

Actions As Objects: A Novel Action Representation

Alper Yilmaz

Mubarak Shah

University of Central Florida

Orlando, FL-32828, USA

Abstract

In this paper, we propose to model an action based on both the shape and the motion of the object performing the action. When the object performs an action in 3D, the points on the outer boundary of the object are projected as 2D (x, y) contour in the image plane. A sequence of such 2D contours with respect to time generates a spatiotemporal volume (STV) in (x, y, t) , which can be treated as 3D object in the (x, y, t) space. We analyze STV by using the differential geometric surface properties, such as peaks, pits, valleys and ridges, which are important action descriptors capturing both spatial and temporal properties. A set of motion descriptors for a given is called an action sketch. The action descriptors are related to various types of motions and object deformations. The first step in our approach is to generate STV by solving the point correspondence problem between consecutive frames. The correspondences are determined using a two-step graph theoretical approach. After the STV is generated, actions descriptors are computed by analyzing the differential geometric properties of STV. Finally, using these descriptors, we perform action recognition, which is also formulated as graph theoretical problem. Several experimental results are presented to demonstrate our approach.

1 Introduction

Recognizing human actions and events from video sequences is very active in Computer Vision. During the last few years, several different approaches have been proposed for detection, representation and recognition, and understanding video events. Some popular approaches for action recognition include Hidden Markov Models [1], Finite State Machines [2], neural networks and Context Free Grammars. The important question in action recognition is which features should be used? Therefore, the first step in action recognition is to extract useful information from raw video data to be employed in different recognition models. A common approach for extracting relevant information from video is visual tracking. Tracking can be performed by using only a single point on the object. Single point tracking generates a motion trajectory, and there are several approaches employing motion trajectories for action recognition [3]. It is common to use

changes in speed, direction, or maxima in the spatio-temporal curvature of a trajectory to represent important events in an action. However, a single point trajectory only carries motion information. It does not carry any shape or relative spatial information, which may be useful in action recognition. Some other approaches either track multiple points on the object, or track a bounding box enclosing the complete object, which provides some shape information [4]. The bounding boxes are suitable for representing the shape information of rigid or semi-rigid objects, but they are approximations, since not all object shapes can be accurately described by bounding boxes. Complete shape information can be captured by tracking the object contours. Given the object contours in different frames, the crucial issue is how to compute the relevant features to be employed in action recognition. That is, should the features be based on only shape and appearance, or only motion or both. Due to the non-rigid nature of human motion, there is no one to one correspondence between contours in different frames. Most approaches use features computed in individual frames. In [5], Laptev and Lindeberg extended the 2D Harris detector to (x, y, t) and find temporal interest points. Their approach is purely based on the observed intensities, such that intensity changes that do not belong to the object signify false action characteristics.

In this paper, we propose to use spatiotemporal features in order to simultaneously exploit both shape and motion features. When the object performs an action in 3D, the points on the outer boundary of the object are projected as 2D $((x, y))$ contour in the image plane. A sequence of such 2D contours with respect to time generates a spatiotemporal volume (STV) in (x, y, t) . This volume can be treated as 3D object in the (x, y, t) space. This STV can be analyzed by using the differential geometric surface properties, such as peaks, pits, valleys and ridges, which are important action descriptors capturing both spatial and temporal properties. A set of motion descriptors for a given is called *action sketch*. The action descriptors are related to various types of motions and object deformations.

STV has several advantages. First, it captures both spatial and temporal information of an action in one unified manner. Second, since this is a continuous representation, two sequences of the same action but of different lengths will generate the same STV, therefore, there is no need of time

warping. Third, the descriptors in action sketch are either related to the convex and concave parts of the object contours and or to the maxima in the spatiotemporal curvature of a trajectory, which are view invariant, therefore the action sketch is also view invariant.

We assume that the tracking problem has already been solved, and we are given a sequence of object contours. The first step in our approach is to generate the STV, which is achieved by solving the correspondence problem between the contours in the consecutive frames. Generating a volume from a set of images has been previously considered in [6] for walking persons. Their method fits a “manually generated” walking volume, which consists of two surfaces (right and left body parts), to walking sequence. In their approach, volume fitting involved a large number of intricate steps, and fronto-parallel motion assumption. Here, we propose to “automatically generate” the volume for an action viewed from any viewing direction using a graph theoretic approach. The main contribution of our paper is analysis of differential geometric surface properties of this volume using the Weingarten mapping to determine the action descriptors to be used in action sketch, and the use of such descriptors in action recognition, which is also formulated as a graph theoretical problem.

