Human Action Recognition by Extracting Features from Negative Space

Shah Atiqur Rahman[1], M.K.H. Leung[2], and Siu-Yeung Cho[1]

[1] School of Computer Engineering, Nanyang Technological University, Singapore 639798
shah0018@ntu.edu.sg, davidcho@pmail.ntu.edu.sg
[2] FICT, Universiti Tunku Abdul Rahman (Kampar), Malaysia
asmkleung@gmail.com

Abstract. A region based technique is proposed here to recognize human actions where features are extracted from the surrounding regions of a human silhouette termed as negative space. Negative space has the ability to describe poses as good as the positive spaces (i.e. silhouette based methods) with the advantage of describing poses by simple shapes. Moreover, it can be combined with silhouette based methods to make an improved system in terms of accuracy and computational costs. Main contributions in this paper are two folded: proposed a method to isolate and discard long shadows from segmented binary images, and generalize the idea of negative space to work under viewpoint changes. The system consists of hierarchical processing of background segmentation, shadow elimination, speed calculation, region partitioning, shape based feature extraction and sequence matching by Dynamic Time Warping. The recognition accuracy of our system for Weizmann dataset is 100% and for KTH dataset is 95.49% which are comparable with state-of-the-art methods.

Keywords: Human action recognition, Negative space, Silhouette, Dynamic time warping, complex activity, fuzzy function.

1 Introduction

In the field of computer vision, human action recognition is an attractive research topic due to its application area and challenging nature of the problem. Cluttered background, camera motion, occlusion, shadows, viewing angle changes, and geometric and photometric variances are the main challenges of human action recognition. Application of human action recognition includes virtual reality, games, video indexing, teleconferencing, advance user interface, video surveillance etc.

Despite the fact that good results were achieved by traditional action recognition approaches, they still have some limitations [1, 2]. Tracking based methods [3] suffers from self-occlusions, change of appearance, and problems of re-initialization. Methods based on key frames or eigen-shapes of silhouettes [4] do not have motion information which is an important cue for some actions. Local features are extracted in bags-of-words methods [5], lacking temporal co-relation between frames. Optical flow based techniques [6] face difficulties in case of aperture problems, smooth surfaces, and discontinuities.

G. Maino and G.L. Foresti (Eds.): ICIAP 2011, Part II, LNCS 6979, pp. 29–39, 2011.

Since region based methods are relatively robust to noise [2], we proposed a region based approach which extract features from the surrounding regions (negative space) of the silhouette rather than the silhouette itself (positive space) [7]. Negative spaces have the ability to describe poses as good as the positive space with the advantage of natural partitioning of negative space into regions of simple shapes. This approach also performed well in case of partial occlusion and small shadows [7]. However, the system could not show good performance under viewing angle change and long shadows. Here, we extend the idea of negative space to recognize actions in case of viewpoint change by modifying the computation of motion feature and incorporating different viewing angle model data. Moreover, we also propose a method to handle long shadows in segmented images which is one of the major challenges of human action recognition.

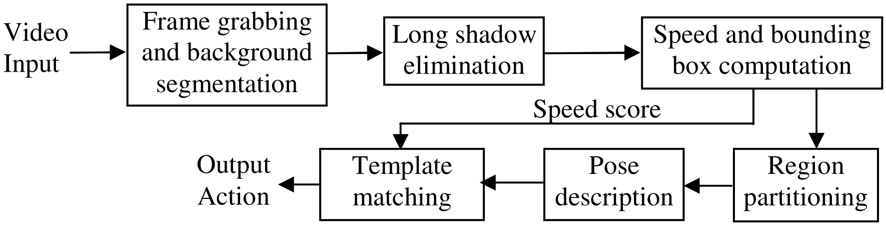


Fig. 1. Block diagram of the system

2 The Proposed System

Our system block diagram is shown in Fig. 1. Input of our system is a video containing single person performing an action. Multi-person activity recognition is left as the future work of current study. The input video is first background segmented which is done by Li et al. [8] algorithm in our system. Long shadows are discarded from segmented images in *long shadow elimination* step. Speeds of the human body are computed and negative spaces are captured in the next step. Complex regions are partitioned into simple regions in *region partitioning* step. Positional and shape based features are extracted in *pose description* step to describe each pose. Finally, pose sequences are matched by Dynamic Time Warping (DTW) and input action is recognized by Nearest Neighbor classifier based on the speed and DTW score.

