Object Tracking and Recognition Based on CPCA

Kap-Ho Seo and Ju-Jang Lee

Department of Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology
373-1, Guseong-dong, Yuseong-gu, Daejeon, 305-701, Republic of Korea
Tel.: +82-42-869-5432, Fax.: +82-42-869-3410
E-mail: khseo@odyssey.kaist.ac.kr

***Abstract*— In recent years, computer vision research has witnessed a growing interest in subset analysis techniques. In particular, eigenvector decomposition has been shown to be a highly effective tool for problems which has high-dimensional signal formats (e.g., an image array) but, nevertheless, represent visual phenomena which are intrinsically low-dimensional. Subspace analysis is heavily used in appearance-based modelling and recognition where the principal modes or the characteristic degrees-of-freedom are extracted and used for description, detection, and recognition.**

The complex nonlinear appearance manifold expressed as a collection of subsets, and the connectivity among them. The connectivity encodes the transition probability between images in each manifold and is learned from a training video sequences.

When we track and recognize the object, a single frame image is used for that tasks. In this case based on PCA, the undesired classification/recognition results often occur.

In this paper, Condensation PCA (CPCA) presentation is introduced, which can be used for spatio-temporal alignment in tracking and recognition tasks.

I. INTRODUCTION

Many imaging applications are based on successful image classification. It is evident that the performance of these algorithms is directly linked to the performance of the classification. Thus, in order to be effective, image classification algorithms need features that well express relevant image properties.

An important aspect of modelling and visualizing high dimensional data involves the removal of redundancies by finding a lower dimensional subspace that best captures the data characteristics. One of the simplest and most general-purpose ones is a statistical method known as Principal Component Analysis (PCA), which finds the linear embedding subspace that maximizes the variance of the projected data. Principal component analysis has already successfully been implemented in image classification for many tasks.

In recent years, computer vision research has witnessed a growing interest in subset analysis techniques. In particular, eigenvector decomposition has been shown to be a highly effective tool for problems which has high-dimensional signal formats (e.g., an image array) but, nevertheless, represent visual phenomena which are intrinsically low-dimensional. Subspace analysis is heavily used in appearance-based modelling and recognition where the principal modes or the characteristic degrees-of-freedom are extracted and used for description, detection, and recognition.

Subspace methods are often critical in machine learning where they are used to extract low-dimensional manifolds comprised of statistically uncorrelated or independent variables which tent to simplify tasks such as regression, classification, and density estimation. Despite this high-dimensional embedding, the natural constraints of the physical world (and the imaging process) dictate that the data will, in fact, lie in a lower-dimensional (though possibly disjoint) manifold. The primary goal of subspace analysis is to identify, represent, and parameterize the manifold in accordance with some optimality criteria.

There are many kinds of subspace methods. The Karhunen-Loeve Transform (KLT) and Principal Component Analysis (PCA) [1] are examples of eigenvector-based techniques which are commonly used for dimensionality reduction and feature extraction. Independent Factor Analysis [2] and more specifically Independent Component Analysis (ICA) [3] is another linear decomposition which seeks statistically independent and non-Gaussian components, modelling the observed data as a linear mixture of unknown independent sources. Nonlinear PCA (NLPCA) [4], nonlinear Principal Surfaces, Kernel PCA, and nonlinear latent variable models are various extensions of these linear techniques.

Principal components can be used to reduce the number of variables in statistical analysis. Different methods for selecting the number of principal components to retain have been suggested. One simple criterion is to retain components with associated eigenvalues greater than the average eigenvalue. Principal components have a variety of useful properties:

- The eigenvectors are orthogonal, so the principal components represent jointly perpendicular directions through the space of the original variables.
- The principal component scores are jointly uncorrelated. Note that this property is quite distinct from the previous one.
- The first principal component has the largest variance

of any unit-length linear combination of the observed variables. The jth principal component has the largest variance of any unit-length linear combination orthogonal to the first j-1 principal components. The last principal component has the smallest variance of any linear combination of the original variables.
- The scores on the first j principal components have the highest possible generalized variance of any set of unit-length linear combinations of the original variables.
- In geometric terms, the j-dimensional linear subspace spanned by the first j principal components gives the best possible fit to the data points as measured by the sum of squared perpendicular distances from each data point to the subspace. This is in contrast to the geometric interpretation of least squares regression, which minimizes the sum of squared vertical distances. For example, suppose you have two variables. Then, the first principal component minimizes the sum of squared perpendicular distances from the points to the first principal axis. This is in contrast to least squares, which would minimize the sum of squared vertical distances from the points to the fitted line.

