

Action Recognition using Space-time Shape Difference Images

Hao Qu, Liang Wang and Christopher Leckie
Department of Computer Science and Software Engineering
The University of Melbourne, Parkville 3010 Australia
 {lwwang, caleckie}@csse.unimelb.edu.au

Abstract

A common approach to human action recognition is to use 2-D silhouettes in the space-time volume as a basis for further extraction of useful features. In this paper, we present a novel motion representation based on difference images. We show that this representation exploits the dynamics of motion, and show its effectiveness in action recognition. Moreover, experimental results demonstrate that this method is highly accurate and is not sensitive to the resolution of the video.

1. Introduction

Action recognition aims to solve the problem of classifying human actions into certain categories. The importance of human action recognition lies in many areas, such as video surveillance, human-computer interaction and content-based video retrieval.

The difficulty of the task arises in three aspects. First, variations of the same action occur due to environmental differences, such as variation in background. Second, different viewpoints and distances result in very different images. Third, the quality of videos themselves vary in resolution and frame rate. Lastly, videos sequences contain high dimensional data, which poses a challenge to fast recognition. Humans can generally extract relevant information out of video sequences very quickly. Handling these four aspects is natural to humans but not for computers.

Much research has been done in all these aspects in the past few years through the advancement of computer vision techniques. Correct feature extraction has proven to be essential for the task of recognising actions in videos. There are two popular approaches to feature extraction in human action recognition. One is using whole video sequences (i.e., global features). Gorelick et al. [3] employs silhouettes as well, and utilises the Poisson equation to extract space-time features. The

other is analysing certain regions of video sequences (i.e., local features). Dollar et al. [2] uses a number of spatio-temporal cubes to represent interesting features in a video. A similar approach was presented in [4], which focused on regional temporal shape contexts.

However, a limitation of using local features is that it requires an accurate interest point detector. A poor interest point detector will result in low classification rates. Wang et al. [10] use a graph embedding method for exploring various dynamic shapes. In [6], the authors correlate spatio-temporal shapes to video clips that have been automatically segmented to avoid silhouette extraction. However, its accuracy is largely dependent on good flow-based correlation techniques.

Our work in this paper uses sequences of full silhouette frames as our input. We assume that the silhouette is known and the camera view is fixed. These two conditions can hold for many real applications, as in the real world the background is often fixed, which makes silhouette extraction feasible. Therefore, we assume that silhouettes contain sufficient information for action recognition and we use this information as the basis for feature extraction and classification.

We propose a global feature that extracts the difference points between frames. We evaluate our method on a standard dataset and show our approach can capture the characteristics of motion effectively. The overall accuracy (100.0%) is very high compared to other recent methods proposed by several other papers.

Our paper is structured as follows. In Section 2, we describe our approach to feature extraction, representation and classification. Next, we evaluate our method in Section 3. We summarise our work in Section 4.

2. Our Algorithm for Feature Extraction

The framework we use is similar to the algorithm proposed in [2] except that we have a different approach to feature extraction. In the following, we describe the procedure we have followed, as shown in Fig. 1.

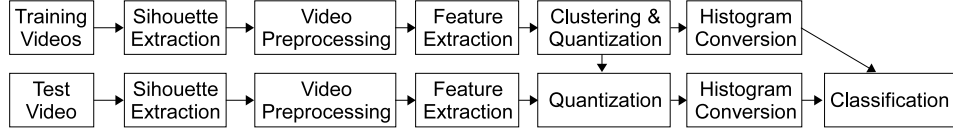


Figure 1. The general framework of human action recognition.

2.1 Motion Feature Extraction

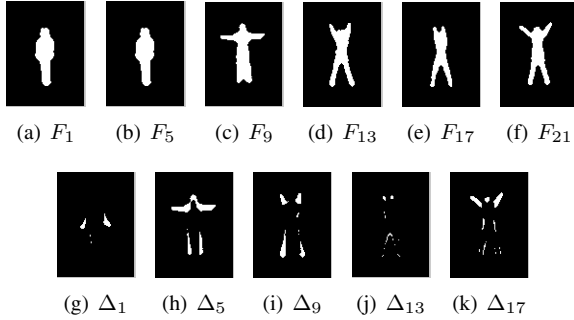


Figure 2. An illustration of our space-time shape difference approach using a jumping-jack action video (Without preprocessing and interleaving of samples for clarity).

Videos can be seen as consecutive frames playing at a constant rate. In order to extract information out of these frames, we need to choose the best frames that can represent the actions. In matrix form, frames can be represented by $F_1, F_2, F_3, \dots, F_n$ etc, where n is the frame count. In our case, these frames are silhouettes of the human figure. Thus each video can be represented by: $V_i = \{F_1, F_2, F_3 \dots F_{n_i}\}$.

For each video, we extract non-consecutive frames first, for example, F_1, F_5, F_9 , etc (if *step size* = 4) (*Fig.2(a) ... (f)*). Here, we assume that we can still observe each action after skipping a certain number of frames.