The organization of the paper is as follows. In the next section, we describe how to generate STV from a sequence of contours. Section 3 deals with action sketch, details how action descriptors are obtained from STV, and presents a discussion on the relationship between action sketch and various motions. In Section 4, a discussion on the view invariance is given. We discuss how the action recognition is performed in Section 5. Finally, we demonstrate the recognition performance in Section 6, and conclude in Section 7.

2 Generating the Action Volume

Spatiotemporal volume to stack a sequence of frames in a video for constructing a cube has been widely used in Computer Vision. In our representation, instead of stacking whole frames, we stack only the *object regions* in the consecutive frames. Object regions can be segmented from the background by means of the background subtraction, layering or contour tracking. In this paper, we use the contour-based representation for objects, due to its simple parametric representation. We assume that the object contours Γ^t are provided by a contour tracking method. In particular we use [7] (see Fig. 1a). Given a set of tracked object contours Γ_t , our aim is to establish the correspondence between the points in consecutive contours to generate the STV.

Computing Correspondences To simplify the problem, we will consider two consecutive contours Γ^t and Γ^{t+1} , and compute point correspondences between them. Matching of two point sets, either in 2D or in 3D, is still an open prob-

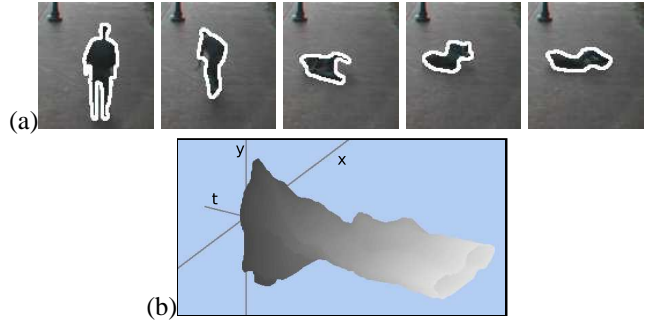


Figure 1: (a) A sequence of tracked object contours for falling action using [7], (b) STV for falling action generated by applying the proposed point correspondence method.

lem. An obvious difficulty in point matching is dealing with points that do not have corresponding points in the other set (homology). Matching contours of nonrigid objects require 1-M (one-to-many) or M-1 (many-to-one) mappings beside homologies. An intuitive approach for point matching is to use the nearest neighbor rule. This approach however performs poorly on contours with high nonrigid motion. Another possibility is to consider articulated rigid motion for subsets of points, and assume small segments of contour perform rigid motion [8]. However, rigidity constraint fails for highly nonrigid motions, e.g. actions performed by humans. Here, however, we propose to use a graph theoretic approach to solve the point correspondence problem, which is motivated by the work of Shapiro and Haralick [9].

Let L and R be two point sets corresponding to Γ^t and Γ^{t+1} respectively. We define a bipartite graph G with $|G| = |L| + |R|$ vertices, where $|\cdot|$ is the cardinality (Fig. 2a). The weights of edges from vertices in L to vertices in R are defined by the proximity, alignment similarity and shape similarity in the spatial domain, as shown in Figure 2b. Let $\mathbf{c}_i = [x_i, y_i, t]^T$ and $\mathbf{c}_j = [x_j, y_j, t + 1]^T$ be vertices in L and R respectively. We compute proximity by

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

The alignment similarity is obtained by considering the angle α between the spatial normal vectors \vec{n}_i and \vec{n}_j corresponding to \mathbf{c}_i and \mathbf{c}_j respectively, which are computed in the neighborhood of the vertices:

$$\alpha_{i,j} = \arccos(\vec{n}_i \cdot \vec{n}_j),$$

where ‘ \cdot ’ is the dot product. Let $\mathbf{T}_{i,j} = \mathbf{c}_i - \mathbf{c}_j$ be the translation and $\mathbf{R}_{i,j}$ be the rotation of the vertex \mathbf{c}_i from frame t to $t + 1$. Shape similarity between the vertices \mathbf{c}_i and \mathbf{c}_j is defined based on how the shape of the neighborhoods N_i and N_j have changed after compensating $\mathbf{T}_{i,j}$ and $\mathbf{R}_{i,j}$:

