Published in IET Computer Vision
Received on 15th February 2011
doi: 10.1049/iet-cvi.2011.0185

ISSN 1751-9632

Recognising human actions by analysing negative spaces

S.A. Rahman[1] *S.-Y. Cho*[1] *M.K.H. Leung*[2]

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

Abstract: The authors propose a novel region-based method to recognise human actions. Other region-based approaches work on silhouette of the human body, which is termed as the positive space according to art theory. In contrast, the authors investigate and analyse regions surrounding the human body, termed as the negative space for human action recognition. This concept takes advantage of the naturally formed negative regions that come with simple shape, simplifying the job for action classification. Negative space is less sensitive to segmentation errors, overcoming some limitations of silhouette-based methods such as leaks or holes in the silhouette caused by background segmentation. Inexpensive semantic-level description can be generated from the negative space that supports fast and accurate action recognition. The proposed system has obtained 100% accuracy on the Weizmann human action dataset and the robust sequence dataset. On KTH dataset the system achieved 94.67% accuracy. Furthermore, 95% accuracy can be achieved even when half of the negative space regions are ignored. This makes our work robust with respect to segmentation errors and distinctive from other approaches.

1 Introduction

Human action recognition attempts to detect, track and identify people, and more generally, to interpret human behaviours, from image sequences involving humans [1]. Automatic categorisation of human activity is highly interesting for a variety of applications such as virtual reality, games, virtual studios, character animations, teleconferencing, choreography, advance user interface and video surveillance. However, challenges of common computer vision tasks still remain, which include cluttered background, camera motion, occlusion, viewing angle changes and geometric and photometric variances of objects.

Within the domain of human motion analysis, many methods have been proposed. Good results were achieved but there are still limitaions. Some of the techniques involve computation of optical flow [2], which is difficult to derive because of, for example, aperture problems, smooth surfaces and discontinuities. Feature-tracking methods [3, 4] face difficulties in cases of self-occlusions, change of appearance and problems of re-initialisation. Some approaches are based on key frames or eigen-shapes of foreground silhouettes (e.g. [5, 6]), lacking information about the motion. Techniques based on periodicity analysis (e.g. [7]) are limited to cyclic actions. Radon transform (RT)-based techniques [6, 8, 9] are not scale invariant and computationally expensive which restrict it from real-time applications. In 'bag-of-words' methods [10, 11], the interest points are local in nature, hence longer term temporal correlations are ignored. Methods using space–time intensity volumes [12, 13] require careful segmentation of background and the foreground.

The region-based approaches work reasonably well as described in [1]. Most region-based approaches work with the regions that are part of the human body. We propose a novel approach focusing on regions surrounding the human body. In art study, such regions are termed as negative space, that is, the space between an object and the edges of the canvas, which contains the object [14]. In our study, the canvas is the smallest rectangle that encloses the human body. By means of negative space, simple and natural regions formed between the canvas and human body can be used for pose or action description as shown in Fig. 1. We show that, from the negative space, a stable semantic-level description of human shapes which is less sensitive to foreground segmentation errors can be obtained. This avoids derivation of sophisticated (but possibly non-robust) scheme to obtain consistent region description from positive space that can be afflicted with leaks and holes inside the silhouette because of image segmentation problem. The triangle or quadrangle shapes that can be extracted from the negative space are not only robust to the low-level segmentation errors, they also provide an implicit semantic description, that is, the description has perceptual meaning of human poses and can be used to reconstruct the original human poses. Semantic-level descriptors provide important and robust information of human action [15], and they are less related to the selection of training samples. Shorter version of this study can be found in [16].

The remainder of this study is organised as follows. In Section 2, previous related studies are reviewed. In Section

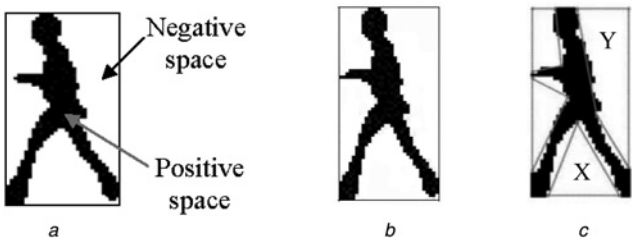


Fig. 1 *Pose description by means of negative space regions*

a Input pose
b After region partitioning process
c Negative space regions represented as triangle or quadrangle

3, we describe our system in details, to compute moving speeds, partition negative space, extract semantic features and perform shape sequence matching. Section 4 shows experimental results of our system and compares system performance with other methods. Section 5 concludes the study.

2 Related works

Existing methods for human action recognition can be broadly categorised into three categories: three-dimensional (3-D) model, two-dimensional (2-D) model and feature-based techniques. In this section, we give a brief review of related studies. Full review can be found in [17–19].

2.1 Feature-based approaches

Feature-based approaches capture the local shape features [20, 21], key points (e.g. SIFT [22]), spatial-temporal features [12, 13], bag-of-words [10, 11] etc. from the human body to recognise actions with classifiers trained by machine learning techniques. In these methods, a set of low-level local features are generated with clustering or learning technique. Then, a strong classifier based on popular machine learning technique is trained for recognition.

'Bag-of-words' approaches originated from text retrieval research are being adapted to action recognition. These studies are mostly based on the idea of forming codebooks of 'spatio-temporal' features. Laptev and Lindeberg [23] first introduced the notion of 'space–time interest points' and use support vector machine (SVM) to recognise actions. Dollar *et al.* [24] extract cubes from poses by using linear filters and then form histogram of these cubes to recognise actions. Niebles *et al.* [10] used pLSA on the extracted cube of [24] for recognition. This method can localise and categorise multiple actions in a single video. Shao *et al.* [25] method extracted interest points by using separable linear filters. They used three tranform-based techniques, discrete Fourier transform, discrete Cosine transform and discrete wavelet transform (DWT), to describe the detected interest points and found superior accuracy for DWT. Wang *et al.* [11] used hierarchical Bayesian models to connect three elements in visual surveillance. Without tracking and human labelling effort, their system was able to complete many challenging visual surveillance tasks including activity recognition. However, in bag-of-words methods, the interest points are local in nature, hence longer-term temporal correlations are ignored.

Silhouette-based (positive space) methods first obtain foreground silhouette by background segmentation and then extract features from the silhouette for recognising actions. Ali and Aggarwal [21] identified the knee and hip by calculating curvature of the silhouette. Then three angles surrounding the hip and knee point were used as features. Their system fails because of large number of points of curvature and false knee and hip detection. Motion history image (MHI) and motion energy image (MEI) are the ways to describe action as temporal template [26] where matching was done by Hu moments [26] or SVM classifier [27]. Liu and Sarkar [28] developed a system where average silhouette was used to represent an action. Later, this average silhouette was termed as gait energy image (GEI) by Han and Bhanu [29], who applied GEI on thermal infrared images. Both of the systems performance degrades because of noisy background, shadow and partial occlusion. Lam *et al.* [30] proposed a method called gait flow image, which was very similar to GEI but achieved better accuracy in gait recognition. Zhang *et al.* [31] proposed an active energy image (AEI), which had the advantage of retaining the dynamic characteristics of gait for recognition. Wang and Suter [20] combined kernel principal component analysis-based feature extraction with factorial conditional random field-based motion modelling. They divided each silhouette image into non-overlapping blocks and the normalised pixel number in each block was used as features. If there are some holes or leaks inside the silhouette because of segmentation error, their system performance may be degraded. Ikizler and Duygulu [32] proposed a hierarchical system which used histograms of oriented rectangles as features. They experimented with three classifiers: nearest neighbour (NN), SVM and dynamic time warping (DTW) and found perfect accuracy for DTW on Weizmann dataset. However, the system accuracy can drop because of imperfect silhouettes and partial occlusion. RT was used as a descriptor for human actions recognition. Boulgouris *et al.* [9] used RT for gait recognition which Singh *et al.* [8] used RT for tracking hands or feet positions. Chen *et al.* [6] selected key postures in an action sequence based on RT. Key postures were then combined to construct an action template for each sequence. RT is not very sensitive to noise, but it is not scale invariant and computational complexity is high.

Some researchers have considered the analysis of human actions by looking at video sequences as space–time intensity volumes. Gorelick *et al.* [12] represented actions as space–time shapes and extracted space–time features for action recognition, such as local space–time saliency, action dynamics, shape structures and orientation. This method cannot handle dynamic backgrounds. Shechtman and Irani [13] proposed a behaviour-based correlation to compute the similarity between space–time volumes that allowed finding similar dynamic behaviours and actions. This method requires significant computation because of the correlation procedure between every patch of the testing sequence and the video database.

Feature-based techniques extract local region of interest (features) from raw pixel data and adaptive with machine learning methods. The extracted features have less power for semantic-level representation. Meanwhile, such methods require large training data for learning and are sensitive to scale variation, lighting changes, cloth variety, complex backgrounds etc.

2.2 Three-dimensional model-based approaches

For accurate description of human poses in a natural way as physical human body structure, 3-D models of the human

doi: 10.1049/iet-cvi.2011.0185

body (e.g. the rigid cylinders connected by joints) have been built and the matching of the specific pose (3-D body part configuration) to the images have been investigated. The 3-D model-based pose representation can provide accurate description of kinematics properties for human action recognition. 3-D approaches can be broadly categorised into two groups based on camera input: single camera system [33–35] and multi-camera system [36–38]. Cheung *et al.* [36] developed a multi-camera-based algorithm which reconstructs 3-D poses by voxel-based reconstruction algorithm. After reconstructing, tracking was done by ellipsoid matching and their system can track motion in real time. Menier *et al.* [37] recognised motion by using 3-D skeletal model. 3-D poses were estimated from medial axes of the visual hull, which were obtained from multiple views foreground silhouettes. In the work of Hofmann and Gavrila [38] estimated upper body pose from three cameras where candidate pose was generated from one camera image and verified by other two camera images. However, this method cannot handle occlusion. Single camera system can be further divided into two types: generative and learning-based methods. In generative methods of [34, 35], 3-D pose is inferred from 2-D poses. In these methods, 2-D poses are estimated by a bottom-up approach, that is, searching different limbs first then construct the whole pose by using kinematics constraints.

One of the major drawbacks of 3-D approaches is the initialisation problem. Learning-based approaches (e.g. [33]) address this problem and initialise automatically. These methods are fast and conceptually appealing. However, learning-based methods may fail because of wide variations of pose and external parameters. 3-D model-based techniques are accurate and invariant under viewing angle changes but the computational cost is high. Multiple cameras may be required for reconstruction of the 3-D body configuration. Hence, it is difficult to apply for human action recognition on video from real world scenes.

