A Novel Approach for Fast Action Recognition using Simple Features

Hossein Ragheb, Sergio Velastin, Paolo Remagnino, Tim Ellis

A Novel Approach for Fast Action Recognition using Simple Features

Hossein Ragheb*, Sergio Velastin, Paolo Remagnino and Tim Ellis
Digital Imaging Research Centre, Kingston University, Kingston Upon Thames, UK.

Abstract

We propose a new method for human action recognition from video streams that is fast and robust to random noise, partial occlusions and large changes in camera views. We extract features in the Fourier domain using the bounding boxes containing the silhouettes of a human for a number of frames representing an action. After preprocessing, we divide each space-time volume into space-time sub-volumes and compute their corresponding mean-power spectra as our feature vectors. Our features result in high classification performance using a weighted variant of the Euclidean distance. We require no camera calibration or synchronization and make use of multiple cameras to enrich the training data towards view-invariance. We test the robustness of our method using a variety of experiments including synthetic data generated in a virtual environment and real-world data used by other researchers. We also provide an experimental comparison, using the same data, between our method and two recent alternatives.

1. Introduction

Human action recognition from video streams is a challenging problem in computer vision. Recent advances in camera, computer and data communication technologies have made it more attractive to conduct research towards robust action recognition algorithms. Further, public places are increasingly more equipped with visual surveillance systems without sufficient regulations and solutions on how to process the data corresponding to so many cameras. Researchers are hence more motivated to help managers, law enforcers and control room operators by making it possible to assign parts of the boring watching jobs to computers. There have been a number of interesting works on this topic. For instance, the motion energy image and motion history image templates were introduced by Bobick and Davis [1] to recognize human motions. Rao et al [7] proposed a view-invariant representation of human actions to capture the corresponding dynamics using spatio-temporal curvature of 2-D trajectories. Efros et al [2] used a motion descriptor based on optical flow of a volume. Yilmaz and Shah [11] have used neighboring masks to match corresponding body points on boundaries and extract distinguishing trajectories. Weinland et al [10] have proposed an exemplar-based HMM method that needs multiple training cameras and at least one test camera, where human pose estimations form an action. Similarly, Lv and Nevatia [5] have recognized human poses using a set of learned 3D exemplars but using deterministic action graph models. In the work of Niebles and Fei-Fei [6], features are extracted from static and dynamic interest points in video frames. A hierarchial model delivers the features to a SVM classifier. Finally, Laptev and Perez [4] have trained and tested on real movies using key-frame priming to combine discriminative models of human motion and shape within an action.

We propose a new method and compare it with two existing methods published in [3] and [9]. As in our case, these two methods are based on human body silhouettes. The method by Gorelick et al [3] deals with short space-time volumes and solves a version of the 3D Poisson equation to assign mean values to the points inside the volume. Global features are extracted using local weights in a weighted moments formula. The process is computationally expensive though it is claimed to be robust when recognizing small periods (about one third of a second) of consecutive poses. The method by Wang and Suter [9] makes use of dimensionality reduction approaches to learn the dynamics of each human activity manifold in a compact trajectory form. Space-time points are projected into a low-dimensional space in an attempt to preserve the geometric structure. The median Hausdorff distance is chosen as the similarity measure. Their method seems to be relatively less time-consuming but more sensitive to isolated noise compared to that in [3].

Our method is aimed at dealing with the main challenges that exist in practical conditions, in order to be used in real-time visual surveillance systems in conjunction with tracking algorithms. We require no cam-

*Part of the REASON project funded by the EPSRC and sponsored by the Home Office in the United Kingdom.

era calibrations and make use of multiple cameras during the training phase towards achieving robustness to camera view changes. Using a single training camera, we still achieve reasonable robustness to view changes and to silhouette corruptions. We utilize a large body of robust features in the frequency domain corresponding to power spectra of integrated pixel-time signals for actions represented by space-time volumes. We also show that simple distance measures, such as a weighted Euclidean distance, work well with our features for classification. The method is tested and evaluated both on synthetic and real-world data using a variety of action classes, actors and camera views. Our synthetic data consist of a large number of human action samples generated in a virtual environment [8]. We provide sufficient evidence for the performance evaluation of the method and comparisons with the alternatives.

2. The Action Recognition Algorithm

We only contribute to the mid-level problem of action recognition towards the high-level problem of behavior recognition. We assume that the low-level problems of human tracking and human foreground segmentation are dealt with using existing approaches. In Sections 2.1 we outline how we extract novel features in the Fourier domain. This is followed in Section 2.2 by describing our classification methodology that works well in conjunction with these robust features.

