Action Recognition based on View Invariant Spatio-temporal Analysis

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

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

Tanveer Syeda-Mahmood
K57/B2, 650 Harry Road
IBM Almaden Research Center
San Jose, CA
stf@almaden.ibm.com

Abstract

In this paper, we propose an approach that retrieves actions from the videos based on the dynamic time warping of view invariant characteristics. Action is represented as a sequence of dynamic instants and intervals, which are computed using the spatiotemporal curvature of a trajectory. Dynamic Time Warping matches action trajectories using a view invariant similarity measurement. The nearest distance clustering approach is used to retrieve human actions without any training. The system is able to incrementally learn different actions without any initialization model. This paper makes two fundamental contribution to view invariant action recognition: (1) Dynamic Instant detection based on multiple motion characteristics (2) View invariant Dynamic Time Warping is used to measure similarity between two trajectories. We show successful recognition of sixty actions performed by different individuals from different viewpoints.

1. Introduction and related work

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. One of the key challenges is view invariance. While humans can recognize actions from various views easily, finding view invariant cues for recognition has been difficult to replicate in computational vision systems. Some approaches to solving this problem have proposed complex 3D recognition systems [3]. We argue that finding view invariant representation makes the problem of recognition far more tractable. Furthermore, in order to generalize view invariant recognition of actions, we lay emphasis on the ability of the system to learn unsupervised. The recognition system we propose consists of three layers: motion capturing, action representation, and learning. An action in our system is represented as a sequence of dynamic instants and intervals. 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. A system has been successfully implemented, which is able to handle actions from different viewing directions so that extensive training, context knowledge, or camera calibration is not needed. Moreover, the system can autonomously build up a recognition category database.

The issue of view invariance is addressed in previous work [1]. This work has also three layers: tracking, representation, and recognition. Previously, the system only tracked the centoid position of the hand in each frame. The representation layer detected dynamic instants only based on x,y,t information. The recognition system classified the actions only based on the spatial information of the instants, the interval information was ignored during processing, and the system assumed the instant detection was perfect and the correspondences were made just using the dynamic instants.

This study broadens the previous framework in two fundamental aspects: (1) Rather than studying point motion at dynamic instants, more motion characteristics are incorporated into the detection of instants, and we improve the anisotropic diffusion method to remove the noise in the trajectory; and (2) The motion information contained in the interval between two instants is measured and used to enhance recognition. By including the continuous information describing actions, the system recognition rate is improved greatly.

At the first layer (motion capture), body movement during actions is recorded with respect to time providing action primitives to be analyzed.

The representation layer takes the results from the motion capture layer and transforms it into a physically meaningful form; a sequence of instants and interval. We use spatio-temporal 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 layer has a central role, since representation of action primitives determines the architecture of the recognition system. A 'good' representation system should illustrate the actual event during the action.

In learning layer, we propose a matching method, such that a similarity measurement is generated from the spatio-temporal information of the action representation. Based on this similarity measurement, 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 system, and the view invariant matching algorithm, the system can take a relatively simple learning approach to achieve high recognition rate.

Early approaches to this problem 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 measurement to find the repeating pose of walking people and the reoccurrence of position of turning points.

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 measurements 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 recognize the same action at different viewing directions.

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

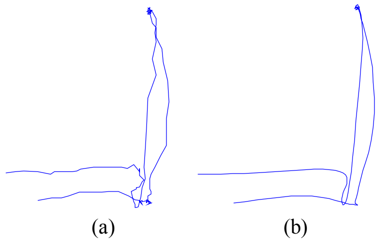


Figure 1: a) the raw input data. b) The smoothing result by PCA-Perona-Malik method.

2. Motion capturing system

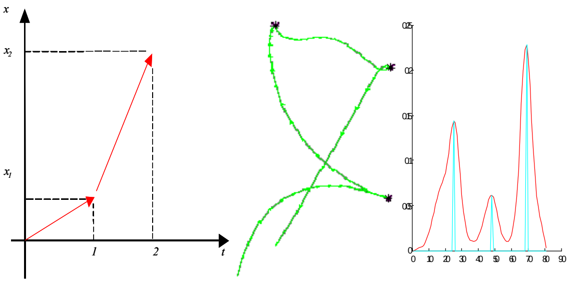
The motion capture layer detects and tracks motion of action primitives. During motion capture there are two steps: tracking and smoothing. The output of this layer is action represented as motion trajectories.

