MVA2005 IAPR Conference on Machine VIsion Applications, May 16-18, 2005 Tsukuba Science City, Japan

Appearance Tracker Based on Sparse Eigentemplate

Takeshi Shakunaga Yasuharu Matsubara Kiyoshi Noguchi
Department of Computer Science, Okayama University
Okayama-shi, Okayama 700-8530, Japan
shaku@cs.okayama-u.ac.jp

Abstract

A novel scheme is proposed for the efficient object tracking by using partial projections of a sparse set of pixels to eigenspaces. This paper shows a theoretical framework of the sparse eigentemplate matching and its application to a real-time face tracker. The sparse eigentemplate matching is formalized as a partial projection onto an eigenspace. Only using a small number of pixels, it facilitates an efficient template matching. In the application, a condensation framework is combined with the sparse eigentemplate matching in order to make a robust and efficient tracker. Experimental results show that the condensation tracker can track a face in real time even when the lighting condition changes.

1 Introduction

This paper discusses an object tracking method characterized by partial projections of a sparse set of pixels to eigenspaces. While eigenspaces are useful for the object/face detection as well as for the object/face recognition, it is not so easy to apply them for efficient and robust tracking. Some robust algorithms have proposed for the object/face detection using eigenspace techniques [1]. In these methods, however, iterative projections have been made along with outlier detection. The iterative projection approaches often suffer from efficient implementation and the "breakdown point" problem. In order to solve the problem, this paper proposes a novel tracking scheme without using the iterative projections.

2 Object Representation

2.1 Normalized Image Space

Normalized Image Space(NIS) is introduced by Shakunaga and Shigenari[4] for realizing a robust face recognition using eigenspaces. In the present paper, an appearance-based tracking is discussed using eigenspaces in NIS because illumination-insensitive eigenspaces can be easily constructed in NIS. The NIS concept is briefly summarized as follows: Let an n-vector $\mathbf{X}$ denote an original image with n pixels, and $\mathbf{1}$ denote an n-vector of which every element is 1. Let n-IS denote the image space consisting of all images with n pixels. The normalized image $\mathbf{x}$ of an original image $\mathbf{X}$ is then defined as

$$\mathbf{x} = \mathbf{X}/(\mathbf{1}^T\mathbf{X}). \tag{1}$$

Thus, $\mathbf{x}$ is normalized in such a way that $\mathbf{1}^T\mathbf{x} = 1$. Normalized Image Space (NIS) is the image space consisting of all normalized images of a given image space. Let n-NIS denote the NIS of n-IS. Then any nonzero image $\mathbf{X}(\neq \mathbf{0})$ in n-IS can be mapped to a point in n-NIS.

NIS provides an image representation that is invariant to changes in light intensity as long as the original image includes neither saturated points nor shadows.

2.2 Normalized Eigenspace

When an image class is given, an m-dimensional eigenspace is constructed in n-NIS by the conventional PCA from the mean vector and covariance matrix

$$\overline{\mathbf{x}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k \quad \text{and} \quad \Sigma = \frac{1}{K}\sum_{k=1}^{K}(\mathbf{x}_k - \overline{\mathbf{x}})(\mathbf{x}_k - \overline{\mathbf{x}})^T,$$

where K is the number of images in the class.

Let Λ denote a diagonal matrix in which diagonal terms are eigenvalues of Σ in descending order, and Φ a matrix in which the i-th column is the i-th eigenvector of Σ. Then PCA implies $\Lambda = \Phi^T\Sigma\Phi$. Using a submatrix Φ_m of Φ, which consists of m most significant eigenvectors, the projection $\mathbf{x}^*$ of $\mathbf{x}$ onto the eigenspace is given by

$$\mathbf{x}^* = \Phi_m^T(\mathbf{x} - \overline{\mathbf{x}}).$$

In our problem, m is a small number because most object surfaces are approximated by a Lambertian surface. Let $\mathbf{x}^\sharp$ denote the residual of the projection,

$$\mathbf{x}^\sharp \quad = \quad \mathbf{x} - \overline{\mathbf{x}} - \Phi_m\mathbf{x}^*. \tag{2}$$

Let us call the m-dimensional eigenspace the Normalized Eigenspace (NES). We also use another notation, $\langle\overline{\mathbf{x}}, \Phi_m\rangle$, which explicitly specifies $\overline{\mathbf{x}}$ and Φ_m.

