



A differential geometric approach to representing the human actions

Alper Yilmaz ^{a,*}, Mubarak Shah ^b

^a *Photogrammetric Computer Vision Laboratory, Department of Civil and Environmental Engineering and Geodetic Science, The Ohio State University, Columbus, OH 43210, USA*

^b *School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816, USA*

Received 8 May 2006; accepted 18 September 2007

Abstract

This paper presents a novel representation for human actions which encodes the variations in the shape and motion of the performing actor. When an actor performs an action, at each time instant, the outer object boundary is projected to the image plane as a 2D contour. A sequence of such contours forms a 3D volume in the spatiotemporal space. The differential geometric analysis of the volume surface results in a set of action descriptors. These descriptors constitute the action sketch which is used to represent the human actions. The action sketch captures the changes in the shape and motion of the performing actor in a unified manner. Since the action sketch is obtained from the extrema of the differential geometric surface features, it is robust to viewpoint changes. We demonstrate the versatility of the action sketch in the context of action recognition, which is formulated as a view geometric similarity problem.

© 2007 Elsevier Inc. All rights reserved.

Keywords: Human Actions; Differential geometry; Epipolar geometry; Action representation; Action recognition

1. Introduction

The analysis of the human actions in a video stream is an active research area in the Computer Vision community. Over the past decade, the action content in a video clip has been analyzed by using different techniques, such as the Hidden Markov Models [1], Finite State Machines [2], Neural Networks and Context Free Grammars [3]. Despite the success achieved by using these approaches, their main limitation is the requirement of a controlled environment, such as a fixed camera viewpoint and controlled actor motion. This limitation stems from the features used in the action analysis. Thus the important question of “Which features are suitable to represent an action?” remains unresolved. Considering an action as a space-time construct, important events during an action can be best described

by the instants when the motion and posture of the actor change.

There are various approaches to represent human actions. A common form of action representation is to use a set of motion trajectories. The motion trajectories can be analyzed to find the instants that characterize the action, such as the changes in the speed and direction of a hand. In [4], Rao et al. extract the spatiotemporal curvature minima and maxima of a single motion trajectory (specifically the hand trajectory). In their approach, the minima and maxima in the curvature values correspond to sudden changes in the speed and direction of the tracked body part. Gritai et al. [5] use 13 motion trajectories obtained by tracking the body landmarks, such as the head, arms and legs. When compared to the use of a single trajectory, multiple trajectories provide a stronger constraint. Moreover, the relative positions of these trajectories implicitly represent the actor’s posture during an action.

Another possible action representation is to use a set of motion features extracted from the bounding box enclosing the actor. These features include but are not limited to the optical flow of each pixel or the affine motion of a bound-

* Corresponding author. Fax: +1 614 247 4323.

E-mail addresses: yilmaz.15@osu.edu (A. Yilmaz), shah@cs.ucf.edu (M. Shah).

URLs: <http://dpl.ceegs.ohio-state.edu> (A. Yilmaz), <http://www.cs.ucf.edu/vision> (M. Shah).

ing box around the object. In [6], Polana and Nelson generate the statistics of the normal flows computed in consecutive frames to represent an action. Efros et al. [7] use the optical flow for the same purpose. For representing the facial expressions, Black and Yacoob [8] analyze the variations of the affine motion parameters computed from the bounding boxes around the eyes, eyebrows and mouth. Similarly, Yang et al. [9] use the relative position between the head and hands, along with the affine motion of each hand region to represent the American sign language gestures. Instead of the optical flow, Zelnik-Manor and Irani [10] use the magnitude of the temporal image gradient at various scales for representing the human actions. In [12], Hsu and Harashima proposed to extract the motion discontinuities by analyzing the spatiotemporal energy computed from the responses of the 3D steerable filters. In their approach, the motion discontinuities are assumed to occur only on the object boundaries.

In addition to the motion templates, the appearance of the actor, in the form of the edge maps, shape templates or skeletal models, has also been used to represent the actions. Bobick and Davis [13] model the changes in the actor's posture by generating temporal templates from a set of tracked silhouettes. The moments computed from these templates are then used to represent the actions. For recognizing the American Sign Language, Starner and Pentland [14] use the appearance templates defined by the bounding boxes enclosing the hand regions. In [15], Cutler and Davis use the color similarity computed in a bounding box enclosing the actor to find the periodicity of the human motion. Instead of using the color observations, Mori et al. [16] use a sequence of shape histograms generated from the edges inside the bounding box. Sullivan and Carlsson [17] also use the edge maps, however, instead of a histogram, they use a voting matrix indexed by the gradient angles of both edge maps. Syeda-Mahmood et al. [18] use the local 2D shapes (contour parts) in the spatial coordinates around a set of points placed on the contours enclosing the object region. The stack of such contours is referred to as the generalized cylinder. The skeletal representation, which models the actor's posture [19], is obtained by applying the medial axis transform to the actor silhouette at a given time instant. From a set of skeletons, the characteristic information of the action is extracted by analyzing the changes in the relative orientations between the skeleton segments over time.

In this paper, we propose a novel action representation that simultaneously exploits the changes in the object shape and motion in a unified manner. The proposed action representation is generated in two steps: the creation of a continuous volume from the object silhouettes or contours, and the extraction of the action descriptors from this volume. The first step assumes that the object tracking has already been performed, and a sequence of object silhouettes or contours is extracted (see Fig. 1a). Using the contours, we generate a spatiotemporal volume by establishing point correspondences. The resulting volume is referred to as the "action volume". Since the action volume is continuous in the spatiotemporal space (see Fig. 1b), we can synthesize an action at different execution rates by discretizing the volume at different sampling rates. The second step extracts descriptors of an action by analyzing the differential geometry of the volume surface using Weingarten mapping. This analysis results in labeling the patches on the volume surface as peaks, pits, valleys and ridges. However, it is well known that computing the differential quantities is sensitive to noise. Hence, prior to evaluating the differential geometry, we smooth the action volume using the level set formalism. This process results in reliable computation of the differential features related to:

- the convex or concave parts of the (spatial) object contours,
- The minima or maxima of the spatiotemporal curvature of the motion trajectories.

The existence of the concavity or convexity in the object-contour as well as the existence of the curvature minima or maxima on the motion-trajectory are preserved with the changes in the viewpoint. Thus, the proposed action descriptors are robust to viewpoint changes. The set of action descriptors is referred to as the *action sketch*.

We demonstrate the proposed action representation in the context of action recognition. Two videos of the same action can be considered as two different views of the same scene related by the epipolar geometry. The epipolar geometry requires correspondences between the action descriptors in both views are known. For this purpose, we use a graph theoretical approach where the correspondences are established using the maximum matching of a bipartite graph. Once the point correspondences are established, the epipolar geometry between the two actions is evaluated by

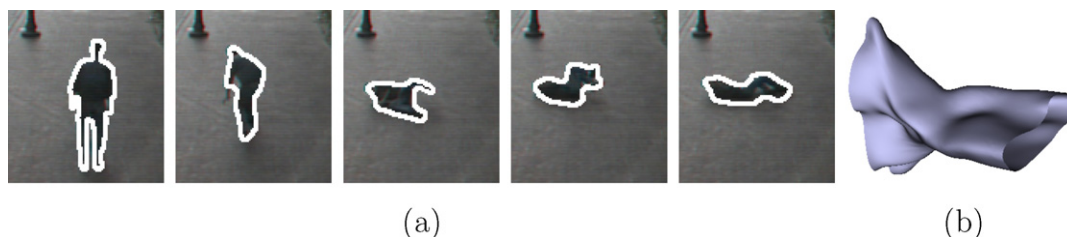


Fig. 1. (a) A sequence of object contours for a falling person and (b) the corresponding spatiotemporal volume.

computing the symmetric epipolar distance. The accumulated symmetric epipolar distances are then used as the matching score between the two actions.

The organization of the paper is as follows: in the next section, we discuss the generation of the action volume from a noisy sequence of object contours. Section 3 describes the extraction of the action sketch from an action volume (Section 3.1), the effect of noise on the proposed representation (Section 3.2), and its relationship to various types of object motions (Section 3.3). In Section 4, we discuss the effect of the changes in the viewpoint on the action sketch (Section 4.1), and propose a recognition scheme based on the epipolar geometry (Section 4.3). The experiments demonstrating the performance of the proposed approach on 30 actions is given in Section 5. Subsequently, we conclude in Section 6.

2. Generating the action volume

The spatiotemporal volume, which is commonly generated by stacking a sequence of video frames, has been widely used in the computer vision community [10,11,29]. In our representation, instead of stacking the video frames, we stack only the object contours (boundaries) and generate a volume from this stack. Generating a volume from a set of object silhouettes has been investigated in [20] specifically for the walking action. In their approach the authors fit a manually generated walking volume, that is comprised of two surfaces (the right and left body parts), to a sequence of silhouettes. Due to the use of a manually generated volume, this approach requires the object moves parallel to the camera. The main limitation of using manually generated volumes is that generating a different volume for different actions observed from different viewpoints is not practical. In this paper, we propose to generate the volume *automatically* for any action viewed from any viewing direction.

The contour of an object can be extracted and tracked by means of background subtraction [21] or contour tracking [22]. In this paper, we use the contour tracking approach discussed in [22], which provides us with a set of closed object contours Γ^t (see Fig. 1a).

Given a sequence of object contours, which provides a dense point cloud in the three-dimensional spatiotemporal coordinates, is a difficult task and an ongoing research in computer graphics [24,25]. In general, the volume generation methods in literature require proximal surface points. This requirement, however, is not satisfied in the domain of the human actions, where the sudden motion of an actor may cause the surface points to be distant. To address this problem, we propose a surface reconstruction technique tailored for the domain of human actions and provide a qualitative comparison with the marching cubes method, which is a baseline surface reconstruction method. Given a point cloud, the marching cubes method approximates isosurfaces by estimating the local geometry from neighboring points [23].

2.1. Point association in consecutive contours

We simplify the isosurface generation to point matching between two consecutive contours Γ^t and Γ^{t+1} . Associating two point sets obtained from the contours of a non-rigid object is still an open problem. An obvious difficulty arises due to the points in one set that do not have correspondences in the other set. In such cases, 1–M (one-to-many) or M–1 (many-to-one) mappings become important. An intuitive approach to find point matches is to use the nearest neighbor criterion [26]. However, as shown in Fig. 2a–c, associating the points to their nearest neighbors results in an incorrect topology. Another possible approach is to compute the rigid motion between two consecutive contours [27]. However, this approach cannot handle a large amount of 1–M mappings. Point matching can also be achieved by using a local shape similarity metric. Sullivan and Carlsson [17] defined the local shape at a given point by its tangent direction. For the same purpose Mori and Malik [16] used the shape histograms for each point in the object’s edge map, which includes the points inside the object boundary. Neither of these features are discriminative, therefore, they may miss points or establish incorrect correspondences (see [16, Fig. 2] and [17, Fig. 3]).