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

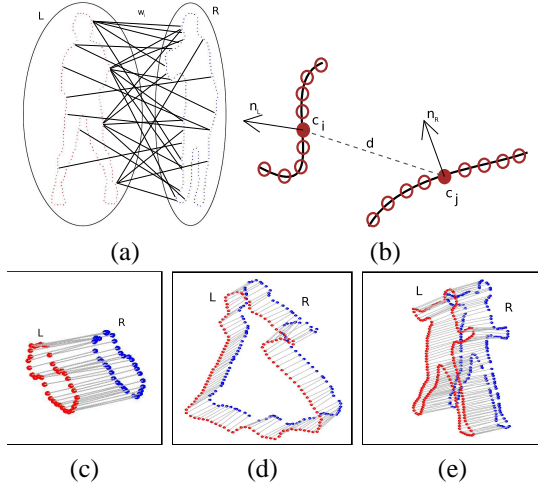


Figure 2: (a) Sets of nodes and edges in two consecutive frames from the tennis sequence, (b) local contour neighborhoods from two consecutive frames used in defining the weights for the central node. Resulting matchings between contours for (c) falling, (d) dance, and (e) tennis stroke.

where $\hat{\mathbf{x}}_i = \mathbf{R}_{i,j}\mathbf{x}_i + \mathbf{T}_{i,j}$, and $\mathbf{x}_i \in N_i$ vertex corresponding to \mathbf{x}_j obtained by using the spatial relationship (note that $|N_i| = |N_j|$). Using these three terms, the weight $w_{i,j}$ from \mathbf{c}_i to \mathbf{c}_j is given by:

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

where σ_d , σ_α and σ_ξ control the distance between the vertices, the alignment and the degree of shape variation respectively (We chose $\sigma_d = 15$, $\sigma_\alpha = 0.5$ and $\sigma_\xi = |N_i|$).

We solve the point correspondence problem by computing the maximum matching of the weighted bipartite graph. In our case maximum matching will provide the 1-1 (one-to-one) mappings from L to R , such that $\sum_i \sum_j w_{i,j}$ is maximized. However, due to the object motion there may be 1-M or M-1 matchings., and these initial associations are not usually correct. In addition, maximum matching does not guarantee to maintain the spatial relations. For instance, mappings like $\mathbf{c}_i \rightarrow \mathbf{c}_j$ and $\mathbf{c}_{i-2} \rightarrow \mathbf{c}_{j+3}$ can not hold simultaneously. For consistent matching, we perform an additional step which iteratively removes outliers and reassigns correct matchings based on the confidence of correspondences in the first step. In Figure 2c,d and e, we show the final vertex matching for three different actions.

Properties of STV

- STV can be considered as a manifold, such that it is nearly flat for small scales defined by a small neighborhood around a point. Based on this observation a continuous action volume, \mathbf{B} , is generated by computing plane equations in the neighborhood around a point.

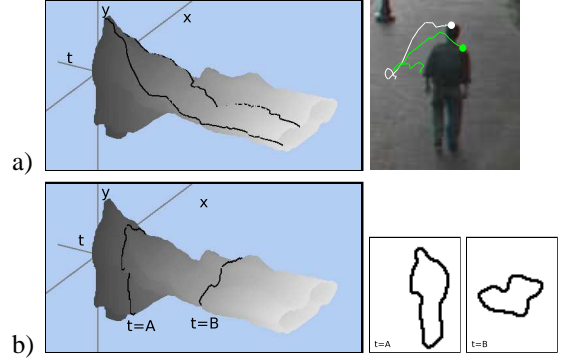


Figure 3: Projections of STV. (a) Motion trajectories generated by fixing the t parameter in $f(s, t)$. (b) Object contours generated by fixing s parameter in $f(s, t)$.

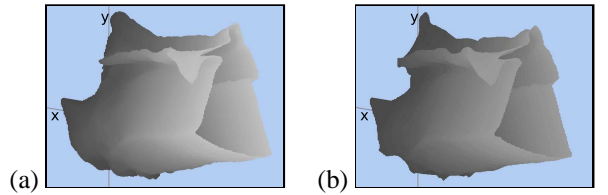


Figure 4: STVs for (a) dancer sequence with 40 frames, (b) synthetic dancer sequence with 20 frames, generated by randomly removing frames.

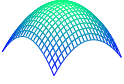
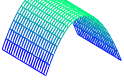
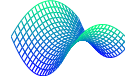
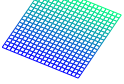
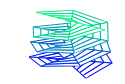
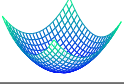
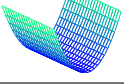
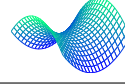
- Since STV is generated from a set of contours, instead of using an implicit three dimensional (x, y, t) representation, we can define a 2D parametric representation by considering arc length of the contour, s , which encodes the object shape, and time, t , which encodes the motion:

$$\mathbf{B} = f(s, t) = [x(s, t), y(s, t), t]. \quad (1)$$

Using s and t , we can generate trajectories of any point on the object from this volume by fixing the s parameter (see Figure 3a). Similarly, fixing the t parameter we can generate the object contours at time t (see Figure 3b).

