International Journal of Computer Vision 50(2), 203–226, 2002

View-Invariant Representation and Recognition of Actions

CEN RAO, ALPER YILMAZ AND MUBARAK SHAH
Computer Vision Laboratory, School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816, USA
rcen@cs.ucf.edu
yilmaz@cs.ucf.edu
shah@cs.ucf.edu

Received November 15, 2001; Revised February 27, 2002; Accepted March 3, 2002

Abstract. Analysis of human perception of motion shows that information for representing the motion is obtained from the dramatic changes in the speed and direction of the trajectory. In this paper, we present a computational representation of human action to capture these dramatic changes using spatio-temporal curvature of 2-D trajectory. This representation is compact, view-invariant, and is capable of explaining an action in terms of meaningful action units called *dynamic instants* and *intervals*. A dynamic instant is an instantaneous entity that occurs for only one frame, and represents an important change in the motion characteristics. An interval represents the time period between two dynamic instants during which the motion characteristics do not change. Starting without a model, we use this representation for recognition and incremental learning of human actions. The proposed method can discover instances of the same action performed by different people from different view points. Experiments on 47 actions performed by 7 individuals in an environment with no constraints shows the robustness of the proposed method.

Keywords: action recognition, view-invariant representation, view-invariant matching, spatio-temporal curvature, human perception, instants

1. Introduction

Recognition of human actions from video sequences is an active area of research in computer vision. Possible applications of recognizing human actions include video surveillance and monitoring, human-computer interfaces, model-based compression and augmented reality.

Natural actions can be classified into three categories: *events*, *temporal textures* and *activities* (Polana, 1994). Events do not exhibit temporal or spatial repetition and they are described by low-level and high level descriptions. Low-level descriptions can be a sudden change of direction, a stop, or a pause, which can provide important clues to the type of object and its motion; while high level descriptions can be "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 events. Examples of motion verbs are the characterization of moving vehicles' trajectories (Koller et al., 1991) or normal/abnormal behavior of the heart's left ventricular motion (Tsotsos et al., 1980). The temporal texture category exhibits statistical regularity. Examples of temporal textures are ripples on water, the wind in 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 of activities are walking, running, jumping, etc.

Recognition of human actions from video sequences involves *extraction* of relevant visual information from a video sequence, *representation* of that information in

a suitable form, and *interpretation* of visual information for the purpose of recognition and learning human actions. Video sequences contain large amounts of data, but most of this data does not carry much useful information. Therefore, the first step in recognizing human actions is to extract relevant information which can be used for further processing. This can be achieved through visual tracking. Tracking involves detection of regions of interest in image sequences, which are changing with respect to time. Tracking also involves finding frame to frame correspondence of each region so that location, shape, extent, etc., of each region can reliably be extracted.

Representation is a very important and sometimes difficult aspect of an intelligent system. The representation is an abstraction of the sensory data, which should reflect a real world situation, be view-invariant and compact, and be reliable for later processing. Once the representation has been defined, the first obvious thing to do is to perform a comparison so that classification or recognition can take place. The methods usually involve some kind of distance calculation between a model and an unknown input. The model with smallest distance is taken to be the class of motion to which the input belongs. The problem with this is that the system can only recognize a predefined set of behaviors. This kind of system needs a large number of training sequences, does not have capability to explain what a particular behavior is, and can not learn and infer new behaviors from already known behaviors.

Therefore, it is desirable to build a system that starts with no model and incrementally builds models of activities by watching people perform activities. Once these models are learned, the system should be able to recognize similar behaviors in the future. This is probably similar to how children learn actions by repeatedly watching adults perform different actions.

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 white-board, etc. Since an action takes place in 3-D, and is projected on a 2-D image, the projected 2-D trajectory may vary depending on the viewpoint of the camera. This creates a problem in interpretation of trajectories at the higher level. In most current works on *action recognition*, the issue of view-invariance has been ignored. Therefore, proposed methods do not succeed in general situations. We propose a view-invariant representation of action consisting of *dynamic instants* and *intervals*, which is computed using the spatio-temporal curvature of a 2-D trajectory. The dynamic instants are atomic units of actions and are also of substantial value to human perception. They result from a change in the force applied to the object during an activity, and cause a change in the direction and/or speed, and can be reliably detected by identifying maxima in the spatio-temporal curvature of the action trajectory. In this paper, we formally show that the dynamic instants are view-invariant, except in the limited cases of accidental alignment. The proposed representation is then used to automatically learn and recognize human actions. In order to match two representations for action recognition purposes, we use a view-invariant matching function, which employs the eigenvalues of a matrix formed from the dynamic instants of two actions. This matching function depends on the rank of the matrix and it is interesting to know under what conditions two actions will match. Towards that end, we restate a theorem given in Seitz and Dyer (1997) in the context of matching actions. In order to demonstrate our ideas, we have experimented with video sequences depicting seven different people performing roughly 47 different actions. The system is able to learn them automatically.

The organization of the rest of the paper is as follows. In the next section, we summarize related work on action representation and recognition. In Section 3, the psychological and theoretical aspects of motion and actions are analyzed. Specifically, in Section 3.1, the details on psychological research on human actions are given. In Section 3.2, we propose a mathematical model to overcome problems of previous approaches, which is followed by analysis of the proposed method's ability to find instants. In Section 3.4, a comparison between our method and the previously proposed approaches is given. Section 3.5 details generating and smoothing of hand trajectories from video sequences. Next, a view-invariant representation of action based on the *dynamic instants* is presented in Section 4. In this section we also show that in addition to the existence of the dynamic instants, the sign of instant is also view-invariant, which is a very useful characteristic for action recognition. Section 5 deals with learning human actions. In particular, we discuss how the representations can be matched using eigenvalues of a matrix formed from the dynamic instants of two actions. Finally, we present experimental results for the proposed system in Section 6.

2. Related Work

Izumi and Kojiama (2000) proposed an approach to generate natural language descriptions of human behavior from real video sequences. First, they extract the head region of the human from each frame. Then, using a model-based method, the 3-D pose and position of the head is estimated. Next, the trajectory of the head is divided into segments, and the most suitable word from the language is selected. To generate text descriptions of actions, a hierarchy of actions is constructed called a case frame. For example, a person can be moving and walking, or moving and running. So moving is higher level description; it can be known with more certainty. Case frames are developed for each body part and each object.

Siskind and Moris (1996) proposed an HMM based system to classify 6 gestures: *pick up*, *put down*, *push*, *pull*, *drop*, and *throw*. Their method requires training, and the features used by their system are not view-invariant. A similar HMM-based action recognition approach for American Sign language was also presented by Starner and Pentland (1996).

Davis et al. (2000) proposed a motion recognition method by fitting a sinusoidal model. The sinusoidal model contains amplitude, frequency, phase, and translation parameters. Their method first estimates the translation, which is 1-D information, then estimates the frequency of x and y to get 2-D information, then estimates phase, and so on. Based on the sinusoidal model coefficients, the motion can be classified into different categories. Each category has consistent underlying structural descriptions.

Polana (1994) used normal flow to recognize activities like walking, running, skipping, etc. He divided each image into a spatial grid of divisions. Each activity cycle is divided into time divisions, and motion is totaled in each temporal division corresponding to each spatial cell separately. A feature vector is formed from these spatio-temporal cells, and used in a nearest centroid algorithm to recognize activities.

Madabushi and Aggarwal (2000) presented an approach to recognize activities using head movement. Their system is able to recognize 12 activities based on nearest neighbor classification. The activities include: standing up, sitting down, bending down, getting up, etc.

Seitz and Dyer (1997) proposed an affine-based view-invariant method to analyze cyclic motion. Their main contribution is the matching method to find the period of repeating body posture. Tsai et al. (1994) used FFT to find the period of cyclic motion, which is captured by a large impulse in the Fourier magnitude plot of the spatio-temporal curvature.

Bobick and Davis (1997) described an approach to recognize aerobic exercises from video sequences. Their method needs training and multiple views to perform recognition.

All the methods described above suffer from applicability to general situations. Most of them are either view variant or have very limited invariance capabilities.

3. Perception of Motion

In this section, we first review some psychological research on how humans perceive motion. Then we propose a mathematical model to capture important information (*instants*) from the trajectory of motion. The proposed model is close to how humans perceive the motion.

3.1. Human Perception

Johansson's experiment on "Moving Light Displays" (MLDs) shows the importance of motion information in human perception. MLDs consist of bright spots attached to the joints of an actor dressed in black and moving in front of a dark background. The collections of spots carry only spatial information without any structural information. A set of static spots remained meaningless to observers, while their relative movement created a vivid impression of a person walking, running, dancing, etc.

Contemporary psychology has provided an instructive analysis of the *atomic units* of actions that are of substantial value to perception. These atomic units of actions are defined as motion events due to the significant changes in motion trajectories (Jagacinski et al., 1983). Examples of these changes include *start*, *pause*, or *stop* of motion and a sudden *change in the direction or the speed* of the motion (Rubin and Richards, 1985). *Start* is the boundary (*time instant*) at which the object changes from the stationary state to the moving state. Similarly, *stop* is the change from the moving state to the stationary state. *Dynamic instant* results due to a change in the force applied to the object during the activity and causes a change in the direction and/or speed. Since *pause* is a combination of stop and start, we will not treat it as an additional class of motion boundary.

In Zacks and Tversky (2001) showed that people tend to divide activities at locations that correspond to changes in the physical features (*speed* and *direction*), and this division of activities constitutes basic actions that are primitive actions. This conclusion is strengthened by a set of studies on the role of events in action comprehension (Parish et al., 1990; Newtson and Engquist, 1976). Parish et al. (1990) described American Sign Language sequences in term of the activity index, which is obtained from the changes in position of the hands. They selected the frames corresponding to local minima of the activity index as critical event boundaries of the sequences. Newtson and Engquist (1976) conducted experiments on human perception and organization of events. In their experiments, they selected representative frames (shot boundaries) from movies and analyzed the descriptions of observers about the actions from these representative frames. When the representative frames were presented to the observers in sequence, they had more accuracy and confidence in their description compared to the presentation of these frames out of order.

There can be two types of forces applied to the object: *continuous* and *discontinuous*. A discontinuous force (force being a function of time) can be either a *step* (Fig. 1(a)) or an *impulse* (Fig. 1(b)) function; whereas a continuous force can be a *non-smooth* (*ramp*, Fig. 1(c)) or a *smooth* function (Fig. 1(d)). According to the Newton's second law of motion

$$\mathbf{F} = m\mathbf{a} \tag{1}$$

where $\mathbf{F}$ is force, m is mass and $\mathbf{a}$ is acceleration. Assuming mass remains constant, any type of change in force results in the same type of change in acceleration. Since force is not a measurable quantity in images, we focus on speed and acceleration in our discussion.

Analysis of human perception shows that humans successfully perceive *start* and *stop instants* emerging from any type of acceleration (continuous or discontinuous) applied to the object (Rubin and Richards, 1985). Similarly, humans are also able to perceive *dynamic instants* resulting from a discontinuous (step or impulse) change in acceleration. However, *dynamic instants* that result from a continuous change in acceleration are not observed by humans (Rubin and Richards, 1985). We summarize this discussion in Table 1, where the first and second columns show the possible speed functions: continuous and discontinuous respectively, and the rows show the instants: *start*, *stop*, and *dynamic*.

Table 1. Classes of motion boundaries, based on the types of speed functions, which are detectable or undetectable by human observers.

	Continuous speed	Discontinuous speed
Start instant	Detectable (Fig. 3(a) and (b))	Detectable (Fig. 3(c))
Stop instant	Detectable	Detectable
Dynamic instant	Undetectable	Detectable (Fig. 5)

Among the six categories of motion tabulated in Table 1, in Figs. 3 and 5 we show the five categories of motion detected by human observers (stop instants can be categorized as inverse of start instants).

In the next section, we propose a mathematical model to detect the *instants* described in this section.

3.2. Spatio-Temporal Curvature

A *motion trajectory* in 3-D is composed of positions of the object for consecutive time instants in the position vector, given by

$$\mathbf{r}(t) = [x(t)\ \ y(t)\ \ z(t)] \tag{2}$$

This vector describes the motion in space and time. The quantitative measure of motion is given by the first derivative of position, *velocity*, $\mathbf{v}(t)$, and the second derivative of position, *acceleration*, $\mathbf{a}(t)$, vectors.