2.1 Long Shadow Elimination

Some system assumes that the shadow is discarded by the segmentation process [4], otherwise those system performances would be degraded. In our case, during segmentation shadow is not discarded which implies our input segmented image may contain long shadows. Negative space processing has the advantage of recognizing actions effectively in presence of small shadow and partial occlusion [7], but in case of long shadow it may fail to recognize. Long shadows have the characteristics that they are connected to the lower part of the body and are projected on the ground (Fig. 2(a)). Motivating from these characteristics, a histogram based analysis is proposed to isolate shadows from the body and then discard it. Number of bins in the histogram is same as the number of columns and frequency of each bin is the pixel count of foreground pixels along the corresponding column (Fig. 2(b)). Empirically we found that the lower foreground pixels (lower than 10% of silhouette height) of columns (bins) that satisfy equation (1) belong to the shadows.

$$\frac{|c_i|}{\max\limits_{k=1}^{n} |c_k|} \leq 0.20 \tag{1}$$

In (1), $|c_i|$ is the pixel count of foreground pixels of the i^{th} column and n is the number of columns in the input image. Hence, we remove lower pixels of those columns and then take the biggest blob from the image. Eventually shadows are discarded from the input image (Fig. 2 (c)).

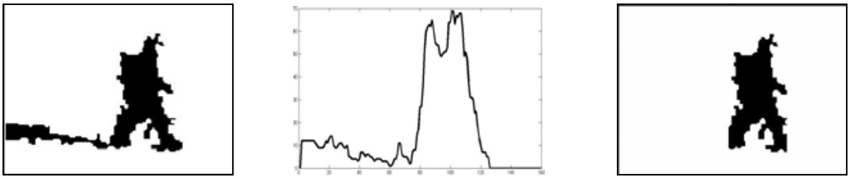

(a) (b) (c)

Fig. 2. Discarding long shadow. (a) Segmented image with shadow. (b) Histogram of foreground pixels of (a). (c) After discarding the shadow.

2.2 Speed and Bounding Box Computation

For human action recognition, the speed of a person is important cues since different actions are performed in different speeds (e.g. running is performed in a faster speed than walking). Speed of a person can be calculated as

$$hor_{sp} = \frac{ds}{dt} = \frac{|x_i - x_{i-1}|}{1/fr\_rate} = |x_i - x_{i-1}| \times fr\_rate \tag{2}$$

where x_i is the X-axis coordinate of the centroid of human body of i^{th} frame and *fr_rate* is the frame rate of the sequence. To remove the scaling effect we normalize equation (2) by the height of the bounding box of the person. Expressing with respect to the average value we have

$$hor_{sp} = \frac{t\_hor_{disp}/(n-1) \times fr\_rate}{t\_height/(n-1)} = \frac{t\_hor_{disp} \times fr\_rate}{t\_height} \tag{3}$$

where $t\_hor_{disp} = \sum_{i=2}^{n} |x_i - x_{i-1}|$, n is the total number of frames in the sequence and *t_height* is the sum of all bounding box height excluding the first frame. Equation (3) can calculate speed effectively where the action is performed in parallel to the image plane but in case of displacement actions (e.g. walk), not performed in parallel with the image plane (i.e. viewpoint is changed), it calculates the observed speed rather than the actual speed of the body (Fig. 3). In Fig. 3 action is performed in a plane which makes an angle (δ) with the image plane. To calculate the actual speed in this scenario, we need to compute the actual displacement of the person which can be calculated by equation (4) (Fig. 3).