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, isolated tracking of centroids of hands do not provide sufficient information, e.g. making gesture, turning a knob, etc. Therefore, the *orientation* of hand is also tracked in our system with skin detection method as follows[6]. A small sequence of images of performer (3 to 5 frames) are used for training to generate the color predicate. The system 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.*[rs]. 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 T: $R^1 \rightarrow R^3$.

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]. 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 principal component analysis (PCA) 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 smoothing motion trajectory, which represents a hand picking up a telephone handset and then putting it back.



(a) (b) (c)

Figure 2: a) spatiotemporal curve of a 1D motion. b) An opening cabinet action trajectory with instants and intervals. c) The spatiotemporal curvature values and the peak detection results of the trajectory.

3. Action representation

In this layer, the motion trajectory recovered by the motion capture layer is interpreted into 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 curvature. 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 spatio-temporal curvature will capture this turning (figure 2a). This idea has been 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 rotation 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:

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

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 |·| denotes the determinant, and

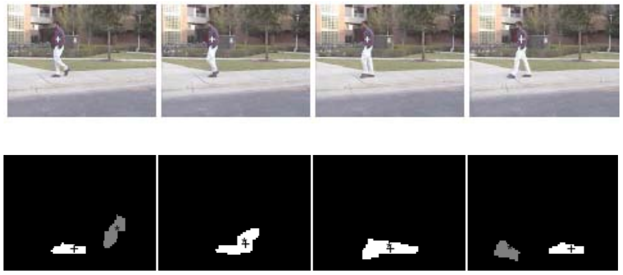


Figure 3: frame 174, 176,178, and 180 of a walking sequence and the foot tracking and labeling results. The gray color represents left feet and white color is right feet. The middle two frames have occlusion, but labeling is solved when occlusion is over.

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

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. Moreover, we can generalize this formula to hold other motion characteristics that change with respect to time.

Consider an opening overhead cabinet action (Figure 2.b). 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 represent 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". We compared the results with the detection using only x,y,t information. Using only x,y, and t we can only get two instants consistently.

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 view invariance

Dynamic instants are places where 'significant' changes occur during the actions. Significant change are defined such that the first derivative of the motion characteristics has a discontinuity. A dynamic instant in 3D is always projected as a dynamic instant in 2D. 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. 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 represent a peak even when the object is under smooth motion. This too may generate a false detection. Fortunately, the viewing direction only affects the intervals that have continuous second derivatives, and does not affect the intervals along straight lines. Experiments show that human beings always choose the straight path during daily life, since straight paths save energy and time. A simple example is that when a person wants to pick up an object, the hand approaches to the object along a straight line. It is against intuition that the hand will travel along a circle to approach the object. Therefore, outliers caused by projection are rarely gross errors. To handle these outliers we propose the use of dynamic time warping method, which provide an efficient and reliable basis to suppress the outliers and find correspondence between instants from different action trajectories.

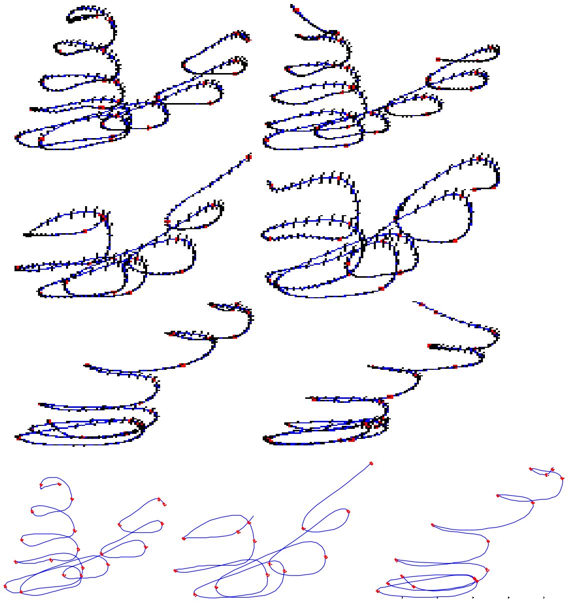


Figure 4. The trajectories three walking sequences, the left hand side is left foot and the right hand side is right foot. The small lines display the orientation value, and the '*' is the instants detected by spatiotemporal curvature. The last row is the trajectories and instant detection results, which using only *x*,*y*,*t* information

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 same permutation of signs for the corresponding instants.