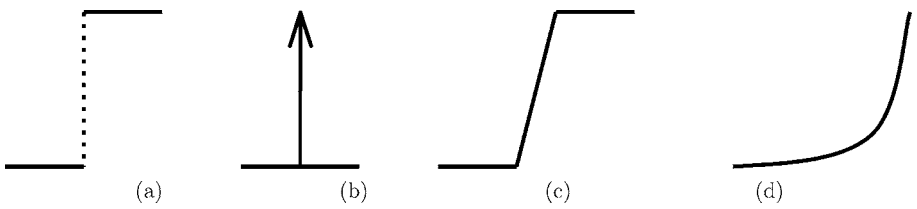

Figure 1. Possible functions for force and speed; (a) non-smooth-discontinuous (step function); (b) non-smooth-discontinuous (impulse function); (c) non-smooth-continuous (ramp function); and (d) smooth-continuous.

Velocity is the tangent vector to the 3-D curve of the motion trajectory at position $\mathbf{r}(t)$ given in Eq. (2).

Due to the kinematics of the human muscle structure, the forces applied by the muscles have discontinuities in time (Fig. 1(a) and (b)). According to the Newton's second law, given in Eq. (1), the discontinuities in force are reflected as discontinuities in acceleration. Since the velocity is the integral of the acceleration, impulse and step acceleration functions result in the step and ramp functions in velocity.

In our approach, following the theorem stated in Rubin and Richards (1985),

Theorem 1. *The continuities and discontinuities in position, velocity and acceleration in the* 3*-D trajectory of a moving object are preserved in* 2*-D image trajectories under a continuous projection function.*

We consider the 2-D projection of the 3-D trajectory using the affine projection model, which is a valid assumption for most surveillance systems and suits the purpose of obtaining view-invariant characteristics. For the proof of this theorem, we refer readers to read (Rubin and Richards, 1985). In Fig. 2, the affine projection of the 3-D motion trajectory of the opening cabinet action is shown, where x and y axis are *spatial axes* and the vertical axis is the *time axis*. Since each point on the 2-D trajectory represents the position of the object in consecutive time instants, it is a spatio-temporal curve, which is defined by,

$$\mathbf{r}_{st}(t) = [x(t),\ y(t),\ t] \tag{3}$$

where $z(t) = t$ and $1 \leq t \leq n$, n being the number of frames in the sequence. Using the definition of Eq. (3), the spatio-temporal velocity and acceleration are defined by

$$\mathbf{v} = \mathbf{r}'_{st}(t) = [x'(t)\ \ y'(t)\ \ 1], \tag{4}$$

$$\mathbf{a} = \mathbf{r}''_{st}(t) = [x''(t)\ \ y''(t)\ \ 0]. \tag{5}$$

To detect elementary components of a motion trajectory, which we call *instants*, it is important to find the discontinuities in velocity $\mathbf{v}(t)$, acceleration $\mathbf{a}(t)$ and position $\mathbf{r}(t)$. The proposed approach uses a measure that encapsulates all this information in one quantity. Besides the position $\mathbf{r}(t)$ on a curve, another important measure is the curvature, κ, at time t, which is given by

$$\kappa(t) = \frac{\|\mathbf{r}'(t) \times \mathbf{r}''(t)\|}{\|\mathbf{r}'(t)\|^3}, \tag{6}$$

where '$\times$' represents the cross product. In Eq. (6), $\mathbf{r}'(t)$, $\mathbf{r}''(t)$ and $\|\mathbf{r}'(t)\|$ respectively represents velocity, acceleration and speed. Since speed $\|\mathbf{r}'(t)\|$ is the arclength, Δs, travelled by the object in unit time $\Delta t = 1$, and is given by

$$\Delta s = \sqrt{(\Delta x)^2 + (\Delta y)^2 + 1} \tag{7}$$

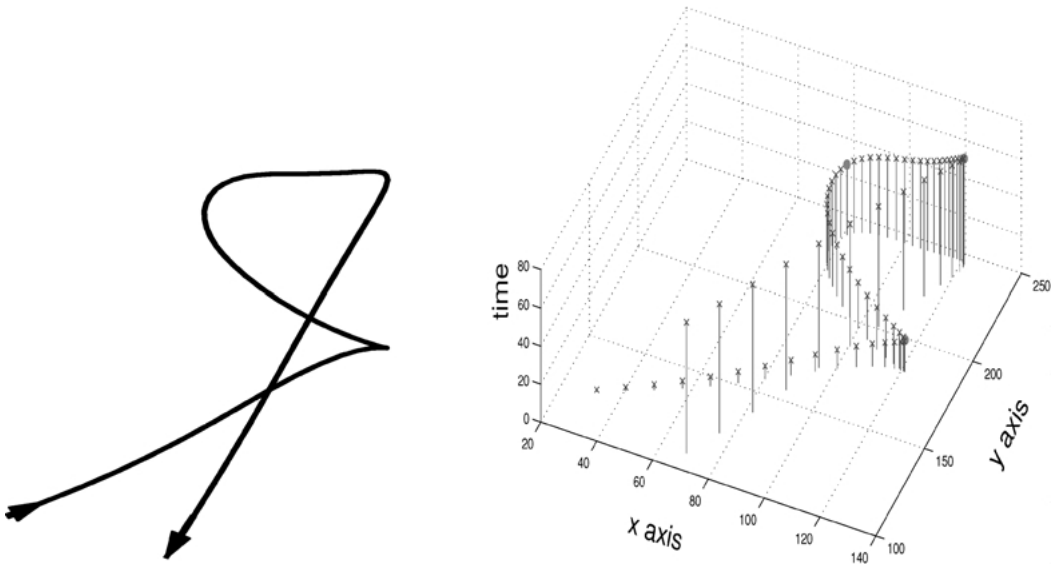

Figure 2. Spatio-temporal representation of "opening a cabinet action": Left is spatial view, right is spatio-temporal view where each time instant is an impulse and the length of the impulse is the position in time.

which captures the position change of the moving object.

Substituting the definitions of velocity and acceleration from Eqs. (4) and (5) in the curvature function given in Eq. (6), we obtain the spatio-temporal curvature:

$$\kappa = \frac{\|[x'(t)\ \ y'(t)\ \ 1] \times [x''(t)\ \ y''(t)\ \ 0]\|}{\|[x'(t)\ \ y'(t)\ \ 1]\|^3} \tag{8}$$

After some manipulations, the spatio-temporal curvature can be rewritten as:

$$\kappa(t) = \frac{\sqrt{y''(t)^2 + x''(t)^2 + (x'(t)y''(t) - x''(t)y'(t))^2}}{(\sqrt{x'(t)^2 + y'(t)^2 + 1})^3}. \tag{9}$$

In the next section, we will show how the spatio-temporal curvature can capture the *instants* that humans are capable of perceiving. We also present some synthetic examples supporting our argument.

3.3. How Spatio-Temporal Curvature Captures Motion Boundaries

Instants, which are elementary components of motion, segment the motion trajectory into *intervals*. In Section 3.1, it was discussed that human observers are able to perceive *start*, *stop* and *dynamic instants* that stem from discontinuities in velocity or acceleration during the activity. However, it was also noted that human observers fail to observe *instants*, which happen due to the continuous speed change during the activity. In the following discussion, we analyze the spatio-temporal curvature's ability to capture *instants* that humans are able to perceive.

For simplicity, we continue the analysis of spatio-temporal curvature in the one dimensional case. One-dimensional temporal curvature, using Eq. (9), is given by

$$\kappa_{1D} = \frac{|x''(t)|}{(x'(t)^2 + 1)^{\frac{3}{2}}}, \tag{10}$$

where $y(t)$ is set to a constant value, i.e. $y'(t) = y''(t) = 0$. A quick analysis of Eq. (10) can be done by looking at the effect of speed vector, $\mathbf{x}'(t)$, on the curvature with respect to the acceleration. Due to the higher exponent of speed ($\frac{3}{2} > 1$), and acceleration being the derivative of speed, an increase in speed will lower the value of curvature exponentially. To see the effect of speed and acceleration on detecting the motion boundaries (*instants*), we analyze five possible motion classes, which are observed by humans as motion boundaries (*instants*) listed in Table 1. These boundaries are shown in Figs. 3 and 5.

In Fig. 3, we show examples of *start instant* due to continuous and discontinuous speed changes. In Fig. 3(a) and (b), before the object starts its motion the spatio-temporal curvature given in Eq. (10) is $\kappa_{1D} = 0$. At the time instant when the object starts moving, the curvature becomes $\kappa_{1D} > 0$. Since the effect of increase in the speed is exponential in Eq. (10), the curvature reduces to $\kappa_{1D} \approx 0$, which results in a peak in κ_{1D}. A similar effect also holds for Fig. 3(c), where the motion starts due to a discontinuous force on the object. Peaks in curvature for both of start instants relate to the

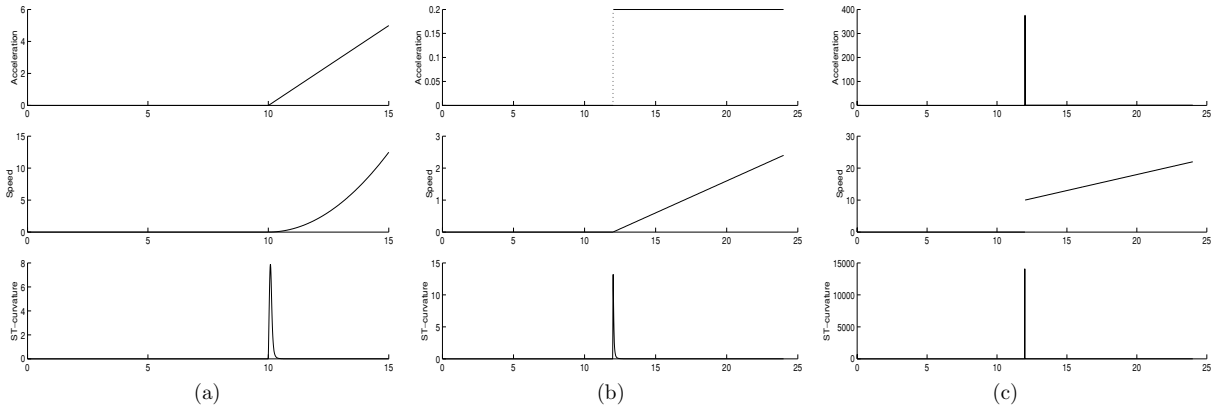

Figure 3. Three examples of *start instants* due to smooth continuous (a), non-smooth continuous (b), and discontinuous (c) speed, which changes the state of object from rest to active. Corresponding accelerations and spatio-temporal curvatures (κ_{1D}) are also shown.

motion boundaries (*instants*) that humans are also able to perceive.

So far we have shown how spatio-temporal curvature captures the psychological motion boundaries that occur due to *start* or *stop instants*. Another class of motion boundaries, which is independent of starts and stops, is the *dynamic instant* that happens due to the force applied to a moving object (active state of the motion). Humans, however, are only able to perceive one type of *dynamic instant*, as was discussed in Section 3.1. The diagrams in Fig. 5 show the types of *dynamic instants* that humans are able to perceive, which are also captured by the spatio-temporal curvature κ_{1D}. In Fig. 5, we show a complete set of speed discontinuities of an object that result in *dynamic instants*, rather than showing a complete set of the infinite number of ways that force changes can be applied to an object.

The construction of this complete set of speed discontinuities is obtained as follows: let s_a and s_b, be the speed discontinuity values as shown in Fig. 4. The speed function before or after the discontinuity can be either increasing or decreasing. We represent increasing speed by $s^{\uparrow}$ and decreasing speed by $s^{\downarrow}$. Thus one of the speed discontinuities can be given by: $(s^{\uparrow}s^{\uparrow}, s_a < s_b)$, which is interpreted as an increase in the speed before the discontinuity, an increase after the discontinuity, and at the discontinuity $s_a < s_b$. Other discontinuities are: $(s^{\uparrow}s^{\uparrow}, s_a > s_b)$, $(s^{\downarrow}s^{\uparrow}, s_a > s_b)$, $(s^{\downarrow}s^{\uparrow}, s_a < s_b)$, $(s^{\downarrow}s^{\downarrow}, s_a > s_b)$ $(s^{\downarrow}s^{\downarrow}, s_a < s_b)$, $(s^{\uparrow}s^{\downarrow}, s_a < s_b)$, and $(s^{\uparrow}s^{\downarrow}, s_a > s_b)$.