$$t\_hor_{disp} = t\_obs_{disp} / \cos\delta \tag{4}$$

Here δ is calculated as [9]

$$\delta = \frac{1}{2}\tan^{-1}\left[\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right] \tag{5}$$

where, μ_{ij} is the i, j^{th} order centralized moment of the centroids (the black dots in Fig. 3) of the human silhouette in the sequence. Hence, the actual speed with viewpoint normalization is

$$ac\_hor_{sp} = \frac{t\_obs_{disp} / \cos\delta \times fr\_rate}{t\_height} = \frac{hor_{sp}}{\cos\delta} \tag{6}$$

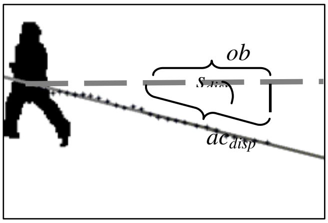


Fig. 3. Viewpoint normalization. Dots are silhouette centroid of each frame, solid line is major principle axis along the centroids, dashed line is the X-axis, δ is the angle between action performing plane and image plane. obs_{disp}, and ac_{disp} are observed and actual displacement respectively.

During speed calculation, we first calculate the speed hor_{sp}, using equation (3). If the human body has significant movement (e.g. walk, run), we apply equation (6) to normalize the speed with respect to viewpoint change and compute the actual speed ac_hor_{sp}. Otherwise (e.g. waving, clapping) ac_hor_{sp} is same as hor_{sp} since for non-moving actions viewpoint change does not affect the speed calculation.

Next an upright bounding box is cut containing the human body to capture negative space regions. Human can perform action in both directions (i.e. moving to the left or right of the image frame). To make the computation easier, we alter all moving action sequences (e.g. walk, run) into one direction (i.e. move to the left of the image frame) by flipping pixels of all the bounding boxes moving left to right about the Y-axis. A movement is seen as from left to right, if the X-coordinate of human body centroid increases over time. For non-moving actions, if the limbs movement is asymmetric (e.g. box), we employed two sets of training data for a single sequence: the flipped and non-flipped images of the sequence, whereas for symmetric movement actions (e.g. handclapping) we employed the training data as it is.

2.3 Region Partitioning

Same pose of same person but captured at different time may not share same number of regions due to the continuous movement of the body as shown in Fig. 4, where pose 4(a) and 4(b) are taken from same pose group but they do not share same number of regions. To overcome the situation and simplify the matching process, region partitioning is applied. We employed the same region partitioning technique as in [7] where for each region, peninsula growing from the silhouette is identified by line growing process. If the peninsula is valid for partition, which is identified by protrusive measure of three distances (Fig. 4(c)), the region is partitioned into two by the tip of the peninsula (Fig. 4(d), 4(e)).

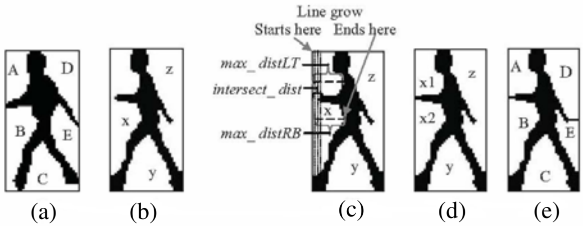


Fig. 4. Region partitioning scenarios. (a) No partition is needed, (b) partition is desired (c) partitioning measures taken for region 'x' of (b), (d) partition output of region 'x'. (e) Final partition output of (b).

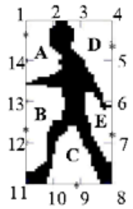


Fig. 5. Positional feature. Numbers represent the anchoring points and letters represent the region, '*' represents the mid-point for each region.

2.4 Pose Description and Matching

Two types of features are extracted to describe the poses: positional feature and shape based features which describe the location and the shape of each negative space region respectively.

2.4.1 Positional Feature

To define the location of a region, we label the bounding box with 14 anchoring points (Fig. 5). For each region, mid-point on the side of the bounding box is computed ('*' in Fig. 5) and the region is assigned a positional label with respect to the nearest anchoring point from that mid-point. For example, anchoring points for regions A, B, C, D and E are points 1, 12, 9, 5 and 7 respectively.

