MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Trajectory Pattern Detection by HMM Parameter Space Features and Eigenvector Clustering

Fatih Porikli

TR2004-032 January 2004

Abstract

We develop an object trajectory pattern learning method that has two significant advantages over past work. First, we represent trajectories in the HMM parameter space which overcomes the trajectory sampling problems of the existing methods. The proposed features are more expressive and enable detection of trajectory patterns that cannot be detected with the conventional trajectory representations reported so far. Second, we determine common trajectory paths by analyzing the optimal cluster number rather than using a predefined number of clusters. We compute affinity matrices and apply eigenvalue decomposition to find clusters. We prove that the number of clusters governs the number of eigenvectors used to span the feature similarity space. We are thus able to automatically compute the optimal number of patterns. We show that the proposed method accurately detects common paths for various camera setups.

ECCV 2004

TRAJECTORY PATTERN DETECTION BY HMM PARAMETER SPACE FEATURES AND EIGENVECTOR CLUSTERING

Fatih Porikli

Mitsubishi Electric Research Laboratories
fatih@merl.com

ABSTRACT

We develop an object trajectory pattern learning method that has two significant advantages over past work. First, we represent trajectories in the HMM parameter space which overcomes the trajectory sampling problems of the existing methods. The proposed features are more expressive and enable detection of trajectory patterns that cannot be detected with the conventional trajectory representations reported so far. Second, we determine common trajectory paths by analyzing the optimal cluster number rather than using a predefined number of clusters. We compute affinity matrices and apply eigenvalue decomposition to find clusters. We prove that the number of clusters governs the number of eigenvectors used to span the feature similarity space. We are thus able to automatically compute the optimal number of patterns. We show that the proposed method accurately detects common paths for various camera setups.

1. INTRODUCTION

Past work on automatic detection of events using trajectory based features has mostly consisted of extraction of trajectories followed by a supervised learning based classification. There are several attempts to interpret the object activity. For example, in [1] an activity recognition method based on view-based template matching techniques is developed. In this method, action is represented by a temporal template which is a static vector-image computed from accumulative motion properties at each point of the image sequences. An action is recognized by matching this template with the templates of known actions. Davis et al. [2] represent simple periodic events (e.g., walking) by constructing dynamic models of periodic pattern of people's movements and is dependent on the robustness of tracking. The Hidden Markov Model (HMM) has also been applied to activity recognition. Starner et. al [5] use an HMM to represent a simple event and recognize this event by computing the probability that the model produce the visual observation sequence. The distributions of object trajectories are clustered in [4]. The resulting model included hundreds of clusters of object trajectories. Stauffer et al. [6] estimated a hierarchy of similar distributions of activity based using a co-occurrence clustering. Though both of these systems learned clusters corresponding to similar activity, they describe an objects entire path through the environment.

Note that the above algorithms only use abstract representations of trajectories sometimes combined with other cues such as skin color, etc. Although the extraction of trajectories and boundaries is well studied, little investigation on the secondary outputs of a tracker has been done.

In this paper, we target a more attainable goal of detecting common patterns using improved tracking features. Since existing trajectory-based features are insufficiently expressive, they cannot be used to identify certain events such as starts and stops of motion that require a more detailed characterization. We are thus motivated to develop more expressive features. In addition to trajectory, we introduce parameter space representations of tracked objects. We find however that our proposed features have high dimensionality. Since conventional learning methods are adversely affected by high dimensionality, we are motivated to develop a new approach to clustering that is much more robust to increase in the dimensionality of the feature space.

Furthermore, unlike the past work cited above we employ an unsupervised learning method. It is based on eigenvector decomposition of the feature similarity matrices. We prove that the number of clusters governs the number of eigenvectors used to span the feature similarity space. We are thus able to automatically compute the optimal number of clusters. The computational complexity is lower than the k-means in case the dimensionality of the features is respectively higher than the number of features.