2.1. Feature Extraction

The input to our algorithm consist of the binary mask images corresponding to a human body for F frames starting at $t = 1$ and ending at $t = F$, where t is the frame number. The body mask pixels are represented by '1's in each binary image. The size of each image may vary in the sequence of F frames depending on the body pose at each instant, and so the size of the bounding box is variable. We compute the 2D centre of mass (CoM) corresponding to each mask and shift the mask so that its CoM is located on the centre of an extended image. The identical (and minimum) size of all F extended images is determined so that all pixels of all F body masks are bounded. Obviously, the extended image corresponding to each body mask may contain empty rows and columns next to the body mask. A resizing step may also be used to normalize all extended images. Although some transitional information could be lost, we believe that this process results in the extraction of sufficient and distinct information corresponding to individual action samples. Moreover, we do not aim for an expensive method with perfect performance but for a cost effective one.

The output of the CoM aligning process is an action volume containing F extended images of identical size with M rows and N columns. The next step is to divide each extended image into $U \times V$ non-overlapping windows. Depending on the image size, values of U and V are chosen so that each window contains a sufficient number of rows X and columns Y, where $X = M/U$ and $Y = N/V$. Here we vary the values of U and V to check different combinations and choose the optimum settings. We use the notation $w_{u,v}[t_i]$ to refer to the window located in the vertical division u and horizontal division v, while it is part of the extended image at frame t_i. Further $w_{u,v}$ refers to the space-time sub-volume (a set of windows through time) in which F windows with identical space indices are included. Each space-time sub-volume (STSV) is very likely to correspond to motions in which specific body parts are used. For instance the top-left STSV, i.e. $w_{1,1}$, may correspond to actions such as waving during which the left hand is raised by an actor facing the camera. Such correspondences between the windows and body parts or actions are implicit and we do not attempt to assign or model any explicit correspondences.

To account for the contribution of each STSV in our feature vector, we treat each STSV as a set of $X \times Y$ discrete signals. For simplicity we drop u and v indices from our notations inside the STSV $w_{u,v}$. Here, each pixel through time defines a discrete signal containing a number of shifted unit impulses $\delta[t-t_i]$ where the pixel is '1'. Each pixel value $B^{x,y}[t_i]$, in the STSV $w_{u,v}$ at frame t_i and in the (x, y) location within the window, corresponds to the presence '1' or absence '0' of body pixels in that location. So the pixel presence signal is

$$\phi^{x,y}[t] = \sum_{t_i=1}^{F} \{B^{x,y}[t_i]\delta[t - t_i]\} \tag{1}$$

Next, we compute our power spectrum features corresponding to each STSV. We use the linearity property and first compute the frequency spectrum corresponding to each individual presence signal using the Fourier transform so that

$$\Phi^{x,y}(\Omega) = \sum_{t_i=1}^{F} \{B^{x,y}[t_i] \exp(-j\Omega t_i)\} \tag{2}$$

In practice (using a fast Fourier transform), the number of frequencies K required to be included in the spectrum is an important factor. Using the video rate as K is sufficient to include the major frequencies corresponding to the presence signals. Using larger values of K does increase the precision of the Fourier transform but does not necessarily result in further distinct features. Once the frequency spectrum $\Phi^{x,y}(\Omega)$ is available, the power spectrum $P^{x,y}(\Omega)$ is computed by convolving $\Phi^{x,y}(\Omega)$ with its complex conjugate form:

$$P^{x,y}(\Omega) = \Phi^{x,y}(\Omega) * Conj\{\Phi^{x,y}(\Omega)\} \tag{3}$$

We integrate all the power spectra corresponding to the pixel presence signals in each STSV and compute the mean-power spectrum corresponding to the STSV $w_{u,v}$, which is given by

$$P_{u,v}(\Omega) = \frac{1}{XY} \sum_{x=1}^{X} \sum_{y=1}^{Y} P^{x,y}(\Omega) \tag{4}$$

The mean-power spectrum appearing in Eq. (4) provides us with K feature values for each STSV. Since the number of STSVs in the whole action volume is $U \times V$, the number of elements in the feature vector will be $n_R = U \times V \times K$. These features are stored for each video sample corresponding to an action class included in the training data. The number of action classes considered is n_C. To account for a variety of human body shapes and irregularities in action performance by different people, we include n_A different actors in the training data. To be able to recognize actions captured from any camera view, we include a large number of camera view directions n_D in our training data. We will mention typical values for these method parameters used in our experiments. In what follows $n_S = n_C \times n_A \times n_D$ refers to the number of action video samples included in the training data for which n_S feature vectors are extracted with n_R elements in each vector, making the total number of $n_R \times n_S$ training features.