We propose a graph theoretic approach to find the matching points, which is similar in spirit to the work of [16] and [28]. Let L and R be two point sets corresponding to Γ^t and Γ^{t+1} , respectively. We define a bipartite graph $G(V, E)$ with $|V| = |L| + |R|$ vertices, where $|\cdot|$ is the cardinality of the set (see Fig. 3a).

The weight of each edge from a vertex in L to a vertex in R is defined by the spatial proximity, the angular distance of the normal directions, and the similarity of the shape of the corresponding vertices (see Fig. 3b). The combination of these three constraints assures that the vertex associations are spatially as close as possible, and the geometry and orientation of vertices are similar such that the problems shown in Fig. 2 are not observed. Let $\mathbf{c}_i = [x_i, y_i, t]^T$ and $\mathbf{c}_j = [x_j, y_j, t + 1]^T$ be the vertices in L and R , respectively. We compute the spatial proximity between the corresponding vertices by:

$$d_{i,j} = \|\mathbf{c}_i - \mathbf{c}_j\|_2. \quad (1)$$

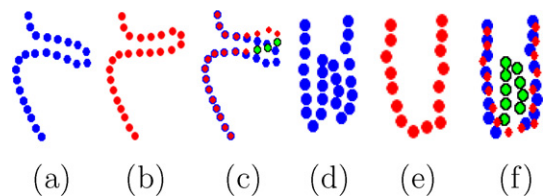


Fig. 2. (a and b) The contour points around the hand region in two consecutive frames. (c) The points in (b) superimposed on the points in (a). The points in green color denote the wrong associations due to the nearest neighbor criterion. (d and e) The contour points around the leg region in two consecutive frames. (f) The points in (d) superimposed on the points in (e). As shown, due to the leg motion, the boundary between the legs disappear, such that the green points in frame t have no correspondences in frame $t + 1$.

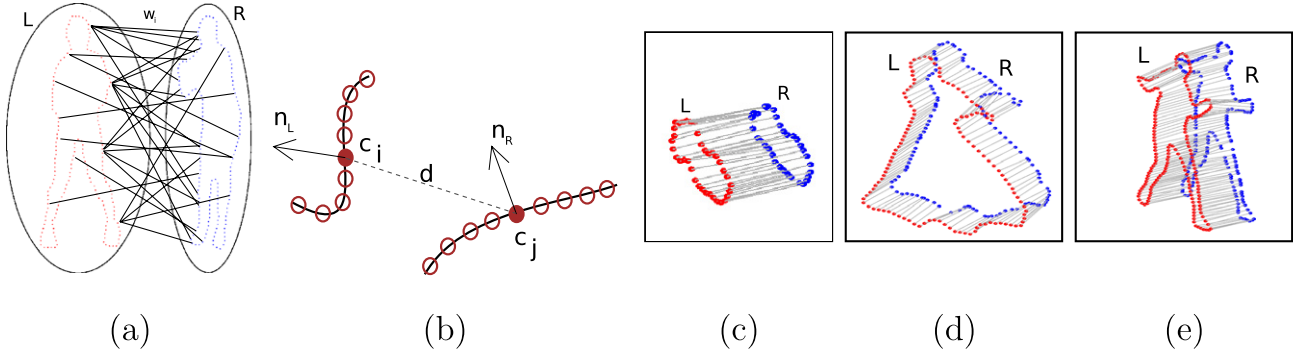


Fig. 3. (a) The set of vertices (the nodes on the object contour) and the edges in two consecutive frames from the tennis sequence, (b) the local contour neighborhoods in two consecutive frames used in defining the weights for the matching between the vertices i and j . The resulting vertex matchings between the contours of (c) the falling, (d) dance, and (e) tennis stroke actions.

The distance between the normal vectors \vec{n}_i and \vec{n}_j , which respectively correspond to \mathbf{c}_i and \mathbf{c}_j , is obtained by considering the angle $\alpha_{i,j} = \arccos(\vec{n}_i \cdot \vec{n}_j)$. Let $\mathbf{T}_{i,j} = \mathbf{c}_i - \mathbf{c}_j$ be the translation and

$$\mathbf{R}_{i,j} = \begin{pmatrix} \cos \alpha_{i,j} & \sin \alpha_{i,j} \\ -\sin \alpha_{i,j} & \cos \alpha_{i,j} \end{pmatrix},$$

be the rotation of the vertex \mathbf{c}_i from the frame at time t to the frame at time $t + 1$. The similarity of the shape between the vertices \mathbf{c}_i and \mathbf{c}_j is defined in the neighborhoods N_i and N_j after compensating $\mathbf{T}_{i,j}$ and $\mathbf{R}_{i,j}$:

$$\xi_{i,j} = \sum_{\mathbf{x}_j \in N_j} \|\hat{\mathbf{x}}_i - \mathbf{x}_j\|_2, \quad (2)$$

where $\hat{\mathbf{x}}_i = \mathbf{R}_{i,j}\mathbf{x}_i + \mathbf{T}_{i,j}$, and $\mathbf{x}_i \in N_i$ is the vertex corresponding to \mathbf{x}_j (note that $|N_i| = |N_j|$). We assume these three measures are distributed by zero mean normal distributions: $d_{i,j} \sim N(0, \sigma_d)$, $\alpha_{i,j} \sim N(0, \sigma_\alpha)$ and $\xi_{i,j} \sim N(0, \xi_d)$. The association hypothesis ($w_{i,j}$ from \mathbf{c}_i to \mathbf{c}_j) is tested by means of the joint probability computed from these three distributions:

$$w_{i,j} = \exp\left(-\frac{d_{i,j}^2}{\sigma_d^2}\right) \exp\left(-\frac{\alpha_{i,j}^2}{\sigma_\alpha^2}\right) \exp\left(-\frac{\xi_{i,j}^2}{\sigma_\xi^2}\right). \quad (3)$$

The distribution parameters σ_d , σ_α and σ_ξ control the contribution of the distance between the vertices, the angle between the normals and the degree of the shape variation respectively. In our experiments, we fix: $\sigma_d = 15$, $\sigma_\alpha = 0.5$ and $\sigma_\xi = |N_i|$.

We solve the point correspondence problem by computing the maximum matching of a bipartite graph with the weights $w_{i,j}$. A matching of a graph is a set of edges such that no two edges share a common vertex. The maximum matching provides 1–1 (one-to-one) mappings from L to R such that $\sum_i \sum_j w_{i,j}$ is maximized [30]. Prior to the matching, the edges with low confidences, due to the non-rigid object motion, are pruned. Upon establishing the correspondences, the spatial relations between the points are usually not maintained. For instance, the following matchings $\mathbf{c}_i \rightarrow \mathbf{c}_j$ and $\mathbf{c}_{i-2} \rightarrow \mathbf{c}_{j+3}$ can not hold simultaneously. Following this observation, we apply a post process which

iteratively prunes the outliers and creates new associations that are 1– M or M –1. Fig. 3c–e show the final vertex matchings for three different actions. In Fig. 4, the action volumes generated for the tennis stroke, dance and walking actions are shown.

To see the effectiveness of the proposed method, we generated a volume for the falling action using both the marching cubes and the proposed method. For fairness, we smooth both volumes using the same method with the same parameters. In order to qualitatively compare the results, we computed the differential geometric properties defined by the Gaussian and mean curvatures of the volume surfaces. In Fig. 5, we show zoom-in views for the same part from two volumes where the actor motion and shape significantly change. Due to the sudden motion of the actor, the neighboring points are distant from each other. As shown in Fig. 5b, the marching cubes method cannot handle the sudden changes in object motion, hence generates an ambiguous action sketch. In our experiments, 3D Delaunay triangulation also had poor performance around the body parts with sudden motion.

2.2. The properties of the action volume

The action volume generated from the point matches between the contours can be considered a manifold, $\mathbf{B}(x, y, t)$, such that a continuous surface can be approximated by computing the surface equations for each small surface patch. An important advantage of this approximation is its ability to provide the contour at any given time instant t , such as the contours at time $t = 2.3$ or 10.7 . Since the action volume is generated from a set of contours, instead of using the three dimensional (x, y, t) representation, we can define a 2D parametric representation by considering the arc-length s of the contour and the time t :

$$\mathbf{B} = f(s, t) = [x(s, t), y(s, t), t]. \quad (4)$$

In this representation arc-length encodes the object shape and the time encodes its motion, such that, fixing the s parameter generates 2D motion trajectories of any point on the object boundary. Similarly, fixing the t parameter generates the object contours at time t .

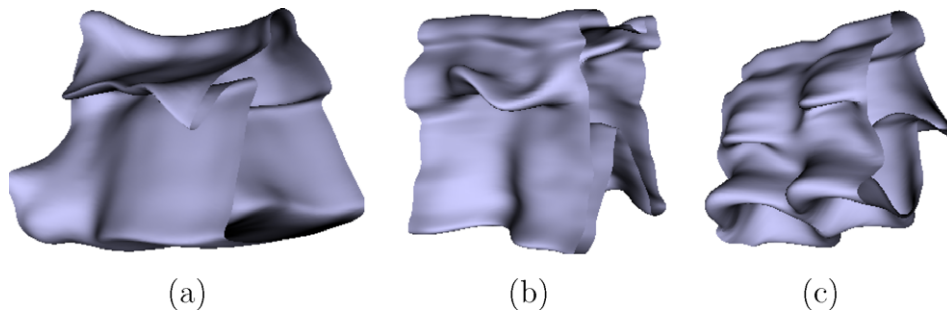


Fig. 4. The volumes for (a) the dance, (b) tennis stroke and (c) walking actions.