2.3 Two-dimensional model-based approaches

The 2-D region-based methods describe the human poses and actions according to the 2-D planar configuration of body parts and limbs. 2-D models can provide a semantic-level description of human poses which is easy to be understood by human beings. The model is represented by stick figure [39], ribbons [40], blobs [41] etc. Guo *et al.* [39] proposed a 2-D stick figure by obtaining the skeleton of the silhouette of the walking human and matching it to a model stick figure. In the study by Leung and Yang [40], the subject's outline was estimated as edge regions represented by 2-D ribbons that were U-shaped edge segments. Wren *et al.* [41] represented model by blobs which used multi-class statistical model of colour and shape to obtain the model. In the later works, tree structure model representation is used for pose estimation [42, 43]. In these models body parts are represented by node and edges of graph where node represent limbs and edges represent kinematics constraints between connected limbs. As the descriptions of body limbs are local, these methods can fail to track different limbs with same image data (e.g. two legs of a person). Part-based tree-structured 2-D model-based human tracking has been proposed [44, 45]. Sigal and Black [44] used a graphical model where each node of the graph represents different limbs and edges represent kinematics constraints between different limbs. Their method could recover 2-D poses even in the presence of occlusion but the method needs manual initialisation. Later Ferrari *et al.* [45] proposed very similar method to [44] method where the initialisation was done automatically. 2-D models are relatively computationally simple, usually work on the positive space and provide semantic-level description of poses. However, these methods based on positive regions are sensitive to the details of human shape variations, noise and segmentation errors.

2.3.1 Negative space approaches: Negative space-based method was originated from art theory and worked on the surrounding regions of silhouette instead of the silhouette itself. Generally, a bounding box is cut containing the human silhouette and features are extracted from the empty regions inside the bounding box. These empty regions are naturally simple in shape that makes feature extraction computationally inexpensive. Moreover, these methods are less sensitive to self-occlusion and blur limbs as no tracking of limbs is required. In addition, they are insensitive to segmentation errors inside the silhouette (e.g. holes or intrusion inside silhouette). Vuuren and Jager [14] have investigated to exploit negative spaces to represent human poses. They grouped empty regions based on the sides of bounding box included in the region and defined 10 region types (facing left, right, top, bottom, top-left, top-right, bottom-left, bottom-right sides of bounding box, facing no side of bounding box and the human silhouette). Area occupied by each region type and bounding box ratio are used as features that did not provide spatial description of the regions, consequently no semantic description of poses. Poses are clustered by using K-means clustering and string matching was performed to recognise actions. They evaluated their system with self-collected datasets that were captured in a controlled environment. Their pose description is inadequate, as only 10 region types are used and no shape descriptions are used to describe all the poses. In contrast, our system preserves the shape and spatial location of each empty region to generate semantic-level description of the poses. Region-based feature description faces difficulties in case of shadow and partial occlusion. Our system describes each empty region individually that are combined later to form global description of poses. If shadow or partial occlusion occurs, it will affect only one or two negative space regions, other remains unaffected. Hence, pose description will be partially altered because of shadow or partial occlusion in our system that makes our system relatively robust to these types of difficulties.

3 Proposed system

The block diagram of the proposed work is illustrated in Fig. 2. Input of this system is a video that captures the human action. Presently, we assume that each video contains only one person doing some activity. Multi-person activity recognition is left as the future work of this study. The video is converted into image frames and the background is segmented from each image. At this moment, our focus is not on background segmentation. There are many popular background segmentation algorithms. Among them background subtraction algorithms are simple and fast. We used one of the background subtraction algorithms [46] in this study. After segmentation, we obtain a binary image that contains only the human body. Speed of the person and direction of the action are then calculated. Meanwhile, a bounding box is cut from the image, containing the whole

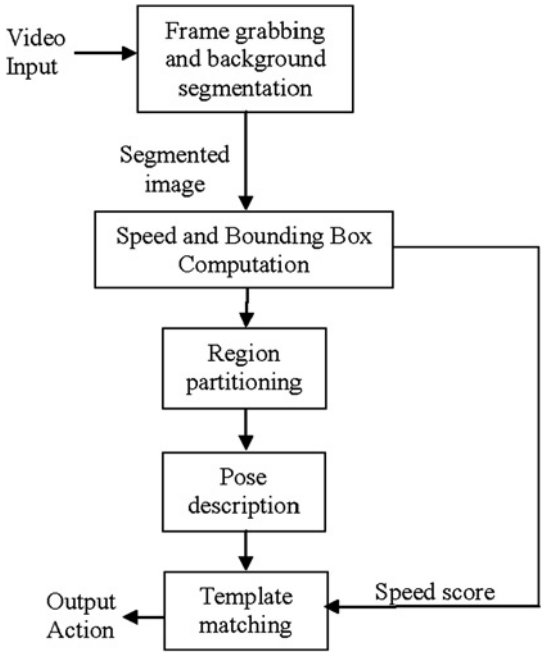


Fig. 2 *Block diagram of the system*

silhouette of the person. Inside the bounding box, attention is focused on empty regions (negative spaces) that are not part of the human silhouette. By means of negative space, simple and natural description can be obtained without further processing after segmentation. For example, the human silhouette (positive space), shown in Fig. 1*a*, cannot be described by simple geometric shapes. However, if the negative space regions are partitioned as shown in Fig. 1*b*, each negative space forms one natural region descriptor that can be described by simple geometric shapes such as triangle or quadrangle as shown in Fig. 1*c*. Consequently, the pose description process becomes simpler. After the region partitioning step, features of the empty regions are extracted. These include the shape and positional features. Finally, we compare test sequences with model sequences by the DTW algorithm to classify an input action.

3.1 Speed and bounding box computation

The human actions shown in Fig. 3 can be grouped into three categories (Fig. 4) according to their motion trajectory. In one

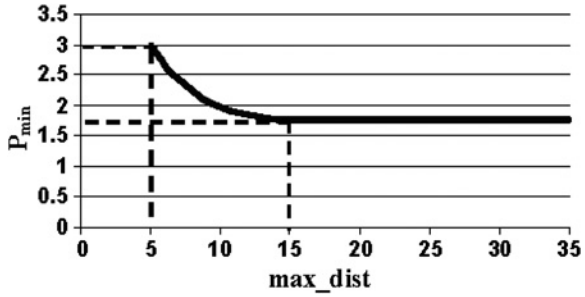


Fig. 4 *Graph for selecting P_{min} to partition a region*

group of actions, the person is stationary (e.g. one-hand-wave, bending-in-front) whereas in other groups of actions the person is moving (e.g. walk, run). Sometime, the upper body part of a person moves up and down (e.g. place-jumping), whereas in other actions it has a steady trajectory (e.g. walk). For grouping the actions, we calculate two types of speeds, that is, the horizontal and vertical speeds. To compute horizontal speed, the absolute value of horizontal (X-axis co-ordinate) displacement of the upper-left corner of the bounding boxes between two consecutive frames is divided by the time difference between these two frames. Let frame rate be the number of frames captured per second. The horizontal speed can be expressed as

$$u_{\text{hor}_i} = s_i \times \text{frame\_rate} \tag{1}$$

where $s_i = |x_i - x_{i-1}|$ and x_i is the upper-left corner's X-axis co-ordinate of the bounding box of the ith frame. To remove the scale effect in the video, we normalise the displacement by the height of the bounding box. Expressing with respect to the average value, we have

$$u_{\text{hor}} = \frac{s_{t\_\text{hor}}/(n-1) \times \text{frame\_rate}}{h_{\text{total}}/(n-1)} \tag{2}$$

where n is the total number of frames in the video, $s_{t\_\text{hor}} = \sum_{i=1}^{n-1} s_i$, and h_{total} is the sum of heights of bounding boxes excluding the first image frame.

The height value is used as the normalisation factor because in most actions, height of the bounding box remains unchanged. Height changing actions, for example, bending-in-front, will not be affected too much as the

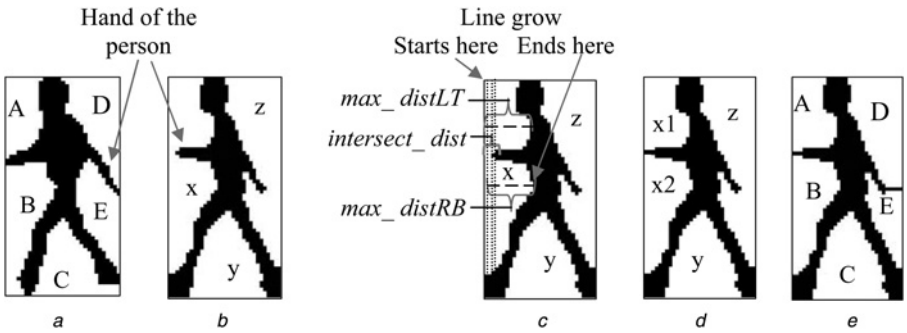


Fig. 3 *Region partitioning scenarios*

a No partition is needed
b Partition is desired
c Partitioning process for vertical side of the region '*x*' of *b*
d Partition output of region '*x*'
e Final output image with new region names of *b*

doi: 10.1049/iet-cvi.2011.0185

average height is used. Simplifying (2), we obtain

$$u_{\text{hor}} = \frac{s_{t\_\text{hor}} \times \text{frame\_rate}}{h_{\text{total}}} \tag{3}$$

Similarly, for vertical speed, we obtain

$$u_{\text{ver}} = \frac{s_{t\_\text{ver}} \times \text{frame\_rate}}{h_{\text{total}}} \tag{4}$$

In next step, each human body is separated from the whole image frame by cutting the smallest up-right bounding box that contains the entire person. By cutting the bounding box, we capture the negative space regions to recognise the input action. Human can perform action in both directions (i.e. left to right or right to left). To simplify the work we consider actions in one direction only from right to left. Hence, if the action direction is from left to right, we flip the pixels in the bounding box about the *Y*-axis. An action is from left to right if the *X*-coordinate of a bounding box corners is increasing over time.

3.2 Region partitioning

Owing to continuous movement, same pose of same person but captured in different time, may not share the same number of empty regions. Figs. 3*a* and *b* show one example of this scenario.

In Fig. 3*a* both hands of the person have touched the bounding box and hence create five negative regions inside the bounding box. However, none of the hands touches the side of the bounding box in Fig. 3*b*. As a result, regions 'A' and 'B' in Fig. 3*a* have been merged in Fig. 3*b*. Similarly, regions 'D' and 'E' of Fig. 3*a* are merged in Fig. 3*b* too. Consequently, when we compare the empty regions from these two images, they are likely to be classified as not similar. To overcome this and simplify the matching process, we propose to partition a region if a peninsula (growing out from the human body) can be found to point to the bounding box. For example, in Fig. 3*b*, if we partition regions '*x*' and '*z*' using the arms as peninsulas as shown in Fig. 3*e*, it would be easier for the system to compute the similarity between Figs. 3*a* and *e*.

The partition process is illustrated using Figs. 3*b*–*e*. Fig. 3*b* is the input image with three empty regions '*x*', '*y*' and '*z*'. Region partitioning is performed in each region separately. For region '*x*', the boundary includes two sides from the bounding box. For each side of bounding box, we search for the top of the nearest peninsula as shown in Fig. 3*c* by scanning along the bounding box boundary in a layer-by-layer (line-by-line) manner. When one of the lines intersects a peninsula (the silhouette), the scanning process stops. To examine validity of the peninsula, a protrusive measure is computed from three distances, max_distLT, intersect_dist and max_distRB, with respect to the bounding box as shown in Fig. 3*c*. Here, max_distLT is the maximum distance from the bounding box side to a silhouette pixel belong to the top half of 'x' with respect to the peninsula (if bounding box side is one of the vertical lines) or to the left half of 'x' (if bounding box side is one of the horizontal lines). max_distRB is defined similarly to a silhouette pixel at bottom or to the right of the peninsula. The protrusive measure is then defined as

$$\text{protrusive} = \text{max\_dist}/\text{intersect\_dist} \tag{5}$$

with max_dist = min(max_distLT, max_distRB). A minimum threshold $P_{\min}$ is set. If protrusive $\geq P_{\min}$, the negative region is divided into two regions. This is done by connecting the peninsula to the nearest pixel to the bounding box. Fig. 3*d* gives one example of the partition where region '*x*' is partitioned into '*x*1' and '*x*2'. The threshold $P_{\min}$ is set dynamically according to Fig. 4. For example, $P_{\min}$ is 3 and 1.75, if the max_dist value is 5 and $\geq$15, respectively. We have performed an appearance-based experiment where the $P_{\min}$ value is set by visually observing the resultant pose. This experiment is performed on 300 images but later in our experiment on full dataset we find that the values set in the graph give optimal action recognition result. Incorrect selection of this value will result in over partitioning or no partition in the region.