2.2. Action Classification

In order to classify a number of unknown test actions during the recognition phase, each in one of the n_C action classes, we should extract n_R features from each of the corresponding test video samples. Here one should use a distance measure such as the Euclidean distance and compute the distances between the n_R features corresponding to each of the n_S training samples and n_R features corresponding to each test sample. The action class giving the minimum distance is then chosen as the estimated class for the test action, no matter that the matching action sample corresponds to which training actor or camera view. In conjunction with our mean-power spectrum features, we describe a weighted Euclidean distance that provides us with fast recognition and improved results. It is of course possible to use other distance measures while noting that the time spent during the recognition phase is an important factor in a real-time system where video streams are delivered with no break. It is important to note that the power spectra corresponding to most human actions contain large magnitudes at low frequencies that mainly correspond to slow parts of the body. For higher frequencies (over 90% of the spectrum), the power spectra contain highly distinct features but with smaller magnitudes corresponding to body parts involved in fast motions and actions. Hence, we weight the Euclidean distance (ED) so that the contributions from these distinguishing features are equally taken into account. The weighted Euclidean distance (WED) between the *test* sample and the training sample r is

$$\Delta_r^{test} = \frac{1}{UVK} \sum_{u=1}^{U} \sum_{v=1}^{V} \sum_{\Omega=1}^{K} \{\mu(\Omega)[P_{u,v}^{r}(\Omega) - P_{u,v}^{test}(\Omega)]^2\} \tag{5}$$

Here we weight each squared distance using a function of the frequency Ω. We use the Gaussian distribution function $\mu(\Omega) = 1 - \exp[-(\Omega/\Omega_0)^2]$ with zero mean and the standard deviation Ω_0. We found that the method is not very sensitive to the value of Ω_0 and approximate values are sufficient. The weight is large at high frequencies where the mean-power values are very small, while it is small at low frequencies where the mean power is strong. This ensures that the important features whose magnitudes are small in nature are magnified. Hence, when comparing a *test* sample with the action sample r from the training data, we take into account both the slow and the fast motion dynamics to achieve reasonable matches. Using the mean-power spectrum corresponding to integrated neighboring presence signals ensures that weighting in favor of high frequency features does not weaken our recognition in the presence of noise. Note that use of Mahalanobis distance did not improve the recognition rates obtained using the weighted Euclidean distance.

3. Experiments

We aim to evaluate our method using several experiments and also by critically comparing our results with those obtained using other methods. In Section 3.1 we make use of the ViHASi data that we have generated in a virtual environment simulating multiple cameras and actors for a large number of action classes [8]. Such virtual action videos make it possible to evaluate the strengths and weaknesses of our method by performing a variety of quantitative experiments under controlled conditions. Our experiments with real-world data are presented in Section 3.2 which provide a direct comparison with two other existing methods.

3.1. Synthetic Data

We used the MotionBuilder software to generate videos of 20 different virtual actions performed by 8 different virtual actors. These actions were captured at 30 fps with resolution 640×480 and using a variety of virtual cameras with up to 40 different viewing directions (shown in Fig. 1). Each action class provides identical action motions and durations for all actors. These motion data correspond to the actions performed previously by human actors using optical or magnetic

motion capture (mocap) hardware so that skeletal motions could be collected with reasonable precision. For each actor, we create 20 action sequences using the mocap data that come with the software. The names and numbers of these action classes are listed in Table 1. However anyone using our approach can use other mocap data, an important facility that we shall utilize in the future. We use some alternative 3D actors in addition to those provided by the software. These actors consist of very detailed 3D structures including skeleton, skin, cloth and face components giving realistic simulations of male and female humans.

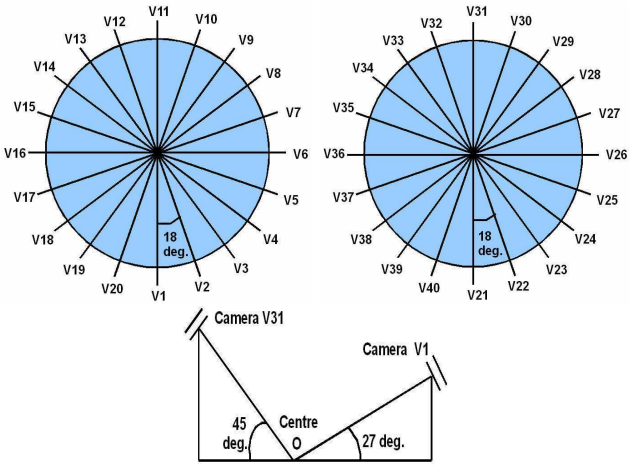

Figure 1. Configurations for 40 cameras: top view of the first set of 20 cameras (left), and the second set of 20 cameras (middle); side view of sample cameras V1 from the first set and V31 from the second set (right) which is located on the opposite side of V1.