PCA is reformulated in a probabilistic framework, called the term Probabilistic Principal Component Analysis (PPCA) [5]. PPCA essentially augments the linear mapping from the PCA space to the observed data space by assuming the observed data to be corrupted by isotropic Gaussian noise, and the PCA coefficients to follow an isotropic Gaussian distribution in the embedding subspace. In the limiting case where the observation noise variance goes to zero, standard PCA is recovered.

II. Problem Statement

Recognition in video offers the opportunity to integrate information temporally across the video sequence, which may help to increase the recognition rates. Our framework exploits temporal coherence in the following ways. The appearance model is composed of a collection of pose manifolds, and a matrix of transition probabilities to connect them. The transition probabilities among the pose manifolds are learned from training videos each one characterizes the probability of moving from one pose to another pose between any two consecutive frames. We use the transition probability to implicitly infer the appropriate pose for each incoming video frame, and then integrate this information by Bayes' rule to perform the recognition task. Therefore, our method effectively captures the dynamics of pose changes and thereby exploits the temporal information in a video sequence for recognition.

Usually, there are two kinds of schemes in performing tracking and recognition task: tracking-then-recognition and tracking-and-recognition. The tracking-then-recognition approach attempts to resolve uncertainties in tracking and recognition sequentially and separately. There are several unresolved issue in the tracking-then-recognition approach: criteria for selecting good frames and estimation of parameters for registration. Also, still-to-still recognition does not effectively exploit temporal information. A common strategy that selects several good frames, performs recognition on each frame and then votes on these recognition results for a final solution might be ad hoc. To overcome these difficulties, we propose a tracking-and-recognition approach, which attempts to resolve uncertainties in tracking and recognition simultaneously in a unified probabilistic framework. To fuse temporal information, the temporal probabilistic PCA is adopted in this paper.

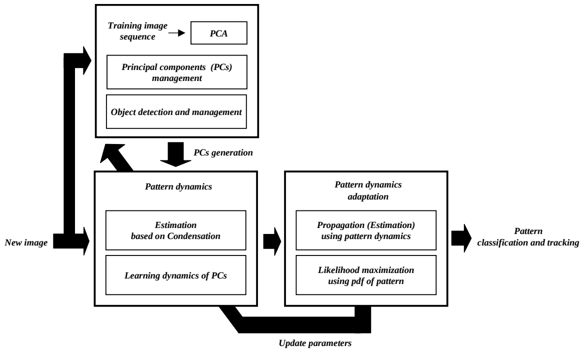

Training image sequence
PCA
Principal components (PCs) management
Object detection and management
PCs generation
New image
Pattern dynamics
Estimation based on Condensation
Learning dynamics of PCs
Pattern dynamics adaptation
Propagation (Estimation) using pattern dynamics
Likelihood maximization using pdf of pattern
Pattern classification and tracking
Update parameters

Fig. 1. Flow chart of Condensation PCA

III. Condensation PCA

As we considered the problem of the frame-based recognition, Condensation PCA (CPCA) is proposed to solve this task. The algorithm and operational flow are introduced.

We have developed a framework for prediction and update which propagates the probabilities of recognition and the recognition parameters over time. The prediction-update requires the use of a filter. We have used the Condensation algorithm by Isard and Blake [6]. The probability distribution over all the recognition parameters is represented by random samples. The distribution then evolves over time as the input data change. Kalman filters have been traditionally used for the purpose of incorporating the temporal aspect. However, Kalman filtering is based on Gaussian densities that are unimodal and, hence, cannot represent non-Gaussian densities and simultaneous multiple hypotheses, which the Condensation tracker can. The Condensation filter uses dynamical models to propagate the recognition parameters over time by incorporating recognition information at each stage. By combining it with Factored sampling, we only propagate samples with higher probabilistic weights over time. Thus, Condensation filtering together with the factored sampling is appropriate for our purpose.