After each partition, the same procedure is repeated on the newly formed regions. For regions with two sides from the bounding box, the partition process will be applied on each side separately and partition results will be combined using an AND operation. One example of the final result is shown in Fig. 3*e*.

3.2.1 Fine tuning: It is observed that small regions can form inside the bounding box because of noisy segmentation, loose clothing of the person etc. These small regions do not provide vital information for pose recognition. By discarding these small regions, unnecessary complexity for recognition can be avoided and the system can be made faster. One example of the same pose with/without noise removal is shown in Fig. 5.

A region is seen as small, if its area is $<$1/50 of the total empty area of the bounding box (empty_*bb*_area). The threshold value 50 is set according to observations (Fig. 6) on 2000 images. On the *X*-axis of Fig. 6, we record the normalised size of empty regions as empty_region_area/empty_*bb*_area, on the *Y*-axis, the normalised accumulative

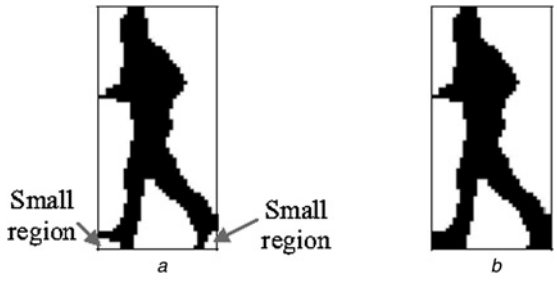


Fig. 5 *Removal of small regions*

a Before removal of small regions
b After removal of small regions

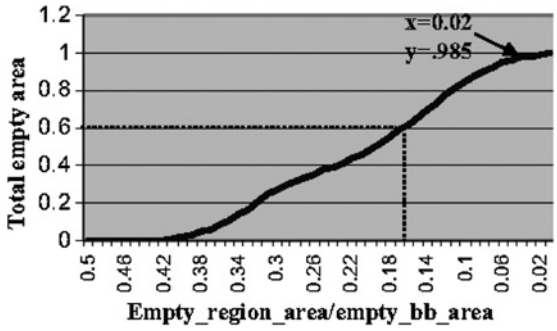


Fig. 6 *Information loss with respect to area of empty regions*

area of the empty regions of sizes $\geq X$. For example, the graph indicates that the total area of empty regions with area ≥ 0.16 of the empty bounding box area, will occupy 60% space of the total empty area. The *Y*-axis value can be seen as the amount of information of the empty regions. From Fig. 6, it can be seen that about 98.5% of pose information is contained in the regions whose empty_region_area/empty_*bb*_area value is >0.02. Thus, if we remove the regions with area $<1/50$ (or 0.02) of the bounding area, we can avoid unnecessary calculation without losing much information.

3.3 Pose description

To compare input poses, discriminative features should be extracted from each empty region to model a pose. We extract two types of features here, the positional and shape information of the regions.

3.3.1 Positional feature: To extract positional information of regions, the bounding box is first labelled with 14 anchoring points such as the anchor points along vertical side divide the side into four equal parts and the anchor points of horizontal side divide it into three equal parts as shown in Fig. 7. Afterwards each empty region is associated with one of these points by first computing the mid-point (*) that bisects the boundary on the bounding box and then finding the nearest anchoring point to the mid-point. For example, in Fig. 7, the anchoring points of regions A, B, C, D and E are points 1, 12, 9, 5 and 7, respectively. The employment of large number of anchoring points on the bounding box can make the region's location more accurate, but it can increase unnecessary computation. In contrast, if we use less number of anchoring points, the location of the region will not be specific enough. Empirically, we found that 14 anchoring points are good enough as we do not expect to find many large empty regions.

3.3.2 Region-based features: There are two types of features available to describe the shape of a region. One is the boundary-based and the other is region-based. It is better to use region-based features, as boundary-based features can be affected more by noise and occlusion than region-based features [47]. In general, descriptors are some set of numbers that are produced to describe a given shape. Good descriptors should have the ability to reconstruct the shape described by the features values. Moreover, the reconstructed shape should be an approximation such that similar shapes can give same approximation and hence similar values of the features

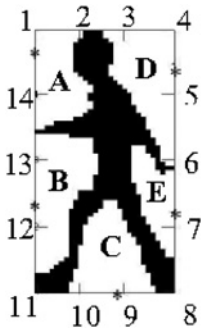


Fig. 7 *Region positioning with numbers representing the anchoring points and letters representing the region*

'*' represents the location of the mid-point on the boundary

or descriptors. After region partitioning, most empty regions are simple in shape. We propose to model each empty region by a triangle or quadrangle as shown in Fig. 1*c* where an empty region is modelled by a triangle if it faces only one side of the bounding box (region 'X' in Fig. 1*c*). A quadrangle is used if it faces two sides otherwise (region 'Y' in Fig. 1*c*). The extracted shape features are expected to facilitate reconstruction of either triangle or quadrangle. To reconstruct a triangle, we need the base and the top point of a triangle. The base is known if we know the length and position of the side on the bounding box. The top point can be estimated approximately by the base length, area and the principal orientation of the triangle. Similarly, a quadrangle can be reconstructed if the lengths of the two sides on the boundary, the area and orientation are known. Furthermore, we employ two additional features to enhance the description. Some of these features can be calculated using statistical moments. Below, all these features are described in detail together with a brief description of statistical moments.

Cartesian moments: The 2-D Cartesian moment, m, of order $p+q$, of a density distribution function, $f(x, y)$, is defined as

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y)\, \mathrm{d}x\, \mathrm{d}y \tag{6}$$

The discrete version of the Cartesian moment for an image consisting of pixels P_{xy}, replacing the integrals with summations, is

$$m_{pq} = \sum_{x=1}^{M} \sum_{y=1}^{N} x^p y^q P_{xy} \tag{7}$$

where M and N are the image dimensions and the monomial product $x^p y^q$ is the basis function. For binary image, m_{00} is the area of the region and the two first order moments, are used to locate the centre of mass (COM) of the object. In terms of moment values, the coordinates of COM are

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \bar{y} = \frac{m_{01}}{m_{00}} \tag{8}$$

Centralised moments: The definition of a discrete centralised moment as described by Hu [48] is

$$\mu_{pq} = \sum_{x=1}^{M} \sum_{y=1}^{N} (x - \bar{x})^p (y - \bar{y})^q P_{xy} \tag{9}$$

where $\bar{x}$ and $\bar{y}$ are the COM of the region. This is essentially a translated Cartesian moment, which means that the centralised moments are invariant under translation.

Area: For binary image, m_{00} denotes the area of a region. We normalise each empty region area by the total empty area (discarding the silhouette area) in the bounding box.

Eccentricity: The simplest eccentricity is the ratio of the major and minor axes of an object [49]. The first and second order moments which define an inertial equivalent approximation of the original image, referred to as the image ellipse [50], can be used to compute lengths of these axes. The image ellipse is a constant intensity elliptical disk with the same mass and second order moments as the original image. If the image ellipse is defined with semi-major axis length α and semi-minor axis length β then α

doi: 10.1049/iet-cvi.2011.0185

and β may be determined from the second order moments using (10) and (11). Consequently, 'eccentricity' of the region can be expressed as β/α.

$$\alpha = \left(\frac{2\left[\mu_{20} + \mu_{02} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}\right]}{\mu_{00}} \right)^{1/2} \quad (10)$$

$$\beta = \left(\frac{2\left[\mu_{20} + \mu_{02} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}\right]}{\mu_{00}} \right)^{1/2} \quad (11)$$

Orientation: The second order moments can be used to calculate the orientation of the shape. In terms of moments, the orientation of the principal axes, ϕ, is given by [48]

$$\phi = \frac{1}{2}\tan^{-1}\left[\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right] \quad (12)$$

In (12), 'ϕ' is the angle of the nearest principal axis (either major or minor) of the shape to the positive X-axis and is in the range $-\pi/4 \leq \phi \leq \pi/4$. To compute the orientation of the shape specifically, we will determine the angle of the major principal axis to the positive X-axis. The angle of major principal axis, denoted by, θ can be determined from the values of μ_{11} ($\mu_{20} - \mu_{02}$) and ϕ [48]. The range of θ is $-\pi/2$ to $\pi/2$. To make θ's range from 0 to 1, $\pi/2$ is added to θ and normalised by π.

Rectangularity: Rectangularity is a measure of how close a shape to a rectangle. A simple rectangularity measure is the ratio of region area over the bounding rectangle area of that region, as shown in (13). This value is 1 for a rectangle and for other shapes it will be in the range of 0–1.

$$\text{rectangularity} = \text{reg\_area}/\text{bounding\_rec\_area} \quad (13)$$

Horizontal and vertical side length: For each region there must be at least one side of the bounding box and at most two sides are included in the region. The length and position of each side are used as two features of the region description. The positional feature has been described in detail in Section 3.3.1. Here, we record the length of each side as the desirable feature. The lengths along the horizontal and vertical sides are recorded as *hor-len* and *ver-len* normalised by the width and height of the bonding box, respectively, to give scale invariant measures. For regions with a corner point, both *hor-len* and *ver-len* will be present. Otherwise one of the lengths will be zero.

In summary, six region-based features are extracted. They are area, eccentricity, orientation, rectangularity, *hor-len* and *ver-len*. All the features are scale invariant with values ranges from 0 to 1. As we have 14 anchoring points and for each region there are six features, the dimension of a feature vector will be $6 \times 14 = 84$. Feature values can be displayed as a grey scale image as shown in Fig. 8 with black colour representing the value of zero. There are 14 blocks [$v(1)$, $v(2)$, . . . , $v(14)$] representing the 14 anchoring points and each block is further divided into six smaller rectangular cells ($v(i)_l$: $l = 1 - 6$) to represent the six feature values. A fully black block means that the corresponding anchoring point is not associated with any negative space region.

3.4 Distances between feature vectors

As the association of an anchoring point can be shifted for two similar poses as shown in Fig. 9 where the anchor point has shifted from 5 to 4 for two similar regions 'A' and 'B', we need to develop an algorithm which can calculate distance between two poses even in presence of anchor point shifting. It is assumed that the shift can at most move the mid-point, '*', to one of its two nearest anchor points. Hence, the matching of the two feature vectors from Figs. 9*a* to *b* should match block 5 in (a) to block 4 in (b) as shown in Fig. 9*c*.

