A View-Invariant Representation of Human Action

Cen Rao and Mubarak Shah
Computer Vision Lab
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816
{rcen, shah}@cs.ucf.edu

Abstract

The representation plays an important role in recognition and understanding of human action from video sequences. A view-invariant representation of action consisting of *dynamic instants* and *intervals*, which is computed using spatiotemporal curvature of a trajectory, is presented. In order to validate our representation, we report experiments on several different actions performed by different people, and captured in different viewpoints.

Keywords: Video Understanding, Action Recognition, View-invariant Representation, Spatiotemporal curvature, Events, Activities

1. Introduction

What do we mean by an action? Webster's dictionary defines action: the doing of something; state of being in motion; the way of moving organs of the body; the moving of parts: guns, piano; military combat; appearance of animation in a painting, sculpture, etc. More or less, hand gestures, sign language, facial expressions, lip movement during speech, human activities like walking, running, jumping, jogging, etc, and aerobic exercises are all actions. Consider a typical office scene, at a given time a person can be performing either one of the following actions: reading, writing, talking to other people, working on a computer, talking on a phone, opening and closing cabinets, leaving or entering the office.

Actions can be classified into three categories: *events*, *temporal textures* and *activities* [1]. Motion *events* do not exhibit temporal or spatial repetition. Events can be low-level descriptions like a sudden change of direction, a stop, or a pause, which can provide important clues to the type of object and its motion. Or they can be high level descriptions like "opening a door", "starting a car", "throwing a ball", or more abstractly "pick up", "put down", "push", "pull", "drop", "throw", etc. Motion verbs can also be associated with motion events. For example, motion verbs can be used to characterize trajectories of moving vehicles [2], or normal or abnormal behavior of the heart's left ventricular motion [3]. The temporal textures exhibit statistical regularity but are of indeterminate spatial and temporal extent. Examples include ripples on water, the wind in the leaves of trees, or a cloth waving in the wind. Activities consist of motion patterns that are temporally periodic and possess compact spatial structure. Examples include walking, running, jumping, etc.

Recognition of human actions from video sequences is very popular in computer vision. This work has applications in video surveillance and monitoring, human-computer interfaces, model-based compression, and augmented reality. One standard approach for human action recognition is to extract a set of features from each frame of a sequence, and use those features to train Hidden Markov Models (HMMs) to perform recognition. The features can be an image location of a particular point on the object, a centroid of image region, moments of an image region, gray levels in a region, optical flow in a region (used as magnitude of optical flow, or concatenated u and v in a vector), sum of all changed pixels in each column (XT trace), 3-D locations (X_i, Y_i, Z_i) of particular point on the object, joint angels; how the parts of body move with respect to time, muscle actuations, properties of optical flow in a region like curl, divergence, etc, coefficients used in the eigen decomposition of above features, etc. A HMM consists of a set of states, a set of output symbols, state transition probabilities, output symbol probabilities, and initial state probabilities. The model works as follows. The features extracted from video sequences are used to train the HMMs. Matching of an unknown sequence with a model is done through the calculation of the probability that an HMM could generate the particular unknown sequence. The HMM giving the highest probability is the one that most likely generated that sequence. Siskind and Morris gave a good proposal for this kind of approaches [6]. The tracking results of objects, hand, and arm are fitted with ellipses. Then the characteristics of these ellipses are fed into HMM. The

recognition result is from the model, which gives highest likelihood of the action going on.

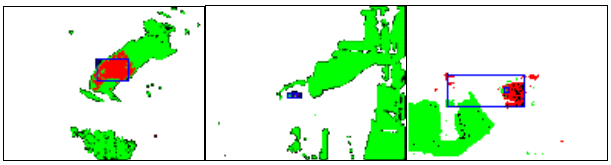
Researchers also use the similar approaches to solve handwriting, speech, and American Sign Language (ASL) recognition problems [8]. The input values can be the curvature and direction of pen trajectory, or even raw pixels for handwriting [7]; the voice data for speech; position, shape, angle of the arm for ASL [8]. And there is a lot of research focus on various HMMs to get better computation performance, for example Hierarchical HMM, tree-based formalism for indexing and searching, neural-network approaches. Pre-processing methods, such as PCA, linear/non-linear transform, wavelet, can also improve the recognition rate.

