View-Invariance in Action Recognition

Cen Rao and Mubarak Shah
Computer Vision Lab
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816

Abstract

Automatically understanding human actions using motion trajectories derived from video sequences is a very challenging problem. 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, 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 actions only captures characteristics, which are view-invariant, then the higher level interpretation can proceed without any ambiguity. In most of the current work on action recognition, the issue of view invariance has been ignored. Therefore, proposed methods do not succeed in more general situations.

In this paper, we first present a view-invariant representation of action consisting of dynamic instants and intervals, which is computed using spatiotemporal curvature of a trajectory. Then this representation is used by our system to learn human actions without any training. The system automatically segments video into individual actions, and computes 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.

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

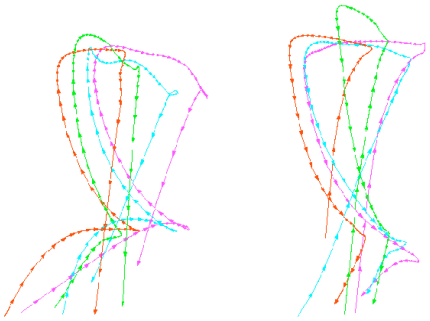
1. Introduction

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.

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 only statistical regularity. Examples include 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 include walking, running, jumping, etc.

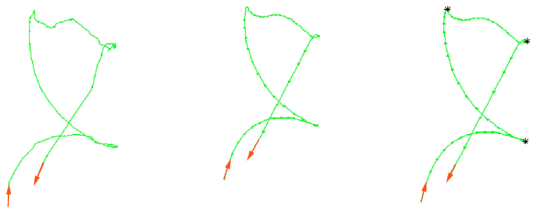
In this study, we focus our attention on human actions performed by a hand. These actions include: opening and closing overhead cabinets, picking up and putting down a book, picking up and putting down a phone, erasing a whiteboard, etc. Since an action takes place in 3-D, and is projected on 2-D image, depending on the viewpoint of the camera, the projected 2-D trajectory may vary. This may create a problem in interpretation of trajectories at the higher level. In most of the current work on action recognition, the issue of view invariance has been ignored. Therefore, proposed methods do not succeed in general situations.

In this paper, we first present a view-invariant representation scheme based on spatiotemporal curvature of a trajectory. A trajectory is represented by a sequence of *dynamic instants* and *interval*s. This representation is then used to automatically learn human actions. The system starts with no model, and incrementally builds models by watching people perform actions. Ultimately the system is able to recognize new actions using the learned actions. We present results on a video sequence depicting five different people performing roughly 60 different actions. The system is automatically able to segment the video into different actions, and learn them.



(a) (b)

Figure 1. Several trajectories of "opening overhead cabinet" (a), and "closing overhead cabinet" (b) actions.



(a) (b) (c)

Figure 2:"Opening overhead cabinet" trajectory (a) smoothed version of the trajectory (b) *dynamic instants* (marked by "*") and *intervals* (c)

2. Related work

Siskind and Morris [7] use HMMs to classify 6 gestures: pick up, put down, push, pull, drop, and throw. This requires training, and features used are not view invariant. Kojima et al [10] propose an approach to generate a natural language descriptions of human behavior from real video images. First, a head region of a human is extracted from each frame. Then, using a model-based method, 3-D pose and position of head are estimated. Next, the trajectory of head is divided into segments, and the most suitable verb is selected. Bobick and Davis [8] describe a method to recognize aerobic exercises from video sequences. They need training, and multiple views to perform recognition. Stauffer and Grimson [9] use simple classification based on aspect ratio of tracked objects. Seitz and Dyer [11] proposed an affine view-invariant trajectory matching method to analyze cyclic motion. Davis et al [14] proposed a motion recognition method by fitting sinusoidal model. The sinusoidal model contains amplitude, frequency, phase, and translation parameters. His method first estimates the translation, which is 1-D information, then estimates the frequency of x and y to get 2-D information, and then estimates phase, and so on. Based on the sinusoidal model coefficients the motion can be classified into different categories, each of them has consistent underlying structural descriptions. Yacoob and Black [15] proposed a method for modeling and recognizing activities. In their paper, they claim that the actions can be captured by motion parameters of body parts, for example, horizontal translation of arm, vertical translation of torso, and rotation of thigh. Each of these parameters is a function of time. And these functions can be represented by coefficients and bases using PCA methods. Their main contribution is the new method for computing PCA. They proposed to use robust regression method, because the traditional PCA method is not robust. Their view-invariant ability is limited to scaling, and temporal warping. And their method is not view-invariant in general cases. If the view point of camera is changed by rotating camera, their method will have problems.