4. Learning system

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 measurement 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 measurement

In [1], the authors reported a similarity measurement 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:

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

where $((u_1^i,v_1^i,),(u_2^i,v_2^i,),\ldots,(u_n^i,v_n^i,))$ and $((u_1^j,v_1^j,),(u_2^j,v_2^j,),\ldots,(u_n^j,v_n^j,))$ are two set of image coordinates of dynamic instants from different viewpoints (interested readers can refer to appendix A for 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×*n* matrix *M* will be *zero*. Therefore, the similarity measurement between action trajectories is determined by the matching error $dist_{i,j} = |\sigma_4|$, where σ_4 is the 4th eigenvalue 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.

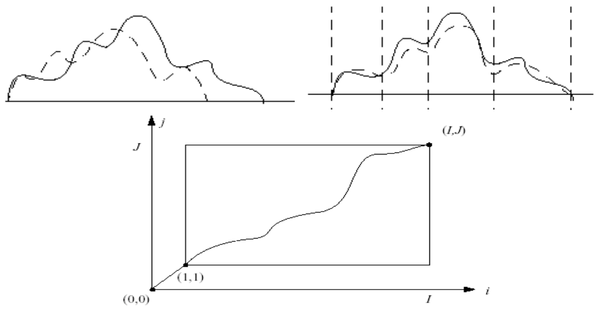


Figure 5: a) two temporal signals, b) after time warping, c) the warping path.

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 system 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 measurement 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.

Dynamic Time Warping (**DTW**) is a widely used technique for matching two temporal signals. It uses an optimum time expansion/compression function to do non-linear time alignment (Figure 5). 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:

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

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 framework can handle only motion information. We now inject shape information into the analysis through the d_{ij} metric.

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

1) For each trajectory, pick up *4* instants from the instant detection result, such that the permutations of signs are the same.
2) Execute the classic DTW algorithm, but replace the distance measurement 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 eigenvalue of matrix ***M***, and ***M*** is defined as:

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

   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 *4* instants in two trajectories separately. (u_i,v_i) is the image coordinate of the *i*th point in one trajectory, (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.

* 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 get more robust measurement.

We find that this algorithm performs DTW without being affected by viewpoint variance since the difference measurement 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 measurement 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.

This measurement can not be applied to the walking sequences (section 3.1), due to that 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, and then computing the match distances. For each action, we select closely matched actions. All the matches 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 easy implemented by Warshall's algorithm. Figure 6 shows some matching results and the correspondence for every 7 points of the trajectories. Please reference to the supplemental file to get the correspondence results.

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 6). 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 representations were interpreted by the system to learn these actions.

Each of these actions was matched using method discussed in section 4.1. The results are shown in Table 2. 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. The table 2 shows the results. We list the matching result from using only instant information for matching in table 2 also, which contains 2 three totally wrong matches and seven partial mismatches. The improvement is significant.