On the basis of Fig. 9, we can construct a matching matrix PM, where one feature vector v_1, is placed along the columns

Fig. 8 *Feature vector for the pose of Fig. 7 with values shown in grey scale*

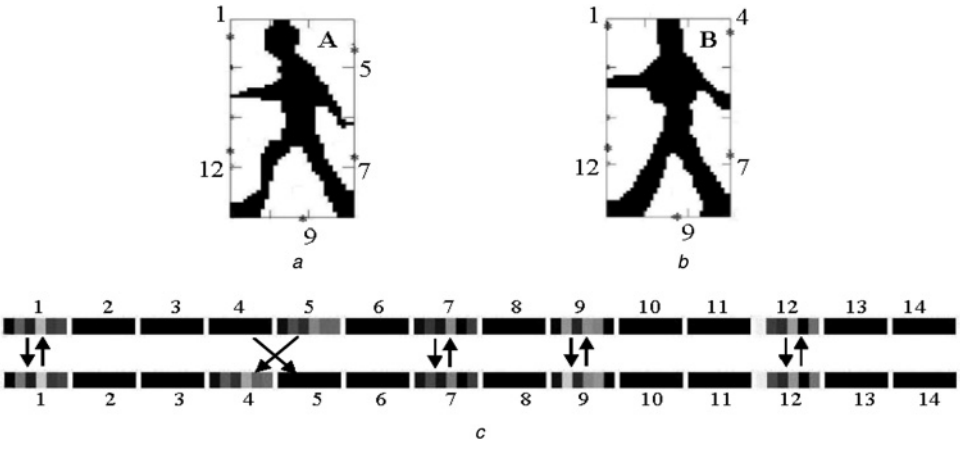
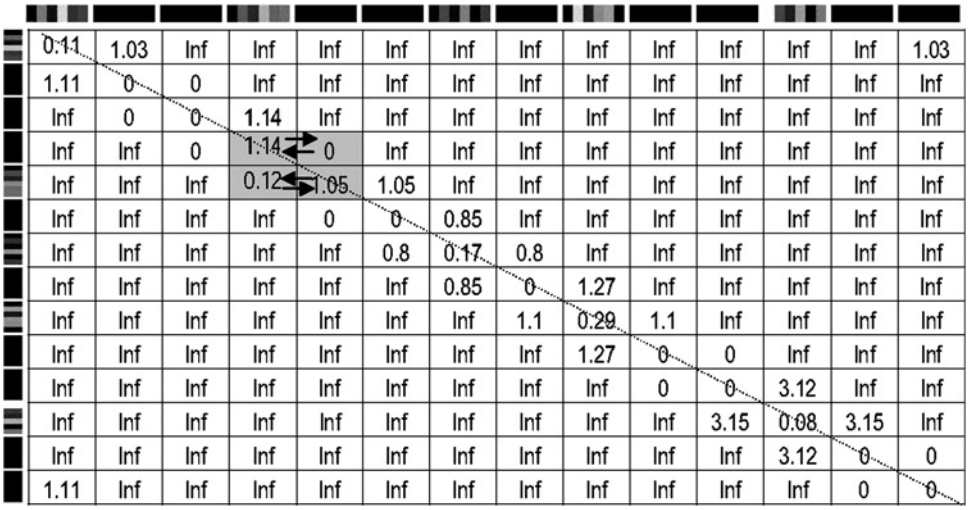


Fig. 9 *Pose matching*

a Pose 1
b Pose 2
c Feature vectors matching from *a* to *b* with one cross matching of regions



0.11	1.03	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1.03
1.11	0	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	0	0	1.14	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	0	1.14	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	0.12	1.05	1.05	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	0	0	0.85	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	0.8	0.17	0.8	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	0.85	0	1.27	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	1.1	0.29	1.1	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1.27	0	0	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	0	3.12	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	3.15	0.08	3.15	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	3.12	0	0
1.11	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	0

Fig. 10 *Matching matrix PM, for the two feature vectors from Fig. 9c (Inf = ∞)*

and another feature vector v_2 is along the rows, with 14 × 14 elements computed as

$$\mathrm{PM}(i,j) = \begin{cases} \text{block\_dist}(v1(i), v2(j)) & |i-j| \le 1 \text{ or } |i-j| = 13 \\ \infty & \text{otherwise} \end{cases} \tag{14}$$

(see (15))
with orientation being $v1(i)_1$ or $v2(j)_1$. The maximum orientation difference is $\pi/2$.

The matching can be done in two steps. At first we match without crossing, that is, $i = j$. Thus, the matching distance is just the sum of the diagonal elements of PM as indicated by the dotted line in Fig. 10. Later, we minimise the distance by allowing anchor point shifting that permits two neighbouring pairs of matrix elements to shift as shown in Fig. 10 (see the shaded cell region). Afterwards the diagonal sum will be lowered. The greedy shifting process continues until no further improvement. In addition, each shifted element is not allowed to obtain shifted again.

3.5 Action recognition

To perform a robust matching of an input action sequence to a model action, the two sequences should be aligned. As the starting frame and even the temporal durations of the same type of actions of a person can be different, we employ DTW for measuring similarity between two sequences. The sequences are 'warped' non-linearly in the time dimension to determine a measure of their similarity regardless of non-linear variations in the time dimension [51].

Let T and R be two sequences of poses of lengths n and m, respectively. The goal of DTW is to find a mapping path (or alignment) $[(p_1, q_1), (p_2, q_2), \ldots, (p_k, q_k)]$, according to a recurrence relation (see (16) below), such that the distance, $\sum_{b=1}^{k} d(T_{p_b}, R_{q_b})$ on this mapping path is minimised, where $d(X, Y)$ is the distance between the X and Y poses calculated as in Section 3.4. This is a dynamic programming problem in nature. In practice, we need to construct a matrix $\boldsymbol{D}$ of dimensions $n \times m$ first and fill in the value of $D(1, j)$ and $D(i, 1)$ according to the boundary constraint as described below. Then by using (17), we fill the whole matrix one element at a time, by following a column-by-column or row-by-row order.

$$D(i,j) = d(i,j) + \min \begin{cases} D(i-1,j) + w_v(i,j) \\ D(i-1,j-1) \\ D(i,j-1) + w_h(i,j) \end{cases} \tag{16}$$

In (16), $w_h(i, j)$ and $w_v(i, j)$ are slope constraints that are described below. In our system, we put test sequence along the horizontal side and model sequence along the vertical side of $\boldsymbol{D}$. For action sequences of training data, we divide each sequence into sub-sequences such that each sub-sequence contain only one cycle of an action. These sub-sequences are then used as model sequences in our system.

DTW finds an optimal match between two given sequences with user defined constraints. In our system, we use the following constraints.

3.5.1 Boundary constraint: Generally, warping path starts from $D(1, 1)$ and ends at $D(n, m)$. As the input and model sequences might not start from the same pose, we use relaxed endpoints constraints, that is, we match the whole model sequence with any part of the test sequence. We initialise $D(i, 1)$ and $D(1, j)$ with the values of ∞ and $d(1, j)$ and, respectively, where $i = 2$ to n and $j = 1$ to m. This will force the warping path to start from $D(1, i_1)$ and ends at $D(n, i_2)$ where $i_1, i_2 = 1$ to m and $i_1 \le i_2$. One example of a warping path is shown in Fig. 11*a*.

3.5.2 Local continuity constraint: Warping path advances along the path as shown in Fig. 11*b*. This local path has fan-in angles of 0°–45°–90°. Using other fan-in path in our system, for example, 27°–45°–63°, may skip some key frames in test sequence that can give rise to inappropriate warping path.

$$\text{block\_dist}(v1(i), v2(j)) = \sqrt{\sum_{l=1}^{6} \begin{cases} (v1(i)_l - v2(j)_l)^2 & l \ne 1 \\ (|v1(i)_l - v2(j)_l|/0.5)^2 & l = 1 \ \textit{and} \ |v1(i)_l - v2(j)_l| \le 0.5 \\ ((1 - |v1(i)_l - v2(j)_l|)/0.5)^2 & \text{otherwise} \end{cases}} \tag{15}$$

doi: 10.1049/iet-cvi.2011.0185

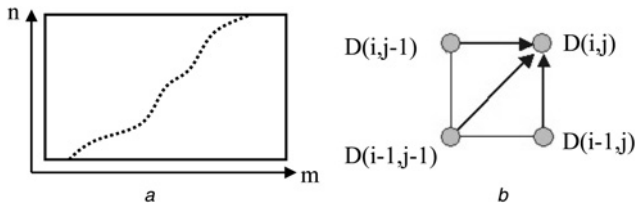


Fig. 11 *Warping path and its local constraints*
a Example of a warping path
b Local constraints

3.5.3 Slope constraint: The ideal warping path should have a slope of 45°. However, same type of action can be performed in different speeds by different persons. Therefore appropriate warping path cannot be strictly 45° in our system. We allow a few vertical and/or horizontal movements of the warping path occasionally. For our system, we empirically find that for same type of action, speed variation of different persons can cause a pose of one sequence to occur at most two frames before or after in another sequence, provided that both sequences are captured at same frame rate. Hence, we add a slope penalty when the path makes more than $l_v(l_h)$ consecutive moves in strictly vertical direction with $l_v = (2 \times \text{ts\_fr\_rate/tr\_fr\_rate})$ [or horizontal direction with $l_h = (2 \times \text{tr\_fr\_rate/ts\_fr\_rate})$] where ts_fr_rate and tr_fr_rate are the frame rates of the test and model sequences, respectively. The penalties are captured by the $w_h(i, j)$ and $w_v(i, j)$ in (16) as

$$w_h(i,j) = \begin{cases} \text{cons\_}m_h(i,j) & \text{if cons\_}m_h(i,j) > l_h \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

$$w_v(i,j) = \begin{cases} \text{cons\_}m_v(i,j) & \text{if cons\_}m_v(i,j) > l_v \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

$$\text{cons\_}m_h(i,j) = \begin{cases} \text{cons\_}m_h(i,j-1)+1 & \text{if } D(i,j) = d(i,j) + D(i,j-1) \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

$$\text{cons\_}m_v(i,j) = \begin{cases} \text{cons\_}m_v(i-1,j)+1 & \text{if } D(i,j) = d(i,j) + D(i-1,j) \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

where cons_m stands for consecutive moves. After calculating the warping distance between test and model sequences, the distance is normalised by the warping path length which is the number of elements of D traversed by the path.

3.6 Doubly matching scheme

When the test sequence contains numerous cycles of the same action, we can perform a robust doubly matching scheme as shown in Fig. 12 where two consecutive matching from the model to the test sequence are performed. This is necessary as certain action type, for example, 'bending-in-front' can be partitioned into two parts with one part being a perfect match with another shorter action type, for example, 'place-jumping'. The average of the doubly matches is taken as the matching score of current model to the test. The first match is found by performning the DTW from current model to the test sequence. The second match is found by performing DTW from current model to the next k frames right after (before) the first match if later (former) part of test sequence is longer than the former (later) part as shown in Fig. 12. To cater for uncertainty (as the speed of a person can vary within an action sequence), we allow an interval of $1.1k$ frames (10% larger than first match) for the second match.

One example of doubly matching scheme is shown in Fig. 13 where the test sequence and model sequence length is 121 and 36 frames, respectively. First DTW match is found from frames 8 to 41. Second DTW matching will be performed between frames 42 and 80 as the later part of test sequence (frames 42–121) is longer than former part (frames 1–7) and the first match length (frames 8–41) is 34 (i.e. $42 + (34 \times 1.1) \simeq 80$). The second match is found from frames 51 to 80. Some example matched frames are shown in Fig. 13.