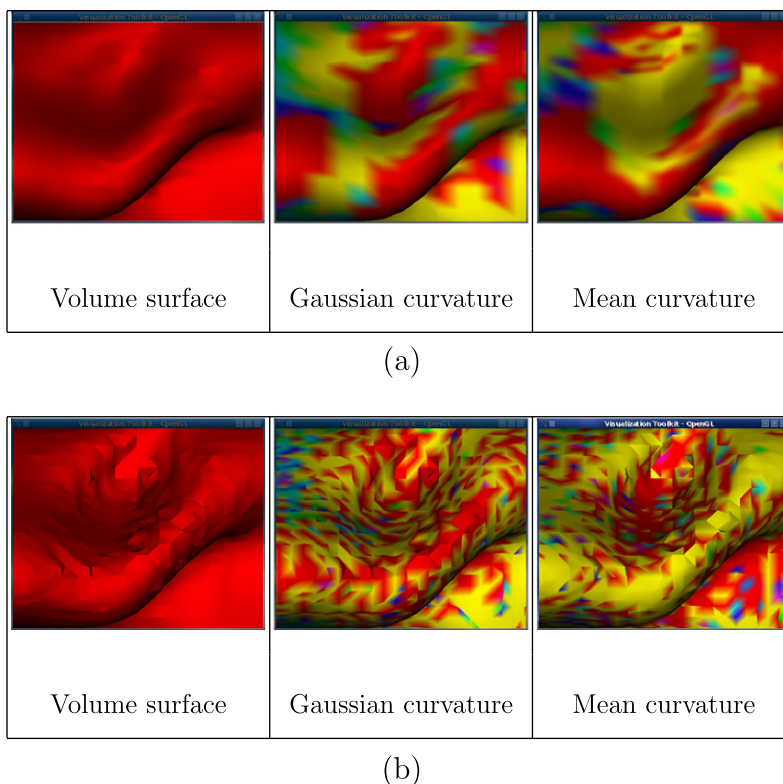


Fig. 5. For comparing the differential surface geometry, we generated the action volume using both the marching cubes method and the proposed method. The same level of smoothing is applied for both methods. The figures display part of the volume surface where sudden motion and shape changes occur. The different colors on the volumes represent the value of the curvatures (red, high; blue, low). (a) Proposed method accurately estimates the Gaussian and mean curvatures. (b) The marching cubes method gives incorrect isosurfaces, hence, the Gaussian and mean curvatures are not correct. We use a standard implementation of the marching cubes method available from the visualization toolkit at www.vtk.org. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

Assuming the pace of the actor does not drastically change,¹ several discrete approximations of the action volume can be generated by using different samplings in time. In Fig. 6, we show an example to demonstrate this property for the dance sequence where a synthetic dance clip is generated by randomly selecting 20 frames from among 40 frames. As seen from the figure, although 50% of the observations are missing, the volumes look very similar. However, we should note that this property is only valid for

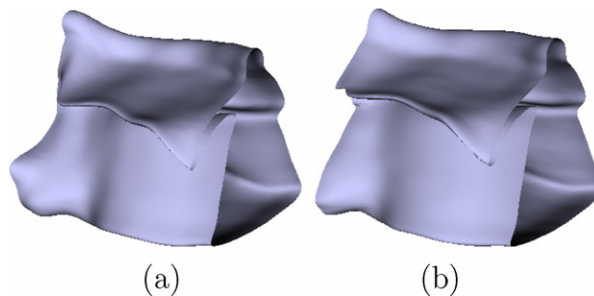


Fig. 6. The impact of the temporal sampling. The action volumes for (a) the dance sequence with 40 frames and (b) the synthetic dance sequence with 20 frames, which is generated by randomly removing frames from (a).

¹ For instance, for an atomic walking action, we assume the pace of the first step is equal to the pace of the second step.

the atomic actions. For instance, this property is satisfied for two walking cycles, but it is invalid for the video clips which may contain different number of walking cycles.

3. The action sketch

In the field of psychology, the changes in the direction and/or the speed of the body parts have been successfully used to characterize a human action [31]. In this formalization, the body part accounts for the shape of the object, and the direction/speed pair accounts for the motion of the object. For instance, grabbing an object with a hand can be characterized by the position of the fingers at various time instants corresponding to “the hand reaches the object”, “the fingers grab the object”, and “the hand recedes with the object”.

In our approach, the action volume encodes three quantities for each point on the contour: the shape, direction of motion and speed. Obviously these quantities are dependent on the camera viewpoint, such that the action volume will be different when the action is captured from different viewpoints. However, the differential properties, which are related to the shape and motion variations, are not dependent on the viewpoint. These properties are implicitly encoded on the volume surface which can be extracted by analyzing the surface geometry. We should note that the differential features used in this paper are different from the differential features used in other papers which extract the local motion [6,7,10]. In particular, the local motion results in a dense set of features which are usually redundant due to the similar motion of the pixels that belong to the same body parts. In contrast, the proposed representation is composed of a sparse set of features that are related to the geometry of the volume surface. These geometric features are not redundant and encode the shape and motion changes simultaneously. In the following sec-

tion, we discuss how the action descriptors can be extracted using the differential geometry.

3.1. Finding the action descriptors

The differential geometric properties of surfaces have commonly been considered in the context of range image analysis. An extension of these features has also been utilized for video segmentation by computing motion discontinuities from the image brightness [12]. Here, we extend these results to a new domain: human actions. The differential geometry of a surface is described by two quantities: The Gaussian curvature K , and the mean curvature H . Both of these curvatures are computed from the Weingarten mapping defined in terms of the first and second fundamental forms of a surface [32]. In particular, the signs of K and H define the surface type and the values of K and H define the surface sharpness. There are eight types of surfaces: peak, ridge, saddle ridge, flat, minimal, pit, valley and saddle valley (see Table 1). These surface types are also known as the fundamental surface types.

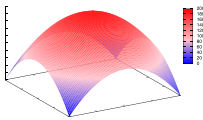
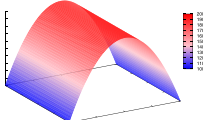
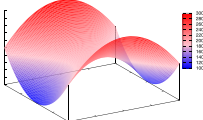
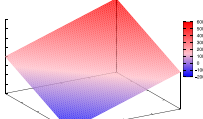
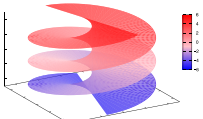
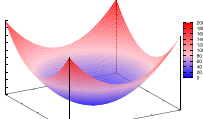
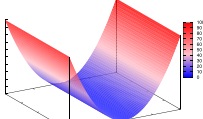
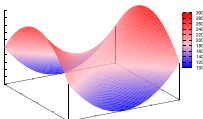
3.1.1. The first fundamental form

The first fundamental form is the inner product of the tangent vector at a given point $\mathbf{x}(s, t)$ and can be computed in the direction (s_p, t_p) by:

$$\mathbf{I}(s, t, s_p, t_p) = [s_p \ t_p]^T \underbrace{\begin{bmatrix} E & F \\ F & G \end{bmatrix}}_{\mathbf{g}} [s_p \ t_p], \quad (5)$$

where $E = \mathbf{x}_s \cdot \mathbf{x}_s$, $F = \mathbf{x}_s \cdot \mathbf{x}_t$, $G = \mathbf{x}_t \cdot \mathbf{x}_t$, and the subscripts denote the partial derivatives with respect to the arc-length and time. The \mathbf{g} matrix is called the metric tensor of the surface and has the same role as the speed function for a spatiotemporal trajectory [33]. In particular, E in (5) encodes

Table 1
The surface types and their relation to the mean, H , and Gaussian, K , curvatures

	$K > 0$	$K = 0$	$K < 0$
$H < 0$	 Peak	 Ridge	 Saddle ridge
$H = 0$	None	 Flat	 Minimal
$H > 0$	 Pit	 Valley	 Saddle valley

the spatial information, whereas F and G encode the velocity.

3.1.2. The second fundamental form

The second fundamental form defines the variation of the normal vector with respect to the surface position. It is computed by the following equation:

$$\mathbf{II}(s, t, s_p, t_p) = [s_p \ t_p]^T \underbrace{\begin{bmatrix} L & M \\ M & N \end{bmatrix}}_{\mathbf{b}} [s_p \ t_p], \quad (6)$$

where $L = \mathbf{x}_{ss} \cdot \vec{n}$, $M = \mathbf{x}_{st} \cdot \vec{n}$ and $N = \mathbf{x}_{tt} \cdot \vec{n}$, \vec{n} is the unit normal vector, and the subscripts denote the second order partial derivatives. In terms of encoding the motion, N in Eq. (6) is related to the acceleration of $\mathbf{x}(s, t)$.

3.1.3. The Weingarten mapping

The Weingarten mapping combines the first fundamental form (5) and the second fundamental form (6) into one single matrix S :

$$S = \mathbf{g}^{-1} \mathbf{b} = \frac{1}{EG - F^2} \begin{bmatrix} GL - FM & GM - FN \\ EM - FL & EN - FM \end{bmatrix}. \quad (7)$$

The Gaussian curvature K , and the mean curvature H are the two algebraic invariants of the Weingarten mapping [34]. In particular, the Gaussian curvature is the determinant of S :

$$K = \det(S) = \frac{LN - M^2}{EG - F^2}, \quad (8)$$

and the mean curvature is the half of the trace of S :

$$H = \frac{1}{2} \text{trace}(S) = \frac{EN + GL + 2FM}{2(EG - F^2)}. \quad (9)$$

In Fig. 7, we show the mean and Gaussian curvatures computed for the falling action.

Once the Gaussian and mean curvatures are computed for each point on the action volume, we apply a non-maximal suppression to $|K|$ and $|H|$ to extract the action descriptors, hence the action sketch. In Fig. 8, we show several action sketches with the action descriptors color coded according to the fundamental surface types.

3.2. The action sketch in presence of noise

The differential quantities which are used to extract the action sketch are known for their sensitivity to noise. In the context of action volumes, which are generated from the tracked object contours, the noise occurs due to imperfect contour parts emanating from poor tracking performance. For instance, tracking methods that rely on the appearance will fail to provide good object regions when the object appears similar to the background. In this case, there will be missing or spurious object regions and the resulting boundaries will be noisy. This effect is shown in Fig. 10A for several sequences from the standard Human ID (HID) database of tracked individuals [35]. The sequences shown in this figure are captured from the same camera viewpoint, while different actors performed walking action with different execution styles. Due to different clothing and the background clutter, in each sequence, different parts of the silhouettes are missing.

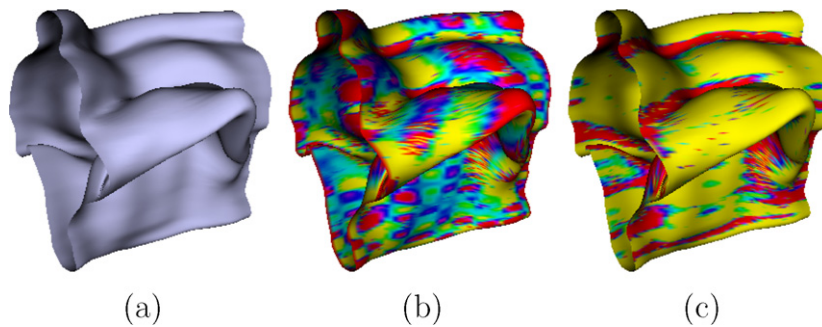


Fig. 7. (a) The action volume, (b) the Gaussian and (c) mean curvatures. Different colors represent the value of the curvature (red, high; blue, low). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

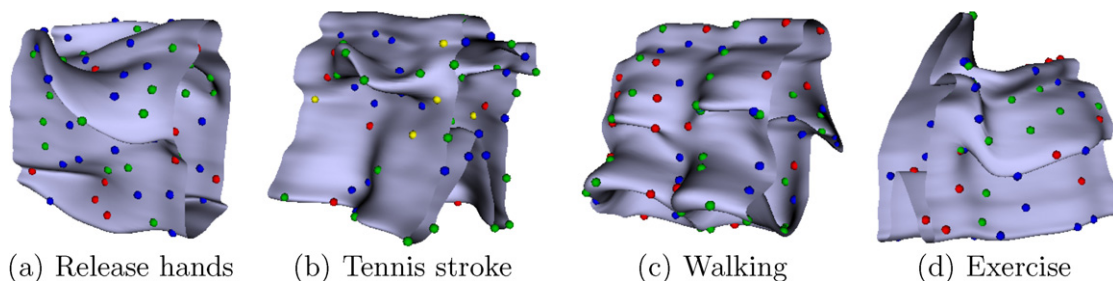


Fig. 8. The color coded action descriptors corresponding to various surface types. The color codes are: blue (peak), jade (saddle ridge), pink (saddle valley). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