- STV is a continuous representation in the normalized time scale ($0 \leq t \leq 1$), that is it does not require any time warping for matching two sequences of different lengths. Several different discrete approximations of STV in terms of contour sequences can be generated by using different samplings in time. In Figure 4, we show an example to demonstrate this property for the dance sequence. A synthetic sequence with 20 frames is generated by randomly removing frames from the original sequence, which has 40 frames. As seen from the figure, the volumes look very similar. However, we should note that, this is only valid for “atomic actions” such as for two walking cycles (it is not valid for two videos containing different number of walking cycles).

Table 1: The surface types and their relation to 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 

3 Action Sketch

Once STV is generated from a sequence of contours, we analyze it to compute important action descriptors which correspond to changes in *direction*, *speed* and *shape* of parts of contour. Changes in these quantities are reflected on the surface of STV, and can be computed using the differential geometry. A set of these action descriptors for an action is called *the action sketch*.

There are eight fundamental surface types: peak, ridge, saddle ridge, flat, minimal, pit, valley and saddle valley. As shown in Table 1, the fundamental surface types are defined by two metric quantities, the Gaussian curvature, K , and mean curvature, H , computed from the first and second fundamental forms of the underlying differential geometry.

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:

$$\mathbf{I}(s, t, s_p, t_p) = [s_p \quad t_p]^T \underbrace{\begin{bmatrix} E & F \\ F & G \end{bmatrix}}_{\mathbf{g}} [s_p \quad t_p], \quad (2)$$

where $\mathbf{x}(s, t)$ is defined in equation 1, $E = \mathbf{x}_s \cdot \mathbf{x}_s$, $F = \mathbf{x}_s \cdot \mathbf{x}_t$ and $G = \mathbf{x}_t \cdot \mathbf{x}_t$, and subscripts denote the partial derivatives. The \mathbf{g} matrix is called the metric tensor of the surface and has the same role as the speed function for spatial curves [10]. In particular, E in (2) encodes the spatial information, whereas F and G contain velocity information.

The second fundamental form, in contrast to the first fundamental form, is dependent on the placement of the surface in the 3D space, but is also invariant to rotation and transla-

tion. The second fundamental form is given by:

$$\mathbf{II}(s, t, s_p, t_p) = [s_p \quad t_p]^T \underbrace{\begin{bmatrix} L & M \\ M & N \end{bmatrix}}_{\mathbf{b}} [s_p \quad t_p], \quad (3)$$

where \vec{n} is the unit normal vector, $L = \mathbf{x}_{ss} \cdot \vec{n}$, $M = \mathbf{x}_{st} \cdot \vec{n}$ and $N = \mathbf{x}_{tt} \cdot \vec{n}$, and subscripts denote the second order partial derivatives. In terms of encoding motion, N in (3) is related to the acceleration of $\mathbf{x}(s, t)$. An important operator defined on a surface using the first and the second fundamental forms is the *Weingarten mapping* given by:

$$S = \mathbf{g}^{-1} \mathbf{b} = \frac{1}{EG - F^2} \begin{bmatrix} GL - FM & GM - FN \\ EM - FL & EN - FM \end{bmatrix}. \quad (4)$$

The Weingarten mapping S is a generalization of the curvature of a planar curve to the surfaces. Gaussian curvature, K , and the mean curvature, H , are two algebraic invariants derived from the Weingarten mapping [10]:

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

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

As shown in Table 1, the Gaussian and mean curvature values can be used to categorize the surface type. We consider these surface types as important action descriptors, and the set of such descriptors for a given action is called *action sketch*. In Fig. 5, we show several STVs with superimposed action descriptors.

3.1 Analysis of Action Descriptors

In this work, we consider 2D contours in the image plane, which are projections of a three-dimensional non-rigid object. When an object moves in 3D, its projected contour moves in the image plane. Similarly, if the object deforms in shape in 3D its projected 2D contour also deforms. Our action descriptors capture both motion and shape changes in a unified manner. Contour motion can be result of different object motions which are reflected by the action descriptors on the STV. For instance, ‘‘closing fingers while forming a fist’’ generates different descriptors compared to ‘‘waving hand’’. In the first case, the hand contour changes dramatically giving rise to different surface types, e.g. saddle valley and pit are generated in the STV. While in the later case, the hand shape does not change, however its motion (change of speed and direction) results in the ridge and saddle ridge.

In order to define the relationship between the action descriptors and the object motion, we will consider three types of contours: concave contour, convex contour and straight contour¹. Depending on the object motion, these contour types may generate the following action descriptors:

¹Other contour shapes are a combination of convex, concave and straight contours.