3. Hand trajectories

In this section, we discuss how to compute motion trajectories from video sequences. In our method, hand is located in each frame, and centroids of the hand in each frame are connected to obtain a trajectory.

3.1. Capturing and smoothing trajectories

We apply skin detection [6] 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. This process is very fast, since only lookup table operations are involved. After skin detection, a connected component algorithm is applied, and fastest moving skin region is identified as hand. Next, the centroid of this skin region is computed for each frame, and trajectory of the hand is created by joining the centroids.

A trajectory is a spatiotemporal curve defined as: $(x[1],y[1],t[1])$, $(x[2],y[2],t[2]),\ldots$, $(x[n],y[n],t[n])$. This trajectory contains some noise due to errors in skin detection, lighting conditions, projection distortions, occlusion, etc. Although there are a lot of filters available in literature to reduce noise, such as low pass and mean filter, they are not suitable for this application, because these filters intend to smooth out all the peaks, which may represent meaningful changes in action. We use anisotropic diffusion to smooth the $x(t)$ and $y(t)$ coordinates of the trajectory. Anisotropic diffusion was proposed in the context of scale space [4]. This method iteratively smoothes the data with a Gaussian kernel, but adaptively changes the variance of Gaussian based on the gradient of a signal at a current point. Figure 2 shows a trajectory (a) and the one after anisotropic diffusion of x and y coordinates (b). Notice that now the trajectory is much smoother and changes of action status are kept.

3.2. Computing spatiotemporal curvature

We use spatiotemporal curvature to compute view invariant representation of an action. The spatiotemporal

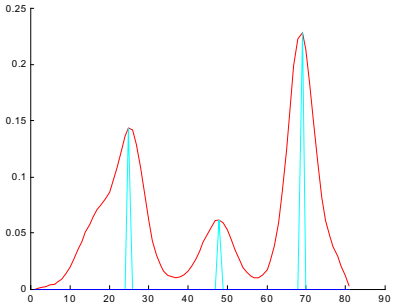


Figure 3. Spatiotemporal curvature, and detected maxima (*dynamic instants)* in "opening overhead cabinet" trajectory.

curvature of a trajectory is computed by a method described by Besl and Jain [5]. In this case, a 1D version of the quadratic surface fitting procedure is used. The spatiotemporal curvature, k is given as follows:

$$k = \frac{\sqrt{A^2 + B^2 + C^2}}{\left((x')^2 + (y')^2 + (t')^2\right)^{3/2}}. \quad (1)$$

where

$$A = \begin{vmatrix} y' & t' \\ y'' & t'' \end{vmatrix}, B = \begin{vmatrix} t' & x' \\ t'' & x'' \end{vmatrix}, C = \begin{vmatrix} x' & y' \\ x'' & y'' \end{vmatrix}$$

The notation $|\cdot|$ denotes the determinant, and

$$\begin{aligned} x'(t) &= x(t) - x(t-1), \\ x''(t) &= x'(t) - x'(t-1). \end{aligned} \quad (2)$$

Since the time interval is constant, i.e. t=1, 2, 3,…, so t'=1, and t''=0.

Spatiotemporal curvature captures both the speed and direction changes in one quantity. When the time information is ignored in the spatiotemporal curvature, we simply get spatial curvature, commonly used in 2-D shape analysis. In this case, time interval is 0, therefore $t' = t'' = 0$, and the equation (1) reduces to the spatial curvature. The spatiotemporal curvature of the "opening overhead cabinet" trajectory is shown in Figure 3.