A. Formulation

1) Principal components generation: PCA can be defined in an intuitive way using a recursive formulation. Having determined the first $k-1$ principal components, the k-th principal component is determined as the principal component of the residual:

$$w_k = \arg \max_{\|w\|=1} E\{[w^T(x - \sum_{i-1}^{k-1} w_i w_i^T x)]^2\}. \tag{1}$$

The principal components are then given by

$$s_i = w_i^T x. \tag{2}$$

Therefore, the new image $\tilde{x}$ is reconstructed by the principal components s_i as following equation,

$$\tilde{x} = \sum_{i}^{N} \tilde{w}_i s_i \tag{3}$$

where N is the number of the principal components. Before the input image is classified to the predefined class, we should calculate the principal components s_i.

In order to reconstruct or recognize the input image, we should select suitable value of w_i. But, as we know, the real image has lots of noise factor, and the image condition is always not good. Since real image is effected by illumination, occlusion, and etc., it is hard to expect that the exact result is inferred from a single image. In order to prevent to misclassification, we should consider the change over time into PCA algorithm. So, in this research, equation 3 is expanded to the temporal PCA model. The concept of temporal PCA is very easy to understand.

Let $\mathbf{w}_t$ be the PCA parameter $\mathbf{w} = \{w_1, w_2, \ldots, w_N\}$ at the time t. Then $\mathbf{w}_t$ is presented by $\mathbf{w}_t = \{w_{t1}, w_{t2}, \ldots, w_{tN}\}$. The parameter $\mathbf{w}$ represent the state of each class. As previously considered, the input image can be represented by the product of principal components and parameter $\mathbf{w}$. Since the principal components are determined in the stage of training the system, we can focus main interest on the variation of $\mathbf{w}$ over time in order to recognize and track the classification of object.

2) Stochastic pattern dynamics: An accurate dynamical model is essential for robust tracking and to achieve real-time performance. The dynamics of an object is described by a Kth order Markov model. A linear-Gaussian Markov model of order K is an autoregressive process (ARP) defined by

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{n}_{t-1}) = \sum_{k=1}^{K} A_k \mathbf{x}_{t-k} + B\mathbf{n}_{t-1} \tag{4}$$

where $\mathbf{x}_t$ is a state vector at time t, and $\mathbf{n}_{t-1}$ is a noise vector with a known distribution that allows to model uncertainties in the system function f.

The dynamics of the corresponding probability function is then described by a mapping that calculates $p(\mathbf{x}_t)$ from $p(\mathbf{x}_{t-1})$ as

$$p(\mathbf{x}_t) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}, \tag{5}$$

where $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the process density describing the stochastic dynamics. This equation is used to estimate the probability distribution for the next time step.

3) Measurement probability function: As the observed scene changes over time, the probability function evolves to represent the altered object states.

We can probabilistically summarize the PCA as follows.

$$\begin{aligned} k^* &= \arg\min_k d_H(I, M_k) \\ &= \arg\min_k \textstyle\int_{M_k} d(x, I) p_{M_k}(x|I) dx \\ &= \arg\min_k \textstyle\sum_{i=1}^{m} p(C^{ki}|I) d_H(I, C^{ki}). \end{aligned} \tag{6}$$

By Bayesian's rule, equation 6 is reformulated as follows,

$$\begin{aligned} k^* &= \arg\min_k \textstyle\sum_{i=1}^{m} p(C^{ki}|I) d_H(I, C^{ki}) \\ &= \arg\min_k \textstyle\sum_{i=1}^{m} \{\alpha p(I_t|C_t^{ki}) \cdot \\ &\quad \textstyle\sum_{j=1}^{m} p(C_t^{ki}|C_{t-1}^{kj}) p(C_{t-1}^{kj}|I_{t-1} I_{0:t-2})\} d_H(I, C^{ki}) \end{aligned} \tag{7}$$

From the above function, I will design the probability function for the adaptation of condensation. I choose the two term, $p(C_t^{ki}|C_{t-1}^{kj})$ and $p(I_t|C_t^{ki})$, in equation 7. The temporal information of pattern dynamics is included in the term $p(C_t^{ki}|C_{t-1}^{kj})$, which is the transition probability between the manifolds. That is the transition probability $p(C_t^{ki}|C_{t-1}^{kj})$ is related implicity to the geodesic distance between C^{ki} and C^{kj}.