2.3 Partial Projection

Let us discuss an optimization of partial projection when a NES is given for the target object and effective pixels for the partial projection are known. For this purpose, we define a part indicator matrix P; an $n \times n$ diagonal matrix, of which each diagonal element is 1 or 0. P indicates which pixels are effective for partial projection. If the j-th diagonal element, p_{jj}, is 1(0), the j-th pixel is effective(ineffective) for partial projection. When all the pixels are effective, P becomes an identity matrix, I. All P satisfy $P = P^T = PP^T$.

Suppose a NES $\langle\overline{\mathbf{x}}, \Phi_m\rangle$ is given. An image $\mathbf{X}$ in n-IS is mapped to n-NIS by $\mathbf{x} = \mathbf{X}/(\mathbf{1}^T\mathbf{X})$. Then, the conventional definition of optimum projection reduces to the minimization of

$$\epsilon_I = (\mathbf{x}' - \Phi_m\mathbf{x}^*)^T(\mathbf{x}' - \Phi_m\mathbf{x}^*), \tag{3}$$

where $\mathbf{x}' = \mathbf{x} - \overline{\mathbf{x}}$. It is well-known that this problem reduces to a linear projection:

$$\mathbf{x}^* = \Phi_m^T\mathbf{x}' = \Phi_m^T(\mathbf{x} - \overline{\mathbf{x}}). \tag{4}$$

Given P, the partial projection problem is defined as the minimization of

$$\begin{aligned}\epsilon_P &= (\mathbf{x}' - \Phi_m\mathbf{x}^*)^T P(\mathbf{x}' - \Phi_m\mathbf{x}^*) \\ &= (P\mathbf{x}' - P\Phi_m\mathbf{x}^*)^T(P\mathbf{x}' - P\Phi_m\mathbf{x}^*). \end{aligned} \tag{5}$$

The minimization of ϵ_P is equivalent to

$$P\mathbf{x}' = P\Phi_m\mathbf{x}^*. \tag{6}$$

Eq.(6) is solved if $\mathbf{x}'$ is known. In general, however, the partial projection problem is not solved because $\mathbf{x}'$ cannot be calculated directly from the given partial image, $P\mathbf{X}$. Assuming $\beta = \mathbf{1}^T\mathbf{X}$, we can get a relation

$$\mathbf{x}' = \frac{1}{\beta}\mathbf{X} - \overline{\mathbf{x}}. \tag{7}$$

Substituting Eq.(7) into Eq.(6) results in a set of linear equations:

$$P\mathbf{X} = [\,P\Phi_m \quad P\overline{\mathbf{x}}\,] \begin{matrix}\beta\mathbf{x}^* \\ \beta\end{matrix} = P\tilde{\Phi}_m\tilde{\mathbf{x}}^*. \tag{8}$$

This is solved by

$$\tilde{\mathbf{x}}^* \quad = \quad (P\tilde{\Phi}_m)^+ P\mathbf{X}, \tag{9}$$

where

$$(P\tilde{\Phi}_m)^+ \quad = \quad (\tilde{\Phi}_m^T P\tilde{\Phi}_m)^{-1}(P\tilde{\Phi}_m)^T.$$

In the solution, the pseudo-inverse matrix is calculated from $\tilde{\Phi}_m$ and P. Once the pseudo-inverse has been formed, the partial projection problem reduces to $(m+1)$ p-vector inner products where $p = \text{tr}(P)$.

3 Sparse Eigentemplate Matching

3.1 Computational cost and performance

As stated in 2.3, the partial projection problem reduces to calculation of $(m + 1)$ p-vector inner products once the pseudo-inverse is calculated, where m-dimensional NES is used and $p = \text{tr}(P)$.

If P is constant in the search, therefore, the computation cost is very low and proportional to $(m + 1)p$. On the other hand, if P is variable in the search, additional computation is rather heavier because the calculation of the pseudo-inverse is much more time-consuming than the calculation of the inner products. This is the first key point for the efficient algorithm design, and a RANSAC algorithm with random point selection is not a possible choice for our purpose.

3.2 Point selection and its feasibility

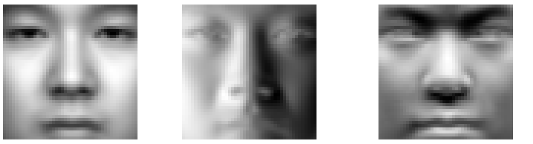
It is well known, a Lambertian object has a 3-d eigenspace for a point light source in infinity. A 2-d eigenspace is given for the object in the NIS. Since the eigenspace is due to the photometric property of Lambertian surfaces[5], we can select a small set of points when we know the object shape.