4. Representation

Representation is very important and sometimes difficult aspect of an intelligent system. The representation is an abstraction of sensory data, which should reflect a real world situation, be view-invariant and compact, and be reliable for later 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 zerocrossing in a first derivative) in the spatiotemporal curvature. An *interval* represents the time-period between any two *dynamic instants*, during which the motion characteristics remain pretty much 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 2.c.

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 change of motion direction 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 2.c). 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.

4.1. View invariance

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 3D is always projected as a dynamic instant in 2D, 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 that case, the centroids of hand in all frames are projected at the same location in the image plane, resulting in a 2-D trajectory, which is essentially a single point. In Figure 1.a, we show trajectories of opening overhead cabinet action from several viewpoints. Even though these trajectories look quite different, three dynamic instants are detected by the proposed method.

In our work we assume the camera is an affine camera, which means that the depth of 3-D trajectory of action is small compared to the viewing distance [12]. This assumption is valid for most actions in surveillance systems.

Assume that the location of a hand in 3-D space at t_1, t_2, t_3 is given by (P_1, P_2, P_3). In this case, we have two vectors $\overrightarrow{P_1P_2}$ and $\overrightarrow{P_2P_3}$ (see Figure 4a). The projection of these three points in image plane is shown in Figure 4b.

It is clear that there is a dynamic instant at t_2 . Assume that the angle between the two vectors is α. The sign of this angle can be determined by computing the sign of the cross product of projection of two vectors in the image plane. We will use the sign of this angle as the sign of the instant. We claim that the sign of instant is view-invariant when the camera viewpoint remains in the upper hemisphere of the viewing sphere. This is explained in the following:

We want to show that the angle is view invariant under affine camera model. The camera translation will not affect the angle α, therefore we will only consider the situation when the camera rotates. Let us assume for simplicity that camera axis passes through P_2 and the distance from the camera to P_2 is D, and $\overrightarrow{P_1P_2}$ is always vertical. It is obvious that camera rotation around the Z axis does not change α. Therefore, the situations that need to be considered are camera rotations around the X-axis (tilt) and the Y-axis (pan).

During the camera panning (Figure 4b), the only part which changes is the projection of P_3 (X_3,Y_3,Z_3). Its image coordinate are (u_3,v_3). 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). Due to camera panning with angle θ, X coordinates of any point are changed to X' as follows:

$$X' = X\cos\theta - Z\sin\theta \tag{3}$$

Now the image coordinates under affine camera are given by:

$$u' = f\frac{X'}{D} \tag{4}$$

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

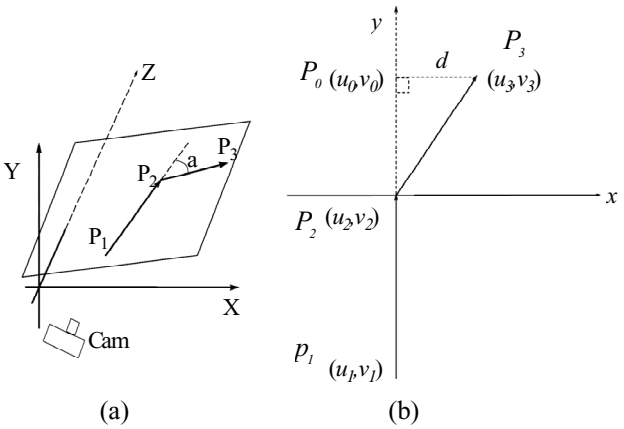
$$\begin{aligned} d &= u'_3 - u'_0 \\ &= f\frac{[(X_3\cos\theta - Z\sin\theta) - (X_0\cos\theta - Z\sin\theta)]}{D} \\ &= f\frac{(X_3 - X_0)\cos\theta}{D} \end{aligned} \tag{5}$$

In the above equation if $\theta \in$ (-90°,+90°), then $d > 0$, so that α is positive. This means that the sign of α is view invariant when the camera is panning within the semicircle.