3.7 Final matching

Given a test sequence T, we calculate speed scores SC_{c_i} as described in Section 4.1 and the warping distance of T from each of the model sequence R_j. The matching distance is computed as

$$\text{matching\_dist}(T, R_j) = \text{DTW}(T, R_j) + (\text{max\_score} - \text{SC}_{c_i}) / \text{max\_score} \quad (21)$$

where $\text{DTW}(T, R_j)$ is the warping distance between T and R_j, C_i is the category for action type of R_j according to speed. Ideally, if T and R_j are from the same action category, SC_{c_i} should give high score (approximately max_score). Afterwards, we apply a NN classifier to recognise the action of the test sequence.

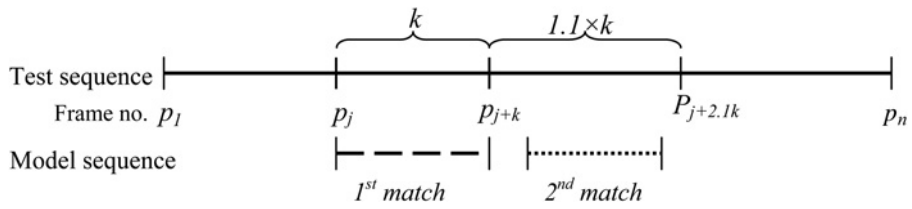


Fig. 12 *Matching process of model sequence with test sequence*

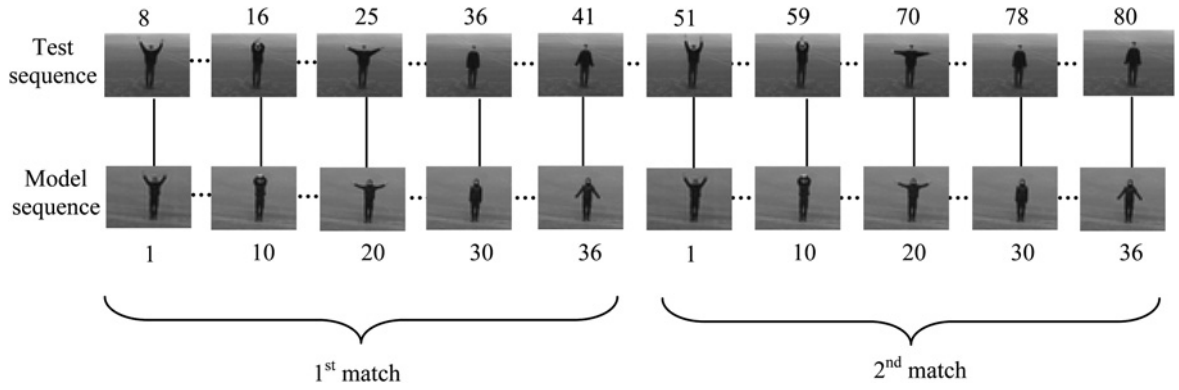


Fig. 13 *Example of doubly matching scheme*

Number represents the frame number within a sequence. First match is from frames 8 to 41 and second match is from frames 51 to 80 of test sequence

4 Experimental results

4.1 Action datasets

In our experiment, we employed the Weizmann human action dataset [52] and KTH [53] dataset to test the efficacy of our system.

4.1.1 Weizmann dataset: In this dataset there are nine persons performing ten actions (90 sequences in total). The actions are 'bending-in-front' (or 'bend'), 'jumping-jack' (or 'jack'), 'jump-forward-on-two-legs' (or 'jump'), 'jump-in-place-on-two-legs' (or 'pjump'), 'run', 'gallop-sideways' (or 'side'), 'jump-forward-on-one-leg' (or 'skip'), 'walk', 'one-hand-wave' (or 'wave1') and 'two-hands-wave' (or 'wave2'). This dataset provides a good testing bed to investigate the performance of the algorithm in presence of relatively large number of action types. We employed Zivkovic and van der Heijden's [46] implementation to subtract the background from the foreground for each video sequence and used a simple threshold to form binary image. As the number of video sequences is low, each training sequence is divided into sub-sequences that contains only one cycle of an action and used these sub-sequences as the model sequences. Leave-one-out (LOO) testing scheme is employed for this dataset as most of the other systems used this scheme.

4.1.2 KTH dataset: This dataset is more challenging than Weizmann dataset because of considerable amount of camera movement, long shadow etc. There are 25 persons performing six actions (box, hand-clap, hand-wave, jog, run and walk). There are four different scenarios in this dataset. We employed only the outdoor scenario (scene 1 as defined by the author of the dataset) and compared our result with other methods for this scenario only. Each video is sub-divided into four sequences, each containing several cycles of one action type. To extract the silhouette we used [46], same as the Weizmann dataset. In addition, LOO testing scheme is employed like the Weizmann dataset. As the number of video sequences is high enough in this dataset, only one cycle of an action from each video sequence is taken as model sequence to avoid unnecessary calculation. Example frames of both datasets are shown in Fig. 14.

For both these datasets, the segmented silhouettes contains 'leaks' and 'holes' because of imperfect subtraction, shadows,

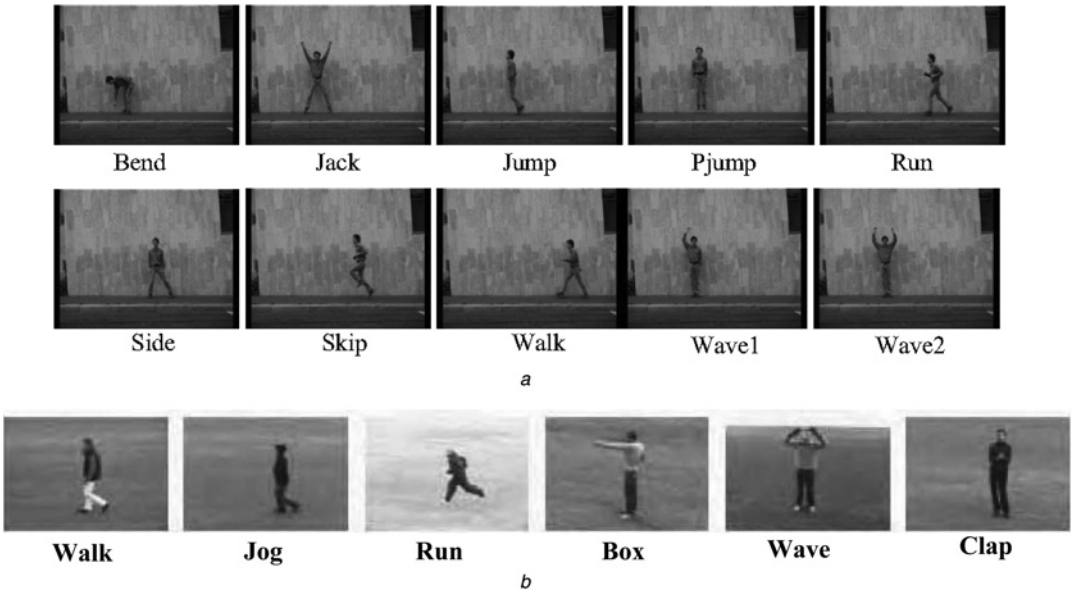


Fig. 14 *Example frames of datasets used in the system*

a Weizmann dataset
b KTH dataset

doi: 10.1049/iet-cvi.2011.0185

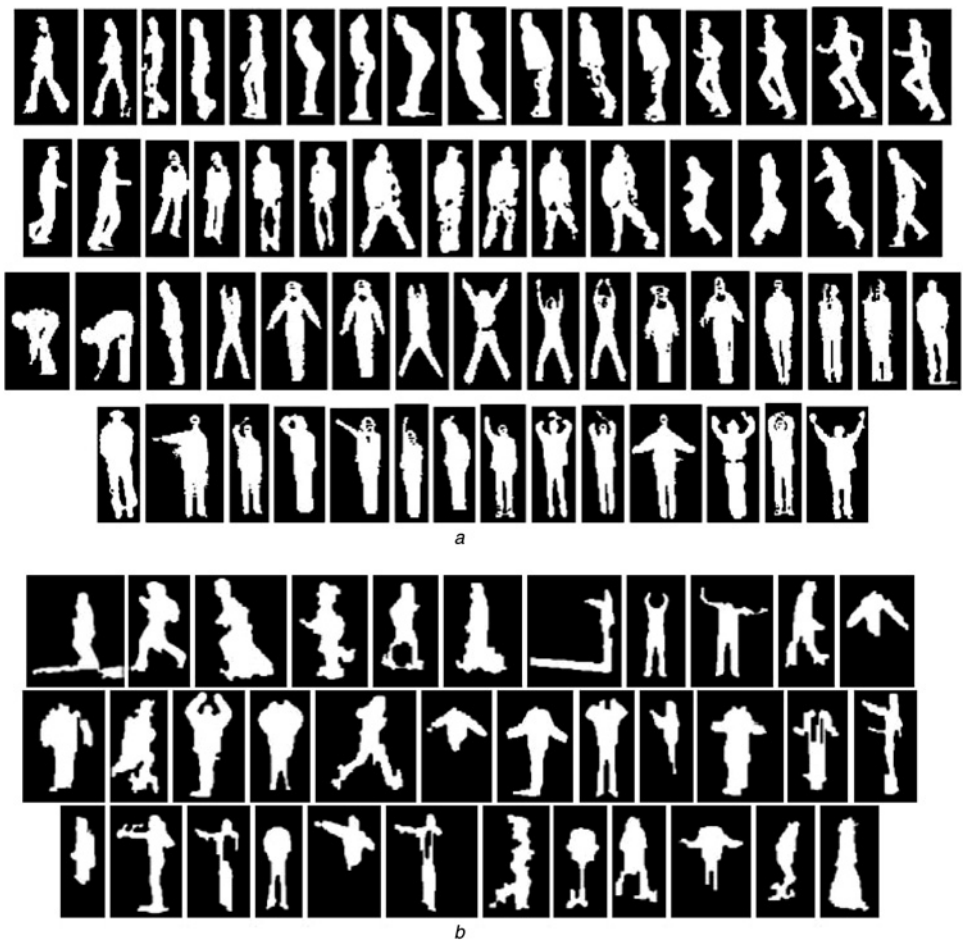


Fig. 15 *Example image frames of segmented images*

a Weizmann dataset
b KTH dataset

and colour similarities with the background but our system can cope with all these complexities well. Example frames of the segmented images, taken from the correctly recognised image sequences are shown in Fig. 15.

To calculate the speed score, fuzzy membership functions are generated empirically. Fuzzy functions are generated for each dataset separately as the types of action are different in two datasets. Table 1 lists the average values of horizontal and vertical speeds of different types of actions for Weizmann dataset. The horizontal speeds of the last five rows of Table 1 should be zero as there is no displacement of the human body. However, as there are some limbs movements inside the bounding box which result in small displacement of upper-left corner of the bounding box, the horizontal speed in-between two frames usually not zero.