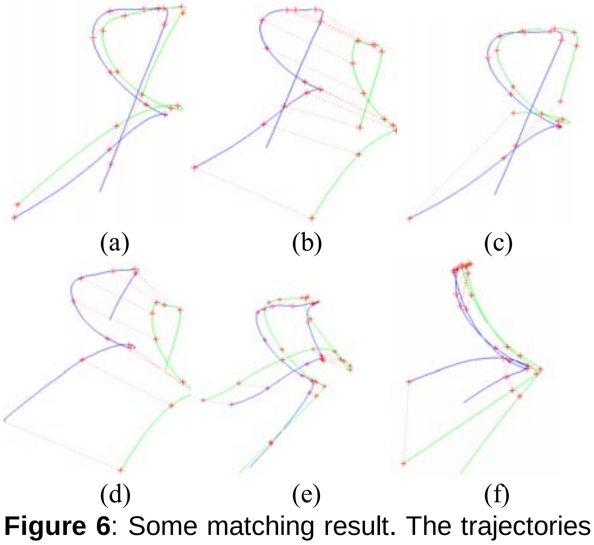


Figure 6: Some matching result. The trajectories are shown in different colors, and the red dot line with + connect the points corresponding each other. a) action 1 and action 29, b) action1 and action 43, c) action 1 and action 38, d) action 29 and action 43. e) action 3 and 6, f) action 3 and 8.

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, 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.

10. References

[1] Removed for blind review, Computer Vision and Pattern Recognition, CVPR 2001, Kauai, Hawaii, Dec 11-13, 2001

[2] Removed for blind review, Detection and Recognition of Events in Video, 2001. Proceedings. IEEE Workshop on , 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] Tanveer Syeda-Mahmood, “Segmenting actions in velocity curve space”, ICPR2002.

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

9. Appendix A

The affine camera is a special case of projective camera and proposed by Mundy and Zisserman. The projection can be represented as:

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 matching
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**

$$P_{aff} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ 0 & 0 & 0 & P_{34} \end{bmatrix}$$

A image coordinate ***u***=(u,v) can be represented as a projection of 3D point ***X***=(X,Y,Z)$^{\mathrm{T}}$:

$$u = NX + t \tag{a}$$

where ***N*** is a 2×3 matrix (with elements N_{ij}=P_{ij}/P_{34}) and ***t***=(P_{14}/P_{34}, P_{24}/P_{34})$^{\mathrm{T}}$ is a 2-vector.

A property of this affine camera is that it retains its form when the scene undergoes a 3D affine transformation. Consider a 3D point ***X*** moves to a new position ***X'*** as $X' = AX + T$, where A is a 3×3 matrix and T is a 3-vector. The new 3D position ***X'*** then projects to ***u'***=(u',v')^{T}, where

$$u' = NX' + t = N(AX + T) + t$$
$$= NAX' + (ND + t) = N'X + t'$$

A second property of the affine camera model is that relative coordinates cancel out translation effects, such that $\Delta X = X - X_0$ and $\Delta X' = X' - X'_0 = A\Delta X$. Furthermore, in the image, the points are:

$$\Delta u = u - u_0 = N\Delta X \text{ , and}$$
$$\Delta u' = u' - u'_0 = N'\Delta X = NA\Delta X$$

Therefore, the image coordinates are independent of ***T***, ***t*** and ***t'***.

The equation of epipolar line is obtained by petitioning ***N*** as (***B***|***b***), where ***B*** is a 2×2 matrix and ***b*** a 2×1 vector. From (a),

$$u = B\begin{bmatrix} X \\ Y \end{bmatrix} + Zb + t \tag{c}$$

and similar for ***N'***

$$u' = B'\begin{bmatrix} X \\ Y \end{bmatrix} + Zb' + t' \tag{d}$$

from (c) and (d), we can eliminate the world coordinates (X,Y)$^{\mathrm{T}}$, and get:

$$u' = \Gamma u + Zd + \varepsilon \tag{e}$$

with $\Gamma = B'B^{-1}$, $d = b' - \Gamma b$ and $\varepsilon = t' - \Gamma t$, and these quantities are depend only on the cameras motion – not on the scene structure. Notice ***Γ*** is a 2×2 matrix, ***d*** and ***ε*** are 2-vectors. Multiply the both sides of equation (e) with ***d***$^{\perp}$, which is the perpendicular to ***d***, and notice that ***d•*** ***d***$^{\perp}$=0, we get:

$$(x' - \Gamma x - \varepsilon)^T d^{\perp} = 0 \tag{f}$$

Then, the equation (f) can be represented as $ax' + by' + cx + dy + e = 0$. Moreover, the difference vector form is:

