Invariance in Motion Analysis of Videos

Cen Rao
Computer Vision Lab
University of Central Florida
rcen@cs.ucf.edu

Mubarak Shah
Computer Vision Lab
University of Central Florida
shah@cs.ucf.edu

Tanveer Syeda-Mahmood
IBM Almaden Research Center
stf@almaden.ibm.com

ABSTRACT

In this paper, we propose an approach that retrieves motion of objects from the videos based on the dynamic time warping of view invariant characteristics. The motion is represented as a sequence of dynamic instants and intervals, which are automatically computed using the spatiotemporal curvature of the trajectory of moving object in the videos. Dynamic Time Warping (DTW) method matches trajectories using a view invariant similarity measure. Our system is able to incrementally learn different actions without any initialization mode, therefore it can work in an unsupervised manner. The retrieval of relevant videos can be easily performed by computing a simple distance metric. This paper makes two fundamental contribution to view invariant video retrieval: (1) Dynamic Instant detection in trajectories of moving objects acquired from video. (2) View-invariant Dynamic Time Warping to measure similarity between two trajectories of actions performed by different persons and from different viewpoints. Although the learning algorithm is relatively simple in our approach, we can achieve high recognition rate because of the view-invariant representation and the similarity measure using DTW.

Categories and Subject Descriptors

I.5.3 [**PATTERN RECOGNITION**] Clustering: Algorithms, Similarity measures.

General Terms

Algorithms, Measurement.

Keywords

View-invariant action representation, spatiotemporal curvature, view-invariant measure, view-invariant dynamic time warping, learning, human actions.

1. INTRODUCTION AND REALATED WORK

Motion information provides an important cue for understanding the video contents. The typical applications include video retrieval, intelligent video surveillance, HCI, and human perception study. Understanding behavior of humans in a scene is a task that humans perform with great ease, allowing us to better interact, communicate with and respond to each other. However, it has been seen that developing computational models of such understanding of behavior has been a persistently difficult problem for video computing. One of the key challenges is view invariance, due to the fact that video is the 2D projection of the 3D world. While humans can recognize actions from various views easily, finding view invariant cues for recognition has been difficult to replicate in computational vision systems. We argue that finding view invariant representation makes the problem of recognition far more tractable. Secondly, people perform the same action differently each time, and even the same person performs the same action at different speeds. Therefore, the system must be able to solve the temporal-invariance problem, such that the same actions with different speeds are matched. Furthermore, the view-invariance and temporal-invariance need to be combined in one framework, so that the system can process general videos. Lastly, we want to emphasis the ability of the our system to learn in an unsupervised manner. The recognition system we propose consists of three modules: motion capturing, action representation, and learning. A system has been successfully implemented, which is able to handle actions from different viewing directions and at different speeds so that extensive training, context knowledge, or camera calibration is not needed. Moreover, the system can autonomously build up a recognition category database.

In the first module (motion extraction), body movement during actions is recorded with respect to time, providing action primitives to be analyzed.

The representation module takes the results from the motion capture module and transforms it into a physically meaningful form: a sequence of instants and interval. A dynamic instant is an instantaneous entity that occurs for only a single frame, and represents an important change in motion characteristics. Intervals are defined as the time period from one instant to the next. We use spatiotemporal curvature to detect instants, effectively capturing speed, direction, and orientation changes during the action within one quantity. Moreover, since actions take place in 3D, then get projected on an arbitrary 2D image, depending on the viewpoint of the camera, our representation is able to recover the characteristics that are consistent from different viewing directions. The representation module has a central role. A "good" representation system should illustrate the actual event during the action and reduce the complexity of recognition/learning module significantly.

In learning module, we propose a matching method, such that a similarity measure is computed from the spatiotemporal

MM'03, November 2-8, 2003, Berkeley, California, USA.

information of the action representation to handle both view-invariance and temporal invariance problems in the videos. Based on this similarity measure, a nearest neighbor clustering approach is applied, so that the recognition database can be incrementally developed without any training. Because of the strength of our action representation module, and the view invariant matching algorithm, the system can use a relatively simple learning approach to achieve high recognition rate.

Earlier approaches to video retrieval were either region-based [8,16,18], temporal trajectory-based [17,19,20,1], part-based [21,22] or a combination of these [9,24,33], and considered either 2d shape or motion alone. These approaches were sensitive to changes in viewpoint, requiring explicit models for handling different viewpoints. Recent attempts have alleviated effects of viewpoint by developing invariants that are insensitive to viewpoint changes using an affine camera model [1], or have explicitly recovered viewpoint transformations using homography [25], or the general perspective case [2]. Seitz and Dyer [11] used view-invariant measure to find the repeating pose of walking people.