For the situation when the camera tilts around the X-axis, the similar argument also holds. Therefore, when the camera tilts within the semicircle ($\varphi \in$ (-90°,+90°)) 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 is for the situations when all the instants are located in one plane, which is restricted. However, we can extend the proof for more general situations as follows.

Assume that there are four instants (P_1,P_2,P_3,P_4), P_1,P_2,P_3 are in one plane with an angle α , and P_2,P_3,P_4 are in another plane with an angle β. Then the signs of α and β are invariant when the camera rotates within the quarter of sphere, which is defined by the two planes.



(a) (b)

Figure 4 .(a) Three points in 3-D (b) 2-D projections

For the situations when more non-planar instants are involved, we can conclude that the sign of an instant will remain the same if the camera view is not in a plane containing three or more co-planar instants.

The sign characteristic of an instant is very useful, because the sequence of signs of instants helps us to distinguish between different actions under different viewpoints. For example, the opening cabinet action (Figure 2c) has five instants, the signs for second, third and fourth instants are (-,+,+). On the other hand, closing cabinet action (Figure 1b) has five instants also, but the signs of the middle three instants are (-,-,+), which are different. In general, for a trajectory with n instants, the number of permutations of signs is $2^{(n-2)}$; here 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 high match score as compared to the trajectories of the 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 and position. The matching algorithm is discussed in detailed in section 5.1.

5. Learning

Once representation has been defined, the next step is to use this representation to learn human actions. As stated earlier, our aim is to start with no model, and incrementally build model of actions by continuously watching. This is

the way, we believe, how children learn to recognize different actions by repeatedly observing adults perform actions.

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

The continuous video stream can be easily segmented into individual actions. One particular action begins as soon as the hand enters the field of view, and ends when the hand goes out of the field of view. When the system detects the hand again, the second action begins, and so on.

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 signs. The system also compares each action with all other actions in its category.

At the higher level of abstraction, the system also determines sets of similar actions based on the match scores. For example, different instances of "opening overhead cabinet" action can be automatically determined to be similar. For each such 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 remain 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 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 equal number and the same sign permutation of instants. We want to note that one action can be a sub-action of the other. In this case, these actions won't have an equal number of instants; however, this match is meaningful. At this point, we are not going to deal with it.

We use a view-invariant matching function that equates a set of images if and only if they represent views of an object in the same configuration as proposed by Seitz and Dyer in [11].

Let us represent an action by a sequence of n instants, where each action is represented by (x, y) image coordinates of each instant: $\boldsymbol{I}=((u_1,v_1),(u_2,v_2),\ldots,(u_n,v_n))$. Assume a particular action is captured in k views, represented by: $I_{v_1}, I_{v_2}, \ldots, I_{v_k}$. Our aim is to automatically determine if these views represent the same action. Let us form a matrix M as follows:

$$M = \begin{bmatrix} \mathrm{I}_{v_1} \\ \mathrm{I}_{v_2} \\ \vdots \\ \mathrm{I}_{v_k} \end{bmatrix}, \quad and \quad \mathrm{I}_v = \begin{bmatrix} \mu_1^v & \mu_2^v & \ldots & \mu_n^v \\ \nu_1^v & \nu_2^v & \ldots & \nu_n^v \end{bmatrix} \quad (6)$$

If the views represent the same action, then we can express M as:

$$M = \begin{bmatrix} \Pi_{v_1} \\ \vdots \\ \Pi_{v_k} \end{bmatrix} S \quad (7)$$

where $\boldsymbol{S}$ (shape) represents the 3-D coordinates of points corresponding to the instants, and $\boldsymbol{\Pi_v}$ is the projection matrix of each viewpoint. Matrix M is the product of two matrices, each having a rank at most 3. Therefore, the rank of M is at most 3. This is due to the rank theorem by Tomasi and Kanade [13]. As a consequence of this result if the views represent the same action, and there are no numerical errors, then all singular values of the matrix M except the first three will be zero. However, these singular values may not be exactly zero. Therefore, Seitz and Dyer [11] use the sum of the squares of singular values of M, except the first three singular values, to match the different views. This distance is given as follows:

$$dist = \sqrt{\frac{1}{2kn}\sum_{4}^{n}\sigma_i^2} \quad (8)$$

where $\sigma_{1..n}$ are the singular values of $\boldsymbol{M}$, k denotes the number of views, and n denotes the number of singular values. This distance gives the average amount necessary to additively perturb the coordinates of each instant in order to produce projections of a single action.