$$ax' + by' + cx + dy = 0 \tag{g}$$

We rewrite equation (f) as matrix form, such that $\begin{bmatrix} u' - u'_0 & v' - v'_0 & u - u_0 & v - v_0 \end{bmatrix} n = 0$, where $n = (a, b, c, d)^T$ and ***n*** is the motion parameters. Since all k points on the object share one set of motion parameters, we can represent the relationship before and after the movement as:

$$\begin{bmatrix} u'_1 - u'_0 & v'_1 - v'_0 & u_1 - u_0 & v_1 - v_0 \\ u'_2 - u'_0 & v'_2 - v'_0 & u_2 - u_0 & v_2 - v_0 \\ \vdots & \vdots & \vdots & \vdots \\ u'_k - u'_0 & v'_k - v' & u_k - u_0 & v_k - v_0 \end{bmatrix} n = M^T n = 0$$

Since ***M***^{T} is a k×4 matrix, in order to obtain a non-trivial solution for ***n***, the rank of matrix ***M*** must be at most 3.

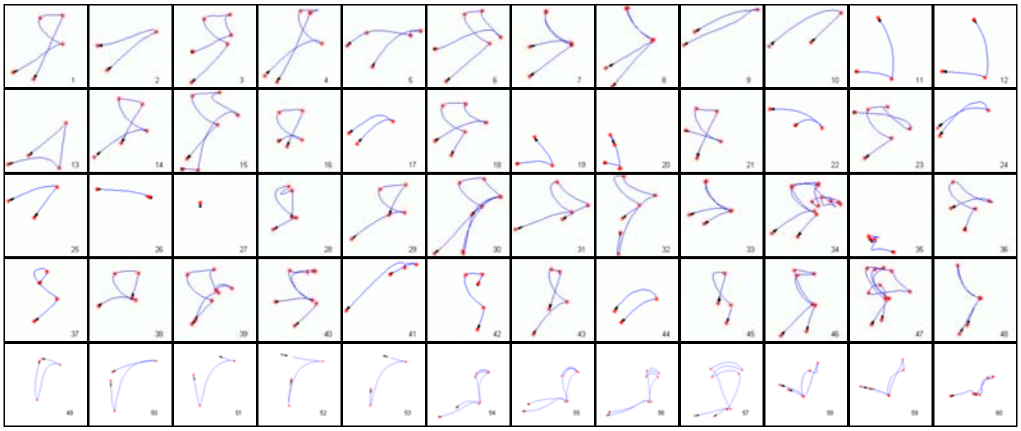


Figure 7. Trajectories of all 60 actions. The instants are shown with red "*".

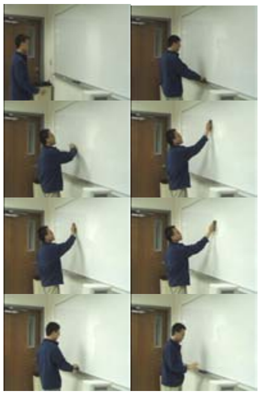


Figure 8. Sequence showing Action 56, erase the white board.

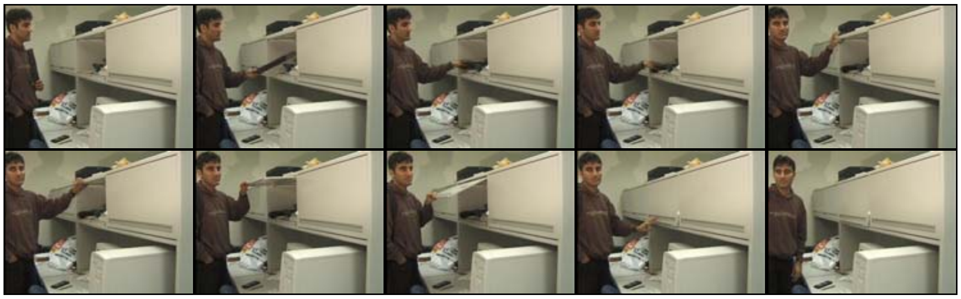


Figure 9. Sequence showing Action 3, put down the object in cabinet, then close the door.