There are two main types of approaches for retrieving sequences: sequence-to-sequence and trajectory-to-trajectory. The sequence-to-sequence approach, which is also called the direct approach, takes the video frames as input and applies the computation over all pixels in the video frames or tracked regions [8,25,38]. The trajectory-to-trajectory approach tracks the movement of the feature points in the field of view, and the computation is based on the information from the trajectories. The advantages of the direct approach include: it determines the spatial transformation between sequences more accurately than the trajectory-to-trajectory approach does, and it does not require explicit feature detection and tracking. On the contrary, since the trajectories contain explicit geometric information, the trajectory-to-trajectory approach better handles the large spatiotemporal variation, can process video sequences acquired by different sensors and is less affected by changes in background. The detailed comparison between these approaches is available in [37,25]. Since the video sequences in most applications contain a significant spatiotemporal variation, we choose the trajectory-to-trajectory approach.

In addition to viewpoint changes, the execution style variations include local changes in velocity and acceleration that are the result of natural variations produced by moving subjects and the effect of surrounding environments.

A popular way to handle execution style variations is through hidden Markov models (HMM) where matching of an unknown sequence with a model is done through the calculation of the probability that a HMM could generate the particular unknown sequence. Siskind and Morris proposed a HMM based system [7]. The recognition system takes the 2D pose stream, such as position, orientation, shape, and size of each participant object, and classifies it as an instance of a given action type. Campbell et al. used 3D measures obtained from a stereo system [3]. Essa et al. [28], Hoey and Little [29] proposed similar systems. In order to model the interactions between subjects, Oliver et al. proposed a more complex architecture -- Coupled Hidden Markov Models (CHHM)[30]. The HMM-based approaches however suffer from the design and training issues relating to the construction of models per action. Moreover, in most of approaches, only view-based features have been used so that the proposed systems do not have ability to retrieve the same action from different viewing directions in videos.

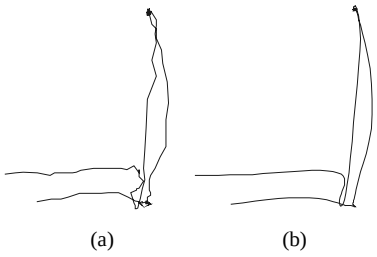


Figure 1: a) the raw trajectory. b) The smoothed trajectory.

From the preceding discussion, we can see that view based methods face difficulty in handling recognition of the same actions from different viewpoint, which makes their applications rather limited. For implicit methods, such as HMM, the results are based on extensive training, and the rules of classification cannot be understood, so that there is no hint to generate new models except using huge number of exemplars

2. MOTION EXTRACTIN MODULE

The motion extraction module detects and tracks motion of action primitives. During motion extraction module there are two steps: tracking and smoothing. The output of this module is action represented as motion trajectories.

2.1 Tracking

For the actions performed by an action primitive (e.g. hand), first, the centroids of the hand regions are computed for each frame. The Mean-shift tracker is applied on the performing subjects (centroids) to get the trajectories of hand motion [31]. However, for more complicated hand actions, tracking of centroids of hands does not provide sufficient information, e.g. making gesture, turning a knob, etc. Therefore, the *orientation* of hand region is also tracked in our system with skin detection method as follows [6]: A small sequence of images of performer (3 to 5 frames) is used for training to generate the color predicate; the module then labels the incoming pixels as either skin or non-skin based on the predicate. Finally, morphologic operations are used to group the skin pixels into region. Correspondence is resolved using the algorithm proposed by Rangarajan *et al.*[34]. As the result of tracking, a motion trajectory is generated, which is a spatiotemporal curve defined as: $\{(x[t_i],y[t_i],\theta[t_i])\}$, i=0, 1, 2,… , where x and y are positions of the centroid, θ is orientation, and t is timestamp. In this way, we can treat a trajectory as a temporal function.

2.2 Smoothing

To remove the noise in the trajectory caused by error from tracking, skin detection, and projection distortions, an anisotropic diffusion algorithm is used for smoothing [4,1]. This method iteratively smoothes the data (I) with a Gaussian kernel, but adaptively changes the variance of Gaussian based on the gradient of a signal at a current point as follows:

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

Where $0 \le \lambda \le \frac{1}{4}$; t represents the iteration number, and

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

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

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

where $g(\nabla I) = e^{-(\|\nabla I\|/k)^2}$. The constant k can be fixed either manually at some fixed value, or can be estimated from the "noise estimator" [4].

The original diffusion algorithm proposed by Perona and Malik only applies to functions that have a 1D co-domain, such that F: $R^n \rightarrow R^1$, rather than trajectory functions: T:$R^1 \rightarrow R^3$, which has 3D co-domain. We need an algorithm that works on the vector data ($x[t_i]$,$y[t_i]$,$\theta[t_i]$) to keep the correlation in the co-domain (x, y, θ). The steps of the empirical method we use are: (1) Apply Hoteling transform to the raw data so that the correlations between different dimensions are minimized; (2) Perform Perona-Malik smoothing on each dimension of the transformed data; (3) transform the smoothed data back to original data coordinates. Figure 1 shows an example of smoothed motion trajectory, which correspond to a hand picking up a telephone handset and then putting it back.