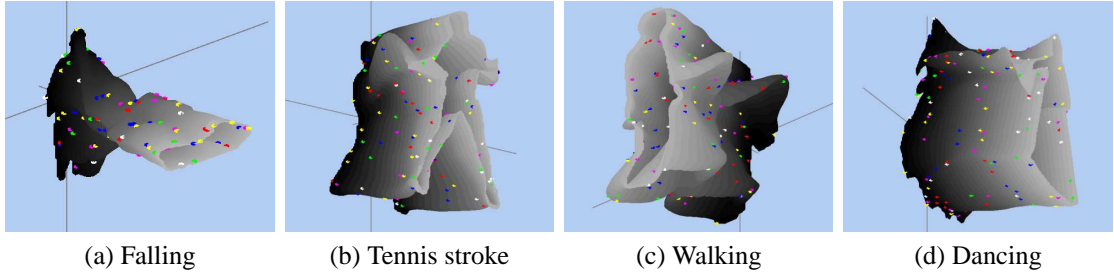


Figure 5: Color coded action descriptors corresponding to ridges, saddle ridges, valley and saddle valleys for various actions. Color codes are: red (peak), yellow (ridge), white (saddle ridge), blue (pit), pink (valley), green (saddle valley).

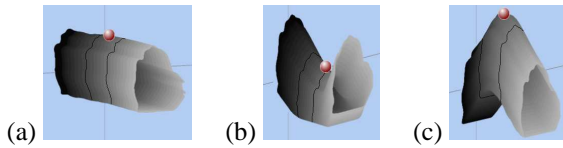


Figure 6: STVs corresponding to the motion of a hand and related action descriptors (shown as red spheres), (a) hand stays stable generating a ridge, (b) hand decelerates in \downarrow direction and accelerates in \uparrow direction generating a saddle ridge, (c) hand first decelerates in \uparrow direction and accelerates in \downarrow direction generating a peak (shown by red sphere).

- Straight contour generates ridge, valley or flat surface,
- Convex contour generates peak, ridge or saddle ridge,
- Concave contour generates pit, valley or saddle valley.

Without loss of generality, let us consider rigid motion, such that, there is no shape deformation due to object motion. In this setting, a contour can undergo three types of motions:

- *No motion:* Contour stays stable,
- *Constant speed:* Contour moves in one direction with a constant speed,
- *Deceleration and acceleration:* Contour moves in one direction while decelerating, than comes to a full stop followed by an acceleration in the opposite direction.

In Figure 6, we show the volume and resulting action descriptor generated for a sequence of hand contours. Note that, in this example only the “concave contour” segment of the hand is considered to generate the action descriptors, such that resulting descriptors are only ridge, saddle ridge and peak depending on the direction of motion. However, of course, there are also other possible ways to generate action descriptors. Below, we summarize and give examples for the hand motion that give rise to various action descriptors.

Peak Surface Peak surface is generated from a sequence of “convex” contours. A typical example of a peak is given in Fig. 6c, where the hand moves first in the direction normal

to the contour then stops and moves in the opposite direction.





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.

Ridge Ridge surface is generated in two different ways based on the motion/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. 6a, we give an example of a ridge surface generated from a sequence of hand contours with zero speed. The second possible way is when a “straight” contour moves first in some direction then comes to a stop and moves in the opposite direction.

Saddle Ridge Similar to the ridge surface, the saddle ridge is generated by the motion of convex contours. An instance of saddle ridge is shown in Fig. 6b, 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 action descriptors related to the ridge, saddle ridge can simply be extended to valley and saddle valley. The difference between the two is that the contour has to be concave for the latter case. The strength of the contour concavity or convexity and the magnitude of contour motion is encoded by the values of Gaussian and mean curvatures. For the peak and pit surfaces 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 t direction, such that when $K > 0$, the object moves in the normal direction of the contour and when $K < 0$ it moves in the opposite direction to contour normal. Similar arguments hold for the action descriptors defined by the saddle valley and saddle ridge surfaces. However, for the valley and the ridge surfaces object shape and motion can be encoded by either the mean or the Gaussian curvature. Depending on the type of the surface, the curvatures can also be used to compute the motion direction and speed at any contour point.

Table 2: Surface types and their relations to the curvature of the trajectory and the contour. Similar results can be derived for the remaining surface types.