To reduce the effect of noise, we first apply a smoothing operation on the action volume generated from the noisy silhouettes. In the context of 3D surfaces, smoothing has received a considerable amount of interest and numerous approaches have been proposed for generating smooth surfaces which are suitable for computing the differential surface properties [36–38]. In this paper, we adapt the level set formalism discussed in [38] for generating a smooth volume. A main advantage of the level sets is its ability to provide robust numerical approximations of the finite differences which are required for computing geometric properties such as normal vector $\vec{n} = \frac{\Delta\phi}{|\Delta\phi|}$ and surface curvature $\kappa = \text{div} \frac{\Delta\phi}{|\Delta\phi|}$. In addition, as discussed by Olver et al. [39], even for non-smooth surfaces like the noisy action volumes, the surface evolution based on differential properties is well-posed and stable. In a 3D level set grid, $\phi(\mathbf{x})$, the action volume \mathbf{B} can be represented by the zero level set where each grid position \mathbf{x} encodes the closest distance from the volume surface [38]. The smoothing operation is performed iteratively by evolving the volume by means of a speed function, s , evaluated at each point around the surface based on the grid quantities. For the surface smoothing, we define the evolution speed by the Gaussian curvature, such that the overall curvature is minimized by iteratively evolving the surface using (see Fig. 9 for smoothing iterations):

$$\kappa(x, y, t) = \left| \begin{aligned} &(\phi_{yy}\phi_{tt} - \phi_{yt}^2)\phi_x^2 + (\phi_{xx}\phi_{tt} - \phi_{xt}^2)\phi_y^2 + (\phi_{xx}\phi_{yy} - \phi_{xy}^2)\phi_t^2 \\ &+ 2\phi_x\phi_y(\phi_{xx}\phi_{yt} - \phi_{xy}\phi_{tt}) + 2\phi_y\phi_t(\phi_{xy}\phi_{tt} - \phi_{yt}\phi_{xx}) \\ &+ 2\phi_x\phi_t(\phi_{xy}\phi_{yt} - \phi_{xt}\phi_{yy}) \end{aligned} \right|^{\frac{1}{3}}, \quad (10)$$

where the single subscript for ϕ denotes the first order derivative and the double subscripts denote the second order derivative of ϕ with respect to corresponding variables x , y and t .

In Fig. 10B, we show the smooth volumes obtained by applying the curvature flow given in Eq. (10). After the smoothing step, we compute the differential geometric properties for each sequence, which include the Gaussian and mean curvatures as shown in Fig. 10C.

To quantitatively assess the sensitivity of these differential measures, we define a metric based on the proximity of the surface patches and the similarity of their curvatures evaluated for the entire volume:

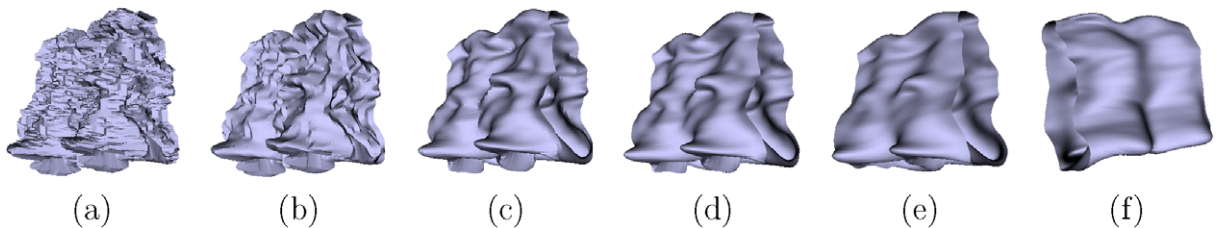


Fig. 9. The smoothing an initial action volume (a) using the level set method. (b–d) The intermediate iterations. (e and f) Two different views of the final smooth volume.

$$s = \sum_{\mathbf{x}_i \in \mathbf{B}_i, \mathbf{x}_j \in \mathbf{B}_j} \sqrt{\frac{e^{-(\kappa_i - \kappa_j)^\top \Sigma_\kappa^{-1} (\kappa_i - \kappa_j)} e^{-(\mathbf{x}_i - \mathbf{x}_j)^\top \Sigma_x^{-1} (\mathbf{x}_i - \mathbf{x}_j)}}{2\pi |\Sigma_\kappa|^{\frac{1}{2}} |\Sigma_x|^{\frac{1}{2}}}}, \quad (11)$$

where \mathbf{B}_i and \mathbf{B}_j denote the two action volumes; the first exponential term measures the curvature similarity and the second exponential term measures the proximity. Rearranging this equation, one can show that the similarity measure given in Eq. (11) is associated with the Bhattacharyya coefficient which is related to the Bayes error under the assumption that the curvatures and the Euclidean distances around each surface point are normally distributed [40]. The similarity results are plotted in Fig. 11 for different actions. In the plots, we have chosen the sequence in Fig. 10a as the reference volume and computed the similarity of this action volume against other walking examples shown in Fig. 10b–g, and two other volumes generated from actions different from the walking action. As shown in the figure, the similarity computed using Eq. (11) between pairs of action volumes (a) as well as between pairs of action sketches (b) are distinctively similar for the same action type and are different for different actions. It is clear to see that the level set smoothing reduces the effect of noise and the extracted action sketches are robust. An important observation at this point is “If both the volume and sketch based measures give similar results, why don’t we simply use the volume alone for action recognition?”. For action sequences captured from the same viewpoint, the volume based measure is adequate, however, since the volumes from different viewpoints are different, it will fail to characterize an action. As discussed in Section 4.1, this is not the case for the action sketch.

3.3. Analysis of the action descriptors

The action sketch encodes both the motion and shape of the performing actor simultaneously. For instance, “closing the fingers while forming a fist” (see Fig. 12a) generates different descriptors compared to “waving a hand” (see Fig. 12b). In the first case, the hand contour changes dramatically giving rise to different surface types; e.g. saddle valleys and pits. In the latter case, the hand shape does not change, however, its motion (change of speed and direction) results in ridges and saddle ridges.

In order to define the relationship between the descriptors in the sketch with the object motion and shape, we

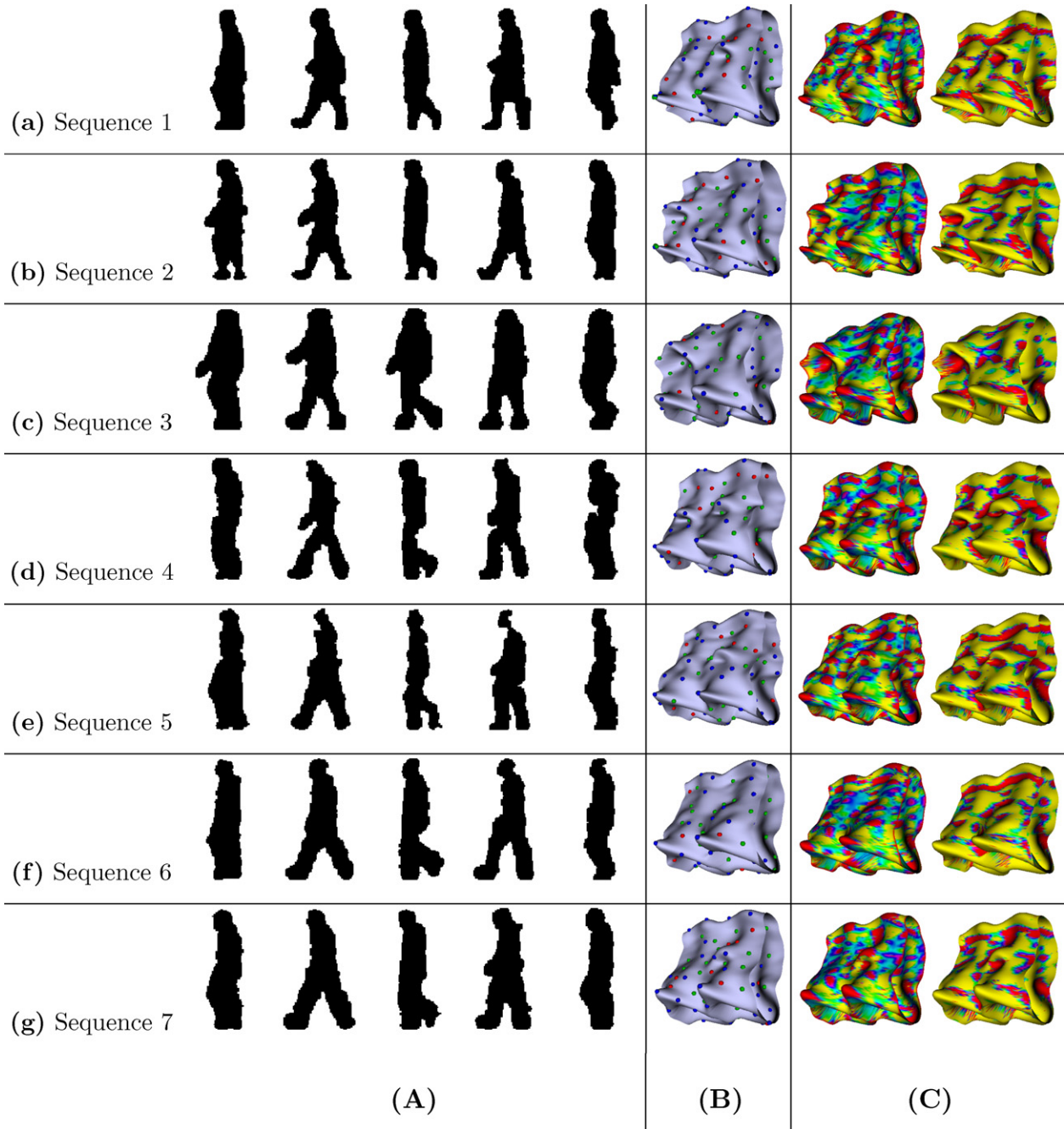


Fig. 10. The noisy silhouettes from the standard HID database [35]. We have selected seven sequences corresponding to different individuals with different durations. (A) The silhouettes of different individuals, (B) associated volume after the level set smoothing, (C) the Gaussian (left) and mean (right) curvatures color coded on the surface (red, high; blue, low). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

consider three basic types of contours: concave, convex and straight. The other contour shapes are a combination of these basic contour types. Depending on the object motion, these contour types may generate the following action descriptors:

- A straight contour generates a ridge, a valley or a flat surface,
- A convex contour generates a peak, a ridge or a saddle ridge,
- A concave contour generates a pit, a valley or a saddle valley.