2. TRAJECTORIES TO HMM FEATURE SPACE

The trajectory $T(t) = [x(t)\ y(t)]$ of an object is represented as the collection of image coordinates that correspond to the center-of-masses of the shape in the consecutive frames. The shape of the object is often parameterized by a certain number of variables, e.g. upper-left and lower-right coordinates in case of a box, and perpendicular cords in case of an ellipsoid. In this paper, we represent trajectories, which we

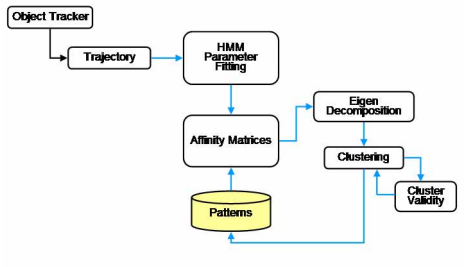

Fig. 1. Flow diagram.

obtained by an object tracker presented in [7], in terms of parameter space features.

Since feature clustering requires feature vectors to have equal lengths (dimensions), features that have varying sizes are transferred to a parameter space. This parameter space is spanned using continuous HMM variables. In order to define an HMM, we used the following elements; 1) A set of prior probabilities π, 2) A set of state transition probabilities B, 3) Mean, variance and weights of mixture models H. We keep the structure of the HMM, the number of states, and number of mixture model constant. By training a separate HMM for each feature, we construct a parameterized representation for a set of W frames, which are included within a moving window along the time axis centered around the current frame t. To decrease computation, we use overlapping windows.

We construct a vector $\gamma_f(t) = (\pi, B, H)_f(t)$ using the HMM state transfer probabilities B, model means statistics and model weights H, and the prior probabilities π. The window size is chosen such that the targeted activity fits in the duration of window. Note that, $\gamma_f(t)$ is the best fit for the features within the window since at the training phase we used those features. We will evaluate the fitness of $f(t_1)$ to $\gamma_f(t_2)$ and $f(t_2)$ to $\gamma_f(t_1)$ when we measure the similarity of the features $f(t_1)$ and $f(t_2)$.

3. FEATURES TO PATTERNS

A flow diagram of the event detection process is shown in Fig. 1. First, the HMM parameters are determined for each trajectory. We compute affinity matrices that represents the similarity of two trajectories. The similarity is computed using parameter space. The affinity matrices are decomposed to determine the largest eigenvectors and this is used to obtain the clusters of trajectories. After clustering, a deviation score is assigned to each trajectory. The trajectories that has marginal scores are then selected as outliers.

For each trajectory, an affinity matrix A is constructed. The elements a_{ij} of the matrix A are equal to the similarity of the corresponding trajectories i and j. The similarity is defined as $a_{ij} = e^{-d(i,j)/2\sigma^2}$. Note that matrix $A \in \mathcal{R}^{n\times n}$ is a real semi-positive symmetric matrix, thus $A^T = A$.

In case of the HMM parameter based features, the distance $d(i,j)$ is measured using a mutual fitness score of the models and input features as

$$\begin{aligned} d(t_i,t_j) &= |P(f(t_1)|\gamma_f(t_1)) + P(f(t_2)|\gamma_f(t_2)) \\ &\quad -P(f(t_1)|\gamma_f(t_2)) - P(f(t_2)|\gamma_f(t_1))| \end{aligned} \quad (1)$$

3.1. Eigenvector Decomposition

The decomposition of a square matrix into eigenvalues and eigenvectors is known as eigenvector decomposition. For the affinity matrix A there are n eigenvalues λ with associated eigenvectors $\mathbf{v}$ which satisfy $A\mathbf{v} = \lambda\mathbf{v}$. To find these eigenvalues, we rewrite the previous equation as $(A - \lambda I)\mathbf{v} = 0$ and determinant is computed $det(A - \lambda I) = 0$.

Let $V \equiv [\mathbf{v}_1 \ \mathbf{v}_2 \ .. \ \mathbf{v}_n]$ be a matrix formed by the columns of the eigenvectors. Let D be a diagonal matrix $diag[\lambda_1, \lambda_2, .., \lambda_n]$. Lets also assume $\lambda_1 \geq \lambda_2 \geq ..\lambda_n$. Then the eigenvalue problem becomes

$$AV = [A\mathbf{v}_1 \ .. \ A\mathbf{v}_n] = [\lambda_1\mathbf{v}_1 \ .. \ \lambda_n\mathbf{v}_n] = VD \quad (2)$$

and $A = VDV^{-1}$. Since A is symmetric, the eigenvectors corresponding to distinct eigenvalues are real and orthogonal $VV^T = V^TV = I$, which implies $A = VDV^T$.