The preprocessing process applied to each frame comprises two steps:

1. We trim the extra space in the horizontal axis so that the width of the frame has the same width as the human figure;
2. We also resize each video with a fixed resolution to mitigate the difference between distinct human frames. Thus, each video is reduced to $V' = \{F_1, F_{1+step}, \dots F_{n'}\}$ where $n' \leq n$.

In the next step, we calculate the difference between these frames, $\Delta_i = F_{step+i} - F_i$. These difference

frames become our intermediate features, for example, $\Delta_1(\text{Fig.2}(g))$ becomes our first frame, $\Delta_5(\text{Fig.2}(h))$ becomes our second frame feature, etc. Each difference frame is then reformed into a one-dimensional vector (of length $n_x * n_y$ for an n_x by n_y frame) and is then passed to the K-means clustering algorithm.

2.2 Action Descriptor

In the next step, K-means clustering is performed on the feature vectors from the videos for all activities in the training set. Thus from k clusters centers, we have k key frames from the training data. Once we have these key frames, we represent each video by these frames so that the number of key frames contained in these videos are calculated to give a histogram for each video, where for k key frames, the histogram is a vector containing k bins, i.e., the distribution of the number of such key frames for each video (see examples in *Fig. 3*). The classification problem then becomes one of the estimating the similarity among these histograms.

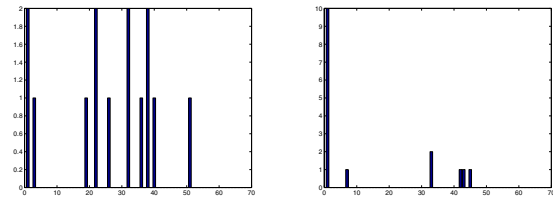


Figure 3. An example of histogram descriptors for a 'jumping-jack' action video (left) and a 'bend' video (right) when $k = 70$ clusters used.

In summary, we use the differences between non-consecutive frames based on the idea that humans are able to distinguish actions by observing the movement differences of body parts over time. Also, large areas of the human body do not provide useful information on actions, e.g., the centre of human body when walking. By using differences between frames we should be able to identify those points of interest while ignoring points that these might only introduce noise.

2.3 Classification

In the classification phase, we use the intermediate features we have generated in the training phase. We use the same preprocessing technique for feature extraction from test videos as from training videos.

From each test video V_t we extract non-consecutive frames with the given step size to obtain the reduced sequence V'_t , from which we calculate the sequence $\Psi_t = \{\Delta_1, \Delta_{1+step}, \dots\}$ of difference frames. Each frame in Ψ_t is matched to the nearest key frame from training, resulting in a histogram $hist_t$ of key frames for the test video:

$$V_t \Rightarrow V'_t \Rightarrow \Psi_t \Rightarrow hist_t$$

Nearest neighbour classification is then used to choose the nearest training video histogram using the χ^2 distance metric. Classification can be summarised as choosing the class that minimises the distance d_{χ^2} between the test video histogram $hist_t$ and the nearest histogram of the training video for that class:

$$Class = \underset{c \in C}{\operatorname{argmin}} \min_{hist_{c_j} \in H(c)} d_{\chi^2}(hist_{c_j}, hist_t)$$

where C is the set of action categories and $H(c)$ is the set of histograms corresponding to the training videos for action c .

3. Evaluation

In this section, we summarise the datasets we used and the results observed from our algorithm.

3.1 Evaluation Dataset

Experiments are performed on the Weizmann dataset containing 93 clips [3], comprising nine different people performing 10 actions such as “run”, “walk”, “skip”, “jumping-jack”, “jump-in-place-on-two-legs”, “jump-forward-on-two-legs”, “wave-one-hands”, “wave-two-hands”, or “bend”. The silhouettes contain noise due to the limits of background subtraction.

3.2 Experimental Procedure

The algorithm parameters include the frame resolution (n_x, n_y pixels in each dimension), the step size between frames, and the number of clusters k . We test a range of combinations of these parameters and we also test the effect of changing each individual parameter. We perform leave-one-out cross validation to measure classification accuracy.

Confusion Matrix, num steps = 4, Clusters = 70, Resolution = 30*60

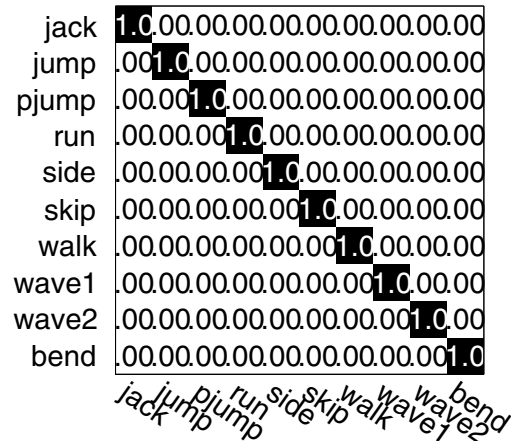


Figure 4. Confusion matrix for our algorithm using the best parameter settings as shown.