The above discussions on detecting discontinuities using curvature of trajectories deals with continuous functions. However in video sequences, we deal with discrete functions, which are sampled version of continuous functions. Therefore issue related to sampling rate, which can be referred as number of frames per second, becomes important.

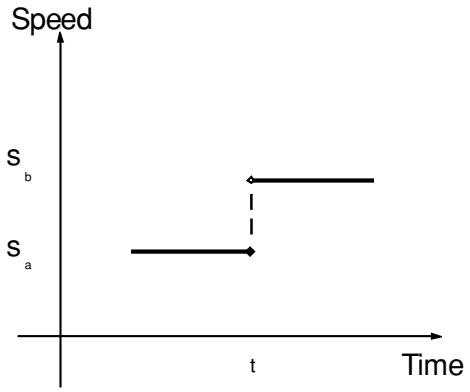

Figure 4. Discontinuity values in the speed function.

In general, high sampling rate improves the performance of the instant detection. If we don't have sufficiently large number of samples, viewing the action from different view points will result in false instant detection, especially when the image plane and the action plane are close to being perpendicular. For the experiments on the actions presented in this paper, we found that 24 fps. is sufficient to achieve view invariance in instant detection.

In the next section, we discuss previously proposed approaches related to detecting *instants* in a motion trajectory and give their drawbacks.

3.4. Previous Approaches

For extracting the *instants* from a 2-D projected motion trajectory, Rubin and Richards (1985) considered the change of velocity in polar coordinates, where the magnitude of velocity vector is the speed, $s(t)$, and the angle is the direction, $\Theta(t)$, of motion. In their approach for obtaining the motion boundaries (which we call *instants*), they compute the zerocrossings of the second derivatives of both $s(t)$ and $\Theta(t)$. Since the changes in velocity and speed are not always temporally aligned, the important problem with this approach is how to combine these two pieces of information in a meaningful manner. For example, the union of speed and direction instants results in too many instants, while the intersection results in too few instants. This issue was never addressed by Rubin and Richards.

Detection of *instants* was also addressed by Gould and Shah (1989). Instead of using the polar coordinates, they used the velocity vector $\mathbf{v}(t) = [v_x, v_y]$ for *instant* detection. They also introduced Trajectory Primal Sketch (TPS), such that significant changes are identified by the strength of zerocrossings of v_x and v_y computed at various scales. This process results in a set of TPS contours, where each contour corresponds to an *instant*. However, union of instants obtained from v_x and v_y also suffers from the temporal alignment problem.

3.5. Generating and Smoothing of Trajectories

In order to find the action performed by the hand, we construct the trajectory of the moving hand by marking

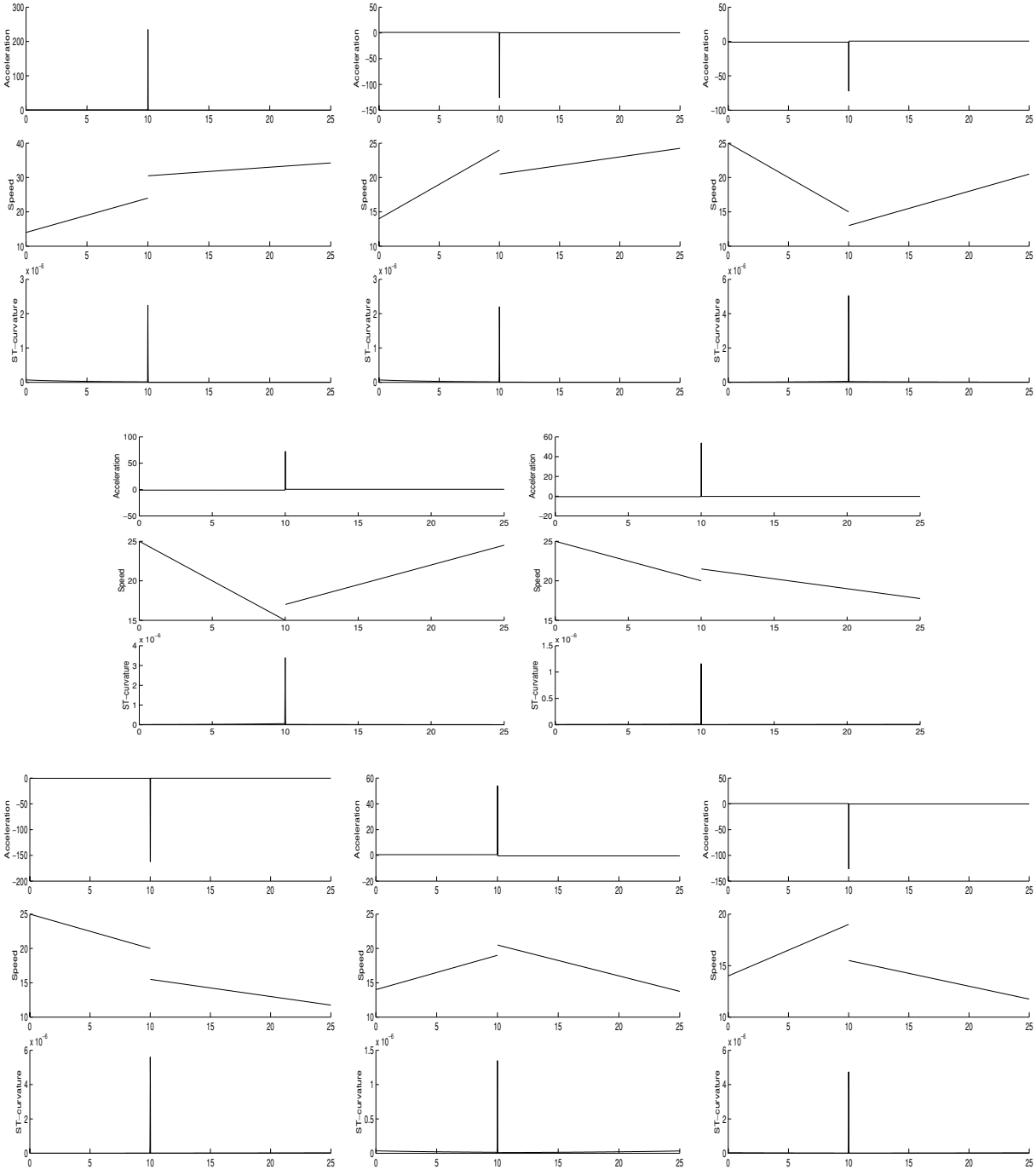

Figure 5. Eight possible *dynamic* boundaries that occur due to the non-smooth step change in speed. Corresponding accelerations and spatio-temporal curvatures are also shown.

the hand position in consecutive frames. For finding the position of the hand we extract the region corresponding to the hand in an image sequence by applying the skin detection method discussed in Kjeldesn and Kender (1996). Skin detection is based on the color predicates of the skin. The color predicates are computed to form a lookup table from a training set of skin and non-skin regions. The incoming pixels are then

labelled as skin or non-skin by using the lookup table operations. The skin detection only gives the approximate hand region. Therefore, after skin detection, we use the mean-shift tracking of Comaniciu et al. (2000) to obtain the spatio-temporal trajectory of the hand. Mean-shift tracking is based on maximizing the likelihood of the model (hand) intensity distribution and the candidate intensity distribution using

$$\rho(\mathbf{m}) = \sum_{u=1}^{n} \sqrt{q_u\, p_u(\mathbf{m})}, \tag{11}$$

where $\mathbf{m}$ is the center of the hand region, n is the number of bins in the distribution, and q_u and p_u are the weighted histograms of the model and candidate respectively. The weights for the histograms are obtained using the Epanechnikov Kernel given by

$$K(\mathbf{x}) = \frac{1}{2} c_d^{-1}(d+2)(1 - \|\mathbf{x}\|^2), \tag{12}$$

where $\mathbf{x}$ is a d-dimensional vector, c_d is the volume of a d-dimensional sphere and $\|\cdot\|$ is the magnitude operator. The center of the hand region in the next frame is found using

$$\mathbf{m}_{new} = \frac{\sum_{\mathbf{x}_i \in S} w_i(\mathbf{m} - \mathbf{x}_i)}{\sum w_i} + \mathbf{m}_{old} \tag{13}$$

where S is the image patch and w_i are the weights computed using

$$w_i = \sum_{u=1}^{n} \delta\,(S(\mathbf{x}_i - u) \sqrt{\frac{q_u}{p_u(\mathbf{m})}}, \tag{14}$$

where δ is the *Kronecker delta* function.

As discussed in Section 3.2, the trajectory is a spatio-temporal curve defined by: $(x[1], y[1], t[1])$, $(x[2], y[2], t[2]), \ldots, (x[n], y[n], t[n])$. The spatio-temporal curve contains noise due to the errors in skin detection, tracking, lighting conditions, projection distortions, occlusions, etc. Although, there are filters available in the literature to reduce noise, such as low-pass, mean filter, etc., they are not suitable for removing the noise in a spatio-temporal trajectory because these filters tend to smooth all the peaks, which may represent important changes in a trajectory. Therefore, we use anisotropic diffusion for smoothing $x(t)$ and $y(t)$ of the trajectory. Anisotropic diffusion was proposed in the context of scale space (Perona and Malik, 1990).

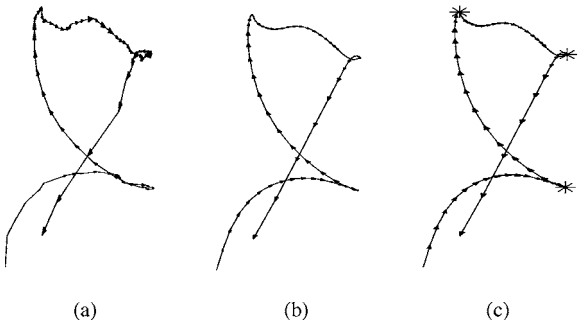

Figure 6. (a) Opening overhead cabinet trajectory; (b) smoothed version using anisotropic smoothing; and (c) *dynamic instants* marked by $*$.

This method iteratively smoothes the data with a Gaussian kernel, but adaptively changes the variance of the Gaussian based on the gradient of a signal at a current point as follows:

$$I_i^{t+1} = I_i^t + \lambda\big(c_N^t \bullet \nabla_N I^t + c_S^t \bullet \nabla_S I_i^t\big), \tag{15}$$

where $0 \le \lambda \le \frac{1}{4}$ is the control parameter (we chose 0.2 in our experiments), t represents the iteration number and

$$\nabla_N I = I_{i-1} - I_i,$$
$$\nabla_S I = I_{i+1} - I_i.$$

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

$$c_N^t = g\big(\nabla_N I_i^t\big)$$
$$c_S^t = g\big(\nabla_S I_i^t\big)$$

where $g(\nabla I) = e^{-(\frac{\|\nabla I\|}{k})^2}$. We chose noise estimator $k = 40$ in our experiments.

We show a captured spatio-temporal trajectory in Fig. 6(a) and the trajectory smoothed using anisotropic diffusion in Fig. 6(b). Notice that the processed trajectory is much smoother and the important events (such as direction changes and speed discontinuities) are still maintained.

4. Representation

Representation, which is an abstraction of the sensory data that reflects a real world situation, is an important and sometimes difficult aspect of an intelligent

system. The representation of data should not only be view-invariant and compact but also be reliable for later processing.

For high level data abstraction, we propose a new representation scheme based on the spatio-temporal curvature of a motion trajectory. Our representation of trajectory includes a sequence of *dynamic instants* and *intervals*, and assigns physical meanings to them.