As shown in Fig. 1, our 40 cameras consist of two sets of 20 camera views. The cameras are located around two circles in a surround configuration. All cameras are directed towards an identical central point (called O in Fig. 1) which is the projection of the centres of the circles on the floor. All viewing directions in each camera set have an identical slant angle with the horizontal plane (floor). The slant angle for the first set (cameras V1 to V20) is about 27° while for the second set (V21 to V40) the slant angle is 45°. Neighboring cameras in each set have a tilt angle of about 18° with each other. The rotation angle of each camera around its optical centre is zero, so that an actor standing on the central point O appears vertical from all 40 camera views. It is straightforward in this environment to extract the silhouettes corresponding to the bodies of actors at each frame, an advantage of the virtual world that is hardly possible in the real world (Fig. 2). Note that actor A1 is an accurate humanoid used mostly as part of the training data. Other actors provide both male (A2, A4, A6, A7) and female (A3, A5, A8) body shapes with a variety of clothes. We have made the ViHASi data publicly available through Internet [8].

For all experiments using synthetic data we set $U = 8$, $V = 4$, $K = 30$ and $\Omega_0 = 4$, and so the feature vector for each sample has $n_R = 960$ feature elements. Also the number of action classes used is $n_C = 20$ for all cases while the number of training cameras n_D and actors n_A vary in different cases. When experimenting with synthetic data we use the weighted Euclidean distance (WED) given by Eq. (5). A comparison with the standard Euclidean distance is made when experimenting with real-world data. We have tried different combinations of actors and cameras in the training and test data. Here we only present the detailed results for the most challenging combination, i.e. novel test actors and cameras. Due to space limit, we briefly report the recognition rates obtained using other combinations. Using the 'leave one out' cross validation on 1200 video test samples, the average recognition rate obtained was 98.92%. With identical training and test actors, and, novel test cameras we achieved 96.42% recognition on 1200 test samples. Finally, with identical training and test cameras, and, novel test actors, we obtained 98.82% recognition on 1440 test samples.

Table 1. Action class names and corresponding numbers (C) for the 20 virtual motions applied to the 8 virtual actors, and the number of video frames (F).

C	action name	F	C	action name	F
1	HangOnBar	76	11	KnockoutSpin	72
2	JumpGetOnBar	64	12	Knockout	36
3	JumpOverObject	52	13	Granade	76
4	JumpFromObject	44	14	Collapse	36
5	RunPullObject	20	15	StandLook	80
6	RunPushObject	20	16	JumpPunch	52
7	RunTurn90Left	44	17	JumpKick	40
8	RunTurn90Right	44	18	Walk	32
9	HeroSmash	76	19	WalkTurnBack	60
10	HeroDoorSlam	32	20	Run	24

Novel Test Actors and Cameras. Here the actors and camera views included in the training data are different from those included in the test data. The training actors are A1 and A2, while the novel test actors are A3, A4, A5, A6, A7 and A8. The actions performed by A1 and A2 during the training phase are captured from 8 camera views of the first camera-set (V2, V5, V7, V10, V12, V15, V17, V20) and 8 of the second camera-set (V22, V25, V27, V30, V32, V35, V37, V40). This gives us $n_S = 640$ training samples half of which correspond to the first camera set and the other half to the second camera set. The test data

contain 24 novel camera views. The actions performed by half of the test actors (A3, A4, A5) are captured from the first set of 12 camera views (V1, V3, V4, V6, V8, V9, V11, V13, V14, V16, V18, V19) while those performed by the other half of the test actors (A6, A7, A8) are captured from the second set of 12 camera views (V21, V23, V24, V26, V28, V29, V31, V33, V34, V36, V38, V39). The test data then contain 1440 action samples. The average recognition rate obtained is 90.69%, i.e 134 false alarms out of 1440 test samples.

Table 2. Number of misclassifications (out of 72 samples per action class) for each action class.

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
6	1	8	6	0	9	3	0	13	17
C11	C12	C13	C14	C15	C16	C17	C18	C19	C20
6	10	0	12	0	3	4	18	14	4

Table 3. Number of misclassifications (out of 120 samples per camera pair) by camera pairs.

V1	V3	V4	V6	V8	V9	V11	V13	V14	V16	V18	V19
V21	V23	V24	V26	V28	V29	V31	V33	V34	V36	V38	V39
26	23	8	0	4	9	17	8	3	3	5	28

Table 4. Number of misclassifications (out of 240 samples per actor) by test actors.