To match two actions $\boldsymbol{I_i}$ and $\boldsymbol{I_j}$, we form matrix $\boldsymbol{M}$ as follows:

$$M = \begin{bmatrix} \mu_1^i & \mu_2^i & \ldots & \mu_n^i \\ \nu_1^i & \nu_2^i & \ldots & \nu_n^i \\ \mu_1^j & \mu_2^j & \ldots & \mu_n^j \\ \nu_1^j & \nu_2^j & \ldots & \nu_n^j \end{bmatrix}$$

We then determine the singular values of M, and compute the distance (equation 8) as $dist_{i,j} = |\sigma_4|$. The distance gives the matching error of two action trajectories.

However, when we match two actions, there are two possible shape matrices S_i, and S_j. In this case the rank theorem may not be valid. We need to prove that if the rank of M is 3, then $S_i = R \cdot S_j$. This means that the rank of $[S_i, S_j]$ is 3, and S_j is a linear transform of S_i, so that S_i and S_j represent the same actions.

In this case matrix M has the following form:

$$M = \begin{bmatrix} \mu_1^i & \mu_2^i & \dots & \mu_n^i \\ v_1^i & v_2^i & \dots & v_n^i \\ \mu_1^j & \mu_2^j & \dots & \mu_n^j \\ v_1^j & v_2^j & \dots & v_n^j \end{bmatrix} = PS = \begin{bmatrix} \Pi_i & 0 \\ 0 & \Pi_j \end{bmatrix} \begin{bmatrix} S_i \\ S_j \end{bmatrix}$$

$$= \begin{bmatrix} a_{i1} & a_{i2} & a_{i3} & 0 & 0 & 0 \\ a_{i4} & a_{i5} & a_{i6} & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{j1} & a_{j2} & a_{j3} \\ 0 & 0 & 0 & a_{j4} & a_{j5} & a_{j6} \end{bmatrix} \begin{bmatrix} x_{i1} & x_{i2} & x_{i3} & \dots & x_{in} \\ y_{i1} & y_{i2} & y_{i3} & \dots & y_{in} \\ z_{i1} & z_{i2} & z_{i13} & \dots & z_{in} \\ x_{j1} & x_{j2} & x_{j23} & \dots & x_{jn} \\ y_{j1} & y_{j2} & y_{j3} & \dots & y_{jn} \\ z_{j1} & z_{j2} & z_{j3} & \dots & z_{jn} \end{bmatrix}$$

Assume R is *3* by *3* matrix and S_i $R \cdot S_j$, so the rank of matrix S is equal to or greater than *4*. Moreover, as we know the rank of M is 3. So under affine camera model the only possibility is that the rank of P is 3. From the P matrix, $rank(P)=3$ *iff* $(a_{l1},a_{l2},a_{l3}) = k(a_{l4}, a_{l5}, a_{l6})$, where k is a scalar, and l is either i or j. Then this implies $(u_{l1}, u_{l2}, \dots, u_{ln}) = k(v_{l1}, v_{l2}, \dots, v_{ln})$. This means that in one of the two actions all the instants are located on a line. This is a situation, which can be easily detected. So we have the following theorem:

Theorem: With affine camera model, if the rank of M is equal to or less than 3, and neither of actions has all instants in a line, then the two actions S_i and S_j match, and $S_i=R \cdot S_j$, where R is a linear transformation.

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

The best matches for individual actions are merged into a compact list using the transitive property. That is, if action 1 is similar to actions 14, 21, and 29; and action 4 is similar to actions 43, 1, and 14; then actions 1, 4, 14, 21, 29, and 43 are all similar actions due to the transitive property.