A *dynamic instant* is an instantaneous entity that occurs for only one frame, and represents an important change in the motion characteristics (speed, direction and acceleration). These changes are captured by the spatio-temporal curvature. We detect *dynamic instants* by identifying maxima in the spatio-temporal curvature. As long as instants are consistently detected for the same action performed by different people, the temporal extent (length) of the action will not change the representation. Therefore, there is no need to make temporal alignment for trajectories, which is commonly used in other recognition system and is a time consuming process.

In the proposed representation, a *dynamic instant* is characterized by its frame number, the image location and the sign. Among these characteristics, the "frame number" represents the time at which the *dynamic instant* occurs and the "image location" provides the spatial position of the hand in the frame when the *dynamic instant* occurs. The last characteristic, "sign", represents the change of the direction of motion at the instant. Examples of *dynamic instants* include: touching, twisting and loosening.

Similarly, an *interval* represents the time-period between any two *dynamic instants*, during which the motion characteristics does not change drastically. Examples for *intervals* include approaching, lifting, pushing, and receding etc.

A remarkable feature of our representation is that it is able to explain an action in a natural language in terms of meaningful atomic units, which can not only be mathematically modelled but can also be detected in real images. In Fig. 7, we show the *dynamic instants* by $*$ on the motion trajectory of the opening overhead cabinet action. This action can be described as: the hand approaches the cabinet (approaching interval), the hand makes contact with the cabinet (touching instant), the hand lifts the cabinet door (lifting interval), the hand twists (twisting instant) the wrist, the hand pushes (pushing interval) the cabinet door in, the hand breaks the contact (loosening instant) with the door, and finally the hand recedes (receding interval) from the cabinet.

Figure 8 displays the trajectory of the "erasing whiteboard" action. This action can be described as: the hand approaches the eraser (approaching interval), the hand makes contact with the eraser (touching instant), the hand picks up the eraser (picking interval), the hand turns (turning 1 instant), the hand wipes the board (wiping interval), the hand turns (turning 2 instant), the hand wipes (wiping interval), the hand turns (turning 3 instant), the hand wipes (wiping interval), the hand turns (turning 4 instant), the hand puts the eraser back (putting down interval), the hand breaks the contact (loosening instant) with the eraser, and finally the hand recedes (receding interval) from the board.

Figure 9 shows the trajectory of "picking up an object from the floor and then putting it down on the desk". The action can be described as: the hand approaches the object (approaching interval), makes contact with the object (touching instant), picks it up (picking interval), breaks the contact (loosening instant), and then recedes (receding interval).

If a representation has view-invariant characteristics, then a higher level interpretation of the information can be performed without any ambiguity. In the next section, we show the view- invariance property of the proposed representation.

4.1. View-Invariance

As discussed in Section 3.2, the discontinuities in 3-D, which are perceived as *instants* by human observers, are always projected as discontinuities in 2-D (Theorem 1). Therefore, *instants*, which are the maxima in spatio-temporal curvature of a trajectory, are also view-invariant, except in limited cases of accidental alignment. By accidental alignment, we mean a view direction which is parallel to the plane where the action is being performed. In this case, the centroids of a hand region in consecutive frames are projected at the same position in the image plane, resulting in a 2-D trajectory which is essentially a single point. Examples of *instants* in trajectories of opening and closing the overhead cabinet action are given in Fig. 10 for different views. Even though these trajectories look quite different, three *dynamic instants* for every view point are correctly detected by the proposed method.

In the following, we formally show the view-invariance of our representation. We will use the

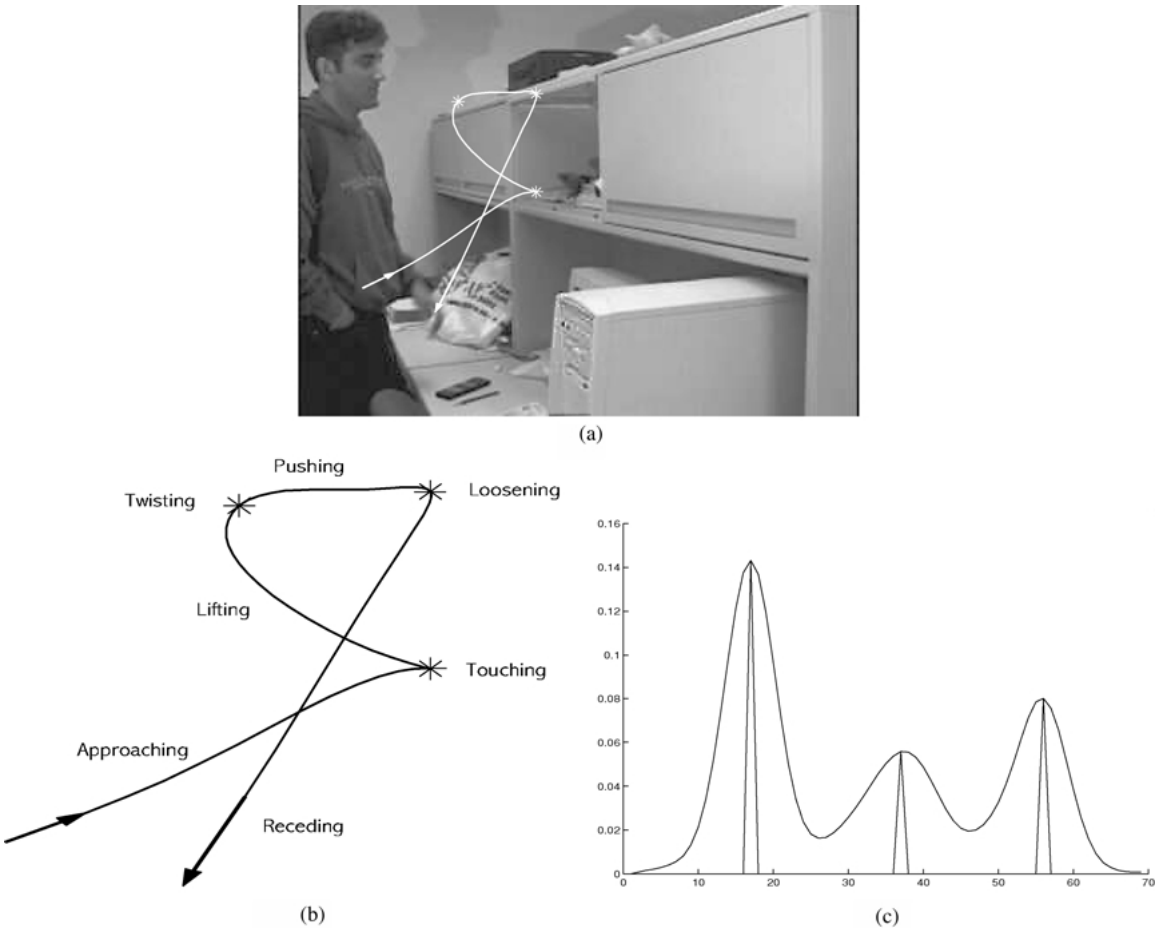

Figure 7. (a) The "opening a cabinet" action, the hand trajectory shown in white is super imposed on the first image; (b) a representation of the action trajectory in terms of *instants* and *intervals*; and (c) corresponding spatio-temporal curvature values and detected maximums (dynamic instants).

affine projection model, which assumes that the depth of the 3-D trajectory of the action is small compared to the viewing distance (Mundy and Zisserman, 1992).

Assume that the location of a hand in 3-D space at times t_1, t_2 and t_3 is given by P_1, P_2 and P_3. In this case, we have two vectors $\overrightarrow{P_1P_2}$ and $\overrightarrow{P_2P_3}$ (see Fig. 11(a)). The projection of these three points in the image plane is shown in Fig. 11(b). It is clear that there is a *dynamic instant* at t_2, due to the significant change in the direction. Assume that the angle between the vectors is α. The sign of this angle can be determined by computing the sign of the cross product of the projection of the two vectors in the image plane. We will use this sign as the sign of the *instant*. We claim that the sign of the *instant* is view-invariant under the affine camera model if the camera viewpoint remains in the upper hemisphere of the viewing sphere. This is explained in the following.

The camera translation does not affect the angle α, therefore we will only consider the situation when the camera rotates. Let us assume, for simplicity, that the camera axes pass through P_2 and is perpendicular to X-Y plane. The distance from the camera to P_2 is D, and $\overrightarrow{P_1P_2}$ is always vertical. It is obvious that the camera rotation around the Z-axis does not change α. Therefore, the situations that need to be considered are the camera rotations around the X-axis (*tilt*) and the Y-axis (*pan*). While the camera pans, the only part that changes is the projection of $P_3(X_3, Y_3, Z_3)$, which becomes $P_3'(u_3', v_3')$ in Fig. 11(b). Note that P_0 is the projection of P_3 on the line P_1P_2 and its image coordinates are (u_0, v_0). When the camera pans by angle Ω, the X-coordinate of any point is changed to X' as

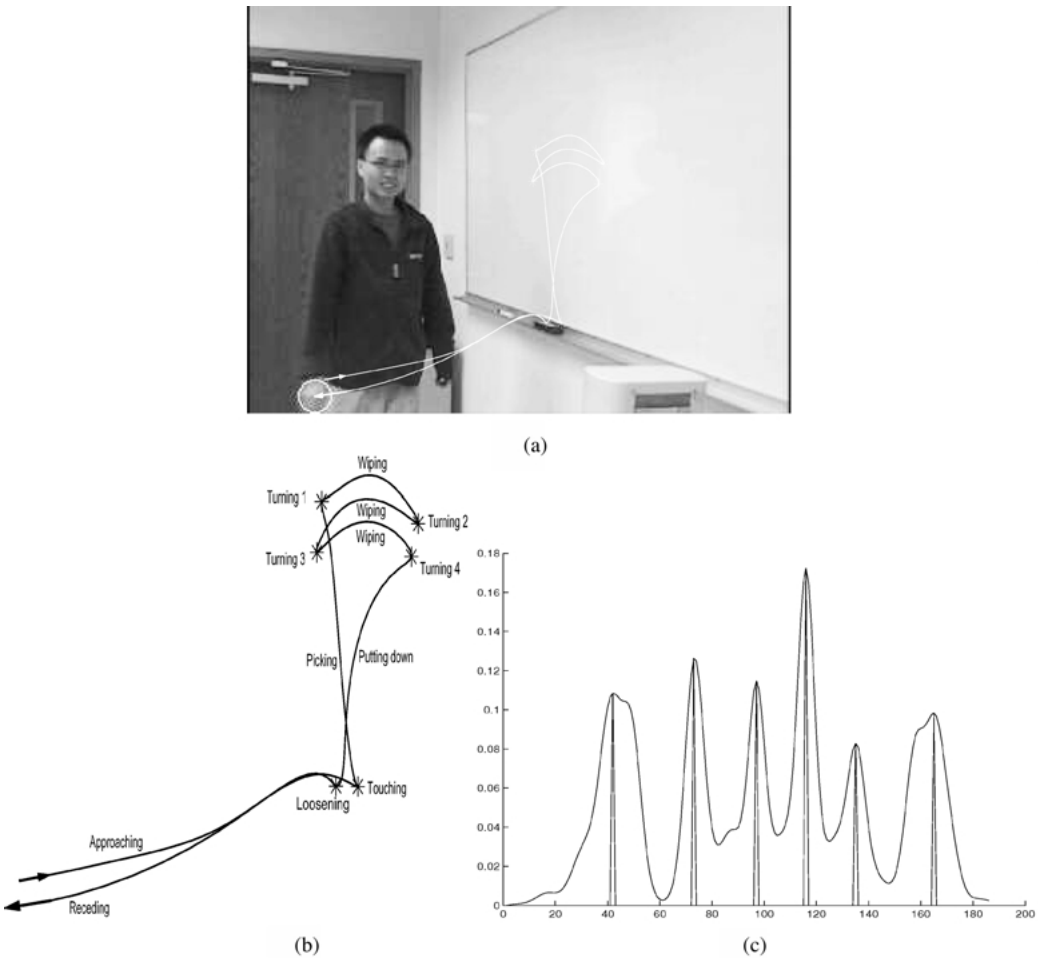

Figure 8. (a) The "erasing whiteboard" action, the hand trajectory shown in white is super imposed on the first image; (b) a representation of the action trajectory in terms of *instants* and *intervals*; and (c) corresponding spatio-temporal curvature values and detected maximums (dynamic instants).

follows,

$$X' = X\cos\Omega - Z\cos\Omega \tag{16}$$

The image coordinate under the affine camera model is given by,

$$u' = f\frac{X'}{D} \tag{17}$$

where f is the focal length and D is the distance from the camera to P_2. The distance between the projections of points P_3 and P_0 in the image plane is given by,

$$d' = u'_3 - u'_0$$

$$d' = f\frac{[(X_3\cos\Omega - Z\sin\Omega) - (X_0\cos\Omega - Z\sin\Omega)]}{D}$$

$$d' = f\frac{(X_3 - X_0)\cos\Omega}{D} \tag{18}$$

In Eq. (18), if $\Omega \in (-90^\circ, +90^\circ)$ then $\cos\Omega$ is positive, and the rest of elements on the right hand side of equation are constant, therefore, d' retains its sign. Furthermore, α retains it sign. This means that the sign of α is view-invariant when the camera pans within a semi-circle.