2.4.2 Region Based Features

We extracted simple region based features to describe triangle or quadrangle since negative space regions can be approximated by those shapes [7]. Our shape based features are area, orientation, eccentricity, rectangularity, horizontal and vertical side lengths of bounding box included in a region.

2.4.3 Distance between Poses

Matching of regions for two similar poses raises the need to shift anchor point, e.g. region ‘D’ of Fig. 5 can be assigned to anchor point 6 or 4 instead of 5. In the matching process, we allow the mid point (‘*’) to move at most once to its left or right neighboring anchor point. Hence, we need to develop a distance metric which may allow the region to shift at most one position without any penalty and calculate the distance between poses effectively. This could be done by similar technique used in [7], where a matrix PM is constructed as equation (7)

$$PM\ (i,j)=\begin{cases} r_{dist}\ (v1(i), v2(j)) & \text{if } |i-j|\le 1 \text{ or } |i-j|=13 \\ \infty & \text{otherwise} \end{cases} \tag{7}$$

where *i, j=1 to14*, *v1* and *v2* are two pose vectors. r_{dist} is defined as

$$r_{dist}(v1(i),v2(j))=\sqrt{\sum_{k=1}^{6}\begin{cases} (v1(i)_k - v2(j)_k)^2 & k\neq 1 \\ (|v1(i)_k - v2(j)_k|/0.5)^2 & k=1\ \&\ |v1(i)_k - v2(j)_k|\le 0.5 \\ ((1-|v1(i)_k - v2(j)_k|)/0.5)^2 & \text{Otherwise} \end{cases}} \tag{8}$$

where *k* is the index variable of 6 features for each region with orientation being $v1(i)_1$ or $v2(j)_1$. The maximum orientation difference is π/2.

Then algorithm 2.4.1 is applied to calculate the distance between two poses.

```
Algorithm 2.4.1. Function f_dist(PM)
Begin
  pose_d=0;
  m_ele=MIN(PM);
  while m_ele≠INF
      [r c]=POSITION(m_ele,PM);
      pose_d=pose_d+m_ele;
      assign INF to all elements of row r in PM
      assign INF to all elements of column c in PM
      if r≠c  //anchor point is shifted
       pose_d=pose_d+PM(c,r);
         assign INF to all elements of row c in PM
         assign INF to all elements of column r in PM
      end
      m_ele=MIN(PM);
  end
  return pose_d
end
```

when *INF=∞*, *MIN(X)* returns the minimum element in matrix *X* and *POSITION(j,X)* returns the location of *j* inside matrix *X* in terms of row and column. If there are multiple *j*, return the location of *j* with lowest row and column values.

2.5 Distance between Sequences

Since the temporal duration of even same type of actions can be different, time warping should be applied to determine the distance between two sequences. We employed DTW algorithm for this purpose. DTW finds a minimized mapping path in terms of pose distance according to a recurrence relation (equation (9)) with respect to some constraints (e.g. relaxed end point, slope constraints) which are same as [7].

$$D(i,j) = d(i,j) + \min\begin{cases} D(i-1,j) + w_v(i,j) \\ D(i-1,j-1) \\ D(i,j-1) + w_h(i,j) \end{cases} \tag{9}$$

$$\textit{Initialization: } D(1,t)=d(1,t),\ \ D(i,1)=\infty$$

where D is the DTW matrix, $i=2$ *to* n, $j=2$ *to* m, $t=1$ *to* m, $d(i,j)$ being the distance between poses calculated as in previous section, m and n are number of poses in input and model sequences respectively, w_h and w_v are the slope constraints: $w_{h(v)=}cons\_m_{h(v)}$ if $cons\_m_{h(v)}>2$ otherwise 0 ($cons\_m_{h(v)}$ is consecutive move of warping path in horizontal (vertical) direction).