Then, the probability function for Condensation PCA is as following equation,

$$\pi(\mathbf{x}_t) = p(C_{\mathbf{x}_t}|C_{\mathbf{x}_{t-1}}) p(I_t|C_{\mathbf{x}_t}). \tag{8}$$

The transition probability $p(C_{\mathbf{x}_t}|C_{\mathbf{x}_{t-1}})$ is defined by counting the *actual* transition between different S_i observed in the image sequences:

$$p(C_{\mathbf{x}_t}|C_{\mathbf{x}_{t-1}}) = \frac{1}{\Lambda_{\mathbf{X}}} \sum_{q=2}^{l} \delta(I_{q-1} \in C_{\mathbf{x}_t}) \delta(I_q \in C_{\mathbf{x}_{t-1}}) \tag{9}$$

where $\delta(I_q \in C_{\mathbf{x}_t}) = 1$ if $I_q \in C_{\mathbf{x}_t}$ and otherwise it is 0. The normalizing constant $\Lambda_{\mathbf{X}}$ ensures that

$$\sum_{j=1}^{m} p(C_{\mathbf{x}_t}|C_{\mathbf{x}_{t-1}}) = 1. \tag{10}$$

where we set $p(C_{\mathbf{x}_t}|C_{\mathbf{x}_{t-1}})$ to a constant κ if $C_{\mathbf{x}_t}$ is equal to $C_{\mathbf{x}_{t-1}}$.

With C^{ki} and its linear approximation L_{ki} defined, we can define how $p(I|C^{ki})$ can be calculated. We can compute the L^2 distances $\hat{d}_{ki} = d_H(I, L_{ki})$ from I to each L_{ki}.

We treat $\hat{d}_{ki}$ as an estimate of the true distance from I to C^{ki}, i.e., $d_H(I, C^{ki}) = d_H(I, L_{ki})$. $p(I|C^{ki})$ is defined as

$$p(I|C^{ki}) = \frac{1}{\Lambda_{ki}} \exp\left(\frac{-\hat{d}_{ki}^2}{2\sigma^2}\right) \tag{11}$$

with $\Lambda_{ki} = \sum_{i=1}^{m} \exp(-\hat{d}_{ki}^2/2\sigma^2)$.

B. The Adaptation of Condensation PCA

When we consider the problem of classifying the static/unmovable object which changes only its shape, it is enough to adapt pattern parameter **x**. But, in order to track the moving object, we should consider the tracking parameters such as position and scale of the object in image.

The recognition and tracking parameters which we need to predict and update are:

- (x, y): position. Tracking parameter. Position ranges over the entire image.
- s: scale. Recognition parameter. We use a discrete range of scales that have been empirically chosen.
- **w**: weight for PCA. Recognition parameter. The product of w parameter and training PCA image is used for generating mask image.

The state vector at time t, S_t is defined to be a vector of parameters $S_t = [x_t, \dot{x}_t, y_t, \dot{y}_t, s_t, \mathbf{w}_t, \dot{\mathbf{w}}_t]^T$. The observations at each stage are the probability values computed by the probability function. Given this conditional probability distribution, a discrete representation of the probability distribution can be constructed over the possible states. Our proposed algorithm is divided into four steps:

1) Step 1: PC generation and Initialization: Before starting to recognize and track the object, the principal components, which will be used for recognizing the object in further process, should be determined.

