M., Thevenon, J., Martinez, F., Velastin, S., 2011. Are current monocular computer vision systems for human action recognition suitable for visual surveillance applications? In: ISVC.

Nowak, E., Jurie, F., Triggs, B., 2006. Sampling strategies for bag-of-features image classification. In: ECCV.

Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A., 2007. Object retrieval with large vocabularies and fast spatial matching. In: CVPR.

Reddy, K., Liu, J., Shah, M., 2009. Incremental action recognition using feature tree. In: ICCV.

Shao, L., Chen, X., 2010. Histogram of body poses and spectral regression discriminant analysis for human action categorization. In: BMVC.

Shao, L., Wu, D., Chen, X., 2011. Action recognition using correlogram of body poses and spectral regression. In: ICIP.

Souvenir, R., Babbs, J., 2008. Learning the Viewpoint manifold for action recognition. In: CVPR.

Srivastava, G., Iwaki, H., Park, J., Kak, A.C., 2009. Distributed and lightweight multi-camera human activity classification. In: ICDSC.

Sung, J., Ponce, C., Selman, B., Saxena, A., 2011. Human activity detection from RGBD images. In: AAAI workshop on Patterns, Activity and Intent Recognition.

Varma, M., Babu, B.R., 2009. More generality in efficient multiple kernel learning. In: ICML.

Weinland, D., 2006. Free viewpoint action recognition using motion history volumes. In: CVIU.

Weinland, D., Boyer, E., Ronfard, R., 2007. Action recognition from arbitrary views using 3d exemplars. In: ICCV.

Weinland, D., Ozuysal, M., Fua, P., 2010. Making action recognition robust to occlusions and viewpoint changes. In: ECCV, 2010.

Wu, X., Xu, D., Duan, L., 2011. Action recognition using context and appearance distribution features. In: CVPR.

Yan, P., Khan, S.M., Shah, M., Florida, C., 2008. Learning 4D action feature models for arbitrary view action recognition. In: CVPR.

Yao, A., Gall, J., Van Gool, L., 2010. A hough transform-based voting framework for action recognition. In: CVPR.

Yo, G., Yuan, J., Liu, Z., 2011. Unsupervised random forest indexing for fast action search. In: CVPR.