3.3 Results and Analysis

Figure 4 show the confusion matrix for our best parameter settings as shown, with the diagonal of this confusion matrix showing the percentage of correctly identified videos and the rest representing the proportion of misclassified videos. Our algorithm correctly classified all classes of videos in the dataset.

3.4 Parameter Evaluation

The parameter configurations we have used in achieving the above results are resolution: $n_x \in [6, 60]$ by $n_y \in [12, 120]$, step difference: $step \in [1, 10]$, number of clusters: $k \in [20, 200]$. On average, the speed of our algorithm is quite fast. Extracting all the features from 93 videos takes 400 seconds and classification on the whole dataset takes around 100 seconds for 10 repeated finds in a Matlab environment operated on Mac OS X 10.6 with CPU: 2.53GHz and 4GB RAM. As can be seen from Figure 5, the step size parameter has the greatest effect on accuracy, thus the granularity of the available key frames is important to our algorithm. While in the resolution test, the error rate (ER) quickly stabilises once the number of pixels in each frame exceeds 300, which shows the robustness of our algorithm to resolution. For the number of clusters k , the error rate quickly improves when k increases until $k = 50$ and our accuracy stabilises.

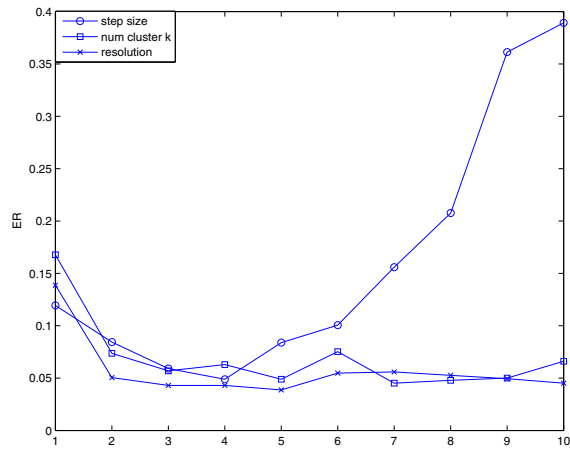


Figure 5. Effect of varying each parameter on error rate (ER).

3.5 Algorithm Comparison

| Methods | # of actions | Accuracy | Classifier |
|---------------------|--------------|--------------|------------|
| Our approach | 9 | 1.000 | 1-NN |
| Liu 2008[7] | 9 | 0.868 | 5-NN |
| Ali 2007[1] | 9 | 0.926 | 5-NN |
| Niebles 2007[8] | 9 | 0.728 | L-SVM |
| Gorelick 2007[3] | 9 | 0.996 | 1-NN |
| Our approach | 10 | 1.000 | 1-NN |
| Hsiao 2008[4] | 10 | 0.972 | NN |
| Jia 2008[5] | 10 | 0.909 | 6-NN |
| Gorelick 2007[3] | 10 | 0.975 | 1-NN |
| Thurau 2007[9] | 10 | 0.867 | 1-NN |

Table 1. Comparison of the classification accuracy for the Weizmann dataset of our methods with the results obtained in recent papers.

As can be seen from Table 1, a comparison is made on two datasets, those with the “skip” action (10 actions) and those without the “skip” action (9 actions). Table 1 also shows the best results reported in recent papers. We can see that the accuracy of our method exceeds all the other methods. The overall recognition rate is 100% in both categories, which is slightly better than the results reported in [3] and much better than the methods reported in [9, 5, 8, 1, 7, 4], whose accuracies are 0.4% to 27.2% lower in 9 action classification, and 2.5% to 13.3% lower in 10 action classification. The

advantages of our method are its simplicity and effectiveness, as it does not require computationally complex interest point detectors and can achieved high accuracy.

4. Conclusion

In this paper we have presented a new method of extracting useful features from human action videos for action recognition. We showed the effectiveness of our method, and compared our results against other well established algorithms, which shows our algorithm has competitive accuracy, is fast, and furthermore, is not very sensitive to video resolution, partial shape deformation of actions nor the number of clusters used.

Future work can include combining other features containing additional shape information, and improving the quality of silhouette extraction.

References

- [1] S. Ali, A. Basharat, and M. Shah. Chaotic invariants for human action recognition. *IEEE International Conference on Computer Vision*, 2007.
- [2] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. *Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005.
- [3] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.
- [4] P. Hsiao, C. Chen, and L. Chang. Human action recognition using temporal-state shape contexts. *IEEE International Conference on Pattern Recognition*, 2008.
- [5] K. Jia and D. Yeung. Human action recognition using local spatio-temporal discriminant embedding. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.
- [6] Y. Ke, R. Sukthankar, and M. Hebert. Spatio-temporal shape and flow correlation for action recognition. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.
- [7] J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.
- [8] J. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance of human action classification. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2007.
- [9] C. Thurau. Behavior histograms for action recognition and human detection. *International Workshop on Human Motion with ICCV*, 2007.
- [10] L. Wang and D. Sutter. learning and matching of dynamic shape manifolds for human action recognition. *IEEE Transactions on Image Processing*, 2007.