On the other hand, eigenspaces can be constructed with a class of objects which may include variations in shape and surface properties. In the cases, eigenspaces cannot be simply interpreted by photometry or physics. Therefore, we have to establish a constructive method for the point selection when an eigenspace is given.

3.3 Sparse point set selection

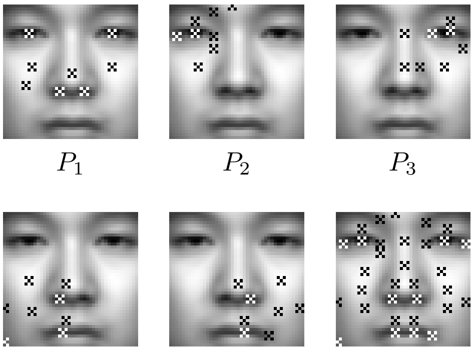
After several heuristic trials, we have arrived at a good heuristic algorithm to the problem: The algorithm selects a point set in the domain of $\overline{\mathbf{x}}$. Let the average image partitioned into s subregions. In each subregion, the algorithm selects two points which provide the regional extremal (maximum and minimum) intensities. The algorithm provides a $2s$-point set in the image. As discussed in the following section, it works well when $4 \leq s \leq 64$.

In our experiments, sparse template matching is implemented using a 2-d eigenface, of which the mean vector and the first and second eigenvectors are as shown in Fig. 1. The 2-d eigenface is constructed from 50 faces under 24 lighting conditions, respectively. Therefore, the 2-d eigenface can span various kinds of lighting conditions as shown in the figure. Figure 2 shows six sparse templates used for the sparse template matching. Five of the six templates show five sets of 8 points, P_1-P_5, which are used for partial projection. Out of 1936 pixels in the entire template, only 8 pixels indicated by "x" are selected by the regional extremal criterion. In P_1, four maximum/minimum pairs are selected in each quadrant. In P_2-P_5, four maximum/minimum pairs are selected in each quadrant of each quadrant. If different point sets, indicated by P_i and P_j, are used for partial template matching, Eq.(5) provides two measures ϵ_{P_i} and ϵ_{P_j} which are independently calculated in the two point sets. Although the two measures formally enable us to judge which point set provides a better result, they do not provide a fair comparison. That is because the measures highly depend on the amount of noise and which point set includes more noise. To prevent the unfair comparison, we use a common point set P^* with any point set P_i. That is, a common measure ϵ^* is calculated over a common point set P^* when a projection $\tilde{\mathbf{x}}^*$ is determined in any point set P_i. The common point set solves the problem of unfair comparison among different sparse sets.In our implementation, P^* is also constructed by the regional extremal criterion. That is, P^* consists of 16 maximum/minimum pairs selected from 16 subregions. Furthermore, a robust norm is used for the evaluation instead of the L-2 norm. That is, residuals for 32 points in P^* are summed up with a robust estimation by Geman-McClure function $\rho(x) = x^2/(c^2 + x^2)$, where $c = 0.5/n$.



$\overline{\mathbf{x}}$ 1st eigenvector 2nd eigenvector

Figure 1: Two dimensional eigenface.



P_1 P_2 P_3

P_4 P_5 P^*

Figure 2: Sparse eigentemplates

3.4 Sparse template condensation

The condensation algorithm is essential for the robust tracking while sparse template matching enables efficient tracking. Robust and efficient tracker can be implemented by appropriately combining the condensation framework and sparse template matching. Figure 3 shows an overview of our condensation tracker which utilizes sparse eigentemplate matching. Let us call the condensation tracker the sparse template condensation. The sparse template condensation is formalized in the three-step condensation algorithm along with sparse eigentemplate matching in the measure step.

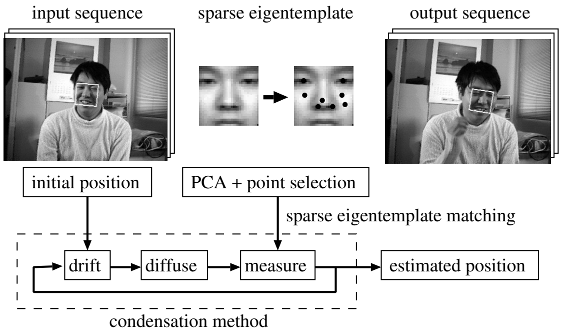


Figure 3: Overview of sparse template condensation.