If the problem is the single object tracking, the possible shape of the desired object is memorized into the principal component from the training video sequence. Especially, the tracking object is rigid and only rotatable on top view, the shape of that object can be generated manually instead of gathering from video sequence. In case of recognize and track the multiple object, each object can be considered as the different class. Also, its shape can be the different manifold of that class. When the shape-changeable object problem, such as hand gesture recognition, is considered, it is possible that the problem is either one class with many manifolds or many classes with one manifold.

We initialize the parameters provided by the detection process for the first frame of the video sequence.

2) Step 2: Selection: We use factored sampling to sample the states at time $t-1$. These are then used for propagation to the next time instant. The sampling method causes the states to be chosen depending on the associated weights. Samples with high probability value are more likely to get propagated.

3) Step 3: Prediction: We use a zero order Markov model for the prediction of a new state and the definition of the state transition probabilities

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{n}_{t-1}) = A\mathbf{x}_{t-1} + B\mathbf{n}_{t-1}. \qquad (12)$$

In equation 12, A and B are the dynamics matrix of the system and object. That means that the processing time and the motion of object affects A and B. These parameter settings are adequate as the motion between subsequent frames of a video is, in general, limited. In case of a rapid motion, the tracker can lose the object. In that case, the parameter for determining $\mathbf{n}_t$ is determined more largely.

4) Step 4: Update: The probabilities of the predicted states are calculated using equation 8. Each predicted sample has the probability value for selecting for next state. The predicted sample with the highest probability value is selected as the next state, then the probability function for next iteration is updated using the selected sample. Especially, $p(I_t|C_{\mathbf{x}_t})$ in equation 8 means the distance between the input image and predicted pattern image.

IV. Experimental Results

We demonstrated the performance of Condensation PCA by applying it to real image sequences. We have implemented the proposed algorithm on a Pentium-IV 2.4 GHz PC with 512MB RAM, and a Logitech QuickCam-Pro USB camera. The sequential image was captured at the rate of 10 frames/sec. Each image has the resolution of 320×240 pixels and depth of 8-bit gray scale.

For simplicity of experiment, the shape of target object is known. Its training image for generating the principal components was gathered from the training video sequence. Since the purpose of this paper is the real-time recognition-and-tracking algorithm, the processing time and performance on real-time operation is most significant. Therefore, after experiments, i will discuss on the performance of Condensation PCA comparing with other algorithm.

The state vector for Condensation algorithm has the following form:

$$S_t = [x_t, \dot{x}_t, y_t, \dot{y}_t, s_t, \mathbf{w}_t, \dot{\mathbf{w}}_t]^T \qquad (13)$$

where (x, y) locates the center of the object, s is the scale factor of the object, and **w** is the pattern parameter of PCA. In the experiments, 100 samples were used at each iteration.

A. Experimental Results

Experiments were performed on three cases. In the first case, experiment was carried out to test the proposed algorithm for tracking a fast moving object on a complex background. Second experiment tested the performance in the presence of camera motion. In third experiments, the simple hand gesture video sequence was used to testing the proposed algorithm. Through the experiments, the effectiveness for real-time operation will be discussed.

Case 1: Single object moving on the complex background

The performance of tracking and recognition in the case of a fast moving object on a complex background is evaluated in case 1. This experiment was implemented in order to demonstrate that the motion information is essential for object tracking, and the temporal information is essential for exact object recognition. In this simple experiment, I will track and recognize the toy car moving on the complex background. In this experiment, the recognition result may not be useful. But, if we hope to know how object moves or rotates in a image, we could use the recognition result. Recognition result contain the information for the shape change of the target object. Before

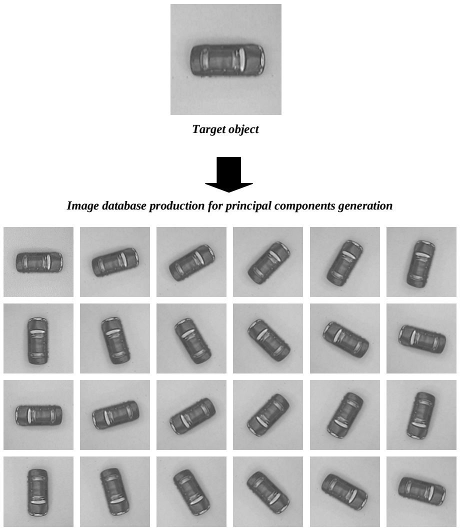

Target object
Image database production for principal components generation

Fig. 2. Target object and its image database for principal components generation. In experiments, the target object is a toy car. Before performing tracking and recognition task, its image without background was captured and image database for generating principal components is constructed. This example shows that 24 images are generated.