In previous research, the most emphasis has been on discovering appropriate features. Therefore, not much work has been done on HMMs; they have been treated as a black box. There are several important issues related to HMMS. First, since HMMs rely on probabilities they require extensive training, therefore one needs to have a large number of training sequences for each activity to be recognized. Second, for each activity to be recognized, a separate HMM needs to be built. Therefore, this approach can only recognize some predefined set of activities. It does not have a capability to learn new activities. Third, since HMM is treated as a black box, it does not explain what a particular activity is? It just outputs the probability an unknown activity is recognized as a model activity. Regarding features, the issue of representation of features has mainly been ignored.

In this paper, we focus our attention on human actions performed by a hand. These actions include: opening and closing overhead cabinets, picking up and putting down a book, picking up and putting down a phone, erasing a whiteboard, etc. While performing an action, a hand essentially generates a 3-D trajectory in (x, y, z) space with respect to time. Our aim is to first compute a 2-D projection of this trajectory from a video sequence, and then to analyze this trajectory to derive a compact representation, which will be useful in recognizing these actions. We propose a new representation scheme based on sptiotemporal curvature of a trajectory. A trajectory is represented by a sequence of *dynamic instants* and *interval*s.

2. Computing Trajectories

In this section, we discuss how to compute motion trajectories from video sequences. In our method, hand is located in each frame, and centroid of a hand in each frame is connected to obtain a trajectory. We apply skin detection to locate a region corresponding to the hand in an image sequence. Skin detection uses pixel color value. Based on the color predicate, the system labels the incoming pixel as skin or non-skin. During the training phase a color histogram is generated. The pixels



(a) (b) (c)

Figure 1. a) Correct detection of hand using color predicate. b) Color predicate is too tight, only few hand pixels are detected as skin pixels. c) Color predicate is too loose, several non-skin pixels are labeled as skin pixels.

are manually labeled as skin or non-skin, and a 3-D Gaussian function for every pixel is generated. If the given pixel is a skin pixel, then a wide Gaussian ($\sigma = 2$) is added to the color histogram. If the pixel is labeled as non-skin pixel, then a narrow Gaussian ($\sigma = 1$) is subtracted from the color histogram. At the end of training, a threshold is applied to color histogram in order to divide histogram bins into skin and non-skin. This way, a predicate is generated, which takes a color pixel value as an input, and outputs the skin or non-skin label, based on which histogram bin the pixel color falls in. Then during detection, we just check the pixel flags in color predicate to decide its label. This process runs very fast, since only lookup table operations are involved.

Determining the exact value of threshold for the color predicate is difficult task. If the the the threshold is too tight, only few pixels corresponding to hand will be detected as skin pixels (Figure1b). In this case, many pixels corresponding to hand will be missed. On the other hand if the threshold is too loose, many non-skin pixels will be detected as skin (Figure 1c). In order to deal with this problem we employ temporal information between two frames. We set up a tracking window in the current frame based on the position of hand in the previous frame. If a pixel is in the tracking window, and its color is close to skin color (not necessarily exactly equal) then that pixel is labeled as skin pixel.

The main reason for the problems in skin detection is that the skin color changes, due to different light conditions, and due to blur created by the hand motion. The use of tracking window improves the quality of skin detection operation greatly, especially for those images, for which we do not have training samples. At the same time does not introduce false positives, which classify non-skin pixels as skin pixels. Our skin detection algorithm is essentially a modified version of the technique described by Kjeldsen and Kender [4].

After skin detection, a connected component algorithm is applied to obtain largest connected component, which is hypothesized to be a hand. We assume that the only skin color object is hand. However, if this is not the case, then we can introduce some other constraint, for example the fastest moving skin region is a hand. Next, the centroid of this skin region is comput-

(a) **(b)** **(c)**

Figure 2. (a)"opening overhead cabinet" trajectory, (b) smooth version of trajectory, (c) *dynamic instants* (marked by "*") and *intervals.*

ed for each frame, and the trajectory of hand is created by joining the centroids.