According to Table 1 we can plot a graph as shown in Fig. 16, where the actions can be grouped into three categories C_1 (bend, jack, pjump, wave1 and wave2), C_2 (walk) and C_3 (jump, run, side and skip). The categories are grouped according to the horizontal speeds, H_1, H_2, H_3 and vertical speeds V_1, V_2 as shown in Table 2 and Fig. 17. The parameters of the fuzzy functions of Fig. 17 are

Table 1 Average speeds of different types of actions

Action	Average speed	
	Horizontal	Vertical
walk	0.687	0.1193
jump	0.9529	0.5393
run	1.3703	0.3116
side	0.9884	0.3558
skip	1.0395	0.433
bend	0.0069	0.3791
jack	0.0207	0.5775
pjump	0.0123	0.6644
wave1	0.0172	0.2442
wave2	0.019	0.2616

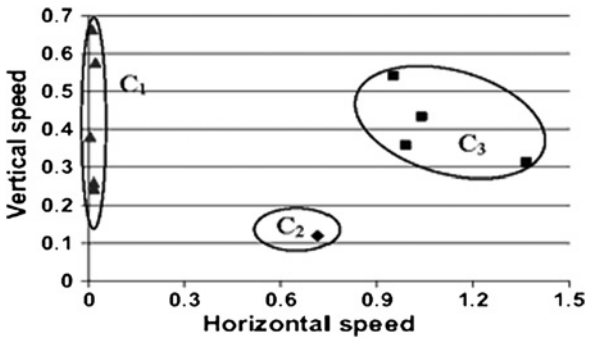


Fig. 16 *Grouping of different type of actions*

Actions in C_1: bend, jack, pjump, wave1 and wave2. C_2: walk. C_3: jump, run, side and skip

Table 2 Action classification based on their speeds

Speed	Sub-group	Action name	Category
horizontal	H_1	bend, jack, pjump, wave1, wave2	C_1
	H_2	walk	C_2
	H_3	jump, run, side, skip	C_3
vertical	V_1	walk	C_2
	V_2	bend, jack, pjump, wave1, wave2, jump, run, side, skip	C_1, C_3

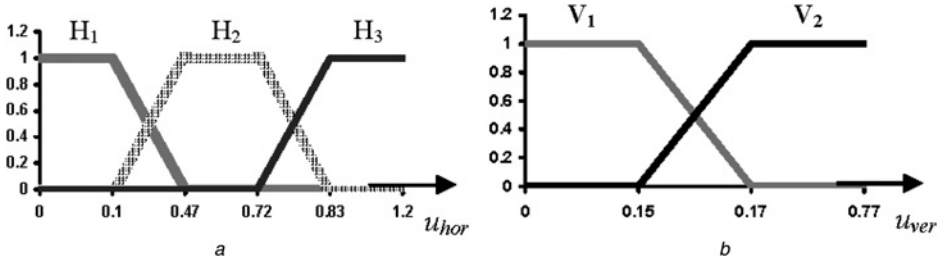


Fig. 17 *Fuzzy membership functions for different speeds*

a Horizontal speed H_i, with $i = 1, 2, 3$
b Vertical speed V_j, with $j = 1, 2$

obtained empirically. According to Table 2, the likelihood scores of C_1, C_2 and C_3 can be computed as, $SC_{C1} = H_1 + V_2$, $SC_{C2} = H_2 + V_1$ and $SC_{C3} = H_3 + V_2$ which are then used in the matching phase.

Fuzzy functions for KTH dataset is also generated by the same method as Weizmann dataset which is shown in Fig. 18*b*. Grouping of actions are shown in Fig. 18*a* from which it is evident that for the actions in KTH dataset vertical speed is not significant. Hence, to avoid unnecessary computation, only horizontal speed is employed for KTH dataset. Likelihood scores of C_1, C_2, C_3 and C_4 are the same as the likelihood scores of h_1, h_2, h_3 and h_4, respectively.

4.2 Action classification

In this experiment, we determine the classification accuracy for the above-mentioned two datasets. Parameters of the system are set based on the training data of Weizmann dataset and applied on both the datasets.

In case of doubly matching, our system has achieved 100 and 94.67% accuracy on Weizmann and KTH dataset, respectively. When we use singly matching (i.e. classification is done based on first match only), the system misclassify 15 action sequences in case of Weizmann dataset (16.67% error rate), where most of the misclassified sequences are from 'bend' action. For KTH dataset, the error rate is 5.33% for singly matching. Comparison of our result with other methods for Weizmann dataset is shown in Table 3. Some of the authors tested their system with another version of the Weizmann dataset by removing the activity 'skip', which makes the classification simpler as 'skip' action has several common poses with 'run' and 'jump' actions. We presented our result for both version of the dataset. Fathi and Mori [54] also achieved 100% accuracy but their method is not robust to non-rigid deformation (e.g. different clothing) and partial occlusion. Ikizler and Duygulu [32] obtained perfect accuracy for nine actions ('skip' was removed) but their method is a hierarchical method with velocity at top level. As a result if their velocity computation is incorrect, their subsequence computation will be wrong which implies incorrect recognition of action.

Fig. 19 shows the average matching distance of the correct and nearest incorrect choices of each action type. Most of the action types have significant differences between these two

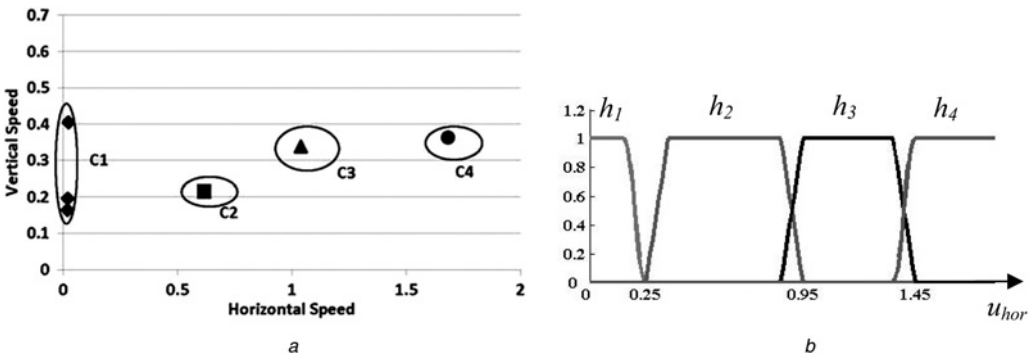


Fig. 18 *Generation of Fuzzy membership functions for KTH dataset*

a Grouping of different types of actions based on the average speed of each action type. Actions in C_1: box, clap and wave, C_2: walk, C_3: jog and C_4: run
b Generated fuzzy membership functions

doi: 10.1049/iet-cvi.2011.0185

Table 3 Comparison of our method with other methods for Weizmann dataset

Method	Without skip	With skip
our method	100%	100%
Fathi and Mori [54]	100%	100%
Ikizler Duygulu [32]	100%	–
Jhuang *et al.* [55]	98.8%	–
Seo and Milanfar [56]	–	97.5%
Wu *et al.* [57]	–	98.9%
Li *et al.* [58]	97.8%	–
Gorelick *et al.* [12]	–	97.8%
Lucena *et al.* [59]	–	98.92
Bregonzio *et al.* [60]	–	96.66%

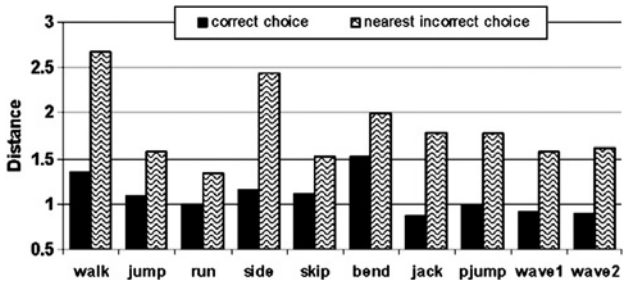


Fig. 19 *Mean distance of correct and nearest incorrect choice of each action*

Table 4 Comparison of our method with other methods for KTH dataset (scene 1)

Methods	Accuracy, %
our method	94.67
Lin *et al.* [61]	98.83
Jhuang *et al.* [55]	96
Schindler and van Gool [62]	93
Ahmad and Lee [63]	90.17

choices except 'jump', 'run' and 'skip' as these three actions have temporal and spatial similarities among themselves.

Comparison of our method with other methods for scene 1 of KTH dataset is shown in Table 4. Here, the system parameters are same as the Weizmann experiment. Our accuracy is comparable with state of the arts methods. The confusion matrix of our system for KTH dataset is shown in Fig. 20*a* and the difference between nearest correct and incorrect choices are shown in Fig. 20*b*.

4.3 Contribution of different features

Contributions of different types of features are shown in Fig. 21 which is generated based on the Weizmann dataset. According to the graph, shape and positional information provide vital information for our system, whereas speed information is needed for only a few action sequences to provide distinguishable information as without speed feature the accuracy is 97.78%.

4.4 Minimum sequence required for action recognition

In this experiment, we analyse the minimum information required by our system to recognise actions on Weizmann dataset. To do this, we use a sliding window technique. For each action, we used a sliding window of frame size f_n with an overlap of $f_n - 2$ frames with the next window. For example, let $f_n = 10$, then for each action sequence frames 1–10 will be used in the first window and frames 3–12 will be used in second window and so on. The range of f_n is $5 \leq f_n \leq 50$ and each time the window size is increased by five frames. For each window size, we compute the recognition rate of each action sequence (90 action sequences in our case). Then we take the average recognition rate for each type of actions. Afterwards the window sizes are converted to cycle lengths for each action type. In Fig. 22*a*, the average recognition rates of each action type are shown for the corresponding cycle lengths.

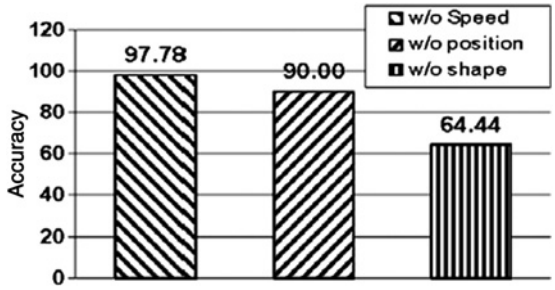


Fig. 21 *Matching results of different features contributions*

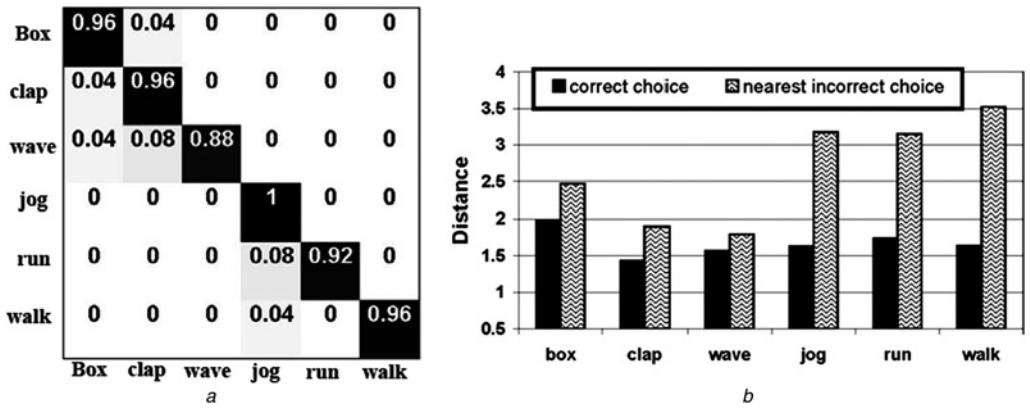


Fig. 20 *Experimental results on KTH dataset*

a Confusion matrix for KTH dataset
b Difference between nearest correct and incorrect choices