6. Experiments

We digitized several video clips recorded at 24 fps. The location of camera was changed from time to time. Seven people performed total of 60 different actions, the complete list of actions is given Table 1. People were not given any instructions, and entered and exited from arbitrary directions, and the location of the camera was changed from time to time. Therefore, the viewpoints of these actions were very different. The system automatically detected hand using skin detection, generated trajectories of actions.

The actions were segmented by the system into 60 actions. Trajectories of these actions were used to generate the view invariant representation proposed in this paper. These representations were interpreted by the system to learn these actions.

Each of these actions was matched using method discussed in section 5.1. The results are shown in Table 2. We are pleasantly surprised to see our simple matching technique worked quite well. Only three matches were completely wrong (actions 31, 36, and 58). Seven matches (4, 8, 41, 43, 48, 59,and 60) were partially incorrect. In action 8, 48, 58, 59, 60, the instants were collinear, therefore they did not provide independent constraint for the measurement. And action 31 and 36 are partially matched with opening action, such as 4.

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

The system was able to learn that actions 1, 4, 14, 16, 21, 29, 43, and 38 are the same. Note that even though trajectories of these actions shown in Figure 6, are different, but due to the strength of our representation, the system was able to learn they represent the same action. Similarly, the system was able to discover that action 3, 18, 6, 23, and 32, which represent "put down the object, and then close the door", are all the same using matching and the transitive property. Therefore, the confidence for this action is quite large.

Several actions were identified as unique, because they did not match well with other actions having the same number of instants. Therefore, their confidence is quite low. Since we assume that the system is continuously watching in its field of view, if more instances of these unique actions are performed, the system will be able to increase the confidence.

7. References

[1] Polana, R., "Temporal textures and activity recognition", Ph.D. thesis, University of Rochester.

[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, Novr, 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] R Kjeldesn andJ Kender, "Finding skin in color images", Int workshop on Automatic face and gesture recogn, pp 312-317, 1996.

[7] Siskind J., M., and Moris, Q., "A maximum likelihood approach to visual event classification", ECCV-96, 347-360.

[8] Aaron Bobick and James W. Davis. Action recognition using temporal templates, pages 125--146.CVPR-97, 1997.

[9] C. Stauffer and W.E.L. Grimson, "Learning patterns of activity using real-time tracking", PAMI, 22(8):747--757, August 2000.

[10] M. Izumi A. Kojiama "Generating natural language description of human behavior from video images", ICPR-2000, 4: 728--731, 2000.
[11] S. M. Seitz and C. R. Dyer. *View-invariant analysis of cyclic motion*. International Journal of Computer Vision, 25:1--25, 1997.
[12] Joseph L. Mundy and Andrew Zisserman, "Geometric Invariance in Computer Vision". The MIT Press, 1992. ISBN 0-262-13285-0.
[13] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *Int. J. of Computer Vision*, 9(2):137-154, 1992.
[14] J. Davis, A. Bobick, W. Richards, "Categorical Representation and Recognition of Oscillatory Motion Patterns", *IEEE Conference on Computer Vision and Pattern Recognition*, June 2000, pp. 628-635.
[15] Y. Yacoob and M. Black, "*Parameterlized Modeling and Recognition of Activities*," International Conf. on Computer Vision, Mumbai-Bombay, India, January, 1998.

Table 1: List of actions.