For the situation when the camera tilts around the X-axis, a similar argument holds. Therefore, when the camera tilts within a semi-circle $\varphi \in (-90^\circ, +90^\circ)$, the sign of φ remains the same. Moreover, the pan and tilt can be combined together to make the camera rotate around an arbitrary axis in the X-Y plane.

The above discussion deals with the situations when all the *instants* are located in one plane, which is restricted. However, we can extend this reasoning for more general situations as follows.

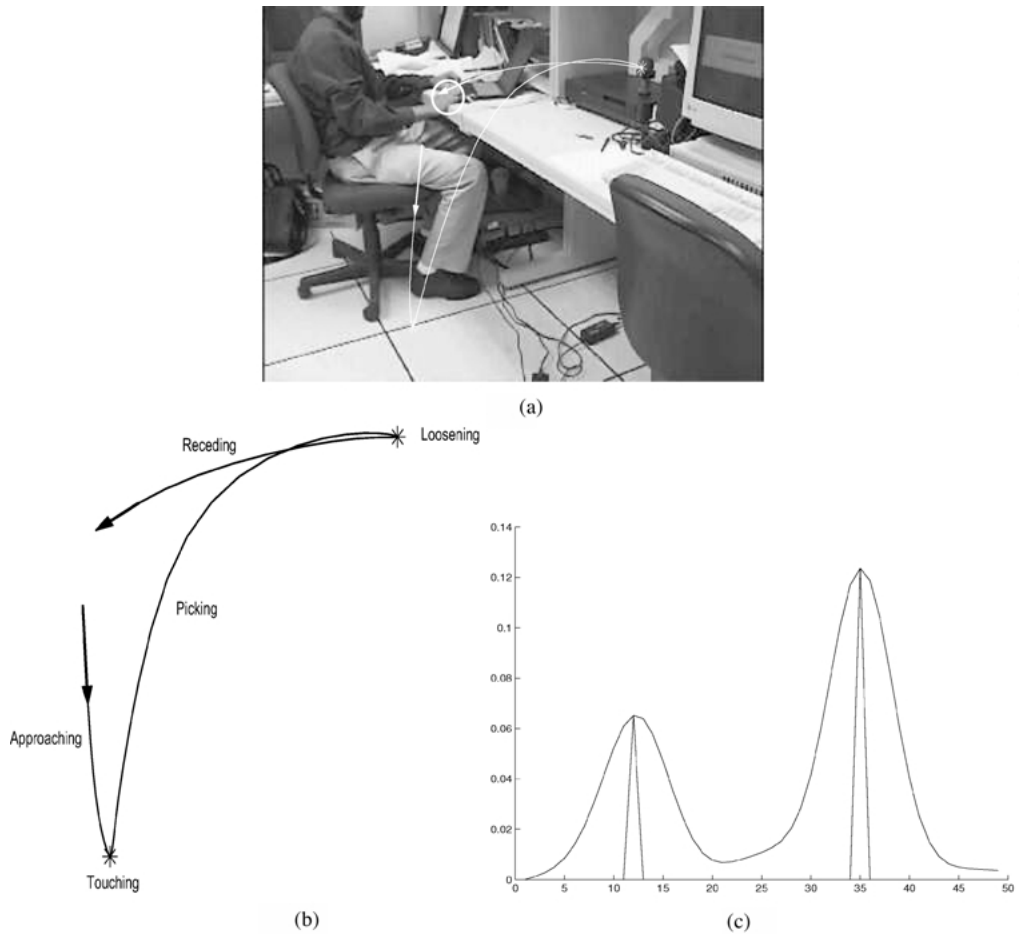

Figure 9. (a) The "picking up and putting down object" action, the hand trajectory shown in white is super imposed on the last image; (b) a representation of the action trajectory in terms of *instants* and *intervals*; and (c) corresponding spatio-temporal curvature values and detected maximums (dynamic instants).

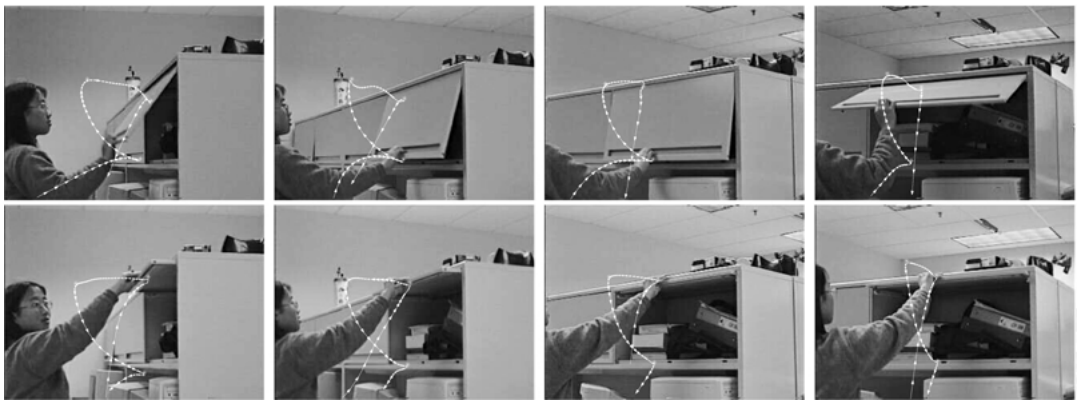

Figure 10. Trajectories from different view points for opening (top) and closing (bottom) overhead cabinet action. Both the opening and closing actions in the same column are taken at the same viewpoint.

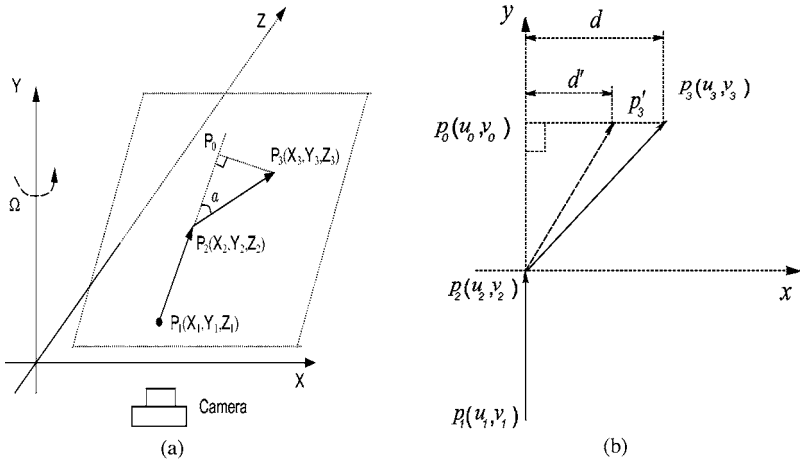

Figure 11. (a) Three points in 3-D and (b) the 2-D projections on the image plane.

Assume that there are four *instants* P_1, P_2, P_3 and P_4. Among these *instants*, P_1, P_2 and P_3 are in one plane and form an angle α, and P_2, P_3 and P_4 are in another plane and form an angle β. Then the signs of α and β are invariant when the camera stays within the space of a sphere defined by the two planes intersecting the sphere. For the situations when more non-planar *instants* are involved, the invariance property of the

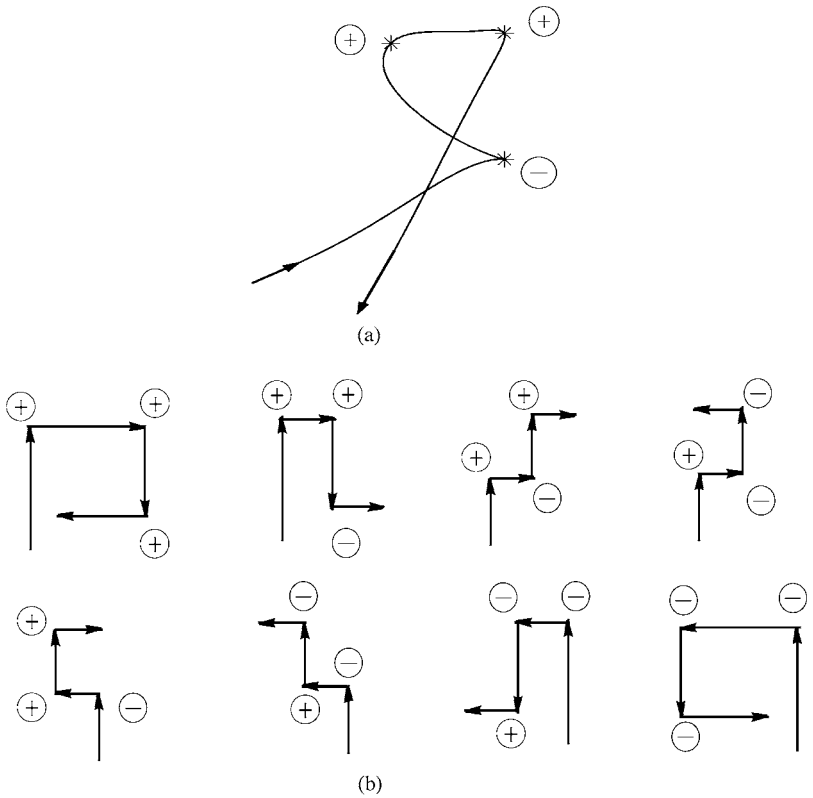

Figure 12. (a) Trajectory of the "opening cabinet" and the signs of the *instants* and (b) Possible permutations of 5-*instant* actions.

method is limited to the region where the camera can move without crossing these planes. However, we believe this representation is adequate, since there are not many cases in human actions which generate many non-planar *instants*.

The sign characteristic of an *instant*, which defines the direction of turns in the action, is very useful in distinguishing different actions captured from different viewpoints. We denote clock-wise turn by "+" and counter clock-wise turn by "−". For example, the "opening cabinet" action (Fig. 6(c)) has five *instants*, and the signs for the second, third and fourth *instants* are $(-, +, +)$ (Fig. 12(a)). On the other hand, the "closing cabinet" action (Fig. 10 bottom) also has five *instants*, but the signs of the middle three *instants* are $(-, -, +)$. In general, for a trajectory with n *instants*, the number of permutations of signs is 2^{n-2} (Fig. 12(b)). Note that, we are not considering the signs of the first and the last *instants*.

From the previous discussion, we can conclude that the number of *instants* and the signs of *instants* in an action are view-invariant. However, these two characteristics of *instants* are not sufficient to uniquely define any action; since two different actions may have the same number of *instants* with the same signs. Therefore, we propose to use a view-invariant method to measure the similarity between two actions that belong to the same category. The trajectories of the same action should give us a high matching score as compared to the trajectories of different actions. Also, the camera viewpoint should not affect the matching scores, that is, the action can be performed in an arbitrary field of view with any camera orientation. The matching algorithm is discussed in detail in Section 5.1.