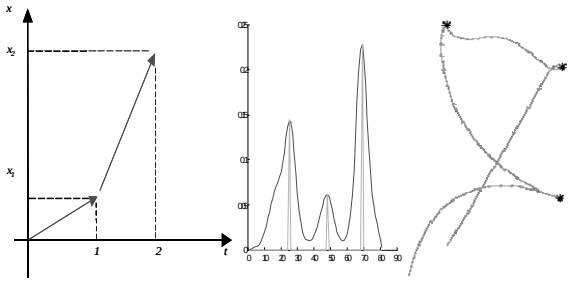
3. ACTION REPRESENTATION

In this module, the motion trajectory recovered by the motion capture module is interpreted as a sequence of dynamic instants and intervals. A dynamic instant is an instantaneous entity that occurs for only one frame, and represents a significant change of any of the motion characteristics: speed, direction, acceleration and rotation. These dynamic instants are detected by identifying maxima (a zerocrossing in a first derivative) in the spatiotemporal curvature. An interval represents the time period between any two adjacent dynamic instants during which the motion characteristics remain fairly constant. In our representation, both instants and intervals embrace certain physical meanings.

3.1 Instants detection

To illustrate the concept of instant detection, consider a 1D motion trajectory $\{x[t_i]\}$, $i=0, 1, 2,\ldots$, where t_i is the uniform sampling index along temporal axis, x is the position along X axis. If there is a change in speed at time t_i, a turning point at $\{x[t_i], t_i\}$ of the x-t curve will be present, and spatiotemporal curvature will capture this turning (figure 2a). This idea is applied to multi-dimensional spatiotemporal curves $\{x[t_i],y[t_i],\theta[t_i]\}$, $i=0, 1, 2, \ldots$, such that changes of speed, direction and orientation will be captured by turning points in the spatiotemporal domain.

The spatiotemporal 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:



(a) (b) (c)

Figure 2: a) spatiotemporal curve of a 1D motion. b) The spatiotemporal curvature values (ordinate) and the peak detection results (blue peaks) of the trajectory, the abscissa is the frame index. c) An opening cabinet action trajectory with instants and intervals.

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

where

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

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

$$x'(t) = x(t) - x(t-1),$$
$$x''(t) = x'(t) - x'(t-1). \quad (2)$$

Here t'=1 and t''=0 since the time interval is constant, i.e. t_0=0, t_1=1, t_2=2,… It is worth noting that the curvature captures all the changes of speed, direction and rotation in one quantity. Moreover, we can generalize this formula to use other motion characteristics that change with respect to time in the videos.

Consider an opening overhead cabinet action (Figure 2.b, Figure 8). 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.

We use this approach to analyze human gait. When a walking person is tracked, his/her foot regions are segmented out by using color predicate, which is generated by the images of shoes. Figure 3 shows some tracking results. Figure 4 shows the trajectories of left and right feet respectively in three walking sequences. The short line segments display the foot orientations at the centroid. The detected instants correspond to three important changes during a walking cycle: "foot touching the ground", "leaving the ground", and then "moving forward". If we use only x,y,t information we can detect only two instants consistently. Therefore, orientation information is important.

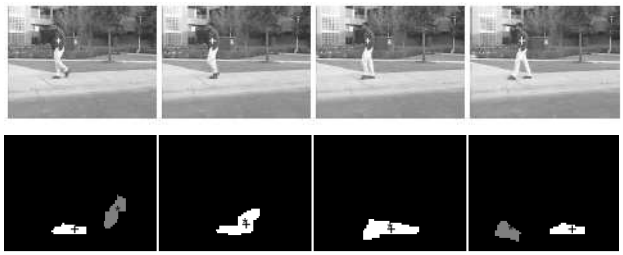


Figure 3: frame 174, 176,178, and 180 of a walking sequence and the foot tracking results. The gray color represents left foot and white color is right foot. The middle two frames have occlusion, but labeling is solved when occlusion is over using the method in [34].

The hands or shoes are uniformly colored in general. If the object of interest is textured, (checkered, striped or has leopard-like markings), we can track the features and represent the motion with its average velocity. The instants can be detected from the characteristics of average velocity curve as proposed in [32].

3.2 Instants and detection results

Dynamic instants are places where "significant" changes occur during the actions. Significant changes are defined such that the first derivative of the motion characteristics have a discontinuity. A dynamic instant in 3D is always projected as a dynamic instant in 2D, which is proposed by Rubin and Richards in [35]. However, while detecting the dynamic instants in a trajectory it is important to handle outliers that may arise. There are two principal sources of outliers during this detection phase.