1st open the cabinet
2nd pick up an object (umbrala) from the cabinet.
3rd put down the object in cabinet, then close the door.
4th open the cabinet, with touching the door an extra time.
5th pick up an object (disks) with twisting hand around.
6th put back the object (disks) and then close the door.
7th open the cabinet door, wait, then close the door.
8th open the cabinet door, wait, then close the door.
9th pick up an object from top the of the cabinet.
10th put the object back to the top of cabinet.
11th pick up an object from the desk.
12th put the object back to the desk.
13th pick up an object, then make random motions.
14th open the cabinet.
15th pick up an object, put it in the cabinet, then close the door.
16th open the cabinet.
17th pick up an object (umbralla) from the cabinet.
18th put the object (umbralla) back to the cabinet.
19th pick up a bag from the desk.
20th make random motions.
21st open the cabinet.
22nd pick up an object (a bag of disks).
23rd put donw an object (a bag of disks) back to the cabinet, then close the door.
24th pick up an object from the top of the cabinet.
25th put the object back to the cabinet top.
26th make random motions with two hands.
27th continue the action 26.
28th close the door, with some random motion.
29th open the cabinet.
30th pick up an object (remote controller) from the cabinet, put it down on the desk, pick up another object (pencil) from the desk, put it in the cabinet, then close the door.
31st open the cabinet door, with the door half pushed, pick up an object (pencil) from the cabinet.
32nd pick up an object (remote controller) from the desk, put it in the cabinet, then close the door.
33rd open the cabinet door, wait, then close the door.
34th open the cabinet door, make random motions, then close the door.
35th pick up some objects.
36th open the door, pick up an object, with the door half opened.
37th close the half opened door.
38th open the cabinet door.
39th pick up an object, move it within the cabinet, pick up another object, move it, then close the door.
40th open the cabinet door, wait, then close the door.
41st pick up an object from the top of the cabinet.
42nd close the cabinet.
43rd open the cabinet.
44th put down a disk.
45th close the half closed door.
46th open the door, wait, then close the door.
47th open the cabinet door, pick up an object, then put it back, then close the cabinet door.
48th open, then close the cabinet door.
49th pick up an object from the floor and put it on the desk.
50rd pick up an object from the floor and put it on the desk.
51rd pick up an object from the floor and put it on the desk.
52nd pick up an object from the desk and put it on the floor.
53rd pick up an object from the floor and put it on the desk.
54th, 55th, 56th, 57th erase the white board.
55th erase the white board.
56th erase the white board.
57th erase the white board.
58th pour water into a cup.
59th pour water into a cup.
60th pouring water into a cup.

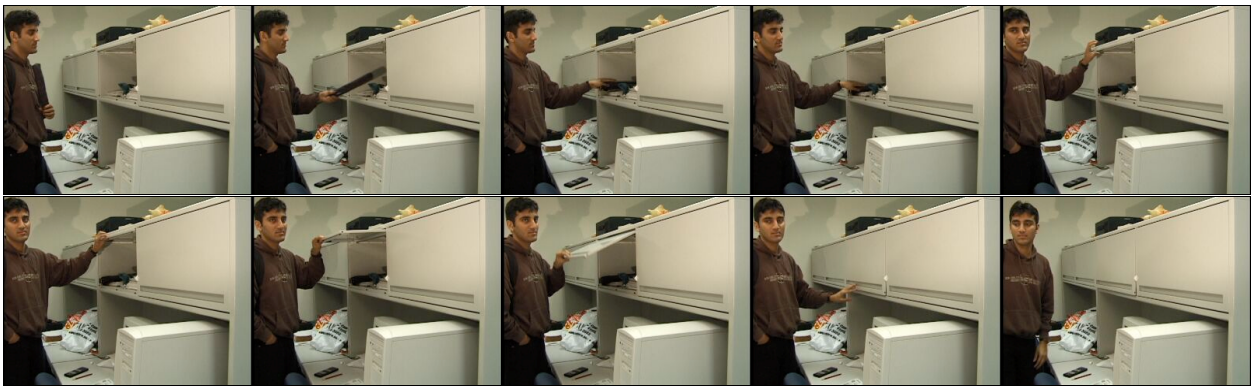


Figure 5. Sequence showing Action 3, put down the object in cabinet, then close the door.