3. Smoothing Trajectories

A trajectory is a spatiotemporal curve defined as: (x[1], y[1], 1), (x[2], y[2], 2),…, (x[n], y[n], n). There are essentially two functions: $x(t)$ and $y(t)$ in the above definition of a trajectory. The trajectory for action "opening overhead cabinet" is shown in the Figure 2a. This trajectory contains some noise due to errors in skin detection, lighting conditions, projection distortions, occlusion, etc. Also, since the centroid of hand region is not always a true centroid of a hand, the trajectory obtained by connecting centroids of skin regions contains some errors. In order to deal with this noise, we use anisotropic diffusion to smooth $x(t)$ and $y(t)$ coordinates of trajectory. Anisotropic diffusion was proposed in the context of scale space [4]. This method iteratively smoothes the data (I) with a Gaussian kernel, but adaptively changes the variance of Gaussian based on the gradient of a signal at a current point as follows:

$$I_i^{t+1} = I_i^t + \lambda[c_N \bullet \nabla_N I + c_S \bullet \nabla_S I]_i^t \qquad (1)$$

where $0 \le \lambda \le 1/4$; we choose 0.2 in our experiments, t represents the iteration number, and

$$\begin{aligned} \nabla_N I_i &\equiv I_{i-1} - I_i \\ \nabla_S I_i &\equiv I_{i+1} - I_i \end{aligned} \qquad (2)$$

The conduction coefficients are updated at every iteration as a function of the gradient:

$$\begin{aligned} c_N^t &= g(\left|\nabla_N I_i^t\right|) \\ c_S^t &= g(\left|\nabla_S I_i^t\right|) \end{aligned} \qquad (3)$$

where $g(\nabla I) = e^{-(\|\nabla I\| / k)^2}$.

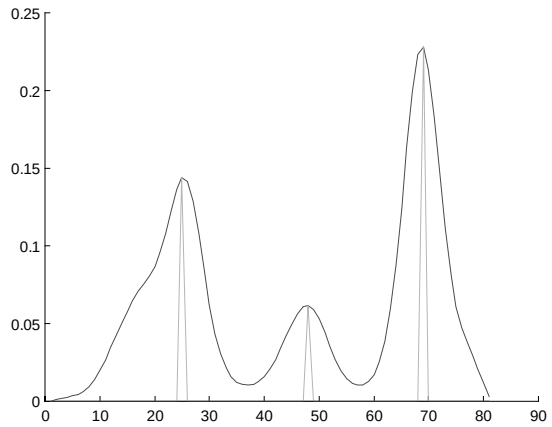


Figure 3. Spatiotemporal curvature, and detected maxima (*dynamic instants)* in "opening overhead cabinet" trajectory.

The constant k can be fixed either manually at some fixed value, or can be estimated from the "noise estimator". We choose $k = 10$ in our experiments. Figure 2b shows a trajectory after anisotropic diffusion of x and y coordinates. Notice that now the trajectory is much smoother.

4. Calculating Spatiotemporal Curvature

We use spatiotemporal curvature to compute view invariant representation of an action. The spatiotemporal curvature of a trajectory is computed by a method described by Besl and Jain [6]. In this case, a 1D version of the quadratic surface fitting procedure is used. The spatiotemporal curvature, k is given as follows:

$$k = \frac{\sqrt{A^2 + B^2 + C^2}}{\left((x')^2 + (y')^2 + (t')^2\right)^{3/2}} \qquad (4)$$

where

$$A = \begin{vmatrix} y' & t' \\ y'' & t'' \end{vmatrix}, B = \begin{vmatrix} t' & x' \\ t'' & x'' \end{vmatrix}, C = \begin{vmatrix} x' & y' \\ x'' & y'' \end{vmatrix}. \qquad (5)$$

The notation $|\cdot|$ denotes the determinant, and $x'(t) = x(t) - x(t-1)$, $x''(t) = x'(t) - x'(t-1)$. Since the time interval is constant, so $t' = 1$, and $t'' = 0$.