A3	A4	A5	A6	A7	A8
11	22	26	22	21	32

Detailed results are shown in Tables 2-4. In Table 2, we show the number of misclassifications for each action class. The number given under each class is out of 72 sample variants corresponding to that class (6 actors and 12 cameras). The two classes with lowest recognition rates are C18 and C10 with 18 and 17 misclassifications, respectively. Interestingly, for the test class C18 (WalkTurnBack), 14 out of 18 misclassifications fall in the class C19 (Walk). This perhaps shows the low sensitivity of the method rather than a low recognition rate. A similar discussion applies to the test class C10 (HeroDoorSlam) where 11 out of 17 misclassifications fall in the class C16 (JumpPunch). In both classes C10 and C16 the right hand is used involving fast body motions. There are more cases where similarities between classes result in misclassifications. For example, all the misclassifications for the test class C6 (RunPushObject) fall in the class C5 (RunPullObject). If hands are self-occluded there is little difference between these two actions. Also, 6 out of 8 misclassifications for the test class C3 (JumpOverObject) fall in the class C2 (JumpGetOnBar). Since most of the transformation information is removed, these two actions become quite similar. One might overcome this problem by utilizing the velocity of silhouette centroid.

In Table 3, the number of misclassifications are given against our test camera views. Each number corresponds to the summation of misclassifications by two neighboring cameras from the two camera sets. The cameras V19 and V39 result in 28 misclassifications in total while the cameras V6 and V26 result in no misclassifications. Table 4 shows the number of misclassifications against our test actors. The female actors A8 and A5 result in the highest number of misclassifications, i.e. 32 and 26 respectively. Note here that some male actors, i.e. A4, A6 and A7 result in a comparable number of misclassifications, i.e. 22, 22 and 21 respectively, compared to the female actor A5. This suggests that when novel test cameras are used, the case of novel actors is the less challenging problem.

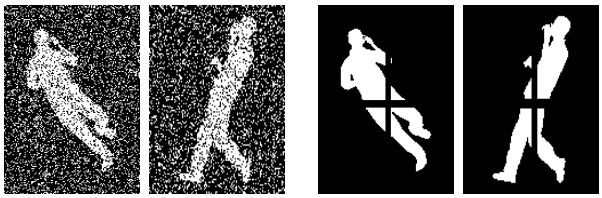

Figure 2. Sample silhouettes of the class C11 captured using camera views V1 and V6: the corresponding corrupted images by adding $\alpha = 40\%$ 'salt and pepper' noise (left) and occlusions, using $\rho_x = 6$ and $\rho_y = 8$ pixels vertical and horizontal bars (right).

Table 5. Average recognition rates using our method by adding α percent 'salt and pepper' random noise to the test data containing 1440 video samples.

$\alpha = 5\%$	10%	15%	20%	50%
89.9%	87.4%	82.0%	73.1%	23.3%

Table 6. Number of misclassifications (out of 120 samples per camera pair) by camera pairs; $\alpha = 20\%$.

V1	V3	V4	V6	V8	V9	V11	V13	V14	V16	V18	V19
V21	V23	V24	V26	V28	V29	V31	V33	V34	V36	V38	V39
38	53	37	13	28	38	41	32	25	11	32	39

Table 7. Number of misclassifications (out of 240 samples per actor) by test actors; $\alpha = 20\%$.

A3	A4	A5	A6	A7	A8
61	66	62	75	55	68

Corrupted Silhouettes. Next, we experiment with silhouettes corrupted either by noise or by partial occlusions. The training and test data used here are identical to those used in the previous experiment. Hence, we can compare the results obtained using clean videos with those obtained using corrupted ones. In Fig. 2, we show sample silhouettes corrupted by noise and partial occlusions (two views) for the class C11.

Isolated Noise. To test the robustness of our method in the presence of noise we follow [9] and add 'salt and pepper' random noise to all test video frames containing actors' silhouettes. Here the density parameter α decides the percentage of the pixels in each image that are changed from '1' to '0' and vice versa. The images for adding $\alpha = 40\%$ noise are shown in Fig. 2.

We list the average recognition rates obtained using several values of α in Table 5. The method is reasonably robust to this type of noise. Further details are given in Tables 6 and 7 when $\alpha = 20\%$ is used. In Table 6, the number of misclassifications are listed against the pair test cameras. Even the camera pair V6 and V26 which give no misclassifications on clean silhouettes result in 13 misclassifications on corrupted ones. It is interesting that noise changes the order of camera pairs that give the highest and lowest recognition rates. For instance here the camera pair V3 and V23 result in the highest number of misclassifications, while it is in third place in Table 3. In Table 7, the number of misclassifications are studied against the actors. Notice that noise has a similar effect here. While the actors A8 and A5 correspond to first and second highest misclassifications in Table 4, here neither A8 nor A5 stands in the first place, but the actor A6 is first.

Table 8. Recognition rates using our method where vertical and horizontal bars (of ρ_x and ρ_y pixels) were added to occlude test samples on all frames.