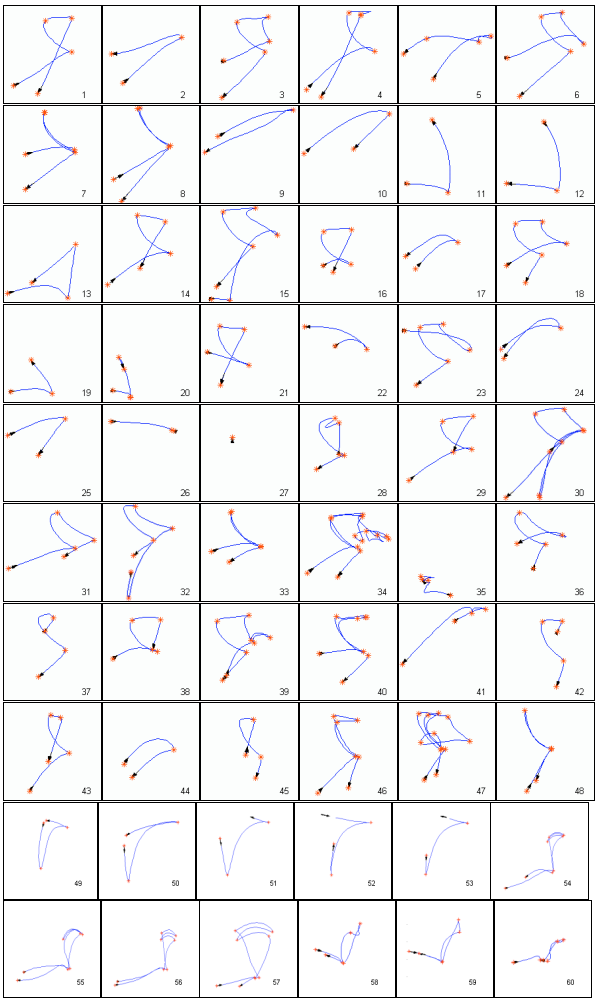


Figure 6. Trajectories of all 60 actions. The instants are shown with red "*".

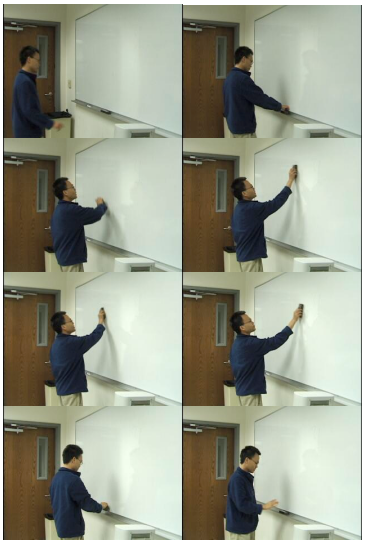


Figure 7. Sequence showing Action 56, erase the white board.

Table 2. Interpretation results. The bold face font in column indicates incorrect match.

Actions	3 Best matches	Evaluation & comments
1	38 29 14	Correct
2	Pick up	Correct
3	18 6 23	Correct
4	**36** 29 14	One wrong
5		Unique action
6	23 3 18	Correct
7	33 8 48	correct
8	33 7 **60**	One wrong
9	Pick up	Correct
10	Put down	Correct
11	Pick up	Correct
12	Put down	Correct
13		Unique action
14	16 1 29	Correct
15		Unique action
16	38 14 29	Correct
17	**Pick up**	Incorrect, object hidden
18	3 23 6	Correct
19	Pick up	Correct
20		Unique random motion
21	14 38 16	Correct
22	Pick up	Correct
23	18 6 3	Correct
24	Pick up	Correct
25	Put down	Correct
26		Unique action
27		Unique action
28		correct
29	1 16 14	Correct
30		Correct
31	**43 16 38**	incorrect
32		Unique action
33	8 7 48	correct
34		Random motion, unique
35	Put down	The action is confusing
36	**38 14 43**	incorrect
37		Unique
38	1 16 29	Correct
39		Correct
40		46 is missing
41	**35**	Unique action
42		Unique action
43	**31** 14 **36**	Two incorrect
44	**Pick up**	Incorrect, object too small
45		Unique action
46		40 is missing
47		Unique action
48	**59** 33 7	One incorrect, collinear points.
49	51 53 50	Correct
50	51 53 50	Correct
51	50 53 49	Correct
52		Unique action
53	51 49 50	Correct
54	56 57	Correct
55	**Incorrect**	One instant missing
56	54 57	Correct
57	56 54	Correct
58	**48 33**	Collinear points
59	**48** 60	Collinear points
60	59 **8 48**	Collinear points