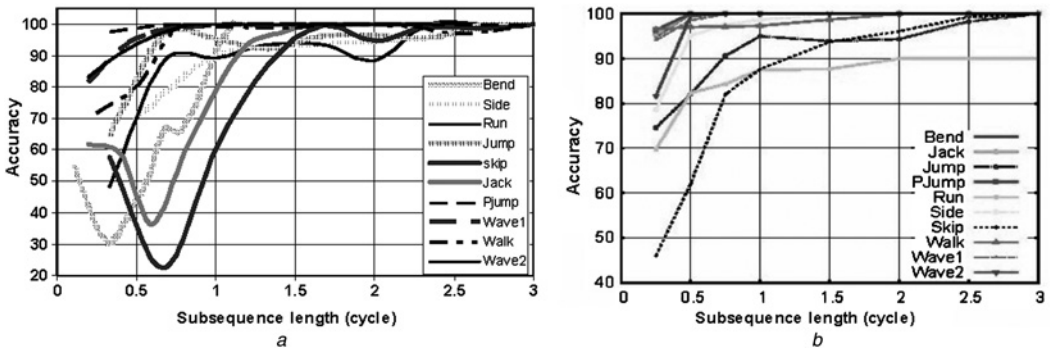


Fig. 22 *Accuracy with respect to sequence length (amount of information)*

a Our method
b Lucena *et al.*'s method [59] (reproduced from [59])

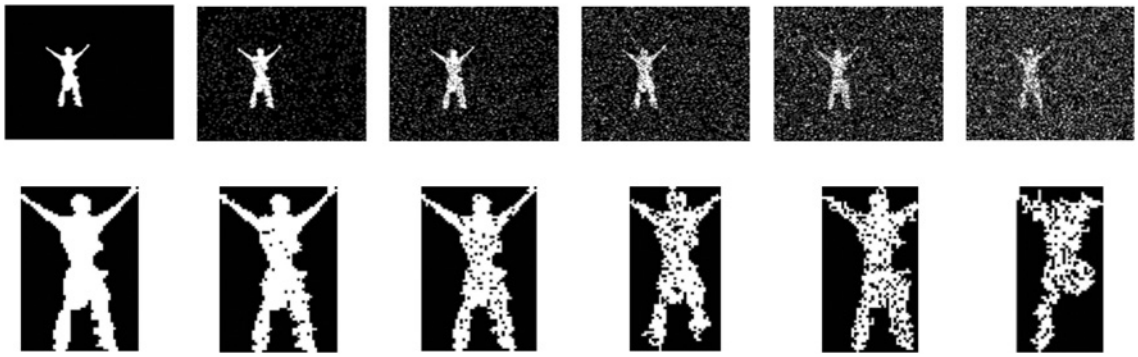


Fig. 23 *Silhouette images with different levels of noise*

Top row is the input image and bottom row is the extracted bounding box image. In both rows, noise levels are 0.0, 0.1, 0.2, 0.3, 0.4 and 0.5 from left to right, respectively

Ideally, in case of doubly matching, all the action sequences should be recognised if two cycles of test sequence data are available but the starting points of all the test sequences are not necessarily aligned with the model sequence. As a result, our system need three cycles of input data for some of the test sequences. We showed the result of Lucena *et al.* [59] in Fig. 22*b*. Our result is found better especially for the 'run' action.

4.5 Feature extraction time

Our descriptors are computationally inexpensive. Feature extraction time of our un-optimized Matlab® code is 4.31 s for a (180 × 144 × 50) segmented sequence on a Pentium 4, 3.4 GHz machine. This is ∼12 frames per second (≃ 50/4.31). The speed can be further improved using C++. Gorelick *et al.* [12] takes about 30 s to extract features for same sequence in similar machine.

4.6 Robustness to noise

In this experiment, the system is evaluated for robustness to noisy segmentation on Weizmann dataset. For this purpose, we explicitly add 'salt and pepper' noise (same as [20]) to the segmented images to simulate noisy segmentation. The amount of added noise is controlled by a parameter 'noise density'. For training purpose, original (uncorrupted) silhouettes are used. During testing, in this experiment, the bounding box is extracted only for the biggest blob from the corrupted image. Inside the bounding box, pixels of biggest blob are treated as foreground pixels and other foreground pixels that are not connected with the biggest blob are removed. Hence, some part of the human body may be discarded in some testing image frames (Fig. 23), which may actually happen in case of noisy segmentation. Some example frames are shown in Fig. 23. Same experiment was performed on Weizmann dataset by Wang and Suter [20]. Comparison of our method with Wang's method for different level of noises is shown in Table 5. It is evident from this experiment that our method is relatively more robust to noisy segmentation.

4.7 Robustness to deformed actions

We also tested robustness of our system with action of high irregularities. For this purpose, we used the Weizmann robust deformation dataset [52]. This dataset contains ten test video

Table 5 Comparison with Wang and Suter [20] method for different level of noise

Noise density	Recognition rate, %	
	Our method	Wang and Suter's [20] method
0.10	100	100
0.20	98.89	100
0.30	95.56	90
0.40	92.22	63
0.50	82.22	27

doi: 10.1049/iet-cvi.2011.0185

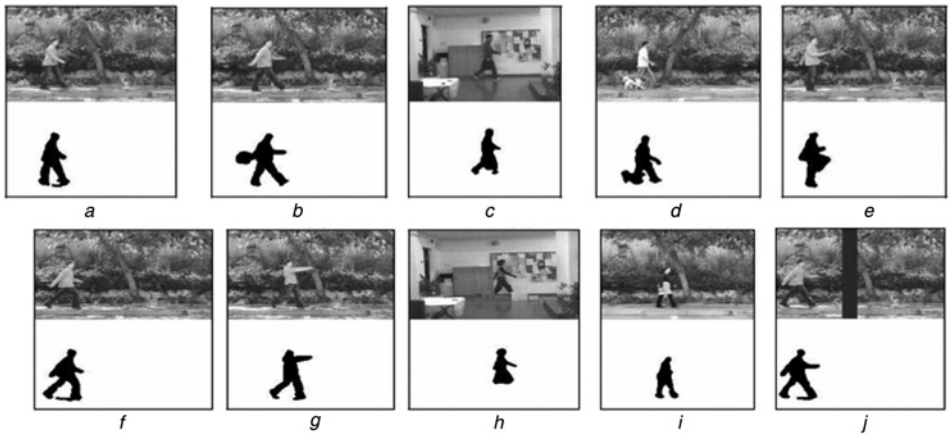


Fig. 24 *Example of image frames used in robustness experiment*

Upper row is the actual image and lower row is segmented image for each frame
a Normal walk
b Swinging bag
c Carrying briefcase
d Walking with a dog
e Knees up
f Limping man
g Moon-walk
h Occluded legs
i Walking in a skirt
j Occluded by a pole

sequences of a walking person in various complex scenarios in front of different non-uniform backgrounds. Some example frames are shown in Fig. 24. For each of the test sequence, we used the normal action dataset as model sequence. All of the actions are recognised as 'walk' and most of them have significant difference with the nearest incorrect choice which is shown in Fig. 25. This implies that our system is relatively insensitive to partial occlusion, non-rigid deformation, shadow and noisy segmentation. Table 6 lists the robustness results of different methods on same dataset.

4.8 Recognition using less information

Under the assumption that the input can be partly corrupted, we used reduced but more robust information from each pose.

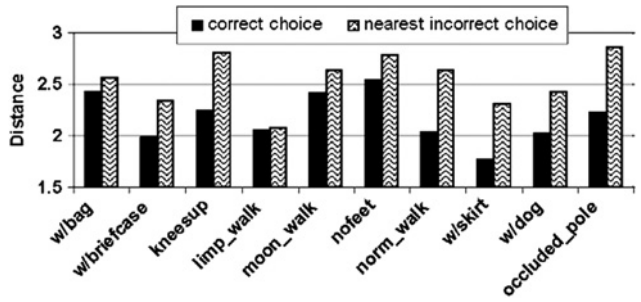


Fig. 25 *Correct and nearest incorrect choice of each distorted action*

Table 6 Robustness result for different methods

Method	Accuracy, %
our method	100
Gorelick *et al.* [12]	100
Bregonzio *et al.* [60]	90
Li *et al.* [58]	80

For each empty region of input pose, we calculate and rank the distance from a model pose. Then we choose the top 50% regions with the lowest distances for matching. We had obtained the recognition rates of 95% (four misclassifications) and 90% (one misclassification) for the Weizmann human action and Weizmann robust deformed dataset, respectively. These results indicate, if 50% regions of a pose are corrupted because of partial occlusion, segmentation error etc., our system can still recognise the action effectively.

5 Conclusion

Human action recognition based on negative space is investigated in this study. Unlike other region-based methods, our method extracts semantic-level features from the negative space regions to recognise actions. By employing negative spaces, we have the advantage to extract naturally formed regions that come with simple shapes. Hence, we can describe poses by simple geometric shapes and overcome the difficulties faced by positive space-based methods. We find that negative spaces can provide sufficient information to describe a pose which is proven by our experimental results. In addition, the semantic-level description of human pose is less sensitive to foreground segmentation errors. Our system obtained 100% accuracy on both Weizmann human action dataset and Weizmann robust deformation dataset. Fathi and Mori's [54] method also achieved 100% accuracy on the human action dataset but their system is not robust to non-rigid deformation of actions (e.g. person walking with skirts). The accuracy achieved on KTH dataset (scene 1) is also comparable with other methods. We showed that our system is robust with respect to variation of clothing, non-rigid deformation, partial occlusion, shadow and noisy segmentation. For all the datasets, we showed that most of the differences between the correct and nearest incorrect choices are well separated, which indicates that the selected

features are appropriate to distinguish between different types of actions. Moreover, we demonstrated that our feature extraction time per action is faster than state-of-the-art method. In summary our contributions are

1. We have enhanced the negative space concept with novel feature descriptors that provide semantic-level description of poses.
2. The recognition rate of our system is same as other state of the art methods for Weizmann dataset (accuracy 100%) and is comparable for KTH dataset (accuracy 94.67%). Furthermore, our system can recognise actions effectively even when half of the feature regions are corrupted. Hence, the system is relatively robust to change of clothing, noisy segmentation and partial occlusion.
3. Computationally simple features have been proposed for the system. Potentially, our system can run in real time in concept.

On the basis of the above discussion, we can say that our system is one of the top-ranked systems to date. One limitation of our approach is that our system is not fully viewpoint invariant. Currently, our system can recognise actions with angle 18° or less between image plane and action performing plane. With the high efficiency in computation, we are working on extending our system for multi-view human action recognition with multi-view models. This limitation can be overcome by shape compacting technique performed on positive space [64]. In contrast, the negative spaces are the complementary regions to positive space. This study has shown that with the features from the negative spaces, accurate and robust human action recognition can also be achieved. In future, we want to combine negative space and positive space descriptor to further improve the recognition power of our system.