4, 6	4, 0	0, 6	6, 8	6, 0	0, 8
85.9%	87.9%	90.1%	70.4%	80.6%	89.4%

Table 9. Number of misclassifications (out of 120 per camera pair) by camera pairs; $\rho_x = 6, \rho_y = 8$.

V1	V3	V4	V6	V8	V9	V11	V13	V14	V16	V18	V19
V21	V23	V24	V26	V28	V29	V31	V33	V34	V36	V38	V39
42	39	27	29	40	38	52	35	27	24	31	43

Table 10. Number of misclassifications (out of 240 samples per actor) by test actors; $\rho_x = 6, \rho_y = 8$.

A3	A4	A5	A6	A7	A8
85	57	71	85	45	84

Partial Occlusions. We test the robustness of our method on videos with silhouettes corrupted by partial occlusions. To simulate some measurable partial occlusions, we use horizontal and vertical bars containing '0' pixels to cut the silhouettes in pieces. We refer to the width of the vertical and horizontal bars using ρ_x and ρ_y parameters in pixel units, respectively. The image corresponding to $\rho_x = 6$ and $\rho_y = 8$ is shown in the right-hand side of Fig. 2. We list the recognition rates using several values of ρ_x and ρ_y in Table 8. Experiments with other types of partial occlusions and random object shapes give satisfactory results, including those presented in Section 3.2. Since the method does not involve recovering any kind of shapes from silhouettes, it is naturally robust to both noise and partial occlusions. Again in Tables 9 and 10, where $\rho_x = 6$ and $\rho_y = 8$, we provide the details about the number of misclassifications against the camera pairs and actors, respectively. The way these occlusions affect images is different from the case of corruption by random noise. Hence, the patterns found in Tables 6-7 are different from those in Tables 9-10. Also, the actors A3 and A6 give the highest number of misclassifications in the test data leaving A5 and A8 behind.

3.2. Real-World Data

In this section we use a real-world data-set used by Gorelick et al. [3] which is also available at *www .wisdom.weizmann.ac.il/∼vision/SpaceTimeActions. html*. We refer to their method as the TPAMI07 method and to the Weizmann real-world human action data as the WRWHA data. Recently, Wang and Suter [9] have also used this data-set to evaluate their method that is referred to here as the TIP07 method. The WRWHA data-set contains the foreground masks of $n_S = 90$ video samples corresponding to $n_C = 10$ actions performed by $n_A = 9$ actors providing a variety of male and female body shapes. These actions are: C1) 'bend', C2) 'jack' (jumping jack), C3) 'jump', C4) 'pjump' (jump on the same spot on two legs), C5) 'run', C6) 'side' (galloping sideways), C7) 'skip', C8) 'walk', C9) 'wave1' (wave one hand), and C10) 'wave2' (wave two hands). The video frame size used is 180×144 at the frame rate 25 fps. We show sample original images of actions 'bend' (C1) and 'run' (C5) in the left-hand side of Fig. 3.

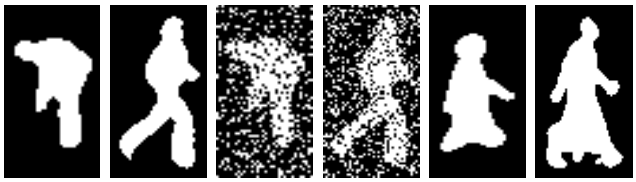

Figure 3. Sample real-world silhouettes of: 'bend' and 'run' actions from WRWHA data (left), corrupted images by adding $\alpha = 40\%$ 'salt and pepper' noise (middle); 'occluded legs' and 'walk a dog' actions from the 'robustness' data used (right).

The parameters of our method used here are $U = 8$, $V = 4$ and $K = 25$, and so the feature vector has $n_R = 800$ feature elements. There is only one training camera $n_D = 1$ and so there are $n_S = 90$ training feature vectors. We used $\Omega_0 = 5$ in Eq. (5) though values in the interval [5,8] give very similar results.

Gorelick et al. [3] have also removed the tricky action 'skip' from the WRWHA data-set leaving $n_S = 81$ training samples. Here, we apply our method to both the WRWHA-90 and WRWHA-81 data-sets. As the software for the TPAMI07 and TIP07 methods are not publicly available, we use their best results reported.

Leave-One-Out. Here we use the 'leave one out' cross validation approach to evaluate the three methods under study. The average recognition rates for the different methods are listed in Table 11. Using the WED classifier (Eq. 5), our method misclassifies 2 out of 81 and 4 out of 90 samples tested. With the WRWHA-81, our method returned 'wave1' for 'walk' and 'side' for 'jump' each in 1 case out of 9. With the WRWHA-90, our method returned 'side' and 'skip' for 2 cases of 'jump' out of 9. Also it returned 'run' for 'skip' and 'wave2' for 'wave1' each in 1 case. From Table 11 it is clear that the weighted Euclidean distance (WED) provides improved results compared to the standard Euclidean distance (ED) which is equivalent to using $\mu(\Omega) = 1$ in Eq. (5). In the rest of the paper we only use the WED in our classifier.