The main idea behind iterative computation is the following. Suppose we have some subspace $\mathcal{K}$ of dimension k, over which the projected matrix A has Ritz value θ_k and a corresponding Ritz vector $\mathbf{u}_k$. Let us assume that an orthogonal basis for $\mathcal{K}$ is given by the vectors $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k$ (already determined eigenvectors).

3.2. Clustering

Although eigenvector based clustering [3] is addressed before in the literature, to our knowledge no one has established the relationship between the optimal clustering of the data distribution and the number of eigenvectors that should be used for spanning before. Here we show that the number of eigenvectors is proportional to the number of clusters.

Let a matrix P_k be a matrix in a subspace $\mathcal{K}$ that is spanned by the columns of V such as $P_k = [\mathbf{v}_1 \ \mathbf{v}_2 \ .. \ \mathbf{v}_k, \ 0]$ where V is the orthogonal basis satisfies $A = VDV^T$. Now, we define vectors $\mathbf{p}_n$ as the rows of the truncated matrix P_k as

$$P_k = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_n \end{bmatrix} = \begin{bmatrix} v_{11} & \cdots & v_{1k} & 0 & \cdots \\ v_{21} & \cdots & v_{2k} & 0 & \cdots \\ \vdots & & & & \vdots \\ v_{n1} & \cdots & v_{nk} & 0 & \cdots \end{bmatrix} \quad (3)$$

We normalize each row of matrix P_k by $p_{ij} \leftarrow p_{ij}/\sqrt{\sum_j^k p_{ij}^2}$. Then a correlation matrix is computed using the normalized

rows by $C_k = P_k P_k^T$. For a given P_k, the value of p_{ij} indicates the degree of similarity between the trajectory i and trajectory j. Values close to one correspond to a match whereas negative values and values close to zero suggest that trajectories are different. Let ϵ be a threshold that transfers values of matrix C_k to the binary quantized values of an association matrix W_k as

$$w_{ij} = \begin{cases} 1 & c_{ij} \geq \epsilon \\ 0 & c_{ij} < \epsilon \end{cases} \quad (4)$$

where $\epsilon \approx 0.5$. Then clustering becomes grouping the trajectories that have association values equal to one $w_{ij} = 1$.

To explain why this works, remember that eigenvectors are the solution of the classical extremal problem $\max \mathbf{v}^T A \mathbf{v}$ constrained by $\mathbf{v}^T \mathbf{v} = 1$. That is, find the linear combination of variables having the largest variance, with the restriction that the sum of the squared weights is 1. Minimizing the usual Lagrangian expression $\mathbf{v}^T A \mathbf{v} - \lambda(\mathbf{v}^T \mathbf{v} - 1)$ implies that $A\mathbf{v} = \lambda \mathbf{v}$. Thus, $\mathbf{v}$ is the eigenvector with the largest eigenvalue.

As a result, when we project the affinity matrix columns on the eigenvector $\mathbf{v}_1$ with the largest eigenvalue and span $\mathcal{K}_1$, the distribution of the a_{ij} will have the maximum variance therefore the maximum separation. Keep in mind that a threshold operation will perform best if the separation is high. To this end, if the distribution of values have only two distinct classes then a balanced threshold passing through the center will divide the points into two separate clusters. With the same reasoning, the eigenvector $\mathbf{v}_2$ with the second largest eigenvalue, we will obtain the basis vector that gives the best separation after normalizing the projected space using the $\mathbf{v}_1$ since $v_1 \perp \mathbf{v}_2$.