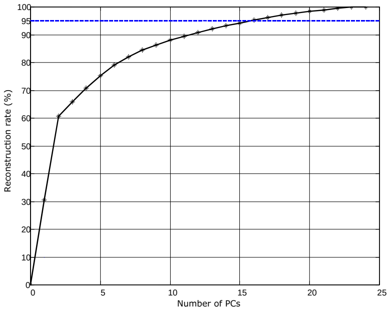
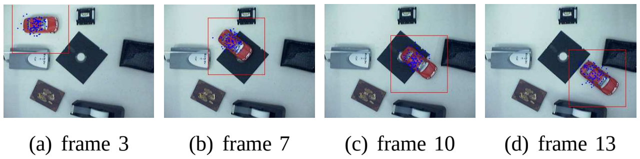
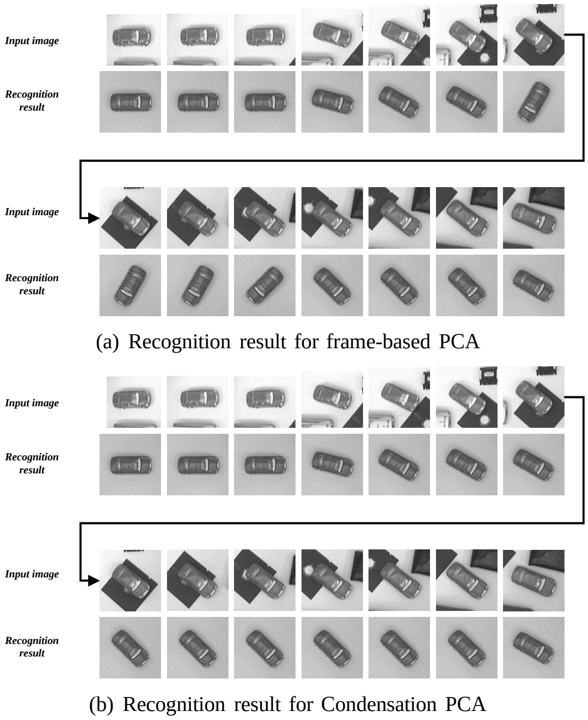


Fig. 3. Plot for reconstruction rate (%) vs number of principal components. Usually, 95% reconstruction rate is sufficient for minimum of the principal components which will be used in recognition task.



(a) frame 3 (b) frame 7 (c) frame 10 (d) frame 13

Fig. 4. Experimental results for Conventional PCA in case 1.



(a) Recognition result for frame-based PCA

(b) Recognition result for Condensation PCA

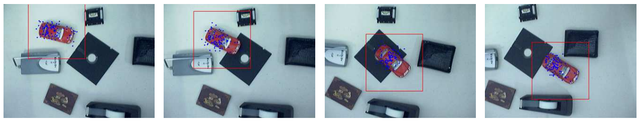
Fig. 5. Recognition results on case 1.

we perform the tracking and recognition task, we should define the shape/database of target object. Figure 2 shows that 24 images are produced for generating the principal components.

Using these images, we should generate the principal component used in Condensation PCA. Usually, 95% reconstruction rate is sufficient for minimum of the principal components which will be used in recognition task. If we use the principal components under the sufficient number, the recognition result will be not good. Otherwise, if we use the principal components over the sufficient number, the calculation time will be long. Usually, the tracking and recognition task should be performed in real-time. And, since I want to design the system to apply the algorithm to real-time system, I selected 16 principal components with tradeoff between calculation time and recognition performance.