6 References

1 Wang, L., Hu, W., Tan, T.: 'Recent developments in human motion analysis', *Pattern Recognit.*, 2003, **36**, (3), pp. 585–601
2 Saleemi, I., Hartung, L., Shah, M.: 'Scene understanding by statistical modeling of motion patterns'. Proc. Computer Vision and Pattern Recognition, 2010, pp. 2069–2076
3 Ikizler, N., Forsyth, D.A.: 'Searching for complex human activities with no visual examples', *Int. J. Comput. Vis.*, 2008, **80**, (3), pp. 337–357
4 Bregler, C.: 'Learning and recognizing human dynamics in video sequences'. Proc. Computer Vision and Pattern Recognition, 1997, pp. 568–574
5 Diaf, A., Ksantini, R., Boufama, B., Benlamri, R.: 'A novel human motion recognition method based on Eigenspace'. Proc. Lecture Notes in Computer Science, 2010, pp. 167–175
6 Chen, Y., Wu, Q., He, X.: 'Human action recognition by radon transform'. Proc. Int. Conf. on Data Mining Workshops, 2008, pp. 862–868
7 Goldenberg, R., Kimmel, R., Rivlin, E., Rudzsky, M.: 'Behavior classification by Eigen decomposition of periodic motions', *Pattern Recognit.*, 2005, **38**, (7), pp. 1033–1043
8 Singh, M., Mandai, M., Basu, A.: 'Pose recognition using the radon transform'. Proc. Midwest Symp. on Circuits and Systems, 2005, pp. 1091–1094
9 Boulgouris, N.V., Hatzinakos, D., Plataniotis, K.N.: 'Gait recognition: a challening signal processing technology for biometric identification', *IEEE Signal Process. Mag.*, 2005, **22**, (6), pp. 78–90
10 Niebles, J.C., Wang, H., Fei-Fei, L.: 'Unsupervised learning of human action categories using spatial-temporal words', *Int. J. Comput. Vis.*, 2008, **79**, (3), pp. 299–318
11 Wang, X., Ma, X., Grimson, W.E.: 'Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009, **31**, (3), pp. 539–555
12 Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: 'Actions as space-time shapes', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2007, **29**, (12), pp. 2247–2253
13 Shechtman, E., Irani, M.: 'Space-time behavior based correlation'. Proc. Computer Vision and Pattern Recognition, 2005, pp. 405–412
14 Vuuren, M.J.V., Jager, G.: 'Art and image processing'. Proc. Conf. Pattern Recognition Association of South Africa, South Africa, 2001, pp. 23–28
15 Yu, E., Aggarwal, J.K.: 'Human action recognition with extremities as semantic posture representation'. Proc. Computer Vision and Pattern Recognition Workshops, 2009, pp. 1–8
16 Rahman, S.A., Liyuan, L., Leung, M.K.H.: 'Human action recognition by negative space analysis'. Proc. Cyberworlds (CW), 2010, pp. 354–359
17 Poppe, R.: 'A survey on vision-based human action recognition', *Image Vis. Comput.*, 2010, **28**, (6), pp. 976–990
18 Moeslund, T.B., Hilton, A., Kruger, V.: 'A survey of advances in vision-based human motion capture and analysis', *Comput. Vis. Image Underst.*, 2006, **104**, (2), pp. 90–126
19 Weinland, D., Ronfard, R., Boyer, E.: 'A survey of vision-based methods for action representation, segmentation and recognition', *Comput. Vis. Image Underst.*, 2011, **115**, (2), pp. 224–241
20 Wang, L., Suter, D.: 'Recognizing human activities from silhouettes: motion subspace and factorial discriminative graphical model'. Proc. Computer Vision and Pattern Recognition, 2007, pp. 1–8
21 Ali, A., Aggarwal, J.K.: 'Segmentation and recognition of continuous human activity'. Proc. IEEE Workshop on Detection and Recognition of Events in Video, 2001, pp. 28–35
22 Scovanner, P., Ali, S., Shah, M.: 'A 3-dimensional sift descriptor and its application to action recognition'. Proc. Int. Conf. on Multimedia, 2007, pp. 357–360
23 Laptev, I., Lindeberg, T.: 'Space-time interest points'. Proc. Int. Conf. on Computer Vision, 2003, pp. 432–439
24 Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: 'Behavior recognition via sparse spatio-temporal features'. Proc. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005, pp. 65–72
25 Shao, L., Gao, R., Liu, Y., Zhang, H.: 'Transform based spatio-temporal descriptors for human action recognition', *Neurocomputing*, 2011, **74**, (6), pp. 962–973
26 Bobick, A.F., Davis, J.W.: 'The recognition of human movement using temporal templates', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2001, **23**, (3), pp. 257–267
27 Meng, H., Pears, N., Bailey, C.: 'A human action recognition system for embedded computer vision application'. Proc. Computer Vision and Pattern Recognition, 2007, pp. 1–6
28 Liu, Z., Sarkar, S.: 'Simplest representation yet for gait recognition: averaged silhouette'. Proc. Int. Conf. on Pattern Recognition, 2004, pp. 211–214
29 Han, J., Bhanu, B.: 'Human activity recognition in thermal infrared imagery'. Proc. Computer Vision and Pattern Recognition, 2005, pp. 17–24
30 Lam, T.H.W., Cheung, K.H., Liu, J.N.K.: 'Gait flow image: a silhouette-based gait representation for human identification', *Pattern Recognit.*, 2011, **44**, (4), pp. 973–987
31 Zhang, E., Zhao, Y., Xiong, W.: 'Active energy image plus 2DLPP for gait recognition', *Signal Process.*, 2010, **90**, (7), pp. 2295–2302
32 Ikizler, N., Duygulu, P.: 'Histogram of oriented rectangles: a new pose descriptor for human action recognition', *Image Vis. Comput.*, 2009, **27**, (10), pp. 1515–1526
33 Agarwal, A., Triggs, B.: 'Recovering 3D human pose from monocular images', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006, **28**, (1), pp. 44–58
34 Lee, M.W., Nevatia, R.: 'Human pose tracking in monocular sequence using multilevel structured models', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009, **31**, (1), pp. 27–38
35 Ramanan, D., Forsyth, D.A., Zisserman, A.: 'Tracking people by learning their appearance', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2007, **29**, (1), pp. 65–81
36 Cheung, G.K.M., Takeo, K., Jean-Yves, B., Mark, H.: 'A real time system for robust 3D voxel reconstruction of human motions'. Proc. Computer Vision and Pattern Recognition, 2000, pp. 714–720
37 Menier, C., Boyer, E., Raffin, B.: '3D skeleton-based body pose recovery'. Proc. Int. Symp. on 3D Data Processing, Visualization and Transmission, 2006, pp. 389–396
38 Hofmann, M., Gavrila, D.: 'Single-frame 3D human pose recovery from multiple views'. Proc. DAGM Symp. on Pattern Recognit., 2009, pp. 71–80
39 Guo, Y., Xu, G., Tsuji, S.: 'Understanding human motion patterns'. Proc. Int. Conf. on Image Analysis and Pattern Recognition, 1994, pp. 325–329

doi: 10.1049/iet-cvi.2011.0185

40 Leung, M.K.H., Yang, Y.H.: 'First sight: a human body outline labeling system', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1995, **17**, (4), pp. 359–377
41 Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P.: 'Pfinder: real-time tracking of the human body', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1997, **19**, (7), pp. 780–785
42 Felzenszwalb, P.F., Huttenlocher, D.P.: 'Pictorial structures for object recognition', *Int. J. Comput. Vis.*, 2005, **61**, (1), pp. 55–79
43 Hua, G., Yang, M.H., Wu, Y.: 'Learning to estimate human pose with data driven belief propagation'. Proc. Computer Vision and Pattern Recognition, 2005, pp. 747–754
44 Sigal, L., Black, M.J.: 'Measure locally, reason globally: occlusion-sensitive articulated pose estimation'. Proc. Computer Vision and Pattern Recognition, 2006, pp. 2041–2048
45 Ferrari, V., Marín-Jiménez, M., Zisserman, A.: '2D human pose estimation in TV shows'. Proc. Statistical and Geometrical Approaches to Visual Motion Analysis, 2009, pp. 128–147
46 Zivkovic, Z., van der Heijden, F.: 'Efficient adaptive density estimation per image pixel for the task of background subtraction', *Pattern Recogn. Lett.*, 2006, **27**, (7), pp. 773–780
47 Yu, H., Huang, T.S., Niemann, H.: 'A region-based method for model-free object tracking'. Proc. Int. Conf. on Pattern Recognition, 2002, pp. 592–595
48 Hu, M.K.: 'Visual pattern recognition by moment invariants', *IEEE Trans. Inf. Theory*, 1962, **8**, (2), pp. 179–187
49 Sonka, M., Hlavac, V., Boyle, R.: 'Image processing analysis and machine vision' (Thomson-Engineering, 1998, 2nd edn.)
50 Teague, M.R.: 'Image analysis via the general theory of moments', *J. Opt. Soc. Am.*, 1980, **70**, pp. 920–930
51 Rabiner, L., Juang, B.-H.: 'Fundamentals of speech recognition' (Prentice-Hall, Inc., 1993)
52 http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html#Database
53 Schuldt, C., Laptev, I., Caputo, B.: 'Recognizing human actions: a local SVM approach'. Proc. Int. Conf. on Pattern Recognition, 2004, pp. 32–36
54 Fathi, A., Mori, G.: 'Action recognition by learning mid-level motion features'. Proc. Computer Vision and Pattern Recognition, 2008, pp. 1–8
55 Jhuang, H., Serre, T., Wolf, L., Poggio, T.: 'A biologically inspired system for action recognition'. Proc. Int. Conf. on Computer Vision, 2007, pp. 1–8
56 Seo, H., Milanfar, P.: 'Action recognition from one example', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011, **33**, (5), pp. 867–882
57 Wu, X., Jia, Y., Liang, W.: 'Incremental discriminant-analysis of canonical correlations for action recognition', *Pattern Recognit.*, 2010, **43**, (12), pp. 4190–4197
58 Li, W., Zhang, Z., Liu, Z.: 'Graphical modeling and decoding of human actions'. Proc. IEEE Workshop on Multimedia Signal Processing, 2008, pp. 175–180
59 Lucena, M., de La Blanca, N.P., Fuertes, J.M., Marín-Jiménez, M.: 'Human action recognition using optical flow accumulated local histograms'. Proc. Fourth Iberian Conf. on Pattern Recognition and Image Analysis, 2009, pp. 32–39
60 Bregonzio, M., Shaogang, G., Tao, X.: 'Recognizing action as clouds of space-time interest points'. Proc. Computer Vision and Pattern Recognition, 2009, pp. 1948–1955
61 Lin, Z., Jiang, Z., Davis, L.S.: 'Recognizing actions by shape-motion prototype trees'. Proc. Int. Conf. on Computer Vision, 2009, pp. 444–451
62 Schindler, K., van Gool, L.: 'Action snippets: how many frames does human action recognition require?'. Proc. Computer Vision and Pattern Recognition, 2008, pp. 1–8
63 Ahmad, M., Lee, S.-W.: 'Human action recognition using shape and CLG-motion flow from multi-view image sequences', *Pattern Recognit.*, 2008, **41**, (7), pp. 2237–2252
64 Leu, J.G.: 'Shape normalization through compacting', *Pattern Recognit. Lett.*, 1989, **10**, (4), pp. 243–250