2.5.1 Doubly Matching

We perform doubly matching scheme (Fig. 6) where two matching is done consecutively since, certain action type, e.g. bending, can contain two parts with one part having perfect match with another short action type, e.g. place jumping. The 1st match is determined by DTW matching from model to test sequence which can occur in any position of test sequence. Let k number of test frames are matched. The 2nd match is determined by performing DTW matching on the subsequent *1.1k* frames right after the first match if later part is larger than the former part (Fig. 6) otherwise matching is done on the preceding *1.1k* frames. For the speed variation within a sequence, we allowed *1.1k* frames for the 2nd match. Average of the two matching scores is taken as the distance between the model and test sequence. For recognition, DTW matching score is computed between test sequence and all the model sequences. Next, speed score is added to the individual matching score and then Nearest Neighbor classifier is employed to recognize the action.

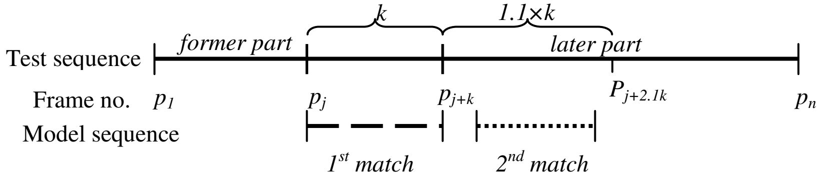


Fig. 6. Matching process of model sequence with test sequence

3 Experimental Results

The proposed system is evaluated by two publicly available datasets: Weizmann human action dataset [10] and KTH action dataset [11].

Weizmann dataset: In this dataset there are 9 persons performing 10 actions. Background segmented images are provided by the author of the dataset and we employed those in our system. Since, the number of video sequence is low, we divide each training sequence into sub-sequences containing only one cycle of an action and we used these sub-sequences as our model sequence. We employed leave-one-out (LOO) testing scheme as most of the other systems used this scheme.

KTH dataset: This dataset is more challenging than Weizmann dataset due to considerable amount of camera movement, long shadow, different clothing and scale and viewpoint variations. There are 25 persons performing 6 actions in 4 different scenarios. Each video is sub-divided into 4 sequences and there are 2391 sequences in the dataset. To extract the silhouette we employ the algorithm by Li et al. [8] without discarding the shadows. Some methods first cut a bounding box either manually [12] or empirically [13] and then perform background segmentation on the bounding box image to reduce noise but in our system we directly feed the input sequence to the segmentation algorithm which means we have less dependency on segmentation process. Since the number of video sequences is high enough, we took only one cycle of an action from each video to avoid unnecessary calculation. Previous systems treat this dataset either as a single dataset (all scenarios in one) or as four different datasets (individual scenarios are treated as a separate dataset, trained and tested separately). Our system is evaluated on both this settings. As a testing scheme some system used Leave-one-person-out (LOO) scheme. Others used split based methods where the

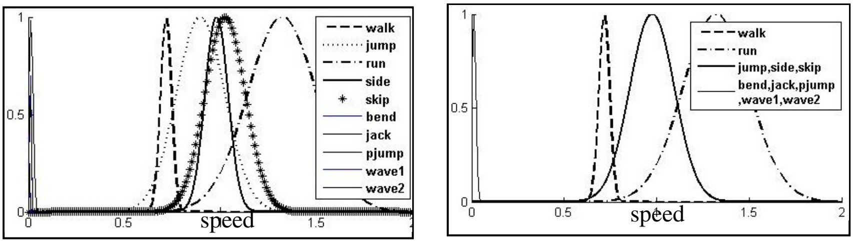


Fig. 7. Membership functions for Weizmann dataset. (a) All 10 membership functions, (b) the final membership functions after merging.

Table 1. Accuracy of different methods for Weizmann dataset

Method	Without skip (%)	With skip (%)
Our Method	**100**	**100**
Fathi et al. [14]	100	100
Ikizler et al. [13]	100	N/A
Kellokumpo et al. [12]	N/A	98.9
Gorelick et al. [10]	N/A	97.8
Lucena et al. [6]	N/A	98.9