5. Learning

Once the representation has been defined, the next step is to use this representation to learn human

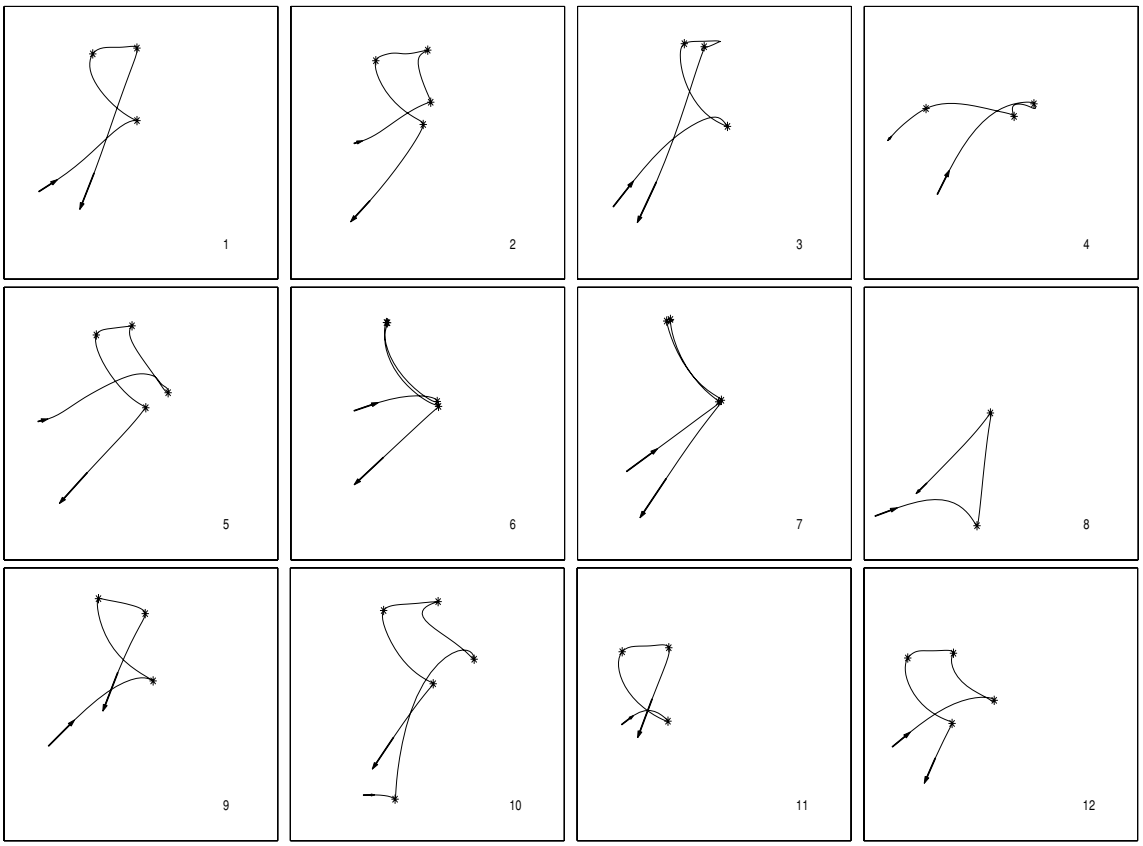

Figure 13. Trajectories of actions 1 to 16. The *instants* are shown with "∗", the definitions of these actions are given in Table 2.

actions. As stated earlier, our aim is to start with no model, and incrementally build a model of actions by continuously watching. We believe, children learn to recognize different actions by repeatedly observing adults perform actions.

We assume that the camera is fixed. However, people can enter the field of view from any side and perform actions with any orientation. The system is continuously analyzing a video stream captured by the camera. The system detects a hand using skin detection, determines the hand trajectory, and computes a view- invariant representation of each action.

For each action, the system builds a view-invariant representation and places it into a corresponding category of actions, depending on the number of *instants* and the permutation of the signs. The system also compares each action with all other actions in its category.

At a higher level of abstraction, the system also determines sets of similar actions based on the match scores. For example, different cases of the "opening overhead cabinet" action can be automatically determined to be similar. For each set, only one prototype representation is maintained, since all other *instances* convey the same information. For each prototype we associate a confidence, which is proportional to the cardinality of the set represented by this prototype. When more evidence is gathered, the confidence of some actions is increased, while the confidence of others remains the same. The prototypes with small confidence can ultimately be eliminated.

5.1. Matching

Given two viewpoint-invariant representations of some actions, how can we determine if these are the same

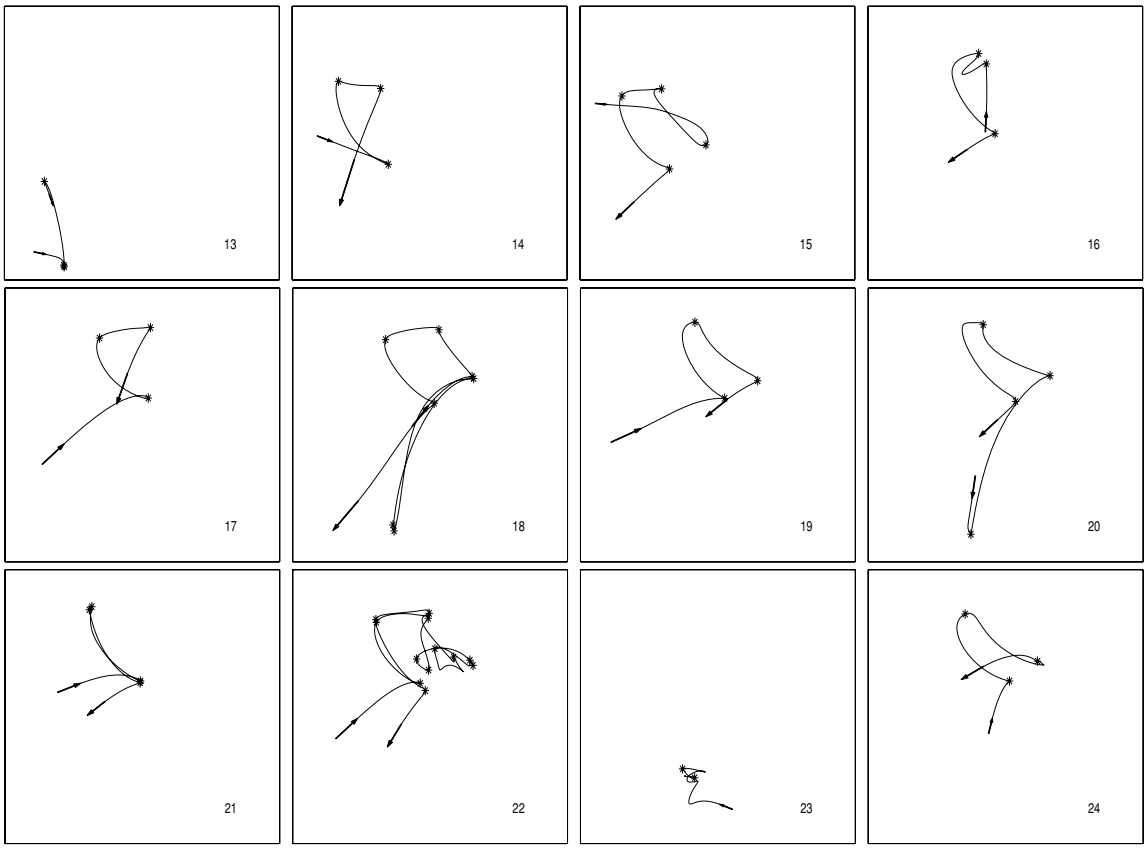

Figure 14. Trajectories of actions 12 to 24. The *instants* are shown with "$*$", the definitions of these actions are given in Table 3.

actions? It is obvious that two actions with a different number of *instants* or different sign permutations cannot be the same. Therefore, we should only match representations with the equal number of *instants* and the same sign permutation. We want to note that one action can be a sub-action of the other. Though these actions do not have an equal number of *instants*; the match is meaningful. However, in this paper, we do not deal with this case.

Our proposed method represents actions as a sequence of *instants*. Let there be n such *instants* denoted by image coordinates $(x_i, y_i)^T$ where $i \leq n, n \geq 5$ and $(x_i, y_i)^T$ is the world-centered coordinates (Tomasi and Kanade, 1992). Assume a particular action is captured in k views represented by: $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_k$ where $\mathbf{M} = ((x_1, y_1)^T, (x_2, y_2)^T, \ldots, (x_n, y_n)^T)$. If we denote this observation matrix by $\mathbf{M} = [\mathbf{M}_1\ \mathbf{M}_2 \ldots \mathbf{M}_k]^T$, we can decompose $\mathbf{M}$ matrix into shape matrix $\mathbf{S}$ and projection matrix $\mathbf{P}$ as proposed Tomasi and Kanade (1992) such that,

$$\mathbf{M} = \mathbf{P} \cdot \mathbf{S} = \begin{pmatrix} \Pi_1 \\ \vdots \\ \Pi_k \end{pmatrix} \mathbf{S} \tag{19}$$

where the shape matrix $\mathbf{S}$ represents the 3-D coordinates of points corresponding to the *instants* and Π_i is the projection matrix of action i. Tomasi and Kanade used this idea in the context of structure from motion problem, whereas we use it for actions which are represented by spatial positions of *instants*. In context of structure from motion, Tomasi and Kanade used the fact that $\mathbf{rank}(\mathbf{M}) = \min(\mathbf{rank}(\mathbf{P}), \mathbf{rank}(\mathbf{S}))$, where the rank is defined in terms of non-zero singular values. Based on the rank theorem stated in Tomasi and Kanade (1991), ideally when there are no numerical errors $\mathbf{rank}(\mathbf{P}) \leq 3$ and $\mathbf{S} \leq 3$, therefore the rank of observation matrix is $\mathbf{rank}(\mathbf{M}) \leq 3$. Practically, these

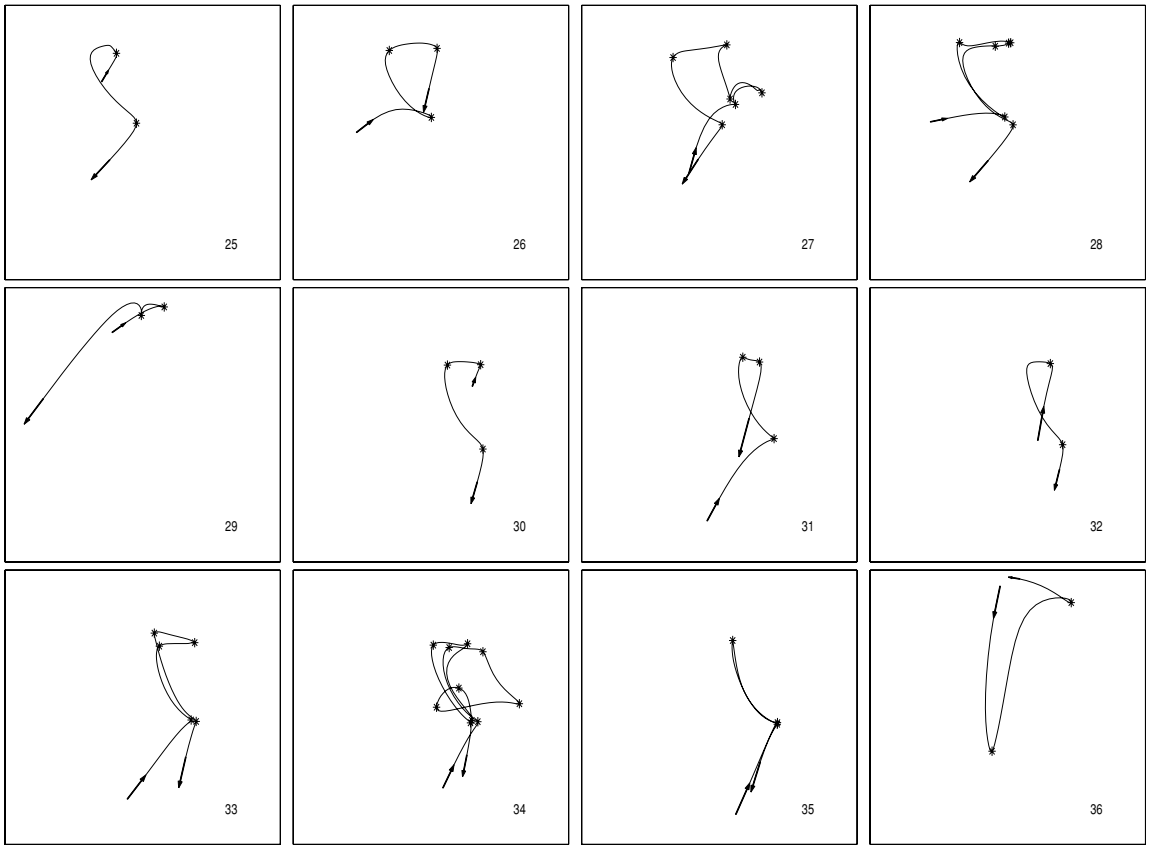

Figure 15. Trajectories of actions 25 to 36. The *instants* are shown with "$*$", the definitions of these actions are given in Table 4.

singular values may not exactly be zero. Based on this fact, Seitz and Dyer (1997) use the sum of the squares of singular values of M, except the first three singular values as the distance measure. This distance measure is formally given by

$$dist = \sqrt{\frac{1}{2kn}\sum_{4}^{n}\sigma_i^2}, \tag{20}$$

where $\sigma_i (i = 1 \ldots n)$ are the singular values of $\mathbf{M}$, k denotes the number of views, and n denotes the number of singular values. In Seitz and Dyer (1997) used this distance function to determine if a set of images match that is they represent different views of the same object. Here, we will use this distance measure to match two different views $\mathbf{M}_i$ and $\mathbf{M}_j$ of the same action, such that the distance computed gives the average amount necessary to additively perturb the coordinates of each *instant* in order to produce two projections of a single action.