In order to illustrate the effect of motion on the surface, let us first consider the rigid motion, i.e., there is no shape deformation. In this setting, there are three possibilities:

- *No motion*: The object contour stays stationary,
- *Constant speed*: The object contour moves in one direction with a constant speed,

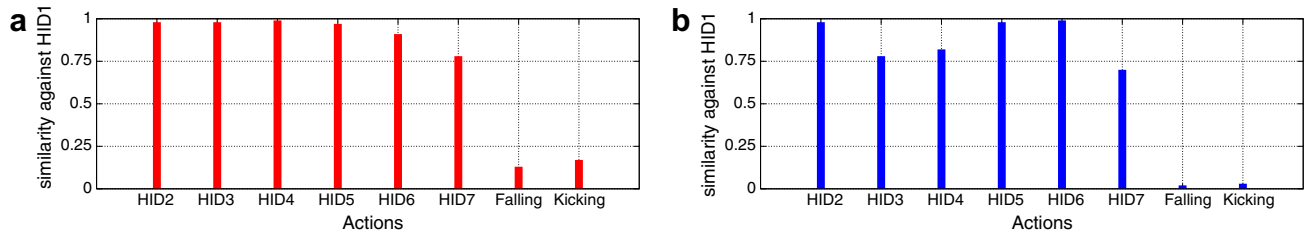


Fig. 11. The similarity of the sequence HID1 shown in Fig. 10a to the other sequences in the same figure using the respective volumes and the action sketches. (a) The similarity is computed using Eq. (11) for the entire volume and (b) the action sketch. Note that both the volume and the sketch effectively match the reference HID1 sequence to other HID sequences.

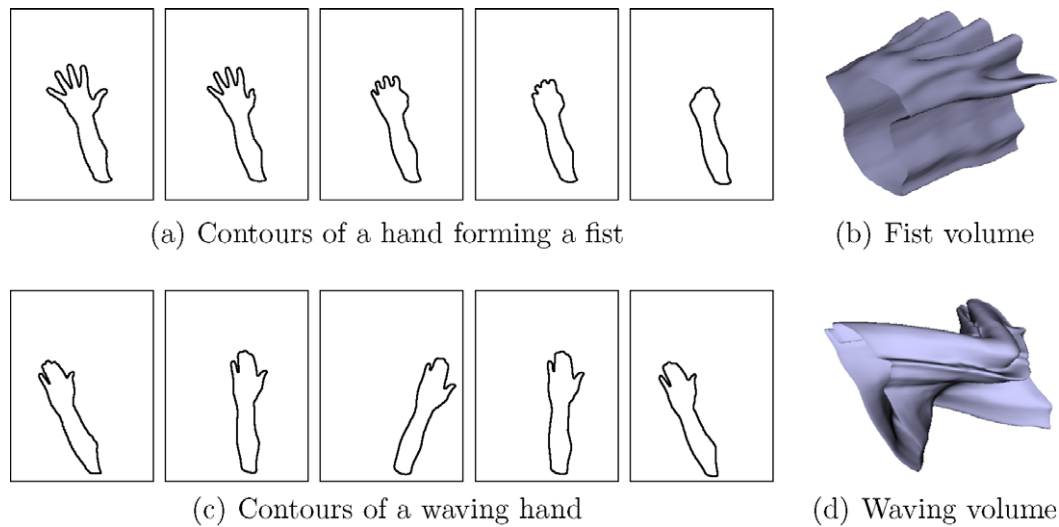


Fig. 12. (a) The contours of a hand forming a fist and (b) the resulting action volume. (c) The contours of a waving hand and (d) the resulting action volume.

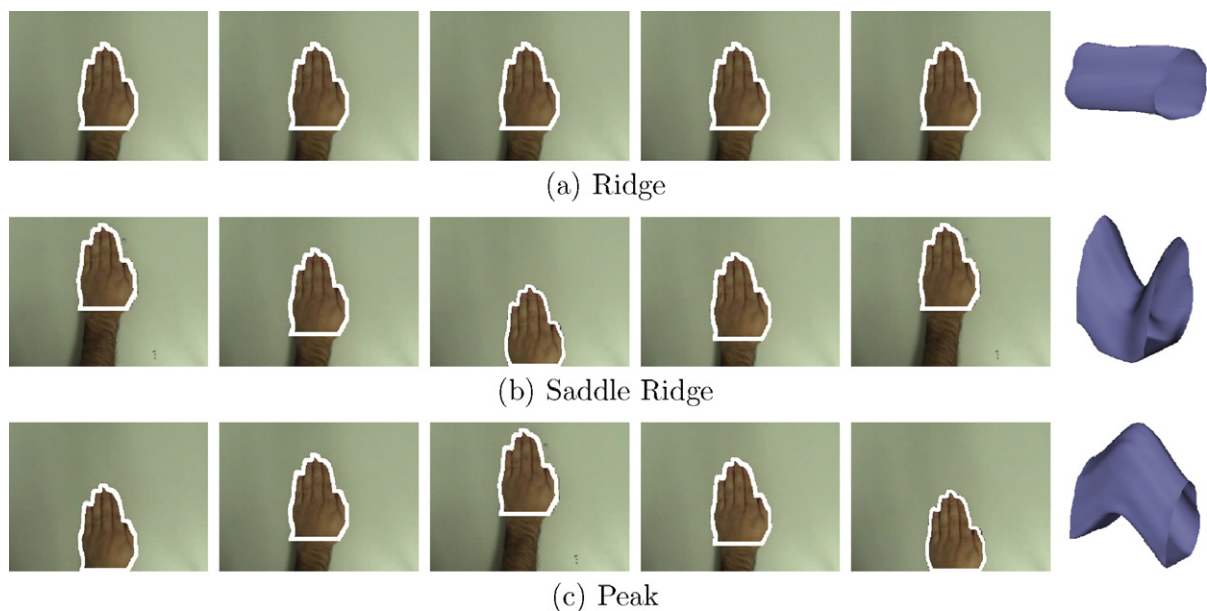


Fig. 13. The motion of a hand resulting in different action descriptors. (a) The hand stays stationary and generates a ridge. (b) The hand decelerates in \downarrow direction and accelerates in \uparrow direction generating a saddle ridge. (c) The hand first decelerates in \uparrow direction and accelerates in \downarrow direction generating a peak.

- *Deceleration and acceleration:* The object contour moves in one direction while decelerating its speed, then comes to a full stop which is followed by an acceleration in the opposite direction.

In Fig. 13, we show the volumes generated from a sequence of hand contours. Note that in this example, only the “concave contour” segment of the hand is used to generate the action descriptors, hence, the resulting descriptors are only ridges, saddle ridges and peaks depending on the direction of the motion. Next, we summarize and give examples for the hand motion that give rise to various action descriptors assuming an affine camera model.

3.3.1. The peak surface

This action descriptor is generated from a sequence of “convex” contours. A typical example of a peak is given in Fig. 13c, where the hand moves first in the direction normal to the contour then stops and moves in the opposite direction.

3.3.2. The pit surface

This is similar to the peak surface, but it is defined for a sequence of “concave” contours. It is generated when the contour first moves in the direction normal to the contour, then stops, and moves in the opposite direction.

3.3.3. The ridge surface

The ridge surface is generated in two different ways based on the motion and shape pair. The first possible way is when a “convex” contour moves in some direction with a constant speed (including no motion case). In Fig. 13a, we give an example of a ridge surface generated from a sequence of hand contours with no motion. The second possible way is when a “straight” contour moves first in some direction, then comes to a stop, and then moves in the opposite direction.

3.3.4. The saddle ridge surface

Similar to the ridge surface, a saddle ridge is generated by the motion of the convex contours. An instance of the saddle ridge is shown in Fig. 13b, where the hand first moves in the direction opposite to the normal of the contour, then comes to a full stop, and moves in the opposite direction.

The discussions for the action descriptors related to the ridge and saddle ridge can be extended to the valley and saddle valley. The difference between the two is that the contour is concave for the latter. The strength of the contour concavity or convexity and the magnitude of the contour motion is encoded by the values of the Gaussian and mean curvatures. For the peak and pit surfaces, the mean curvature encodes the shape of the object (concave: $H < 0$, convex $H > 0$) and the Gaussian curvature controls the bending of the temporal surface in the time direction, such that when $K > 0$, the object moves in the normal direction of the contour while for $K < 0$ it moves in the opposite direction to the contour normal. Similar arguments hold for the action descriptors defined by the saddle valley and the saddle ridge surfaces. However, for the valley and the ridge surfaces, the object shape and motion can be encoded by either the mean or the Gaussian curvature. Depending on the type of surface, the curvatures can also be used to compute the motion direction and speed at any contour point.

4. Recognizing the actions from different viewpoints

Action recognition using the proposed representation can be considered as a 3D object recognition but in the spatiotemporal space. In this line of thought, we first discuss how a change in the viewpoint affect the proposed representation, and then we sketch a matching strategy based on the scene geometry for computing a similarity score between two different actions.

4.1. On the effect of viewpoint change

In our representation, the effect of the viewpoint is directly related to the building blocks of the action volume: the object contours and the trajectories of the points on the contour. For each action descriptor in the action sketch (the minima or maxima of K and H on the action volume), the underlying curve defined by the *object contour* and the *point trajectories* will have a concavity or a convexity (see Table 2).

In this section, we seek an answer to the following question: “What happens to the minima and the maxima of contour and trajectory curvatures when the same action is viewed from another viewpoint?” We first discuss the

Table 2

Various surface types and their relations to the curvature of the trajectory and the contour

	Peak	Pit	Valley	Saddle valley
Contour curvature	Maximum	Minimum	Maximum	Maximum
Trajectory curvature	Maximum	Minimum	Zero	Minimum

effect of viewpoint on the contour, and then extend it to the trajectory.

The object contour Γ at time t is parameterized by its arc-length (Eq. (4)). For the two-dimensional spatial contour, the Gaussian and mean curvatures simplify to a single 2D curve curvature:

$$\kappa = \frac{x'y'' - y'x''}{\sqrt{x^2 + y^2}^3} = \frac{\mathbf{x}'^T B \mathbf{x}''}{(\mathbf{x}'^T \mathbf{x}')^{3/2}}, \quad (12)$$

where $B = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, $x = f(s)$, $y = g(s)$, $x' = \frac{\partial f}{\partial s}$, $y' = \frac{\partial g}{\partial s}$ and s is the contour arc-length.