Table 2. Accuracy of different methods for KTH dataset

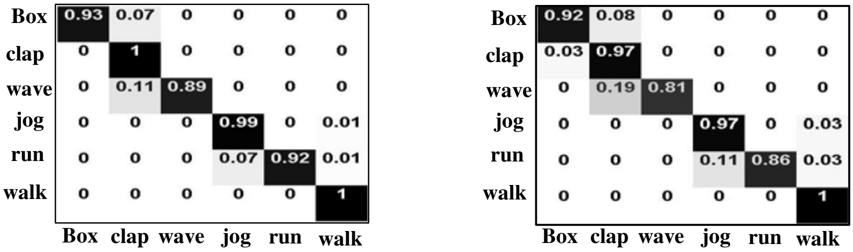
Method	LOO/ Split	Recognition rate (%)	
		Average of all scenes	All scenes in one
Our method	**LOO**	**94.00**	**95.49**
Our method	**Split**	**91.67**	**92.13**
Lin et al. [15]	LOO	95.77	93.43
Kellokumpo et al. [12]	LOO	N/A	95.4
Schindler et al. [16]	Split	90.72	92.7
Fathi et al. [14]	Split	N/A	90.5
Ikizler et al. [13]	Split	N/A	89.4

video sequences are split into training and testing data as [11]. We report results for both testing scheme.

To calculate the speed score, fuzzy membership functions are generated for each type of actions from the training data. Gaussian membership function is chosen as it requires only two parameters (mean and standard deviation) which can be calculated from the training data of each action type. To avoid noisy data, we truncate 5% of data from both highest and lowest boundary, in terms of speed, from each action type. If one membership function is overlapped with another membership function more than 60% of its area, both functions are replaced by a new membership function which is constructed by employing the data of overlapped action types. This merging process goes on until no new membership function is generated. One example is shown in Fig. 7 for Weizmann dataset. Fuzzy functions for KTH dataset is generated by same technique.

Let an input sequence speed is sp_{input} and it is matched with a model of action type 'walk', then the speed score S_C of the input sequence is likelihood score of 'walk' membership function for horizontal speed sp_{input}. Then DTW score is added with speed score by subtracting S_C from 1 since S_C is similarity measure and DTW score is a dissimilarity measure.

Comparison of our system with others for Weizmann dataset is shown in Table 1. Some author evaluated their system with a subset of this dataset (discarding 'skip' action type) which is indicated in Table 1. Our system achieved 100% accuracy on this dataset. Fathi et al. [14] method also achieved perfect accuracy on full dataset but their system requires some parameters to be manually initialized.



	Box	clap	wave	jog	run	walk
Box	0.93	0.07	0	0	0	0
clap	0	1	0	0	0	0
wave	0	0.11	0.89	0	0	0
jog	0	0	0	0.99	0	0.01
run	0	0	0	0.07	0.92	0.01
walk	0	0	0	0	0	1

	Box	clap	wave	jog	run	walk
Box	0.92	0.08	0	0	0	0
clap	0.03	0.97	0	0	0	0
wave	0	0.19	0.81	0	0	0
jog	0	0	0	0.97	0	0.03
run	0	0	0	0.11	0.86	0.03
walk	0	0	0	0	0	1

Fig. 8. Confusion matrix of our system for KTH dataset. (a) leave-one-out scheme (b) split based scheme. For both cases all scenes are taken as a single dataset.

Comparison of our method for different settings with other methods for KTH dataset is shown in Table 2. Our accuracy for this dataset is 95.49% for LOO and 92.13% for split testing scheme which indicates that the amount of training data does not affect too much in our system. Our system performance is better than others for 'all scenes in one'. In case of 'average of all scenes' our system accuracy outperforms most of the other systems. Lin et al. [15] method achieved highest accuracy in this case which is comparable to ours. Lin et al. method is a prototype (key frame) based method which described poses by combination of shape descriptor (describe shape of the silhouette) and motion descriptor (describe motion of different limbs). They need 256-dimensional shape descriptor to describe the shape of pose whereas our pose descriptor is 84-dimensional. Moreover, their system did not employ global motion (speed) of the body to distinguish actions which could significantly improve the accuracy. Hence, by combining Lin et al. [15] method (positive space) with our method (negative space) including speed feature, a new improved system could be obtained in terms of accuracy and computational cost as well. Confusion matrices for both testing scheme is shown in Fig. 8 for 'all-scene-in-one' setting. Most of the misclassification occurs due to the noisy segmentation of background which indicates that if the segmentation process was controlled like other methods [12], our accuracy could be further improved.