Spatiotemporal curvature captures both the speed and direction changes in one quantity. Moreover, a special case of spatiotemporal curvature when $t = 0$, in the above equation is the spatial curvature, commonly used in 2-D shape analysis. The spatiotemporal curvature of "opening overhead cabinet" trajectory is shown in Figure 3.

5. Representation

Representation is very important and sometimes difficult aspect of an intelligent system. The representation is an abstraction of a sensory data, which should reflect a real world situation, be view-invariant and compact, and be reliable for latter processing. We propose a new representation scheme based on spatiotemporal curvature of a trajectory. A trajectory is represented by a sequence of *dynamic instants* and *interval*s. A *dynamic instant* is an instantaneous entity, which occurs for only one frame, and represents an important change in motion characteristic: speed, direction, acceleration, and curvature. An *instant* is detected by identifying maxima (a zero-crossing in a first derivative) in the spatiotemporal curvature. An *interval* represents the time-period between any two *dynamic instants*, during which the motion characteristics pretty much remain constant. In our representation, *instants* and *intervals* have physical meanings. Therefore, it is possible to explain an action as a sequence of meaningful instants and intervals. *Dynamic instants* and *intervals* for "opening overhead cabinet" action are shown in Figure 2c.

A dynamic instant is characterized by a frame number, the image location, and the sign. The frame number tells us precisely in which frame, the dynamic instant occurs; the image location provides the location of the hand in the image when the dynamic event occurs; and the sign represents the sign of change of motion characteristic at the instant. The intervals are described by an average spatiotemporal curvature. Examples of dynamic instants include: touching, twisting, loosening; and the examples of intervals include approaching, lifting, pushing, and receding. Consider an opening overhead cabinet action (Figure 4c). This action can be described as: hand approaches the cabinet ("approaching" interval), hand makes a contact with the cabinet ("touching" instant), hand lifts the cabinet door ("lifting" interval), hand twists ("twisting" instant) the wrist, hand pushes ("pushing" interval) the cabinet door in, hand breaks the contact ("loosening" instant) with the door, and finally hand recedes ("receding" interval) from the cabinet. Similarly, "picking up a phone" action (Figure 4a) can be explained by two *intervals* and one *dynamic instant* as: hand approaches the phone ("approaching" interval), hand touches the phone ("touching" instant), and finally hand lifts up the phone towards the ear ("lifting" interval).

6. View Invariance

It is very important for a representation of action to be view invariant. Since an action takes place in 3-D, and is projected on 2-D image, depending on the viewpoint of the camera the projected 2-D trajectory may vary. Therefore, trajectories of the same action may have very different trajectories, and trajectories of different actions may look the same. This may create a problem in interpretation of trajectories at the higher level. However, if the representation of action only captures characteristics, which are view-invariant, then the higher level interpretation can proceed without any ambiguity. Instants, which are the maxima in spatiotemporal curvature of a trajectory, are view-invariant. A dynamic instant in 3-D is always projected as a dynamic instant in 2-D, except in limited cases of accidental alignment. By accidental alignment, we mean a viewpoint, which is parallel to the plane, where the action is being performed. In that case, the centroid of hand in all frames is projected at the same location in image plane, resulting in a 2-D trajectory, which is essentially a single point. In Figure 5a, we show trajectories of opening overhead cabinet action from several viewpoints. Even though these trajectories look quite different, in all cases three dynamic instants are detected by the proposed method.

The trajectories of the same action from different viewpoints look different even though all of them contain the same number of instants, because the intervals are different. In order to deal with view dependence of intervals, we propose a notion of a *normal view*. For each action, an arbitrary view is selected as a *normal view*, and the representation consisting of instants and intervals is computed. The trajectory of the same action performed under the camera view different from the normal view will still contain the same number of instants, but the characteristics of intervals may vary. We propose to use correspondence between instants in a *normal* and a novel view to fit the affine transformation. Since the order and number of instants in both trajectories are the same, the correspondence can easily be determined. Once the affine transformation is computed, the intervals in the novel view can be transformed into the intervals under normal view using this affine transformation. Figure 6 and Figure 8 respectively show the trajectories of opening and closing overhead cabinet, and picking up a phone, and putting down a phone actions, transformed to the *normal views*. Compare these with trajectories shown in Figure 5 and Figure 7.