Let the action be viewed by two different cameras, such that we have two views of the contour, Γ_L and Γ_R , at time t . Using the affine camera model, the world coordinates \mathbf{X} are projected to the image coordinates \mathbf{x} by $\mathbf{x}_L = A\mathbf{X} + T_L$ and $\mathbf{x}_R = C\mathbf{X} + T_R$, where the subscripts denote the left and right cameras [41]. Thus, the contour curvatures in 2D are related to the world coordinates by:

$$\kappa_L = \frac{\mathbf{X}'^T A^T B A \mathbf{X}''}{(\mathbf{X}'^T A^T A \mathbf{X}')^{3/2}}, \quad (13)$$

$$\kappa_R = \frac{\mathbf{X}'^T C^T B C \mathbf{X}''}{(\mathbf{X}'^T C^T C \mathbf{X}')^{3/2}}. \quad (14)$$

In these equations, $A^T B A = |A|B$ and $C^T B C = |C|B$, where $| \cdot |$ is the determinant. Dividing Eq. (13) by Eq. (14) we have:

$$\frac{\kappa_L}{\kappa_R} = \frac{|A|}{|C|} \frac{(\mathbf{X}'^T C^T C \mathbf{X}')^{3/2}}{(\mathbf{X}'^T A^T A \mathbf{X}')^{3/2}}. \quad (15)$$

Let $\alpha = |C|/|A|$, and by converting \mathbf{X} to the left image coordinates, the relation between the curvature of the contours in the left and the right images is given by:

$$\kappa_R = \alpha \left(\frac{\mathbf{X}'_L{}^T \mathbf{X}'_L}{\mathbf{X}'_L{}^T D \mathbf{X}'_L} \right)^{3/2} \kappa_L, \quad (16)$$

where $D = A^{-1T} C^T C A^{-1}$. This relation shows that the curvature of a point on the right contour, Γ_R , is directly proportional to the curvature on the left contour, Γ_L . Due to this relation, *the minima and maxima of the curvature of Γ_L are still the curvature minima and maxima of Γ_R* . In Fig. 14, we show the same object contour from various viewing directions along with several of the corresponding curvature maxima and minima in each view.

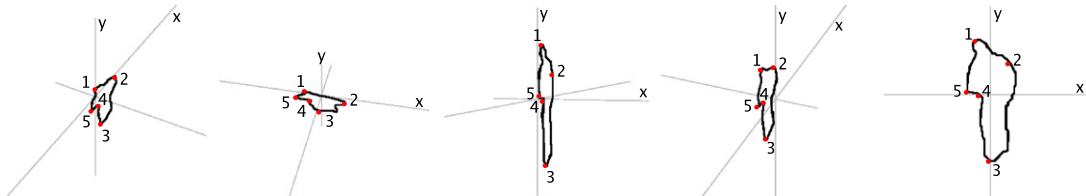


Fig. 14. The projections of the 3D contour on the image plane using the affine camera model from various viewing angles. Different numbers on the contours denote the corresponding minima and maxima points.

The effect of the viewpoint changes to the 2D trajectory of a point has been previously discussed in the context of action recognition [4], and it was shown that the minima and the maxima of the curvature is invariant to the viewing angle. Briefly, trajectory of a point can be viewed as a 3D object in the spatiotemporal space, hence, two views of the this trajectory are related to each other by means of a linear transformation for an affine camera model as discussed above. This relation results in the observation that the minima and maxima of the spatiotemporal curvature on both views are preserved.

4.2. Discussion

Observing the same curvature minima and maxima of the contour and the trajectory from different viewpoints does not apply in cases of an accidental alignment. The accidental alignment happens when a point on a contour moves perpendicular to the viewpoint, such that its trajectory is mapped to a single point in the image plane. This condition may also occur when a corner or a curvature maxima on the contour is mapped to a non-corner point in the image plane.

Due to the articulated motion of the body parts, it may happen that some parts of the contour related to action descriptors get occluded (self occlusion) in a particular view. Hence, the action descriptors corresponding to the occluded parts may not be visible. Thus, it is only meaningful to discuss the existence of the curvature minima and maxima for the parts of the contours which are visible in both views. In Fig. 15a–e, we illustrate this with an example where a walking person is viewed from five different viewing angles. In several views, the right arm is occluded and results in missing action descriptors. However, the remaining object parts are available in all the views and provide adequate number of action descriptors to uniquely represent the action.

4.3. Recognizing actions

Two video sequences of the same action can be considered as two views of the same scene, such that there exists a geometric relation between the two views. Following this observation, we use the epipolar geometry for action recognition. The epipolar geometry inherently provides invariance to viewpoint changes for action recognition [18,42–44]. Under the epipolar geometry constraints, two different

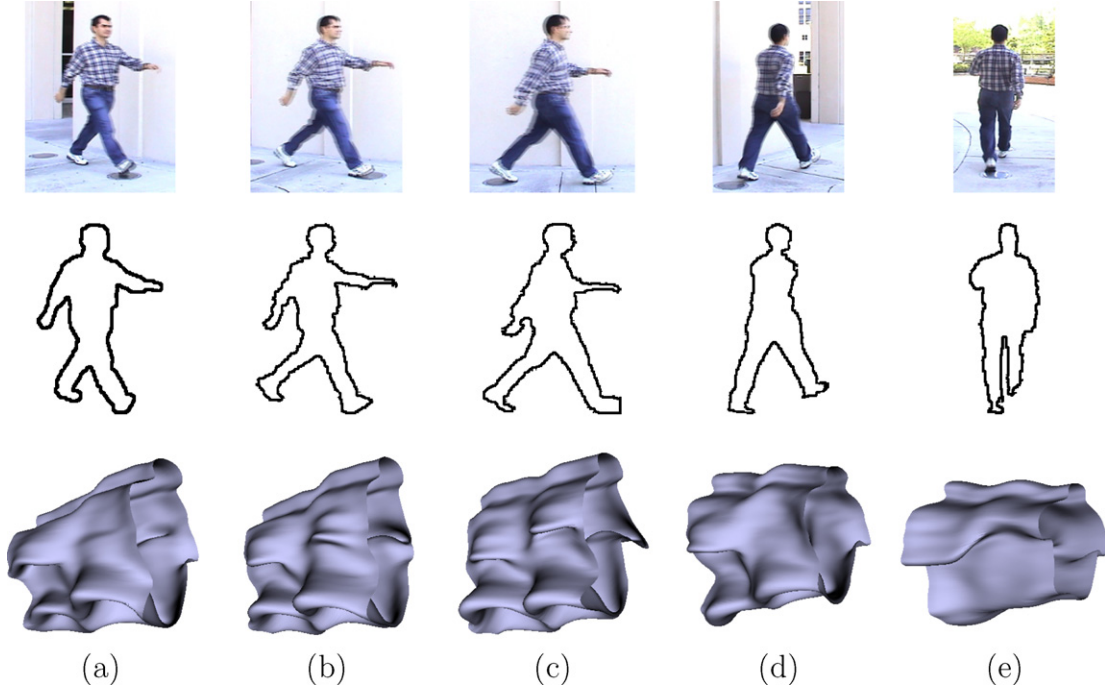


Fig. 15. The walking action captured from five different viewpoints: the first row shows one frame from each viewpoint, the second row shows associated contours and the third row shows the corresponding action volumes. (a) 30°, (b) 60°, (c) 90°, (d) 145°, and (e) 180°.

views of an action are related to each other by the fundamental matrix:

$$\begin{pmatrix} x_{\mathbf{B}} & y_{\mathbf{B}} & 1 \end{pmatrix} \mathcal{F} \begin{pmatrix} x_{\mathbf{D}_i} \\ y_{\mathbf{D}_i} \\ 1 \end{pmatrix} = 0, \quad (17)$$

where, in our context, the points $X_{\mathbf{B}}$ and $X_{\mathbf{D}_i}$ correspond to the location of the descriptors in the action sketch. The fundamental matrix maps the points $X_{\mathbf{D}_i}$ in one view to the epipolar lines $U_{\mathbf{B}} = \mathcal{F}X_{\mathbf{D}_i}$ in the other view, such that the matching point $X_{\mathbf{B}}$ lies on the epipolar line $U_{\mathbf{B}}$. In order to recover the geometry between two actions, the correspondences between two action sketches need to be established. In the context of actions, Syeda-Mahmood et al. [18] use the shape properties of a manually chosen feature point on the action cylinder to search for corresponding feature points. However, use of the shape alone may create ambiguity due to non-rigid object motion. In this paper, we formulate the problem of establishing the correspondences between two action sketches as a graph theoretic problem. In this formalism, the vertices of the graph are the action descriptors, V , from the action sketches corresponding to action volumes \mathbf{B} and \mathbf{D}_i , where $\alpha = \{\mathbf{B}, \mathbf{D}_i\}$. The edges go from one partite set \mathbf{B} to the other partite set \mathbf{D}_i . The weight of each edge is based on the convex combination of the spatiotemporal proximity, $d_i(k, l)$ and the geometric similarity, $g_i(k, l)$:

$$w_i(k, l) = \eta d_i(k, l) + (1 - \eta)g_i(k, l). \quad (18)$$

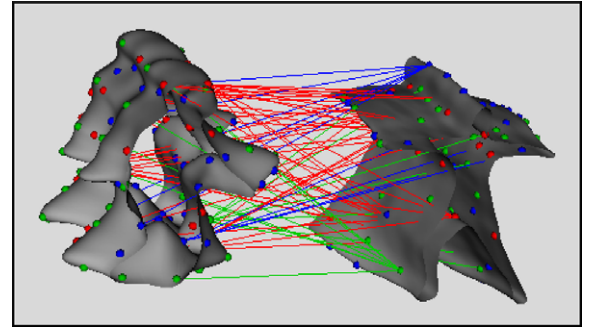


Fig. 16. A subset of possible associations (peak with peak, valley with valley etc.) of the action descriptors between the walking and the exercise actions are shown.