To match two actions I_i and I_j, we form matrix M as follows:

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_i \\ \mathbf{M}_j \end{pmatrix} = \begin{pmatrix} x_1^i & x_2^i & \cdots & x_n^i \\ y_1^i & y_2^i & \cdots & y_n^i \\ x_1^j & x_2^j & \cdots & x_n^j \\ y_1^j & y_2^j & \cdots & y_n^j \end{pmatrix} \tag{21}$$

We then determine the singular values of M, and compute the distance (Eq. (20)) as $dist_{i,j} = |\sigma_4|$. This distance gives us the matching error of two action trajectories.

In our case $\mathbf{M}$ is $4 \times n$, therefore $\mathbf{rank}(\mathbf{M}) \leq 4$. However, if $\mathbf{rank}(\mathbf{M}) = 4$, then there is no linear dependency between the rows of $\mathbf{M}$ and actions I_i and I_j can be labeled as two different actions. Following this observation, we restate the theorem given in Seitz and Dyer (1997) in the context of actions:

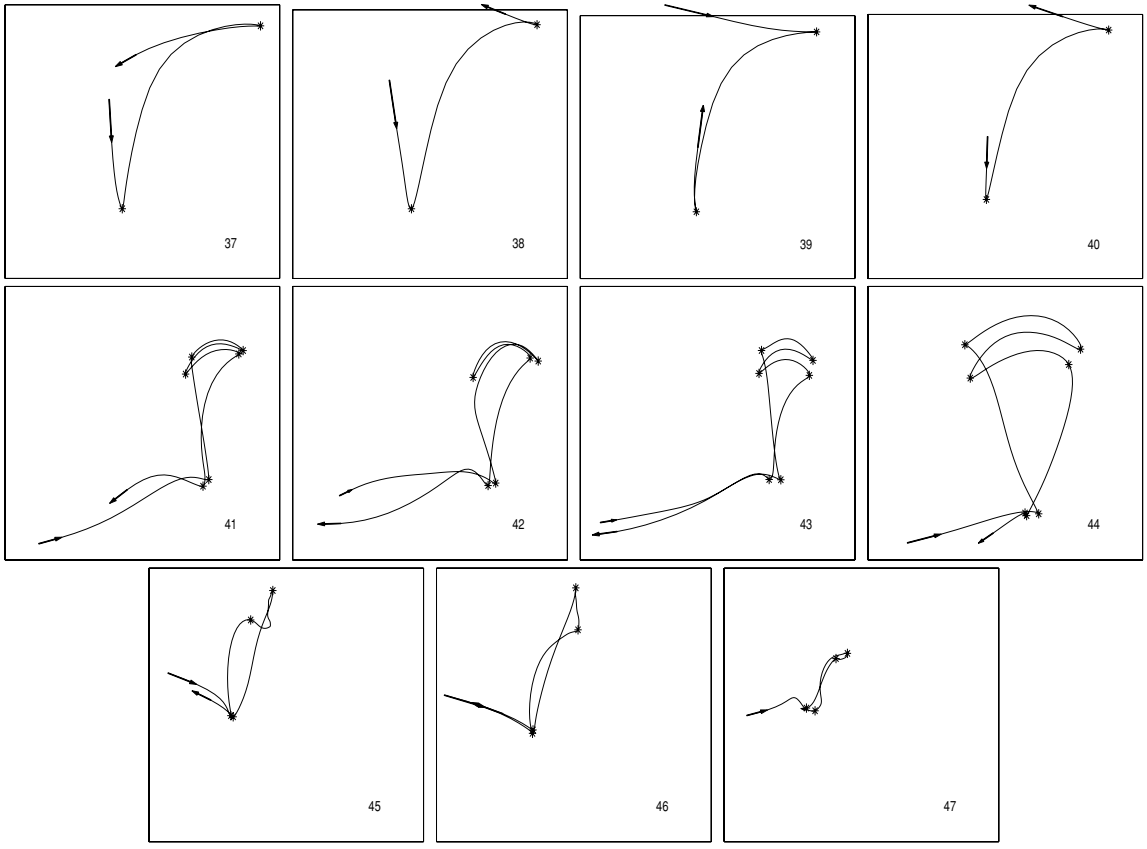

Figure 16. Trajectories of actions 37 to 47. The *instants* are shown with "$*$", the definitions of these actions are given in Table 5.

Figure 17. Action 37: Pick up an object from the floor and put it down on the desk (every 8th frame of the sequence is shown). The hand is highlighted with a white circle, and its trajectory is superimposed on the last frame.

Theorem 2. *Under the affine camera model, if* $1 < \mathbf{rank}(\mathbf{M}) \leq 3$ *and neither of actions has all instants aligned in a straight line, then the two actions* S_i *and* S_j *match and as a consequence actions* S_i *and* S_j *are linearly dependent.*

The matrices $\mathbf{S}$ and $\mathbf{P}$ can be constructed as follows:

$$\mathbf{P} = \begin{pmatrix} \Pi_i & 0 \\ 0 & \Pi_j \end{pmatrix} = \begin{pmatrix} a_{i1} & a_{i2} & a_{i3} & 0 & 0 & 0 \\ a_{i4} & a_{i5} & a_{i6} & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{j1} & a_{j2} & a_{j3} \\ 0 & 0 & 0 & a_{j4} & a_{j5} & a_{j6} \end{pmatrix}$$

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_i \\ \mathbf{S}_j \end{pmatrix} = \begin{pmatrix} X_{i1} & X_{i2} & \cdots & X_{in} \\ Y_{i1} & Y_{i2} & \cdots & Y_{in} \\ Z_{i1} & Z_{i2} & \cdots & Z_{in} \\ X_{j1} & X_{j2} & \cdots & X_{jn} \\ Y_{j1} & Y_{j2} & \cdots & Y_{jn} \\ Z_{j1} & Z_{j2} & \cdots & Z_{jn} \end{pmatrix}$$

where $[X_i, Y_i, Z_i]^T$ are the 3D coordinates of the observed actions.

To give an insight to the above theorem, we can extend the explanation given by Seitz and Dyer (1997). Due to decomposition of the matrix $\mathbf{M}$ into matrices $\mathbf{P}$ and $\mathbf{S}$ the rank of $\mathbf{M}$ is dependent on the ranks of $\mathbf{P}$ and $\mathbf{S}$. Matrix $\mathbf{P}$ is a full rank matrix, i.e. $\mathbf{rank}(\mathbf{P}) = 4$, otherwise one of the projections of the 3D actions i or j will be on a line, which contradicts with the statement of theorem. Under this constraint, there are two possibilities for rank of matrix $\mathbf{M}$: $\mathbf{rank}(\mathbf{M}) = 2$ and $\mathbf{rank}(\mathbf{M}) = 3$. *Case 1*: $\mathbf{rank}(\mathbf{M}) = 2$. Matrix $\mathbf{M}$ is constructed from two submatrices $\mathbf{M}_i$ and $\mathbf{M}_j$, due to the theorem "none of the trajectories should be on a line" rank of both $\mathbf{M}_i$ and $\mathbf{M}_j$ have to be 2. Therefore, the rank of matrix $\mathbf{M}$ is 2 only when $\mathbf{M}_i = \mathbf{K} \cdot \mathbf{M}_j$ where $\mathbf{K}$ is a linear transformation. Since $\mathbf{M}_i = \Pi_i \mathbf{S}_i$ and $\mathbf{M}_j = \Pi_j \mathbf{S}_j$, we have the following: $\mathbf{K} \cdot \Pi_j \cdot \mathbf{S}_j = \Pi_i \cdot \mathbf{S}_i$, which implies that $\mathbf{S}_i$ and $\mathbf{S}_j$ are linearly dependent. In *case 2*: $\mathbf{rank}(\mathbf{M}) = 3$. Since rank of matrix $\mathbf{P}$ is 4, there is only one possibility to make the rank of matrix $\mathbf{M}$ to be 3, that is the rank of matrix $\mathbf{S}$ is 3. In this case, action i is linearly dependent on action j: $\mathbf{S}_i = \mathbf{K} \cdot \mathbf{S}_j$ where $\mathbf{K}$ is a linear transformation. This case is explained

in Shapiro et al. (1995) based on the affine epipolar geometry.

In our approach, we compare each action with all other actions which have the same number of *instants* and the same sign sequence (or permutations), and compute the matching error of Eq. (20). For each action, we need to select closely matched actions. All the matches with the error above a certain threshold are eliminated first, and only three best matches for each action are maintained. Also, if a particular action does not closely match any action of its category then it is declared a unique action. Its label may change as more evidence is gathered.

The best matches for the individual actions are merged into a compact list using the transitive property. That is, if action 1 is similar to actions 9, 14, and 17; and action 3 is similar to actions 31, 1, and 9; then actions 1, 3, 9, 14, 17, and 31 are all similar actions due to the transitive property. We use Warshall's algorithm to group similar actions (Rosen, 1999). Warshall's algorithm computes the transitive closure of a graph, in which each vertex represents a specific action. In our implementation, we modify the original algorithm such that an action is grouped with similar action if its matching error to each of the actions in the group is lower than a threshold.

6. Experiments

We have performed experiments on 47 different action clips performed by seven individuals and the trajectories are given in Figs. 13–16 and described in Tables 2–5 (Please visit www.cs.ucf.edu/∼rcen/

Table 2. Description of actions in Fig. 13.

1st	Open the cabinet.
2nd	Put down the object in the cabinet, then close the door.
3rd	Open the cabinet, touching the door an extra time.
4th	Pick up an object (disks) with twisting the hand around.
5th	Put back the object (disks) and then close the door.
6th	Open the cabinet door, wait,then close the door.
7th	Open the cabinet door, wait, then close the door.
8th	Pick up an object, then make random motions.
9th	Open the cabinet.
10th	Pick up an object, put it in the cabinet, then close the door.
11th	Open the cabinet.
12th	Put the object (umbrella) back in the cabinet, then close the door.

Table 3. Description of actions in Fig. 14.

13th	Pick up a bag from the desk.
14th	Open the cabinet.
15th	Put down an object (a bag of disks) back to the cabinet, then close the door.
16th	Close the door, with some random motion.
17th	Open the cabinet.
18th	Pick up and put down several objects separately, then close the door.
19th	Open the cabinet door, pick up and put down several objects separately.
20th	Pick up an object (remote controller), put it in the cabinet, then close the door.
21st	Open the cabinet door, wait, then close the door.
22nd	Open the cabinet door, make random motions, then close the door.
23rd	Pick up some objects.
24th	Open the door, pick up an object, with the door half opened.

Table 4. Description of actions in Fig. 15.