7. Experiments

We have performed a large number of experiments to validate our view-invariant representation of action. We have considered ten different actions performed by different people, and captured in different view points, and have recorded a large number of video sequences. However, due to space limitation, we are not able present all results here. Please visit our webpage: www.cs.ucf.edu/~rcen for more examples. Figure 4 video shows sequences of four actions (every 20^{th} frame of the sequence is shown), and Figures 5-8 show some representative results.

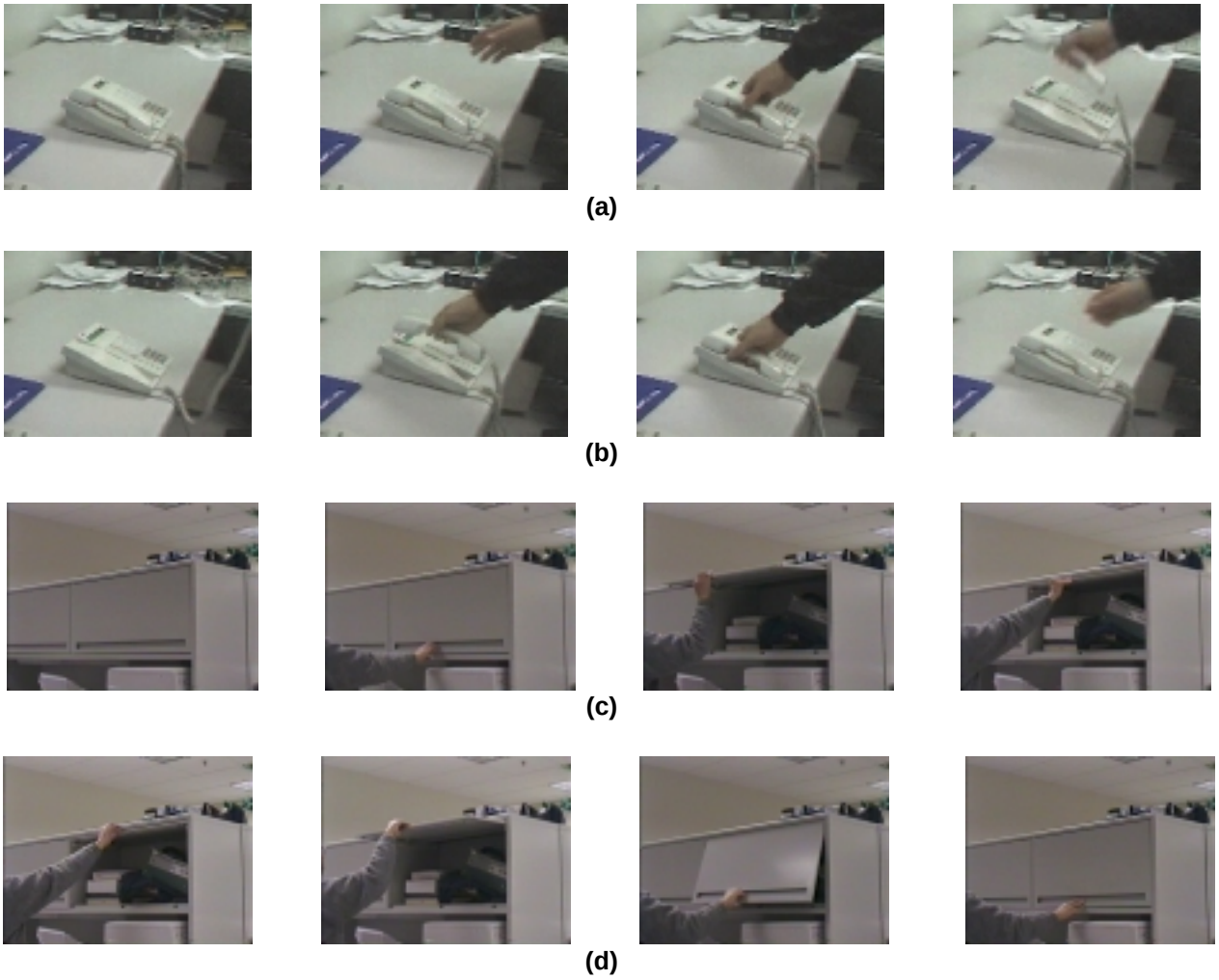


Figure 4.(a) “picking up phone” action. (b) “putting down phone” action.(c) “opening overhead cabinet” action. (d) “closing overhead cabinet” action.