	Surface type	contour	trajectory
	Peak	maximum	maximum
	Pit	minimum	minimum
	Valley	maximum	zero
	Saddle Valley	maximum	minimum

4 View Invariance

Assume that a particular action is captured by videos from two different view points. If the representations of this action derived from these two videos are the same, then this representation is called view invariant. View invariance is very important for action representation and recognition based on 2D information. In our approach the view invariance is directly related to the building elements of STV: object contours and the trajectories of the points on the contour. This is evident from Eq. 1. For each action descriptor in the action sketch (minima/maxima of K and H on the STV), underlying curves defined by the object contour and point trajectory will have a maxima or a minima (see Table 2). Thus, showing the view invariance of our representation is showing that the minima/maxima of the contour and the trajectory are invariant to the camera viewing angle. Invariance of a trajectory has been addressed in the same context in [3]. In the following, we will discuss the view invariance of the contour.

View Invariance of Contour maxima or minima: Object contour Γ at time t is parameterized by its arc-length (see equation 1). For the two-dimensional spatial contour, Gaussian and mean curvatures simplify to a single 2D curve curvature: $\kappa = \frac{x'y'' - y'x''}{\sqrt{x'^2 + y'^2}^3} = \frac{\mathbf{x}'^T B \mathbf{x}''}{(\mathbf{x}'^T \mathbf{x}')^{3/2}}$, 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 contour arc-length.

Let the action be viewed from 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 the world coordinates \mathbf{X} are mapped to image coordinates \mathbf{x} by $\mathbf{x}_L = A\mathbf{X} + T_L$ and $\mathbf{x}_R = C\mathbf{X} + T_R$, where subscripts denote the left and right cameras [11]. Thus, contour curvatures in 2D can be related to the world coordinates:

$$\kappa_L = \frac{\mathbf{X}'^T A^T B A \mathbf{X}''}{(\mathbf{X}'^T A^T A \mathbf{X}')^{3/2}}, \quad \kappa_R = \frac{\mathbf{X}'^T C^T B C \mathbf{X}''}{(\mathbf{X}'^T C^T C \mathbf{X}')^{3/2}}. \quad (5)$$

Note that $A^T B A = |A|B$ and $C^T B C = |C|B$, where $|\cdot|$ is

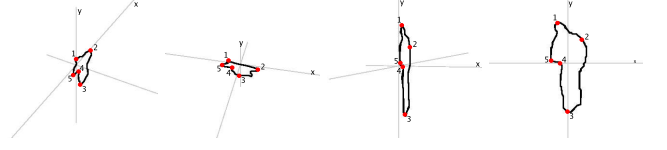


Figure 7: Projections of 3D contour on the image plane using affine camera model from various viewing angles. The numbers denote the corresponding minima and maxima.

the determinant. Dividing κ_L by κ_R we have:

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

Assume $\alpha = |C|/|A|$, when we convert \mathbf{X} to left image coordinates, the relation between curvature of a contours in left and right image will be:

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

where $D = A^{-1T} C^T C A^{-1}$. This relation shows that curvature, κ_R , of a point on right contour is directly related to the curvature, κ_L , on the left contour by only the tangent vector \mathbf{x}'_L . Due to this relation, the maxima and minima of curvature on the Γ_L are the maxima and minima on Γ_R . In Fig. 7, we show the same object contour from various viewing directions along with several of the corresponding curvature maxima and minima in each view.

Discussion

- Note that above arguments for view invariance do not apply to cases of accidental alignment. Accident alignment happens when a point on a contour moves perpendicular to the view point, such that its trajectory is mapped to a single point in the image plane. Accidental alignment may also happen when a corner or curvature maxima in contour is mapped to a non-corner point in the image plane.
- We want to clarify that since only a sequence of (outer) contours is used to generate STV, the STVs of the same action captured from different view point tend to be similar (as shown in Figure 8). If we had used the color inside the contours as well, this may not be true, since persons wearing different cloths will look different from different views. Therefore, a person walking directly toward the camera and a person walking right-to-left approximately generate the same STVs with exception to occluded parts, which is discussed in the next item.
- It is only meaningful to talk about the view invariance of the parts of the contours which are visible in both views.

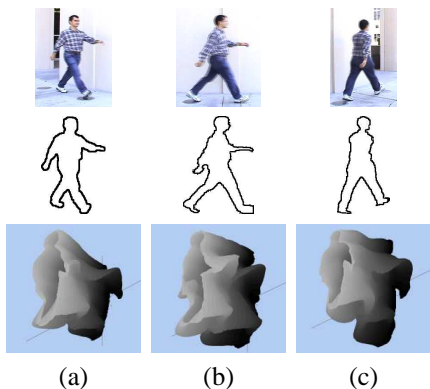


Figure 8: Walking action from three different view points, the first row shows sample frame, the second row shows associated contours and the third row shows the corresponding STVs. (a) 30° , (b) 90° , (d) 145° . The volumes are similar.