Table 11. Average recognition rates obtained by applying the TIP07 and TPAMI07 methods and our new method to two versions of the WRWHA data.

data-set	TPAMI07	TIP07	new,WED	new,ED
WRWHA-81	99.6%	100%	97.5%	95.1%
WRWHA-90	97.8%	100%	95.6%	91.1%

The results for the TIP07 method are perfect since they [9] have temporally segmented cyclic actions and produced 171 training samples where except 'bend' two complete cycles were segmented out of each of the remaining actions. We do not have these 171 segmented samples available and treat each of the 90 samples as a single action. In the case of the TPAMI07 method, the number of misclassifications is somehow ambiguous, since they have reported their rates based on matching so called 'cubes' rather than matching whole actions. Their cubes are 8 frames long and have 4 frames overlap with each other. To find the performance of their method as an 'action' recognition method, only one class number should be decided for a set of at least 3 consecutive cubes (depending on the actions under study). Our method is practical and it can effectively deal with short and long actions (e.g. 20 to 200 frames long). Note that we do not attempt to estimate building blocks of actions, i.e. body pose or partial actions.

Feature extraction process using the TPAMI07 method for a pre-segmented video of 110×70 frame size including 50 frames took about 30 seconds in Matlab on a Pentium 4, 3.0 GHz [3]. On a similar machine (P4, 3.2 GHz) and in Matlab, when our method was applied to all 90 training samples, with average frame size of 114×77 and in average 61 frames per sample, it took about 172.5 seconds to extract all features. This is about $172.5/90 = 1.9$ seconds per average sample, meaning that our feature extraction method is over $30/1.9 \approx 15$ times faster than the TPAMI07 method. Our classifier (without feature extraction) took 2.3 seconds for all 90 samples, i.e. $2.3/90 = 0.026$ seconds per average sample. Further, for a video sample of 2.44 seconds long (61 frames at 25 fps), our method took about 2.33 seconds for both feature extraction and classification. Hence, in a real-time system, there should be about 0.11 seconds left for other tasks.

Isolated Noise. Next, we turn our attention to the robustness of the methods when 'salt and pepper' noise is added to the test videos. Here, the TPAMI07 method is not included as they have not reported this experiment. We show corrupted images in the middle columns of Fig. 3 when $\alpha = 20\%$ noise is added to the original clean images. We list the corresponding average recognition rates in Table 12. Here the recognition rates for the TIP07 method are approximate as we obtained them from their graphs. They add noise up to $\alpha = 20\%$ and not for $\alpha > \%20$ as their graphs show a dramatic decline in the performance of their method. With our method there is no such decline as our features are highly robust to this type of noise. Here, we also show the results using our method for three values where $\alpha > \%20$. With $\alpha = 50\%$ our method gives comparable recognition rate (45%) to that given by the TIP07 method with $\alpha = 20\%$.

Table 12. Average recognition rates using the TIP07 method and new method by adding $\alpha\%$ 'salt and pepper' random noise to the test data WRWHA-90.

method	5%	10%	15%	20%	50%
TIP07	74%	68%	48%	45%	-
new	95.6%	93.3%	90.0%	84.4%	44.4%

Path Angular Deviations. Here we test the robustness of our method to camera view changes and silhouette corruptions due to occlusions by objects in the scene. The WRWHA data also include 'robustness' test samples for the 'walk' action containing 10 samples in which silhouettes are corrupted by occlusions and another 10 samples with different angular deviations from the direction of 'walk' used in the training data. The angular deviations in samples 1, 2, 3, ..., 10 are 0, 9°, 18°, ..., 81°, respectively. This could also show the robustness of the methods against camera-view changes. We apply our method to these 20 samples and list the results in Tables 13 and 14. According to Table 13, our method classifies correctly the first 6 samples with angular deviations up to 45°. For very

large deviations of 72° and 81°, our method classifies them as 'side' actions as the legs are moving while the actor is facing the camera. The TIP07 method is not included since they have not presented these results.