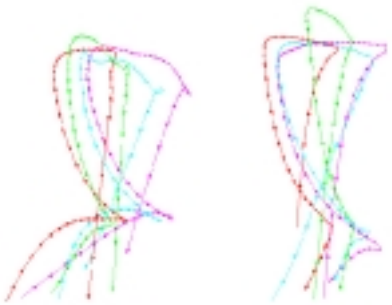
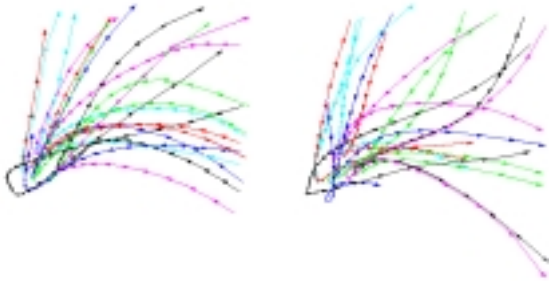


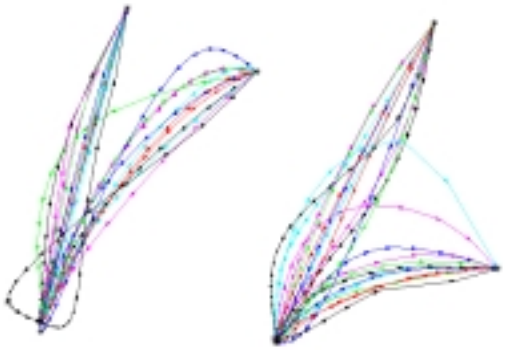
Figure 5: (a) Several trajectories of “opening overhead cabinet”, (b) “closing overhead cabinet” actions.

Figure 6: (a) Several trajectories of “opening overhead cabinet”, (b) “closing overhead cabinets” actions after converting them to the normal view using affine transformation.



(a) **(b)**

Figure 7: (a) Several trajectories of “picking up a phone”, (b) “putting down a phone” actions.



(a) **(b)**

Figure 8: (a) Several trajectories of “picking up a phone”, (b) “putting down a phone” actions after converting them to a normal view using affine transformation.

8. References

[1] Polana, R., and Nelson, R., “Temporal textures and activity recognition”, chapter in *Motion-Based Recognition*, editors: Shah, M., and Jain, R., Kluwer Academic Publishers, 1997.

[2] Koller, D., Heinze, D, and Nagel, H-H. “Algorithmic characterization of vehicle trajectories from image sequences by motion verbs”, CVPR-91, pp 90-95.

[3]. Tsotsos, J. K., et al, “A Framework for visual motion understanding”, IEEE PAMI, 2(6):563-573, November, 1980.

[4]. Pietro Perona and Jitendra Malik, “Scale-space and Edge Detection Using Anisotropic Diffusion”, IEEE PAMI, vol. 12 No. 7. July 1990.

[5] Besl, P. J., and Jain, R. C., “Invariant surface characteristics for 3D object recognition in range images”, CVGIP, 33, 1986, 33-80.

[6] J. M. Siskind and Q. Morris. A maximum-likelihood approach to visual event classification. In Proceedings of the Fourth European Conference on Computer Vision, pages 347--360, 1996.

[7] Bengio, Y., LeCun, Y., and Henderson, D. Globally trained handwritten word recognizer using spatial representation, convolutional neural networks and hidden Markov models. In NIPS 6, pp. 937--944, Morgan Kaufmann, 1993.

[8] T. Starner and A. Pentland. Real-time American Sign Language recognition from video using hidden Markov models. Perceptual Computing Section Technical Report No. 375, MIT Media Lab, Cambridge, MA, 1996.