In the condensation algorithm, a lot of samples (or particles) are propagated in parameter space. This scheme can also be available to selection of part indicator matrices. That is, a particle includes the indicator i as well as six pose parameters.

3.5 Details of sparse template condensation

3.5.1 Pose space and propagation process

In the original condensation algorithm [3], a sample-set $\{\mathbf{s}^{(1)}, \cdots, \mathbf{s}^{(K)}\}$ is generated in the neighborhood of the present pose. In our implementation, an eigentemplate is readily made up, and we assume the template is included in a plane. The template plane is basically transformed in 6-d pose space that covers 3-d rotation and 3-d translation. Thus, a sample (or particle) is generated and propagated in 6-d pose space. When a pose is given in the parameter space, a transformation matrix T can be calculated and it is applied for an input image. That is, $T\mathbf{X}$ is used instead of $\mathbf{X}$ for partial template matching with taking the 6-d pose parameters into account.

The propagation process is set out in terms of discrete time t. The state of the modeled object at time t is denoted by $\mathbf{w}_t$ and its history is $W_t = [\mathbf{w}_1, \cdots, \mathbf{w}_t]$. Similarly, the set of image features at t is $\mathbf{z}_t$ with history $Z_t = [\mathbf{z}_1, \cdots, \mathbf{z}_t]$.

Let $\{\mathbf{s}_t^{(k)}, k = 1, \cdots, K\}$ denote a time-stamped sample-set, and $\pi_t^{(k)}$ is a weight associated with a sample $\mathbf{s}_t^{(k)}$, where $\pi_t^{(k)}$ approximately represents the conditional state-density $p(\mathbf{w}_t|Z_t)$ at time t.

3.5.2 Drift step

In our implementation, when a weighted sample-set $\{(\mathbf{s}_{t-1}^{(k)}, \pi_{t-1}^{(k)}), k = 1, \cdots, L\}$ of $p(\mathbf{w}_{t-1}|Z_{t-1})$ is provided from time-step $t-1$, two types of assumptions are evenly selected for obtaining the sample set: a half of samples are generated from no-move assumption and the other half samples are generated from constant-move assumption in the pose space. Thus, we get a weighted sample-set consisting of $\{(\mathbf{s}_{t-1}^{(k)}, \pi_{t-1}^{(k)}/2), k = 1, \cdots, L\}$ and $\{(\mathbf{s}_{t-1}^{(k)} + \overline{\mathbf{w}}_{t-1} - \overline{\mathbf{w}}_{t-2}, \pi_{t-1}^{(k)}/2), k = L+1, \cdots, 2L\}$ where $\overline{\mathbf{w}}_t$ is an estimated pose in time-step t.

3.5.3 Diffuse step

In the diffuse step, $K(> 2L)$ samples are generated from $2L$ samples by selecting a given sample $\mathbf{s}_{t-1}^{(k)}$ with probability $\pi_{t-1}^{(k)}$. By adding a white Gaussian noise to each selected sample, we can get a new sample set $\mathbf{s}_t^{(k)}$.

3.5.4 Measure step

In the measure step, sparse template matching provides a measure ϵ^* for each sample $\mathbf{s}_t^{(k)}$, where transformation matrix T and sparse template indicator P_i are generated from $\mathbf{s}_t^{(k)}$. Let $\epsilon^{*(k)}$ denote the ϵ^* value for $\mathbf{s}_t^{(k)}$.

After selecting most similar L samples from K samples, a weight for a sample $\mathbf{s}_t^{(k)}$ is calculated by