4 Conclusion

An action recognition system is proposed here which works on negative space. Our earlier work shows that negative space methods are robust to noise, partial occlusion, small shadows, clothing variations etc [7]. Here, we extend the negative space idea to work under viewing angle change. Moreover, a technique is proposed to discard long shadows from segmented binary images. Additionally, a method is proposed to generate fuzzy membership functions automatically from the training data to calculate speed effectively. Our System accuracy is comparable with the state-of-the-arts methods as shown in experimental results. Further, it is theoretically stated that negative space methods could be combined with positive space methods to obtain an improved system. However, our long shadow detection algorithm may fail in case of multiple shadows in one side of the body (e.g. players' shadows in a football match). Also, our system may not able to recognize actions where the action performing plane and image plane are nearly perpendicular to each other. Nevertheless, these could be overcome when it is combined with positive space based pose description.

References

1. Poppe, R.: A Survey on Vision-based Human Action Recognition. Image and Vision Computing 28(6), 976–990 (2010)
2. Wang, L., Hu, W., Tan, T.: Recent Developments in Human Motion Analysis. Pattern Recognition 36(3), 585–601 (2003)
3. Ikizler, N., Forsyth, D.A.: Searching for Complex Human Activities with No Visual Examples. IJCV 80(3), 337–357 (2008)
4. Diaf, A., Ksantini, R., Boufama, B., Benlamri, R.: A novel human motion recognition method based on eigenspace. In: Campilho, A., Kamel, M. (eds.) ICIAR 2010. LNCS, vol. 6111, pp. 167–175. Springer, Heidelberg (2010)
5. Wang, X., Ma, X., Grimson, W.E.: Unsupervised Activity Perception in Crowded and Complicated Scenes Using Hierarchical Bayesian Models. PAMI 31(3), 539–555 (2009)
6. Lucena, M., de la Blanca, N.P., Fuertes, J.M., Marín-Jiménez, M.: Human action recognition using optical flow accumulated local histograms. In: Araujo, H., Mendonça, A.M., Pinho, A.J., Torres, M.I. (eds.) IbPRIA 2009. LNCS, vol. 5524, pp. 32–39. Springer, Heidelberg (2009)
7. Rahman, S.A., Li, L., Leung, M.K.H.: Human Action Recognition by Negative Space Analysis. In: Cyberworlds (CW), pp. 354–359 (2010)

8. Li, L., Huang, W., Gu, I.Y.-H., Qi, T.: Statistical Modeling of Complex Backgrounds for Foreground Object Detection. Image Processing 13(11), 1459–1472 (2004)
9. Prokop, R.J., Reeves, A.P.: A survey of moment-based techniques for unoccluded object representation and recognition. CVGIP 54(5), 438–460 (1992)
10. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as Space-Time Shapes. PAMI 29(12), 2247–2253 (2007)
11. Schuldt, C., Laptev, I., Caputo, B.: Recognizing Human Actions: A Local SVM Approach. In: ICPR, pp. 32–36 (2004)
12. Kellokumpu, V., Zhao, G., Pietikinen, M.: Dynamic Textures for Human Movement Recognition. In: Int. Conference on Image and Video Retrieval, pp. 470–476 (2010)
13. Ikizler, N., Duygulu, P.: Histogram of Oriented Rectangles: A New Pose Descriptor for Human Action Recognition. Image and Vision Computing 27(10), 1515–1526 (2009)
14. Fathi, A., Mori, G.: Action Recognition by Learning Mid-level Motion Features. In: CVPR, pp. 1–8 (2008)
15. Lin, Z., Jiang, Z., Davis, L.S.: Recognizing Actions by Shape-Motion Prototype Trees. In: ICCV, pp. 444–451 (2009)
16. Schindler, K., van Gool, L.: Action Snippets: How Many Frames Does Human Action Recognition Require? In: CVPR, pp. 1–8 (2008)