25th	Close the half opened door.
26th	Open the cabinet door.
27th	Pick up and put down several objects separately, then close the door.
28th	Open the cabinet door, wait, then close the door.
29th	Pick up an object from the top of the cabinet.
30th	Close the cabinet.
31st	Open the cabinet.
32nd	Close the half closed door.
33rd	Open the door, wait, then close the door.
34th	Open the cabinet door, pick up an object, then put it back, then close the cabinet door.
35th	Open, then close the cabinet door.
36th	Pick up an object from the floor and put it on the desk.

Table 5. Description of actions in Fig. 16.

37th	Pick up an object from the floor and put it on the desk.
38th	Pick up an object from the floor and put it on the desk.
39th	Pick up an object from the desk and put it on the floor.
40th	Pick up an object from the floor and put it on the desk.
41st	Erase the white board.
42nd	Erase the white board.
43rd	Erase the white board.
44th	Erase the white board.
45th	Pour water into a cup.
46th	Pour water into a cup.
47th	Pour water into a cup.

Table 6. The matching results and evaluations.

Actions	3 Best matches	Evaluation and comments
1	26 17 9	Correct
2	12 5 15	Correct
3	**24** 17 9	One wrong
4		Unique action
5	15 2	Correct
6	21 7 35	Correct
7	21 6	Correct
8		Unique action
9	11 1 17	Correct
10		Unique action
11	26 9 17	Correct
12	2 15	Correct
13		Unique random motion
14	9 26 11	Correct
15	12 5 2	Correct
16		Unique action
17	1 11 9	Correct
18		Unique action
19	**31 11 26**	**Incorrect**
20		Unique action
21	7 6 35	Correct
22		Unique, random motion
23		Unique, tracking is lost
24	**26 9 31**	**Incorrect**
25		Unique
26	1 11 17	Correct
27		Unique
28		**Incorrect**, an extra instant presents
29	**23**	Unique action, Incorrect match
30		Unique action
31	**19** 9 **24**	Two incorrect
32	**23**	Unique action
33		Unique action
34		Unique action
35	**46** 21 6	One wrong, because of collinear points
36	38 40	Correct
37	38	Correct
38	37 40 36	Correct
39		Unique action
40	38 36	Correct
41	43 44	Correct
42		**Incorrect**, one instant is missing
43	41 44	Correct
44	43 41	Correct
45		**Incorrect**,because of collinear points
46	**35** 47	One wrong, because of collinear points
47	46	Correct

Table 7. The detection of action groups.

Action transitive closure	Evaluation and comments
1 9 11 14 17 26 31 **24**	**24** shouldn't belong to the group,
	3 should have been in the group
2 5 12 15	Correct grouping
6 7 21 35	Correct grouping
23 29 32	**Incorrect grouping**
36 38 40	**37** should also have been in the group
41 43 44	Correct grouping
4	Unique action, correct grouping
8	Unique action, correct grouping
10	Unique action, correct grouping
13	Unique action, correct grouping
16	Unique action, correct grouping
18	Unique action, correct grouping
19	Unique action, correct grouping
20	Unique action, correct grouping
22	Unique action, correct grouping
25	Unique action, correct grouping
27	Unique action, correct grouping
30	Unique action, correct grouping
33	Unique action, correct grouping
34	Unique action, correct grouping
39	Unique action, correct grouping
28	**Shouldn't be a unique action**
	Reason is instant detection
42	**Shouldn't be a unique action**
	Reason is instant detection
45	**Shouldn't be a unique action**
	Reason is false matching
46	**Shouldn't be a unique action**
	Reason is false matching
47	**Shouldn't be a unique action**
	Reason is false matching

research.html for video sequences, results, etc.). People performing the actions were not given any instructions, and entered and exited the field of view from arbitrary directions. While capturing the action clips, the location of the camera was changed from time to time to obtain actions from a different view point. We digitized the clips captured by a video camera at 24 fps. Current implementation of the system only deals with one hand and one head in the scene. For labeling the detected skin blobs as head and hand,

we assume the speed of the hand is higher than the speed of the head. This scheme works for the test sequence we used in our experiments, however one can use more complex schemes for labeling them correctly.

Our system does not require any training step. We start with an empty "known actions database". For each unique action we update the "known actions database" by including the representation of that action to the database. The system automatically detects hands using skin detection and generates trajectories of actions by mean-shift tracking method. In Figs. 17–19, we show two examples of actions from the data set along with the trajectories obtained using the proposed method. Once the trajectories are obtained, we compute the curvature given in Eq. (9) to obtain the view- invariant representation for the action.

The matching of the input action with the actions in the "known actions database" is done using the method discussed in Section 5.1. For the actions that have fewer than 5 instants, we select artificial instants on each interval, so that the total number of instants is bigger than 5, to meet the requirement of Theorem 2. The artificial instants temporally partition the interval into equal subintervals. We present the performance of the method in Table 6 by analyzing the three best matches. Only five of the all actions (actions 19, 24, 28, 42, and 45) has three false matches. Among the rest, five actions (3, 29, 31, 35 and 46) are partially correct, i.e. best three matches include the correct and wrong actions. In actions 35, 45, 46, the *instants* are collinear which are in contradiction to Theorem 2, therefore they do not provide adequate information for the matching.

One of the reasons for wrong matches is the noisy trajectory due to low sampling rate of the continuous hand motion, i.e. for some actions some of the instants may be missing and some may have extra instants. This failed the system to match correct actions for action 28 and action 42. Another reason for degraded performance on some of the actions is based on the affine camera model, which is an approximation of real projection from 3D world to 2D image coordinates. The affine model results in unrealistic matching for some actions. Due to this, action 19 and action 24

Figure 18. Action 2: Put down the object in the cabinet, then close the door (every 15th frame of the sequence is shown). The hand is highlighted with a white circle, and its trajectory is superimposed on the last frame.

Figure 19. Action 43: Erase the white board (every 12th frame of the sequence is shown). The hand is highlighted with a white circle, and its trajectory is superimposed on the last frame. The trajectory and its description are in Fig. 8(a).

are matched with other actions. However, an analysis on actions 19 and 24 shows that, they are partially matched with an opening action, such as 3. We expect using projective model will improve the matching performance, but since this model requires more instants, it is not applicable for actions which have only few instants.

In Table 7, we show the performance of the Warshall algorithm for learning the actions by grouping them using the transitivity property. Actions 4, 8, 10, 13, 16, 18, 20, 22, 25, 27, 30, 33, 34, 37, and 39 are correctly detected as unique actions and are not grouped with any other action. Due to errors in instant detection actions 28 and 42 are incorrectly detected as unique action whereas they belong to other groups. Errors in the matching phase caused actions 45, 46 and 47 to be incorrectly labeled as unique actions. Action 19, which was matched incorrectly (Table 6), is correctly recognized as a unique action, since the modified Warshall algorithm can keep the uniformity of the action closure. For the "opening cabinet action" the proposed system correctly grouped the actions 1, 9, 11, 14, 17, 26 and 31, missed only action 3 and included action 24 as false positive. Note that even though trajectories of these actions, shown in Figs. 13–16, are different, due to the strength of our representation, the system was able to learn that they represent the same action. Similarly, the system was able to correctly match the actions 2, 12, 5, and 15, with "put down the object, and then close the door" action. Actions 6, 7, 21, and 35 are learned as one group of actions, which represent "open the cabinet door, wait, then close the door". Actions 36, 38 and 40 are learned as a group of "pick up an object from the floor and put it on the desk", but algorithm missed action 37 which should have been also a member of this group. Actions 41, 43 and 44 are learned as "erase the white board" action. Actions 23, 29 and 32 are incorrectly learned as one group of actions, due to the constant threshold selection for grouping of all the actions.

Note that all these matches are based on only single instance of an action. Therefore the performance of the proposed approach is remarkable.

7. Conclusions

In this paper, we presented a view-invariant representation of human actions. Our representation of 2-D trajectory of an actions is composed of atomic units called *dynamic instants* and *intervals*. The dynamic instants are important motion events, which capture the significant changes in motion trajectory due to the change in the force applied to the object during the action. This applied force causes a change in the direction and/or speed. We proposed using spatio-temporal curvature of 2-D action trajectory to detect the dynamic instants. This representation was then used by our system to learn human actions without any training. The system automatically segments video into individual actions, and computes the view-invariant representation for each action. The system is able to incrementally learn different actions starting with no model. It is able to discover different instances of the same action performed by different people, and in different viewpoints.

Acknowledgments

The authors wish to express their appreciation to the anonymous referees for their careful reading of the manuscript. Thanks to Dr. Xin Li and Khurram Shafique for their valuable comments and discussions.

References

Bobick, A. and Davis, J.W. 1997. Action recognition using temporal templates. In *CVPR-97*, pp. 125–146.

Comaniciu, D., Ramesh, V., and Meer, P. 2000. Real-time tracking of non-rigid objects using mean shift. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 142–149.

Davis, J., Bobick, A., and Richards, W. 2000. Categorical representation and recognition of oscillatory motion patterns. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 628–635.

Gould, K. and Shah, M. 1989. The trajectory primal sketch: A multi-scale scheme for representing motion characteristics. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, pp. 79–85.

Izumi, M. and Kojiama, A. 2000. "Generating natural language description of human behavior from video images." In *ICPR-2000*, vol. 4, pp. 728–731, .

Jagacinski, R.J., Johnson, W.W., and Miller, R.A. 1983. Quantifying the cognitive trajectories of extrapolated movements. *Journal of Exp. Psychology: Human Perception and Performance*, 9: 43–57.

Kjeldesn, R. and Kender, J. 1996. Finding skin in color images. In *Int. Workshop on Automatic Face and Gesture Recognition*, pp. 312–317.

Koller, D., Heinze, D., and Nagel, H.-H. 1991. Algorithmic characterization of vehicle trajectories from image sequences by motion verbs. In *CVPR-91*, pp. 90–95.

Madabushi, A. and Aggarwal, J.K. 2000. Using head movement to recognize activity. In *Proc. Int Conf on Pattern Recognition*, vol. 4, pp. 698–701.

Mundy, J.L. and Zisserman, A. 1992. *Geometric Invariance in Computer Vision.* The MIT Press. ISBN 0-262-13285-0.

Newtson, D. and Engquist, G. 1976. The perceptual organization of ongoing behavior. *Journal of Experimental Social Psychology*, 12(5):436–450.

Parish, D.H., Sperling, G., and Landy, M.S. 1990. Intelligent temporal sub-sampling of American sign language using event boundaries. *J. Exptl. Psychol.: Human Perception and Performance*, 16:282–294.

Perona. P. and Malik, J. 1990. Scale-space and edge detection using anisotropic diffusion. *IEEE PAMI*, 12(7).

Polana, R. 1994. Temporal texture and activity recognition. Ph.D. Thesis, University of Rochester.

Rosen, K.H. 1999. *Discrete Mathematics and its Applications*. 4th edn. McGraw-Hill: New York.

Rubin, J.M. and Richards, W.A. 1985. Boundaries of visual motion. Tech. Rep. AIM-835, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, p. 149.

Seitz, S.M. and Dyer, C.R. 1997. View-invariant analysis of cyclic motion. *International Journal of Computer Vision*, 25:1–25.

Shapiro, L.S., Zisserman, A., and Brady, M. 1995. "3D motion recovery via affine epipolar geometry." *Int. J. of Computer Vision*, 16:147–182.

Siskind, J.M. and Moris, Q. 1996. A maximum likelihood approach to visual event classification. In *ECCV-96*, pp. 347–360.

Starner, T. and Pentland, A. 1996. Real-time American sign language recognition from video using hidden Markov models. In *Motion-Based Recognition*, M. Shah and R. Jain (Eds.). Kluwer Academic Publishers: Dordrecht. Computational Imaging and Vision Series.

Tomasi, C. and Kanade, T. 1992. Shape and motion from image streams under orthography: A factorization method. *Int. J. of Computer Vision*, 9(2):137–154.

Tsai, Ping-Sing, Shah, M., Keiter, K., and Kasparis, T. 1994. Cyclic motion detection for motion based recognition. *Pattern Recognition*, 27(12).

Tsotsos, J.K. et al. 1980. "A framework for visual motion understanding. *IEEE PAMI*, 2(6):563–573.

Zacks, J. and Tversky, B. 2001. Event structure in perception and cognition. *Psychological Bulletin*, 127(1):3–21.