Depending on the viewing angle, it may happen that some parts of the contour related to action descriptors may get occluded in a particular view, therefore the action descriptors corresponding to the occluded parts will not be available. In this case, other action descriptors available in all views can be used. In Figure 8a-c, we show an example of this phenomena, where a walking person is viewed from three different viewing angles. In several views, right arm is occluded which results in missing action descriptors. However, the remaining object parts which are available in all views provide many useful descriptors common to all views.

- Note that rotating the action volume in the spatio-temporal space is not related to changing the camera view point in the real world. This is evident from the fact that, view point changes directly affect the contours projected onto the image plane, such that contours look different. Examples for changing camera view point and its effect on the contour is given in Fig. 7. STV generated from these contours do not look exactly the same. Regardless of this, action sketches related to different views of the action preserve the same action descriptors.

5 Matching Actions

In general, STV can be considered as 3D rigid object, and the matching problem becomes matching a 3D object with other objects of known types. Based on this observation, we pose the matching problem in terms of matching a set of points in one view of an object with a set of points in another view. In particular, set of points correspond to the action sketches and the views correspond to the projection of STVs to orthographic x-y plane for two different actions. The inverse problem of computing the homography from one view to the

other and recovering 3D structure have been well studied in context of epipolar geometry [12].

In our setting, we have the 3D volume along with a set of points in on the volume. Using this and the results defined for epipolar geometry, one can estimate the relation between two STVs using:

$$\mathbf{x}\mathcal{F}\mathbf{x}' = 0, \quad (6)$$

where \mathbf{x} and \mathbf{x}' are points on left and right action sketches respectively and \mathcal{F} is the 3×3 fundamental matrix defining this relation. Parameters of fundamental matrix can be computed by least squares fitting: $\mathbf{A}\mathbf{f} = 0$, where

$$\begin{aligned} \mathbf{A}_i &= [x_i x'_i, y_i x'_i, x'_i, x_i y'_i, y_i y'_i, y'_i, x_i, y_i, 1], \\ \mathbf{f} &= [\mathcal{F}_{1,1}, \mathcal{F}_{1,2}, \mathcal{F}_{1,3}, \mathcal{F}_{2,1}, \mathcal{F}_{2,2}, \mathcal{F}_{2,3}, \mathcal{F}_{3,1}, \mathcal{F}_{3,2}, \mathcal{F}_{3,3}]. \end{aligned}$$

The solution do this homogeneous system is given by the unit eigenvector corresponding to the minimum eigenvalue (9^{th} eigenvalue) of $\mathbf{A}^\top \mathbf{A}$, which is typically very close to 0 if two sketches are matching. Due to the degenerate cases rank of $\mathbf{A}^\top \mathbf{A}$ may be lower. Thus, using Irani's approach [13], one can compute the rank for noisy measurements and eliminate degenerate cases by not considering the matching. From remaining set of possible matchings between the input action sketch and the known action sketches, we select the corresponding action κ with minimum matching score:

$$\kappa = \arg \min_{0 < j \leq n} \lambda_9^j \quad (7)$$

where n is the total number of actions and λ_9^j is the 9th eigenvalue of $\mathbf{A}^\top \mathbf{A}$ corresponding to action j (note that degenerate cases are eliminated, thus rank of $\mathbf{A}^\top \mathbf{A}$ is exactly 8 if there is a correct matching).

6 Experiments

In order to test the performance of the proposed approach, we have used twenty-eight sequences of twelve different actions captured from different viewing angles. The video sequences include dancing (2), falling (1), tennis strokes (2), walking (7), running (1), kicking (2), sit-down (2), stand-up (3), surrender (2), hands-down (2), aerobics (4) actions (the number in the parenthesis denote the number of videos of a particular action). In Fig. 9, we show the complete set of STV with superimposed action descriptors.

From an input video, we first track the contours of the objects and generate the STV. For each action, the action sketch is generated by analyzing the differential geometric surface properties of the underlying volume. The action sketch is then matched with the representative actions in the database by computing the distance measure discussed in Sec. 5. In Table 3, we tabulate the first two best matches of each action video. Except for five actions the first best matches of all action videos are correct. For the remaining five the second best matches are correct. Usually, the matching criteria