The values of the thresholds should still be computed. We obtained projections that gives us the maximum separation but we did not determine the degree of separation i.e. maximum and minimum values of projected values on the basis vectors. For convenience, we normalize the projections i.e. the *rows* of current projection matrix (V_k) as $\mathbf{p}^T \mathbf{p} = 1$ and then compute the correlation $V_k^T V_k$. Correlation will make rows that their projections are similar to get values close to 1 (equal values will give exactly 1), and dissimilar values to 0. By maximizing the separation (distance) between the points in different clusters on an orthonormal basis, we pushed for the orthogonality of points depending their clusters; $\mathbf{p}_i p_j \approx 1$ if they are in the same cluster, and $\mathbf{p}_i p_j \approx 0$ if they are not.

As opposed to using only the largest or first and second largest eigenvectors (also the generalized second minimum which is the ratio of the first and the second depending the definition of affinity), the correct number of eigenvectors should be selected with respect to the target cluster number. Using only one or two does fail for multiple clusters more than 3 as demonstrated in the next section.

After each eigenvalue computation of matrix A in the iterative algorithm, we compute a validity score α_k using the clustering results as

$$\alpha_k = \sum_{c}^{k} \frac{1}{N_c} \sum_{i,j \in Z_c} p_{ij} \quad (5)$$

where Z_c is set of trajectories included in the cluster c, N_c number of trajectories in Z_c. The validity score gets higher values for the better fits. Thus, by evaluating the local maxima of this score we determine the correct cluster number automatically. Thus, we answer the natural question of clustering; "what should be the total cluster number?"

As a summary, the clustering for a given maximum cluster number k^* includes

1. Compute A, approximate eigenvectors using Ritz values $\lambda_k \simeq \theta_k$, find eigenvectors v_k for $k = 1, .., k^*$,
2. Find $P_k = V_k V_k^T$ and Q_k for $k = 1, .., k^*$,
3. Determine clusters and calculate α_k,
4. Compute $\alpha' = d\alpha/dk$ and find local maxima.

The maximum cluster number k^* does not affect the determination of the fittest cluster; it only limits the maximum number of possible clusters that will be searched.

A question arise that why we preferred the eigen clustering to the ordinary k-means? K-means requires to the total cluster number known before starting the iterative cluster center update mechanism. Existing cluster validity scores of k-means are effective only when the clusters are compact and well separated. It may oscillate between cluster center since it needs a minimum residual threshold. Due to the fact that it models clusters by their centers, it makes a a strict Normal distribution assumption. In eigen-decomposition, mutual inter-feature distance as opposed to center-distance is used. In k-means, different initial values may cause dissimilar clusters. Besides, k-means can stuck to local optima. Therefore, k-means based cluster number estimation is not always accurate. Most importantly, the computational complexity of k-means increases with the larger sizes of the feature vectors.

4. EXPERIMENTS AND DISCUSSION

We first used a generic 1D sequence to demonstrate the properties of eigen decomposition.

We use the silhouette validity measure to evaluate fitness of the clustering results of the k-means. The silhouette index is given as