The spatiotemporal proximity between the k th and l th action descriptor that belong to the action sketches, \mathbf{B} and \mathbf{D}_i , is defined by the following measure:

$$d_i(k, l) = e^{-\frac{\|X_k - X_l V_{\mathbf{D}_i}(s_i, t_i)\|^2}{\sigma_d^2}}. \quad (19)$$

This measure assures that the action descriptors are not distant from one another. The similarity of the underlying geometry between the action descriptors is defined in terms of differential geometric features which measure the concavity and convexity of the local surface around the descriptor. The differential geometric features we used for this purpose are the Gaussian and mean curvatures:

$$g_i(k, l) = \sqrt{e^{-\frac{-(K(k) - K_l(l))^2}{\sigma_K^2}} e^{-\frac{-(H(k) - H_l(l))^2}{\sigma_H^2}}}}, \quad (20)$$

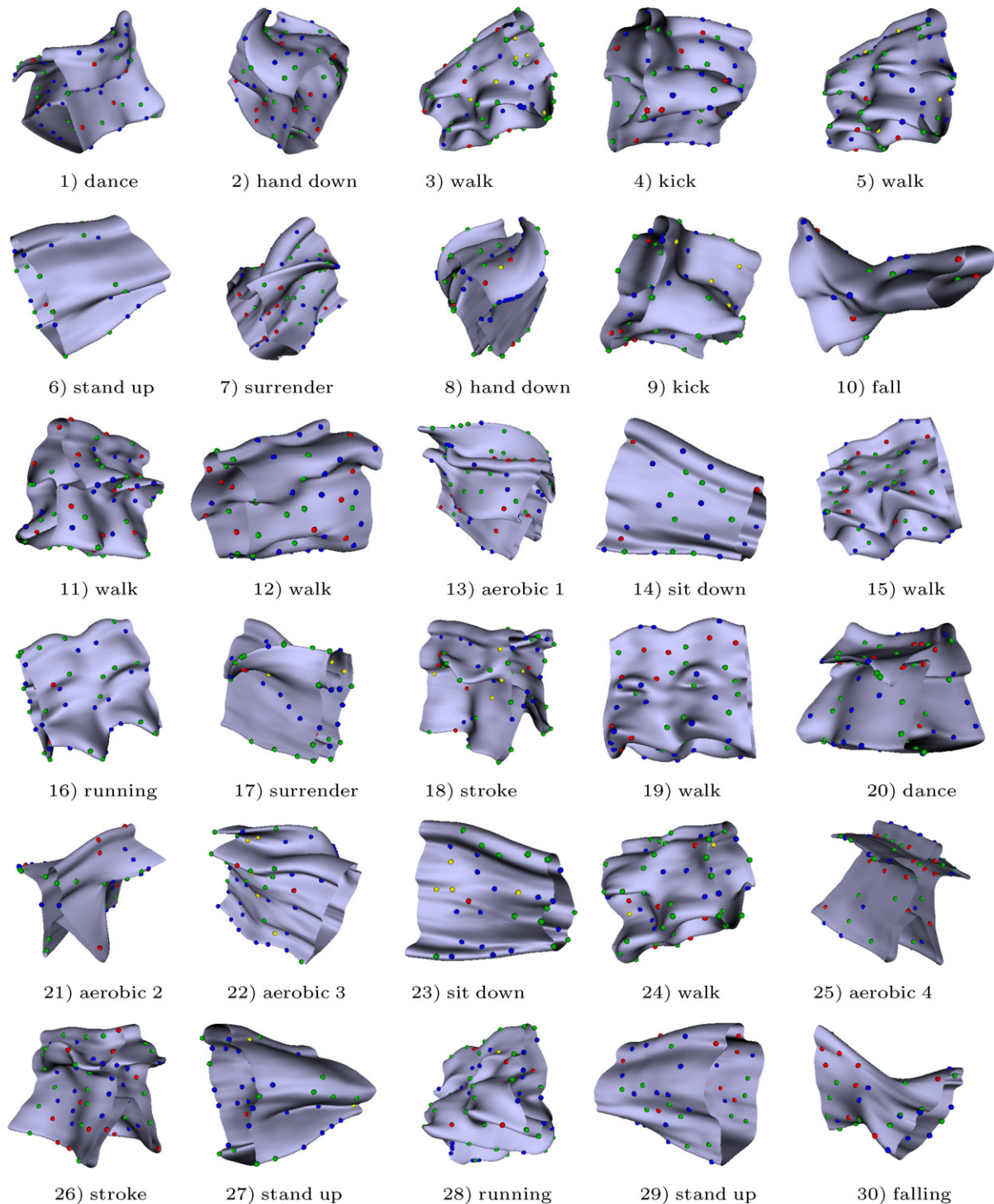


Fig. 17. The action sketches superimposed on the action volumes generated from the video sequences of human actors. The numbers denote the action labels used in Table 3.

where the square root provides the geometric mean of these quantities.² Use of these differential features guarantees not only the similarity of the underlying object contours but

also the similarity of speed and acceleration of the action descriptor. For instance, the peaks will be associated with the peaks, and the valleys will be associated with the valleys, etc.

The maximum matching of the action graph provides only 1-1 correspondences between the action descriptors. In some occasions, some action sketches in one partite

² In our experiments, we have chosen $\sigma_K = 10$, $\sigma_H = 10$ and $\eta = .3$ to emphasize the geometric similarity.

set may not have any correspondence in the other due to self occlusions. In Fig. 16, we show a set of possible matchings between the walking and the exercise sketches whose weights are above some confidence level.

Given the correspondences between the action descriptors, the similarity between two actions can be computed algebraically or geometrically using the relation given in Eq. (17). The geometric score is based on the Euclidean distances of both X_B to U_B and X_{D_i} to U_{D_i} , and measures the quality of the recovered geometry. In contrast, the algebraic matching score is related to the quality of the homogeneous system of equations which are used to compute the fundamental matrix using the least squares fit $(A^T A)f = 0$, where

$$A = [x_{D_i}, x_B, y_{D_i}, x_B, x_B, x_{D_i}, y_B, y_{D_i}, y_B, x_{D_i}, y_{D_i}, 1],$$

$$f = [\mathcal{F}_{1,1}, \mathcal{F}_{1,2}, \mathcal{F}_{1,3}, \mathcal{F}_{2,1}, \mathcal{F}_{2,2}, \mathcal{F}_{2,3}, \mathcal{F}_{3,1}, \mathcal{F}_{3,2}, \mathcal{F}_{3,3}].$$

The details on how to construct A can be found in Hartley's paper [45]. Typically, due to the homogeneous system of equations, the rank of $(A^T A)$ is eight unless all the points (action descriptors) lie on one plane when the rank is less than eight (degenerate cases). Following the results given in [44], we compute the symmetric epipolar distances. The symmetric epipolar distance measures the average distance of each feature point in the left camera view (one action view) to the epipolar line generated from the corresponding feature point in the right (other action view) camera view using the estimated fundamental matrix:

$$d(X_{D_i}, X_B) = \sqrt{\left(\frac{X_{D_i}^T U_B}{|U_B|}\right)^2 + \left(\frac{X_B^T U_{D_i}}{|U_{D_i}|}\right)^2}, \quad (21)$$

where $|\cdot|$ denotes norm 2. Once Eq. (21) is evaluated for all the matching points, a matching score between actions B and D_i is computed by:

$$s(B, D_i) = \frac{\sum_{N_i} d(X_{D_i}, X_B)}{N_i}, \quad (22)$$

where N_i denotes the number of matching points. Normalization by N_i in our context is required due to varying number of feature point correspondences between the actions. Finally, the action which gives the minimum geometric error $\text{argmax}_i s(B, D_i)$ is declared as a match.

5. Experiments

In order to test the performance of the proposed approach, we collected a set of thirty action sequences captured from different viewpoints. Each action class has at least two or more samples performed by different male and female actors. The video sequences include dancing (two sequences, one actor), falling (two sequence, two actors), tennis strokes (two sequences, one actor), walking (seven sequences, three actors), running (two sequences, two actors), kicking (two sequences, two actors), sitting-down (two sequences, one actor), standing-up (three

sequences, two actors), surrender (two sequences, two actors), hands-down (two sequences, two actors), aerobics (four sequences, two actors) actions.

From the input video, we first track the contours of the objects using [22], and generate an action volume (Section 2). For each action volume, the level set based smoothing is performed to compute reliable differential geometric features. The collection of these features provides the action sketch (Section 3). In Fig. 17, we show the complete set of action sketches superimposed on the action volumes. To analyze the recognition performance, we chose ten sequences representing different actions (model set) and used the remaining sequences as the test set. In order to compute a matching action for all the action sequences, we change the model set dynamically. For instance, for the two kicking action sequences K_1 and K_2 , we generate two different model sets. The first model set includes K_1 whereas the second includes K_2 . These selections test the matching of K_1 against the model set K_2 and vice versa. Once the model set is generated, an input action is matched against the actions in the model set by computing the distance measure discussed in Section 4.

In Table 3, we summarize the matching results by tabulating the matching action for each action video. Each row

Table 3
The recognition results for various actions

	#	Matching action	#
Input action			
Dance	1	<i>Dance</i>	20
Hand down	2	Stand up	29
Walking	3	<i>Walking</i>	11
Kicking	4	<i>Kicking</i>	9
Walking	5	<i>Walking</i>	11
Stand up	6	<i>Stand up</i>	29
Surrender	7	<i>Surrender</i>	17
Hands down	8	<i>Hands down</i>	2
Kicking	9	<i>Kicking</i>	4
Falling	10	<i>Falling</i>	30
Walking	11	<i>Walking</i>	11
Walking	12	Sit down	23
Sit down	14	<i>Sit down</i>	23
Walking	15	<i>Walking</i>	11
Running	16	<i>Running</i>	28
Surrender	17	<i>Surrender</i>	17
Tennis stroke	18	<i>Tennis stroke</i>	26
Walking	19	<i>Walking</i>	11
Dance	20	<i>Dance</i>	1
Sit down	23	<i>Sit down</i>	23
Walking	24	<i>Walking</i>	11
Tennis stroke	26	<i>Tennis stroke</i>	18
Stand up	27	<i>Stand up</i>	29
Running	28	<i>Running</i>	16
Stand up	29	Hands down	8
Falling	30	<i>Falling</i>	10

The italics are used to represent the correct matches and boldface is used to represent the incorrect matches. Each row consisted of the input action and the matching action along with their index to the volumes given in Fig. 17.

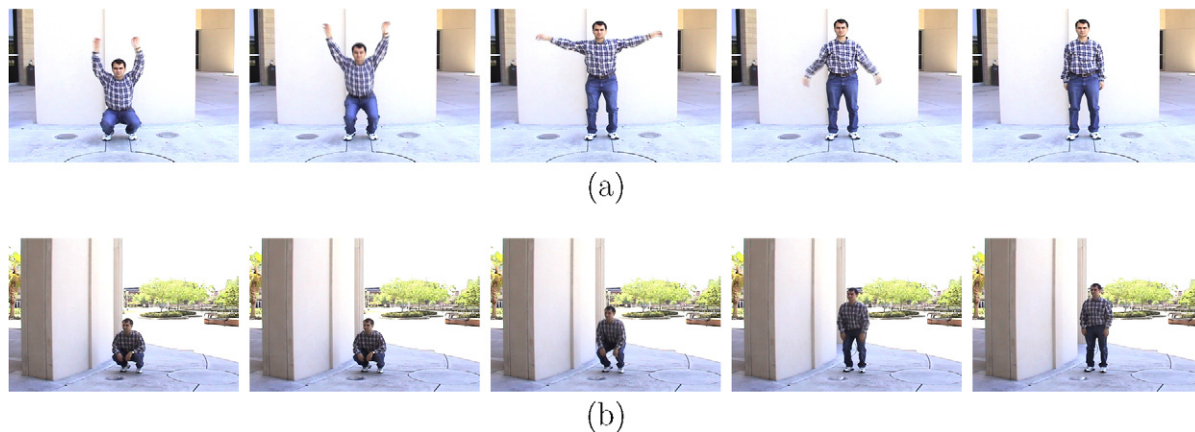


Fig. 18. Selected frames from some of the misclassified actions which appear similar. (a) The hands down action, and (b) the stand up action. In particular an instance of the hands down action in (a) is recognized as (b), and an instance of the stand up action (b) is recognized as (a).

of the table shows the input action, its corresponding index in Fig. 17 and the matching action from the database and its index in Fig. 17. As seen from the table, except for three actions which are printed in boldface, all matches are correct. The incorrect matches include one instance of hands-down, walking and stand-up actions. These incorrect matches are mainly due to the similarity of the input actions to the incorrectly matched actions or due to the confusion of the input actions with the model actions in the database. In particular, the hands down action involves the “standing up” action, and additionally there are other action descriptors that originate from the motion of both hands (see Fig. 18a and b). The walking action is captured from 180° and is confused with the model walking action which is fronto-parallel.