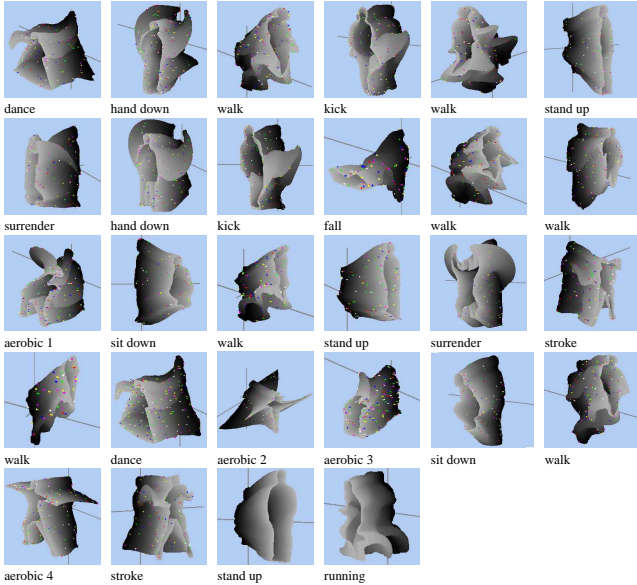


Figure 9: Action sketches superimposed on the STVs, which are generated from video sequences of human actors. Numbers denote the action labels in Table 3.

clusters similar actions during retrieval. Specifically for the 2nd video, in which the action is recognized as the second best match, “hands down” action involves “standing up” and the proposed method retrieved both actions. The hands down action also shows the importance of using both the shape and the motion of the object. 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 proposed method to identify the action.

7 Conclusions

In this paper, we proposed a novel representation for actions using spatio-temporal action volumes. Given the object contours for each time slice, we first generate an action volume by computing point correspondences between consecutive contours using a graph theoretical approach. In order to obtain a compact action representation, we analyze the differential geometry of the local surfaces on the volume. This results in unique action descriptors which are categorized based on the sign of Gaussian and mean curvatures. Set of these action descriptors define the action sketch which is invariant to the viewing angle of the camera. Using these view invariant features, we perform view invariant action recognition.

Table 3: Recognition results for various actions. Bold face represents the correct matches and non-bold represents incorrect match as the first best match.

Action	First best	Second best
Dance	Dance	Walking
Hand down	Stand up	Hand down
Walking	Walking	Kicking
Kicking	Kicking	Stroke
Walking	Walking	Kicking
Stand up	Stand up	Hands down
Surrender	Hands down	Surrender
Hands down	Hands down	Kicking
Kicking	Kicking	Aerobic 1
Falling	Falling	Kicking
Walking	Walking	Kicking
Walking	Sit down	Walking
Aerobic 1	Aerobic 1	Walking
Sit down	Sit down	Falling
Walking	Walking	Kicking
Stand up	Hands down	Stand up
Surrender	Surrender	Aerobic 1
Tennis stroke	Tennis stroke	Walking
Walking	Walking	Aerobic 4
Dance	Dance	Aerobic 3
Aerobic 2	Aerobic 2	Aerobic 3
Aerobic 3	Aerobic 3	Kicking
Sit down	Sit down	Dance
Walking	Walking	Kicking
Aerobic 4	Aerobic 4	Dance
Tennis stroke	Tennis stroke	Kicking
Stand up	Stand up	Release
Running	Running	Kicking

References

- [1] J.M. Siskind and Q. Moris, “A maximum likelihood approach to visual event classification,” in *ECCV*, 1996, pp. 347–360.
- [2] D. Ayers and M. Shah, “Monitoring human behavior from video taken in an office environment,” *IVC Jrn.*, vol. 19, no. 12, pp. 833–846, 2001.
- [3] C. Rao, A. Yilmaz, and M. Shah, “View invariant representation and recognition of actions,” *IJCV*, vol. 50, 2002.
- [4] A. Bobick and J. Davis, “The representation and recognition of action using temporal templates,” *PAMI*, vol. 23, no. 3, pp. 257–267, 2001.
- [5] I. Laptev and T. Lindeberg, “Space-time interest points,” in *ICCV*, 2003.
- [6] S.Niyogi and E.Adelson, “Analyzing gait with spatiotemporal surfaces,” in *Wrks. on Nonrigid and Artic. Motion*, 1994.
- [7] no author, “no title,” *no journal*, p. no pages, no year.
- [8] H. Chui and A. Rangarajan, “A new algorithm for non-rigid point matching,” in *CVPR*, 2000.
- [9] L. Shapiro and R. Haralick, “Structural descriptions and in exact matching,” *PAMI*, vol. 3, no. 9, pp. 504–519, 1981.
- [10] R.C. Jain and A.K. Jain, *Analysis and interpretation of range images*, Springer-Verlag, 1990.

- [11] J.L. Mundy and A. Zisserman, *Geometric Invariance in Computer Vision*, MIT Press, 1992.
- [12] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *IJCV*, vol. 22, no. 7, pp. 161–198, 1998.
- [13] M. Irani, "Multi-frame optical flow estimation using subspace constraints," in *ICCV*, 1999.