$$S_k = 1 - \sum_{i}^{n} \frac{h(i) - g(i)}{\max\{g(i), h(i)\}} \quad (6)$$

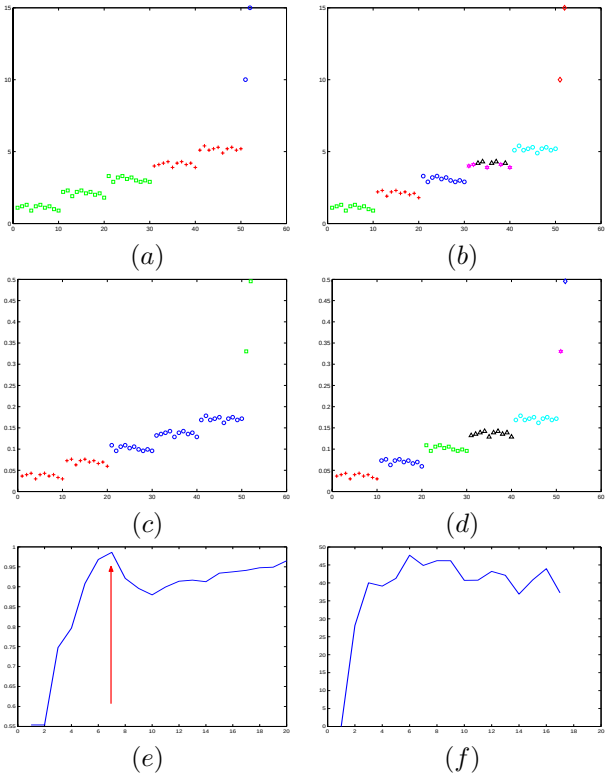

Fig. 2. First row: k-means clustering for the cluster (a) 3, (b) 7 clusters. Second row: (c) eigen decomposition clustering using 2 largest eigenvectors i.e. $k = 2$, (d) for $k = 7$. Third row: cluster validity measures for 2-20 clusters (e) Eigen clustering method evaluated by the validity score proposed in eigen clustering section. K-means clustering measured by (f) silhouette measure.

where $g(i)$ average dissimilarity of trajectory i to all other trajectories in the same cluster; $h(i)$ is the minimum of average dissimilarity of trajectory i to all trajectories in other closest cluster.

In figure 2, we illustrate experimental results that indicate that our technique finds the correct number of clusters even in cases where the K-means will fail. Furthermore, we compensate for the high dimensionality of our proposed features by devising an eigenvector based clustering technique which does not grow with the dimensionality of the feature vector unlike K-means clustering. Figure 3 presents the path detection results for two camera setups using 53 and 39 trajectories respectively.

In conclusion, the main contributions of this paper are:

- We proposed a new set of more expressive features based on trajectory that enable detection of trajectory patterns that could not be detected using conventional representations.

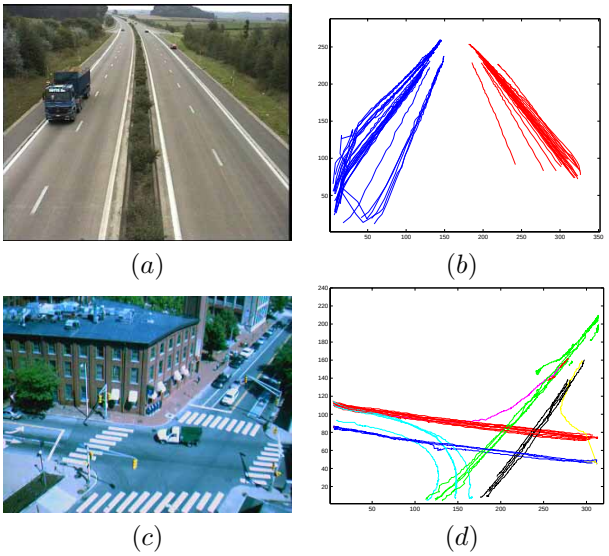

Fig. 3. First row: (a) sample image, (b) detected paths. Second row: (c) sample image, (d) detected paths.

- We showed that the number of largest eigenvalues (in absolute value) to span subspace is one less than the number of clusters.
- We proposed an unsupervised clustering framework based on the above and successfully applied it to pattern detection. The proposed framework is not adversely affected by increases in feature vector dimensionality unlike K-means clustering.

5. REFERENCES

[1] J.W. Davis and A.F. Bobick, "The representation and recognition of human movement using temporal templates", *CVPR*, 928-934, 1997.

[2] L. Davis, R. Chelappa, A. Rosenfeld, D. Harwood, I. Haritaoglu, and R. Cutler, " Visual Surveillance and Monitoring", *Proc. DARPA Image Understanding Workshop*, 73-76, 1998.

[3] G.L. Scott and H. C. Longuet-Higgins, "Feature grouping by relocalisation of eigenvectors of the proxmity matrix" *BMVC*, 103-108, 1990.

[4] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition", *BMVC*, 583592, 1995.

[5] C. Stauffer, W. Grimson. "Learning patterns of activity using real-time tracking", *IEEE Trans. on PAMI*, 22(8), 2000.

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

[7] F. Porikli and O. Tuzel, "Human body tracking by adaptive gaussian models and mean-shift analysis", *ICVS*, Graz, 2003.