We want to emphasize the importance of using both the shape and the motion of the object. This is evident in the hands down action. For instance, if we were modeling the trajectory of only one hand, we would end-up with a line shaped trajectory which is not a characteristic of the action. However, the shape variation around the knees (due to standing up) helps to identify the action.

5.1. A discussion on the implementation

The extraction of the proposed representation is performed in three steps. First, we associate the points on the consecutive object contours, which is followed by a level set smoothing step. The resulting correspondences, once smoothed out, generate piecewise continuous surface patches, which are used to analyze the differential surface geometry. Due to one step being an input to the following step, our current implementation of the system requires offline processing of stored action clips. This requirement results in an increased computational complexity, hence, a poor realtime realization for applications such as realtime surveillance. However, the locality of the establishing correspondences and the level set smoothing can be exploited to implement an online version of the method

which can run in parallel to the object tracking. In particular, starting from the first few instances of tracked object contours or silhouettes that belong to a long action clip, one can associate control points on consecutive contours and perform piecewise smoothing of the resulting correspondences using the level set approach. The analysis of geometric surface features can be performed once the piecewise action volume is available. This speed-up will result in recognition of an action delayed only by the duration of the action.

6. Conclusions

The recognition of human actions requires detecting features corresponding to important changes in the actor shape and motion. In order to facilitate simultaneous use of both quantities, we propose to analyze the action as a 3D object in the spatiotemporal space. The proposed approach considers the action as a 3D volume, which is generated by stacking a sequence of tracked 2D object silhouettes or contours. Given an action volume, the features representing the action are extracted by analyzing the differential geometric properties computed from the volume surface. The differential geometric surface properties reveal the type of motion a body part is going through as well as the related non-rigid shape deformation due to that motion. A collection of the differential geometric properties constitute the action sketch, which is robust to changes in the camera viewpoint. Finally, two action sketches are matched against each other using the geometry between the views. The experiments performed on thirty videos demonstrate the versatility of the proposed representation.

References

- [1] J. Siskind, Q. Moris, A maximum likelihood approach to visual event classification, in: European Conf. on Computer Vision, 1996, pp. 347–360.
- [2] D. Ayers, M. Shah, Monitoring human behavior from video taken in an office environment, *J. Image Vis. Comput.* 19 (12) (2001) 833–846.

- [3] F. Quek, T. Mysliwiec, M. Zhao, *Fingermouse: A freehand pointing interface*, in: *IEEE Conf. on Computer Vision and Pattern Recognition*, 1993.
- [4] C. Rao, A. Yilmaz, M. Shah, *View invariant representation and recognition of actions*, *Int. J. Comput. Vis.* 50 (2) (2002) 203–226.
- [5] A. Gritai, Y. Sheikh, M. Shah, *On the invariant analysis of human actions*, in: *IAPR Int. Conf. on Pattern Recognition*, 2004.
- [6] R. Polana, R. Nelson, *Recognizing activities*, in: *IAPR Int. Conf. on Pattern Recognition*, 1994.
- [7] A. Efros, A. Berg, G. Mori, J. Malik, *Recognizing action at a distance*, in: *IEEE Int. Conf. on Computer Vision*, 2003.
- [8] M. Black, Y. Yacoob, *Recognizing facial expressions in image sequences using local parameterized models of image motion*, *Int. J. Comput. Vis.* 25 (1) (1997) 23–48.
- [9] M. Yang, N. Ahuja, M. Tabb, *Extraction of 2d motion trajectories and its application to hand gesture recognition*, *IEEE Trans. Pattern Anal. Mach. Intel.* 24 (8) (2002) 1061–1074.
- [10] L. Zelnik-Manor, M. Irani, *Event-based analysis of video*, in: *IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.
- [11] E.H. Adelson, J.R. Bergen, *Spatiotemporal energy models for the perception of motion*, *J. Opt. Soc. Am. A2* (1985) 284–299.
- [12] P. Hsu, H. Harashima, *Spatiotemporal representation of dynamic objects*, in: *IEEE Conf. on Computer Vision and Pattern Recognition*, 1993, pp. 14–19.
- [13] A. Bobick, J. Davis, *The representation and recognition of action using temporal templates*, *IEEE Trans. Pattern Anal. Mach. Intel.* 23 (3) (2001) 257–267.
- [14] T. Starner, A. Pentland, *Real-time american sign language recognition from video using hidden markov models*, in: *Motion-Based Recognition*, Kluwer, Dordrecht, 1996.
- [15] R. Cutler, L. Davis, *Robust real-time periodic motion detection, analysis, and applications*, *IEEE Trans. Pattern Anal. Mach. Intel.* 22 (8) (2000) 781–796.
- [16] G. Mori, S. Belongie, J. Malik, *Shape contexts enable efficient retrieval of similar shapes*, in: *IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.
- [17] J. Sullivan, S. Carlsson, *Recognizing and tracking human action*, in: *European Conf. on Computer Vision*, 2002.
- [18] T. Syeda-Mahmood, A. Vasilescu, S. Sethi, *Recognizing action events from multiple viewpoints*, in: *IEEE Workshop on Detection and Recognition of Events in Video*, 2001.
- [19] A. Ali, J. Aggarwal, *Segmentation and recognition of continuous human activity*, in: *IEEE Workshop on Detection and Recognition of Events in Video*, 2001, pp. 28–35.
- [20] S. Niyogi, E. Adelson, *Analyzing gait with spatiotemporal surfaces*, in: *Wrks. on Nonrigid and Artic. Motion*, 1994.
- [21] C. Veenman, M. Reinders, E. Backer, *Resolving motion correspondence for densely moving points*, *IEEE Trans. Pattern Anal. Mach. Intel.* 23 (1) (2001) 54–72.
- [22] A. Yilmaz, X. Li, M. Shah, *Contour based object tracking with occlusion handling in video acquired using mobile cameras*, *IEEE Trans. Pattern Anal. Mach. Intel.* 26 (11) (2004) 1531–1536.
- [23] W. E. Lorensen, H. E. Cline, *Marching cubes: A high resolution 3d surface reconstruction algorithm*, in: *ACM Sigrph Conf.*, 1987.
- [24] N. Amenta, S. Choi, T.K. Dey, N. Leekha, *A simple algorithm for homeomorphic surface reconstruction*, *Int. J. Comput. Geom. Appl.* 12 (2002) 125–141.
- [25] T. Dey, J. Giesen, J. Hudson, *Delaunay based shape reconstruction from large data*, in: *IEEE Symp. in Parallel and Large Data Visualization and Graphics*, 2001, pp. 19–27.
- [26] P. Besl, N. McKay, *A registration of 3d shapes*, *IEEE Trans. Pattern Anal. Mach. Intel.* 14 (2) (1992) 239–256.
- [27] H. Chui, A. Rangarajan, *A new algorithm for non-rigid point matching*, in: *IEEE Conf. on Computer Vision and Pattern Recognition*, 2000.
- [28] L. Shapiro, R. Haralick, *Structural descriptions and inexact matching*, *IEEE Trans. Pattern Anal. Mach. Intel.* 3 (9) (1981) 504–519.
- [29] R.C. Bolles, H.H. Baker, *Epipolar-plane image analysis: a technique for analyzing motion sequences*, in: *Readings in computer vision: issues, problems, principles, and paradigms*, Morgan Kaufmann Publishers, 1987, pp. 26–36.
- [30] H. Kuhn, *The hungarian method for solving the assignment problem*, *Naval Res. Logistics Quart.* 2 (1955) 83–97.
- [31] J. Zacks, B. Tversky, *Event structure in perception and cognition*, *Psychol. Bull.* 127 (1) (2001) 3–21.
- [32] E. Weisstein, *CRC Concise Encyclopedia of Mathematics*, second ed., CRC Press, Boca Raton, FL, 2002.
- [33] R. Jain, A. Jain, *Analysis and Interpretation of Range Images*, Springer-Verlag, Berlin, 1990.
- [34] P. Besl, R. Jain, *Invariant surface characteristics for three dimensional object recognition in range images*, *J. Graph. Model. Image Process.* 33 (1) (1986) 33–88.
- [35] P. Phillips, S. Sarkar, I. Robledo, P. Grother, K. Bowyer, *The gait identification challenge problem: data sets and baseline algorithm*, in: *IAPR Int. Conf. on Pattern Recognition*, 2002.
- [36] W. Li, S. Xu, G. Zhao, C. Wen, *Feature-preserving smoothing algorithm for polygons and meshes*, in: *ACM SIGGRAPH*, 2004.
- [37] G. Hu, Q. Peng, A. Forrest, *Robust mesh smoothing*, *J. Comput. Sci. Technol.* 19 (4) (2004) 521–528.
- [38] J. Sethian, *Level set methods: evolving interfaces in geometry, fluid mechanics computer vision and material sciences*, Cambridge University Press, Cambridge, MA, 1999.
- [39] P. Olver, G. Sapiro, A. Tannenbaum, *Affine invariant detection: edge maps, anisotropic diffusion, and active contours*, *Acta Appl. Math.* 59 (1999) 45–77.
- [40] T. Kailath, *The divergence and Bhattacharyya distance measures in signal selection*, *IEEE Trans. Commun. Technol.* 15 (1967) 52–60.
- [41] J. Mundy, A. Zisserman, *Geometric Invariance in Computer Vision*, MIT Press, Cambridge, MA, 1992.
- [42] C. Rao, A. Gritai, M. Shah, T. Syeda-Mahmood, *View-invariant alignment and matching of video sequences*, in: *IEEE Int. Conf. on Computer Vision*, 2003.
- [43] A. Yilmaz, M. Shah, *Action sketch: a novel action representation*, in: *IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.
- [44] A. Yilmaz, M. Shah, *Recognizing human actions in videos acquired by uncalibrated moving cameras*, in: *IEEE Int. Conf. on Computer Vision*, 2005.
- [45] R. Hartley, *In defence of the 8-point algorithm*, in: *IEEE Conf. on Computer Vision and Pattern Recognition*, 1995, pp. 1064–1070.