Table 13. Action classes obtained by applying the TPAMI07 method and our new method to 10 walk samples from the 'robustness' data with 0, 9°, 18°, ..., 81° angular deviations (left-to-right) from the direction of 'walk' action (C8) in the training data.

method	1	2	3	4	5	6	7	8	9	10
TPAMI07	8	8	8	8	8	8	8	8	8	8
new(81)	8	8	8	8	8	8	6	10	2	2
new(90)	8	8	8	8	8	8	6	10	2	2

Table 14. Action classes obtained by applying the TIP07 and TPAMI07 methods and new method to 10 occluded walk (C8) samples of the 'robustness' data.

method	1	2	3	4	5	6	7	8	9	10
TPAMI07	8	8	8	8	8	8	8	8	8	8
TIP07	8	8	8	8	3	8	8	8	8	5
new(81)	2	8	6	2	8	8	8	6	8	8
new(90)	2	8	6	2	8	8	8	6	8	7

Partial Occlusions. The action classes listed in Table 14 correspond to 10 walk samples containing a variety of occlusions (Fig. 3, right). These are: 1) 'swing a bag', 2) 'carry briefcase', 3) 'knees up', 4) 'limping man', 5) 'sleepwalk', 6) 'occluded legs', 7) 'normal walk', 8) 'occluded by a pole', 9) 'walk in skirt' and 10) 'walk a dog'. The TPAMI07 method classifies all actions correctly in the 'walk' class. The TIP07 method classifies 8 actions as 'walk', while it returns 'run' for 'walk a dog' and 'jump' for 'sleepwalk'. Finally, with the WRWHA-81 data-set our method classifies 6 actions as 'walk', while it returns 'jack' for 'swing a bag' and 'limping man', and, 'side' for 'knees up' and 'occluded by a pole'. The results with our method are promising as we do not expect that actions with large differences are classified in the same class. Despite these examples, dealing with large occlusions remains a challenge for all existing methods.

4. Conclusions

In this paper a fast and robust method has been introduced for recognizing human actions from video sequences. We define space-time sub-volumes within each action space-time volume and treat them as integrated discrete signals for which mean-power spectra are simply computed as our robust features. Since the method is fast and its feature extraction is straightforward, it can be used for visual surveillance applications such as detection of crime and anti-social behavior. The domain of possible actions that can be included in our action classes is relatively large, covering actions that take typically 1 to 4 seconds to complete. When applying our method to a large real-world data-set that is also used by two recent complicated methods, we obtain comparable recognition rates. A critical discussion about the methods and their results are provided.

Our approach based on virtual environments can effectively minimize the cost of the training stages. Also, a variety of real-world situations could be simulated for the purpose of evaluating the action recognition algorithm. Hence, it can provide clear ideas towards the improvement of existing action recognition methods based on their strengths and weaknesses observed. Here, our experiments with multiple cameras confirm that we can relatively achieve camera view invariance while preserving robustness to noise and occlusions.

Our future experiments with synthetic data will focus on testing the method on: a) long video sequences with natural transitions between actions, b) virtual actions generated by applying the mocap data that we plan to collect. Finally, we have recently collected a large real-world data-set of human actions using multiple cameras. We shall report the results in due course.

References

[1] A. Bobick and J. Davis, "The Recognition of Human Movement using Temporal Templates," *IEEE Trans. PAMI*, 23(3), 2001: 257-267.

[2] A.A. Efros, A.C. Berg, G. Mori and J. Malik, "Recognizing Action at a Distance," *ICCV*, 2003: 726-733.

[3] L. Gorelick, M. Blank, E. Shechtman, M. Irani and R. Basri, "Actions as Space-Time Shapes," *IEEE Trans. PAMI*, 29(12), 2007: 2247-2253.

[4] I. Laptev and P. Perez, "Retrieving Actions in Movies," *IEEE ICCV*, 2007: 1-8.

[5] F. Lv and R. Nevatia, "Single View Human Action Recognition using Key Pose Matching and Viterbi Path Searching," *IEEE CVPR*, 2007: 1-8.

[6] J.C. Niebles and L. Fei-Fei, "A Hierarchical Model of Shape and Appearance for Human Action Classification," *IEEE CVPR*, 2007: 1-8.

[7] C. Rao, A. Yilmaz and M. Shah, "View-Invariant Representation and Recognition of Actions," *IJCV*, 50(2), 2002: 203-226.

[8] ViHASi: Virtual Human Action Silhouette Data, REASON Project, *http://dipersec.king.ac.uk/VIHASI/*.

[9] L. Wang and D. Suter, "Learning and Matching of Dynamic Shape Manifolds for Human Action Recognition," *IEEE Trans. Im. Proc.*, 16(6), 2007: 1646-1661.

[10] D. Weinland, E. Boyer and R. Ronfard, "Action Recognition from Arbitrary Views using 3D Exemplars," *IEEE ICCV*, 2007: 1-7.

[11] A. Yilmaz and M. Shah, "Actions Sketch: A Novel Action Representation," *IEEE CVPR*, 2005: 984-989.