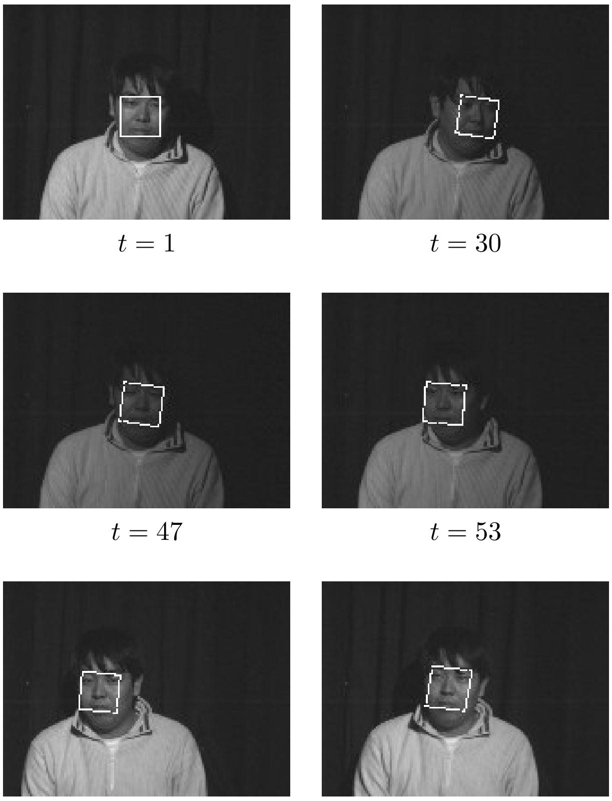
$$\pi_t^{(k)} = \frac{1/\epsilon^{*(k)}}{\sum_{j=1}^{L} 1/\epsilon^{*(j)}}.$$

Thus we can estimate a pose of the object at time-step t by

$$\overline{\mathbf{w}}_t = \sum_{k=1}^{L} \pi_t^{(k)} \mathbf{s}_t^{(k)}.$$

4 Experimental result

Figure 4 shows a tracking result of the sparse template condensation when it is applied for an image sequence



$t = 1$ $t = 30$

$t = 47$ $t = 53$

$t = 75$ $t = 90$

Figure 4: Tracking result using 1-d eigenface.

which includes a 3d rotation and illumination change. The condensation tracker successfully tracks a face using a set of sparse template matching at frame rate. The face tracking is compared among 0-d, 1-d, and 2-d eigenfaces, where the m-th eigenface is given by $\langle \overline{\mathbf{x}}, \Phi_m \rangle$ and $\tilde{\Phi}_0 = \overline{\mathbf{x}}$. Table 1 compares success rates, estimation errors and processing time among 0-d, 1-d, and 2-d eigenfaces. This table shows that success rates are 100 % when 1-d or 2-d eigenfaces are used. Processing time is almost independent of the dimension of eigenfaces. Concerning the estimation errors, the 1-d eigenface provides the best result among the three eigenfaces.

Table 1: Tracking performance vs dimensionality of eigentemplate

m	Success rate [%]	Error[pixel]		Proc. time [msec/frame]
		mean	st.dev	
0	95.7	3.09	0.64	10.5
1	100.0	2.23	0.15	10.6
2	100.0	2.68	0.12	10.7

5 Conclusions

A novel condensation tracking scheme is proposed based on the sparse eigentemplate matching. The basic idea is the use of partial projection instead of iterative robust projection in order to prevent the "breakdown point" problem. We also show a heuristic but practical method to select a small point set which facilitates the efficient tracking. We have confirmed that the proposed scheme works well for the face tracking in 3-d space. We hope the sparse template matching will collaborate with other methods related to the motion tracking [1, 2, 6].

This work has been supported in part by Grant-In-Aid for Science Research under Grant No.15300062 from the Ministry of Education, Science, Sports, and Culture, Japanese Government.

References

[1] M. Black and A. Jepson, "Eigentracking: Robust Matching and Tracking of Articulated Objects using a View-based Representation", International Journal of Computer Vision, vol. 26, no. 1, pp. 63-84, 1998.

[2] G. D. Hager and P. N. Belhumeur, "Efficient Region Tracking with Parametric Models of Geometry and Illumination", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 10, pp.1025-1039, 1998.

[3] M. Isard and A. Blake, "Condensation – Conditional Density Propagation for Visual Tracking", International Journal of Computer Vision, vol. 29, no. 1, pp. 5-28, 1998.

[4] T. Shakunaga and K. Shigenari, "Decomposed Eigenface for Face Recognition under Various Lighting Conditions", Proc. CVPR2001, vol. 1, pp. 864-871, 2001.

[5] A. Shashua, "Geometry and Photometry in 3D visual recognition", Ph.D. Thesis, MIT, 1992.

[6] P. H. S. Torr and C. Davidson, "IMPSAC: Synthesis of Importance Sampling and Random Sample Consensus", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 25, no. 3, pp.354-364, 2003.