The first source of outliers is due to the discrete nature of video sequences. Under ideal continuous conditions if there is a discontinuity, the spatiotemporal curvature will be a Dirac delta function since the numerator of the equation (1) will be infinite. However, for video sequences, the impulse degenerates to a peak in the spatiotemporal curvature values. In addition, the spatiotemporal curvature is not constant; it fluctuates when the motion is changing smoothly. Therefore, there is an ambiguity whether the peak is caused by the discontinuity. The second source of outliers is caused by the projection of the 3D trajectory onto the 2D image plane. The projection of camera may change the property of a smooth 3D curve, such that the spatiotemporal curvature may present a peak even when the object is under smooth motion. This too may generate a false detection. Fortunately, outliers caused by projection are gross errors. To handle these outliers, we propose the use of dynamic time warping method in next section, which provide an efficient and reliable basis to suppress the outliers and find correspondence between instants from different action trajectories.

Once the instants are detected, the properties of the instants are observed. The sign of an instant remains constant when the viewing direction is limited to one of the hemispheres of the viewing sphere. Here, the sign is defined as the turning direction of the trajectory at the instant. This claim is further supported by Burns *et al.* [27]. They studied the variation of relative orientation for two line segments with respect to view. We denote a clockwise turn by "+" and a counter clockwise turn by "-". Therefore, the same action should have the same

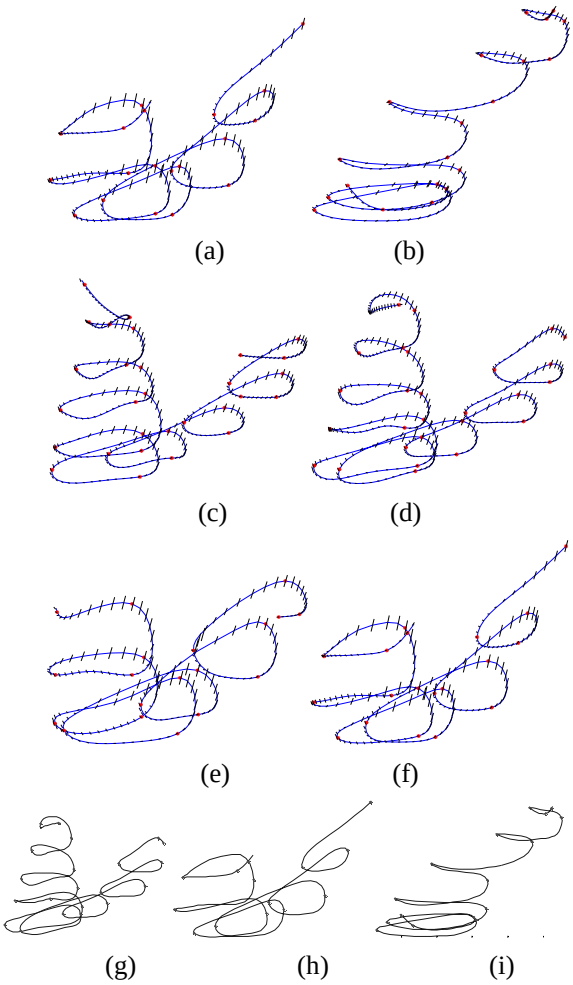


Figure 4. The trajectories of three walking sequences, the left hand side (a,c,e) are the left foot trajectories and the right hand side (b,d,f) are the right foot trajectories. The small lines display the orientation of the foot during walking, and the '*' is the instants detected by spatiotemporal curvature. The last row (g,h,i) shows the trajectories and instant detection results using only *x,y,t* information. In these experiments (g,h,I), only two instants are detected in each cycle. Finally, (g) corresponds to (d) in previous rows, so does (h) to (a) and (i) to (b), therefore, readers can compare the results easily.

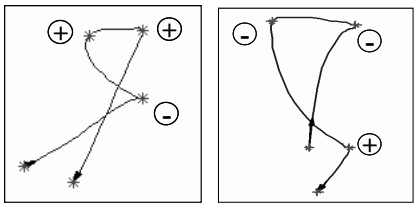


Figure 5. (a) Trajectory of the "opening cabinet" and the signs of the instants. (b) Trajectory of the "closing cabinet" and the signs of the instants.

permutation of signs for the corresponding instants. For example, the "opening cabinet" action (Figure 5a) has five instants, and the signs for the second, third and fourth instants are (-,+,+). On the other hand, the "closing cabinet" action (Figure 5b) also has five instants, but the signs of the middle three instants are (-,-,+).

4. LEARNING MODULE

As discussed in the previous sections, our system is view invariant and does not require any training data. The action database is built incrementally starting from zero and progressively growing by unsupervised learning. Each action trajectory is represented as a sequence of instances and intervals. In section 4.1 and 4.2, we discuss how to measure the similarity of the *intervals* from two different action trajectories and find the correspondence of points on the trajectories by using both spatial and temporal information of actions. Moreover, the measure is view invariant. In section 4.3 an unsupervised learning system is built, such that not only can the system recognize actions that happen before, but it also recognizes new actions.

4.1 View invariant similarity measure

In [1], the authors reported a similarity measure that is not affected by the camera viewpoint changes. They proposed a theorem based on affine epipolar geometry: *two trajectories match if and only if M is of rank at most 3.* Here, the *M* is an *observation matrix* configured as:

$$M = \begin{bmatrix} \mu_1^i & \mu_2^i & \dots & \mu_n^i \\ \nu_1^i & \nu_2^i & \dots & \nu_n^i \\ \mu_1^j & \mu_2^j & \dots & \mu_n^j \\ \nu_1^j & \nu_2^j & \dots & \nu_n^j \end{bmatrix} \quad (3)$$

where $\{(u_1^i, v_1^i), (u_2^i, v_2^i), \dots, (u_n^i, v_n^i)\}$ and $\{(u_1^j, v_1^j), (u_2^j, v_2^j), \dots, (u_n^j, v_n^j)\}$ are two set of image coordinates of dynamic instants from different viewpoints (interested readers can refer to [13] for the proof of this theorem). It's concluded from this results that if two trajectories represent the same action, and there are no numerical errors, the 4^{th} singular value of the $4\times n$ matrix *M* will be *zero*. Therefore, the similarity measure between action trajectories is determined by the matching error $dist_{i,j} = |\sigma_4|$, where σ_4 is the 4^{th} singular value of matrix *M*. The smaller $dist_{i,j}$ is, the more similar two action trajectories are. However, this method requires exact correspondence between all the instants, which is hard to get when false detections of instants are present. Furthermore, since the information during an interval is ignored when matching, the recognition is not particularly robust. Temporal information can be used to ameliorate this problem, by dynamically aligning the trajectories temporally and finding point correspondences.

4.2 View invariant dynamic time warping

There are several methods to measure the similarity between two temporal signals, such as HMM, neural network and dynamic time warping (DTW). DTW is chosen in our approach since research shows that it consistently outperforms HMM when the amount of training data is low [26]. Furthermore, in learning system, based on the similarity measure by DTW between each action trajectory, a nearest neighbor clustering is applied to achieve unsupervised learning, and new action categories are generated when needed. HMM and neural network approaches do not have this capability, because they require large amount of training data for each new model, and the training data must be prepared manually.

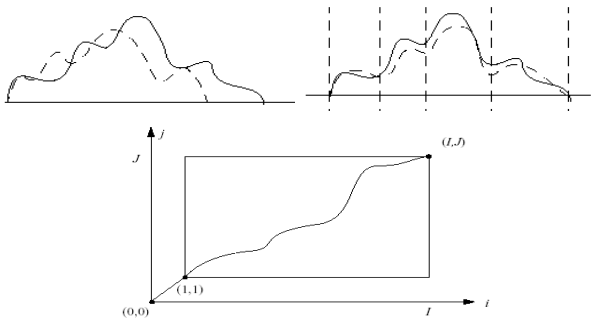


Figure 6: a) two temporal signals, b) after time warping, c) the dynamic warping path.

Dynamic Time Warping (**DTW**) is a widely used technique for matching two temporal signals [36]. It uses an optimum time expansion/compression function to do non-linear time alignment (Figure 6). For two signals ***I*** and ***J***, a distance metric ***C*** is computed to represent the alignment between the two actions, with C_{ij} representing the cost of aligning the actions up to the time instants t_i and t_j respectively. The cost of alignment is computed incrementally using the formula:

$$C_{i,j} = d_{i,j} + \min\{C_{(i-1,j)}, C_{(i-1,j-1)}, C_{(i,j-1)}\} \quad (4)$$

Here d_{ij} captures the cost of making time instants t_i and t_j correspond. The best alignment is then found by keeping track of the element that contributed to the minimization of alignment error at each step and following a path backwards through them from element C_{ij}.

So far, the above DTW approach can handle only motion information from the same viewpoint. We now introduce the shape information into the analysis through the d_{ij} metric.

Based on the view invariant similarity measure in section 4.1, we propose a view invariant DTW as follows:

1) For each trajectory, choose *4* instants from the instant detection result, such that the orders of signs are the same.

2) Execute the classic DTW algorithm, but replace the distance measure between the t_i and the t_j points of two trajectories with the following:

   $d_{(i,j)} = |\sigma_4|$, where σ_4 is the fourth singular value of matrix ***M***, and ***M*** is configured as:

$$M = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_i \\ v_1 & v_2 & v_3 & v_4 & v_i \\ u_1' & u_2' & u_3' & u_4' & u_j' \\ v_1' & v_2' & v_3' & v_4' & v_j' \end{bmatrix} \quad (5)$$

where the $\{(u_1,v_1)(u_2,v_2)(u_3,v_3),(u_4,v_4)\}$ and $\{(u_1,v_1),(u_2,v_2),(u_3,v_3),(u_4,v_4)\}$ are the (x,y) image coordinates of the *4* instants in two trajectories separately. (u_i,v_i) is the image coordinate of the i^{th} point in one trajectory, and (u'_i,v'_i) is the image coordinate of the j^{th} point in the other trajectory*.

Then record this matching distance and the correspondence result. The correspondence results are used for validating the 4 instants matching, since they must be located on the optimum path, otherwise, the result is abandoned.

3) If there are other instants available, go back to step 1 and run DTW again until all the combinations of instants are checked.

4) Find the minimal global distance from step 2, and take the correspondence as the matching of two trajectories.

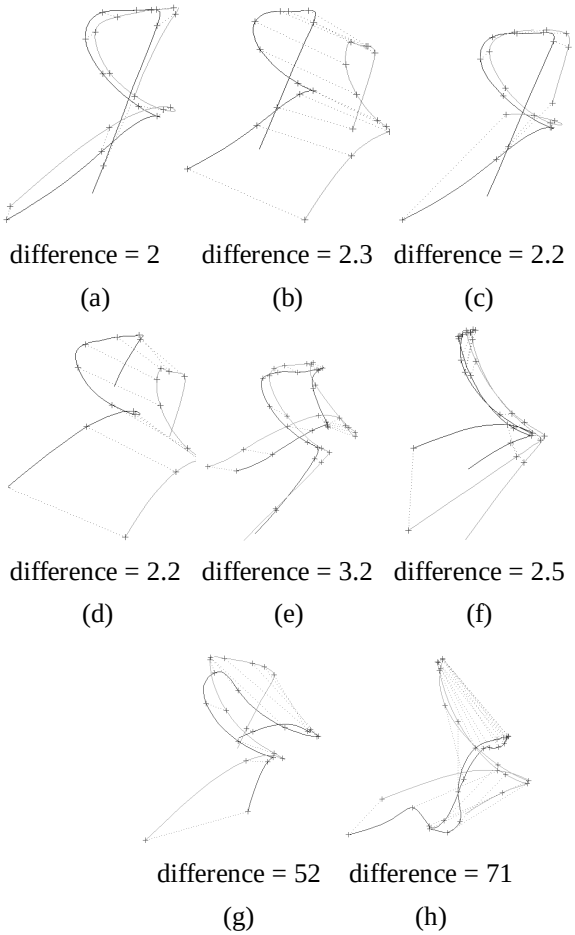


Figure 7: Some matching result. The trajectories are shown in different colors, and the red dot line with + connect the points corresponding each other. a) action 1 and action 29, b) action1 and action 43, c) action 1 and action 38, d) action 29 and action 43. e) action 3 and 6, f) action 7 and 8. All these examples are matches for the same actions, and the difference values are small. g) action 29 and action 31, h) action 7 and action 59. These two are example of view-invariant DTW for two different actions.

* note: the DTW can establish correspondence on the fly, which means that it provides the best warping path to element (i,j). Therefore, we put those corresponding points in the observation matrix M to improve the robustness of the computation.

We find that this algorithm performs DTW without being affected by viewpoint variation since the difference measure itself is not dependant on the viewpoint. Moreover, the instant outliers are suppressed if there are enough correct detections.

The instants outliers are suppressed as following: since only four instants are needed for view invariant measure and DTW, so the system iteratively chooses four pairs of instants. Because wrong correspondence give high error with DTW, and we only choose the correspondence that gives minimal difference, the right four pairs of instants correspondences are kept, and the rest of point correspondence is provided by DTW.

The view-invariant DTW can also compensate the variation of execution style, such that it can shrink the slow motion trajectories, which are longer in temporal axis, and expand the fast motion trajectories on the contrary. More significantly, this approach incorporates view-invariance and temporal-invariance in one framework. Therefore, the learning module can be simplified a lot, because the variations are eliminated.

This measure cannot be applied to the walking sequences (section 3.1), since the camera was moving, and we do not apply global motion compensation yet. The epipolar geometry is not preserved in the sequence.

4.3 Learning

In our approach, we match each action with all other actions by view invariant dynamic time warping. For each action, we select closely matched actions. All the matches whose distances are above a certain threshold are eliminated first, and only the three best matches for each action are maintained. If a particular action does not closely match to any action of its category, then it is declared a unique action. Its label may change as more evidence is gathered (Table 1).

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 29, 43, and 38; and action 29 is similar to actions 43, 38, and 1; then actions 1, 29, 38, and 43 are all similar actions due to the transitive property. This is easily implemented by using Warshall's algorithm from graph theory. Figure 7 shows some matching results and the correspondence for every 7 points of the trajectories.

5. EXPERIMENTS

We digitized several video clips recorded at 24 fps. The location of camera was changed from time to time. Seven people performed a total of 60 different actions (figure 8,9 and 10, and table 2 for descriptions). 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.

Trajectories of these actions were used to generate the view invariant representation proposed in this paper. These

Table 1. **Interpretation results. The bold face font in column indicates incorrect match.**

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

The bold font numbers indicate wrong matches.

actions.

Each of these actions was matched using method discussed in section 4.1. The results are shown in Table 1. We are pleasantly surprised to see our simple matching technique worked quite well. Only two matches were completely wrong (actions 31, 41). Three matches (33, 36, and 59) were partially incorrect. Action 31 and 36 are partially matched with opening action, such as 1. 11 out of 94 matches were wrong. The table 1 shows the results. We list the matching results with and without DTW. The improvement is significant.

Note that these matches are based on only single instance of an action. Therefore the performance of our approach is remarkable. The failures are due to the following reasons: 1) the mistakes in detecting instants. Due to the instability of taking derivatives of tracking data, the false positive and false negative of detection of instants may present. 2) The affine model used in the similarity measure, which is an approximation of real camera projection, cannot capture the activities with large variation in depth well. 3) The variation in execution of human activities. Even the same person performs the same activity differently each time, and we only considered the variation of viewpoint and the speed of the activity among the videos, therefore, further study is needed.

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 7, look very 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, which represent "put down the object, and then close the door", are all the same using matching and the transitive property.

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.

Please visit http://www.cs.ucf.edu/~vision/projects/ViewInvariance/ViewInvariance.html for html for video sequences, results, etc.

6. CONCLUSION

In this paper, we propose a computational representation of human action to capture these changes using spatiotemporal curvature of 2-D trajectories. 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 of the action agent. An interval represents the time period between two dynamic instants during which the action agent's motion characteristics do not change. Starting without a model, we use this representation for recognition and incremental learning of human actions. The Dynamic Time Warping matching is employed to match trajectories of actions using a view invariant similarity measure. The nearest-neighbor clustering approach is used to learn human actions without any training. The proposed method can discover

instances of the same action performed by different people from different viewpoints. Our approach heavily uses the properties of 3D epipolar geometry and employs rank constraints in matching 2-D projections of a 3-D action in order to eliminate the distortion due to this projection, without explicitly constructing the 3-D trajectory. The proposed approaches can be used in many applications, such as video retrieval, action analysis, human activity modeling, and automatic video surveillance.

7. References

[1] Cen Rao and Mubarak Shah, View invariance in action recognition. In IEEE International Conference on Computer Vision and Pattern Recognition, CVPR 2001, Kauai, Hawaii, Dec. 2001.

[2] Tanveer Syeda-Mahmood, A. Vasilescu, and S. Sethi, Recognizing action events from multiple viewpoints. In IEEE Workshop on Detection and Recognition of Events in Video (EVENT'01), Vancouver, Canada, July 2001.

[3] L.W. Campbell, D.A. Becker, A. Azarbayejani, A.F. Bobick, and A. Pentland, ``In-variant Features for 3D Gesture Recognition," in Proceedings, International Conference on Automatic Face and Gesture Recognition, pp. 157-162, 1996.

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

[9] M. Black and A. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation", ECCV 1998

[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..

[16] S. Niyogi and E.H. Adelson, "Analyzing and recognizing walking figures in XYT", cvpr 1994.

[17] A. Nishikawa and A. Ohnishi and F. Miyazaki, "Description and recognition of human gestures based on the transition of curvature from motion images", Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition, 1998.

[18] R. Polana and R.C. Nelson, "Detecting activities", J. of Visual Communication and Image Representation", vol 5, P172-180, 1994.

[19] M. Yang and N. Ahuja, "Extracting gestural motion trajectories", Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition, 1998,

[20] removed for blind review, Proc. IEEE Workshop on Applications of Computer Vision, WACV'98, 1998.

[21] M. Black and Y. Yacoob, "Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion",ICCV 1995.

[22] C. Bregler and A. Hertzmann and H. Biermann, "Recovering non-rigid 3d shape from image streams",CVPR, 2000.

[23] M. Black and A. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation",ECCV 1998.

[24] I. Haritaoglu and D. Harwood and L. Davis, "W4: Real-time surveillance of people and their activities",PAMI 2000 vol 22, num 8,P809-830.

[25] Y. Caspi and M. Irani, "A step towards sequence-to-sequence alignment",CVPR 2000

[26] K.Yu, J.Mason, J.Oglesby, "Speaker recognition using hidden Markov models, dynamic time warping and vector quantisation", IEEE Proceedings- Vision, Image and Signal Processing Vol.142, Issue 5, pg. 313-318, Oct 1995.

[27] J. Brian Burns, Richard S. Weiss, and Edward M. Riseman, "View variation of point-set and line-segment features", IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol. 15, No. 1, Jan. 1993.

[28] D. Moore, I. Essa, M. Hayes, "ObjectSpaces: Context Management for Action Recognition," Proceedings of the 2nd Annual Conference on Audio-Visual Biometric Person Authentication, Washington, D.C.,March 1999

[29] Jesse Hoey and James J. Little, "Representation and recognition of complex human motion". In Proc. IEEE CVPR, Hilton Head, SC, June 2000

[30] Nuria M. Oliver, Barbara Rosario, Alex P. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions", PAMI August 2000 (Vol. 22, No. 8).

[31] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. Intel Tech J Q2, 1998

[32] T. Syeda-Mahmood, "Segmenting Actions in Velocity Curve Space", ICPR 2002.

[33] Vasu Parameswaran and Rama Chellappa, “Quasi-Invariants for Human Action Representation and Recognition”.

[34] Krishnan Rangarajan, and Mubarak Shah. "Establishing Motion Correspondence , Computer Vision, Graphics and Image Processing: Image Understanding, July 1991, pp 56-73.

[35] J.M. Rubin and W.A. Richards. Boundaries of visual motion. In Tech. Rep. AIM-835. Massachusetts Institute of Technology, Apr. 1985.

[36] Trevor J. Darrell, Irfan A. Essa, and Alex P. Pentland. Task-specific gesture analysis in real-time using interpolated views. IEEE Trans. PAMI, 1995.

[37] P. H. S. Torr and A Zisserman. Feature based methods for structure and motion estimation. InW. Triggs, A. Zisserman, and R. Szeliski, editors, International Workshop on Vision Algorithms, pages 278–295, 1999.

[38] Burak Ozer, Tiehan Lv, Wayne Wolf, "A Bottom-Up Approach for Activity Recognition in Smart Rooms", IEEE International Conference on Multimedia and Expo, Lausanne, Switzerland, August 2002.

Table 2: **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.m
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 9. Sequence shows Action 3, put down the object in cabinet, then close the door.

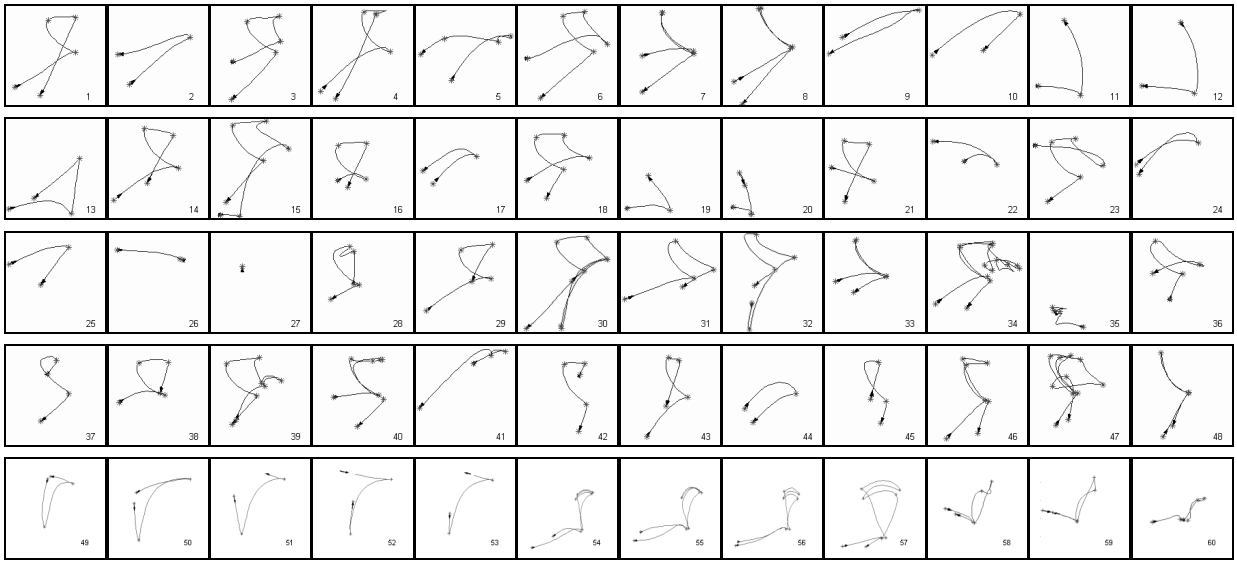


Figure 10. Trajectories of all 60 actions. The instants are shown with red "*".

Figure 8. Sequence shows Action 36, closing an overhead cabinet.