Using the 16 selected principal components, I performed the tracking and recognition task in case 1. As shown in figure 6, a toy car moves from left-top to right-bottom position with rotation. Since a toy car is rigid object, its shape doesn't change. But, complex background affect the recognition result. Before comparing the proposed algorithm with frame-based PCA, the experimental result on case 1 is shown in figure 6. Testing sequence has 14 frames. Images on algorithm testing are used with gray scale. In frame 1, Condensation PCA tried to search a target object on the entire region in image. After first frame, Condensation PCA adapt its parameter using temporal information.

Even though the background is complex, Conventional PCA tracked a toy car without missing any frame. However, the recognition result is different between Condensation PCA and frame-based PCA (Conventional PCA). Frame-based PCA doesn't use the temporal information. It falsely recognize the shape of target object through frame 7 to 10.



(a) frame 1 (b) frame 5 (c) frame 13 (d) frame 17

Fig. 6. Experimental results for Condensation PCA in case 2.

Case 2: With camera motion Second experiment was carried out in order to show the capability of Condensation PCA in real-time tracking while the camera is in motion. This experiment shows the effectiveness to adapt condensation filter for tracking and recognition. Usually, if there are camera motion, there occurs many interesting pixels on input image. Therefore, the computation burden also increases. Especially, the pixel comparison technique such as the optical flow takes long calculation time for only tracking. As mentioned ahead, I want to design the system to operate in real-time at frame rate of 10 frame/sec. Figure 6 shows the experimental results in case 2. As same as case 1, Condensation PCA tracked a toy car through the whole image sequence. Discussion on calculation time for case 1 and case 2 will be considered in section .

V. Discussion

In this section, two considerations are described to evaluate the performance of Condensation PCA compared to other approaches. Those are tracking-and-recognition performance and calculation time. In real-time application, those are the most important factors.

A. Remarks on tracking-and-recognition performance

Usually, other feature tracking approaches such as need additional recognition operator. But, Condensation PCA perform simultaneously tracking task and recognition task since it uses the parameters which has tracking factor and recognition factor.

Since Condensation PCA uses the temporal information within its algorithm, it doesn't notice false recognition alarm frame-by-frame. As described in experiment (case 1), next recognition result is based on the current recognition state. It makes the recognition performance of Condensation PCA to be robust.

B. Remarks on calculation time

In real-time systems, calculation time for the proposed algorithm is of great importance. The computational burden in the real-time implementation of the algorithm for object tracking system is the motion estimation. For the better performance of tracking and recognition, several advanced algorithms for motion estimation has been employed. Motion detection method such as optical flow algorithm has been applied to estimate the object's motion. But, the computation of the optical flow field for the entire area of interest requires considerable computational complexity. It may be only effective for a static camera. As the camera moves, the system generates a considerable variation between two successive images. In presence of camera motion, Condensation filter shows the robust tracking of object's motion.

Table I shows the average calculation time for processing each frame. Experiments are performed in case 1 and 2.

TABLE I
Results of Calculation Time

Case	Recognition method	Tracking method	Time (ms/frame)
Case 1	PCA	Optical flow	143
	PCA	Condensation	85
Case 2	PCA	Optical flow	351
	PCA	Condensation	92

VI. Conclusion and Further work

In this paper, we have introduced a real-time object tracking-and-recognition scheme using the probabilistic approach, called "*Condensation Principal Component Analysis* (CPCA)" which can be applied to object tracking and recognition. Through experiments, the effectiveness of the proposed algorithm was verified.

Through experiments, I have check the performance of Condensation PCA. In case 1 and 2, the effectiveness of Condensation PCA (condensation filter on temporal PCA) was performed. And, case 1 experiments shows the the importance of temporal information in tracking and recognition task. Its use will enhance the performance of the system. Also, simple structure can make the system operate in real time.

Acknowledgment

This paper was partially supported by HWRS-ERC (Human-Friendly Welfare Robot System Engineering Research Center), Republic of Korea.

References

[1] I. T. Jolliffe, "Principal componet analysis," *Springer-Verlag*, 1986.

[2] H. Attias, "Independent Factor Analysis," *Neural Computation*, vol. 11, no. 4, pp. 803-851, 1999.

[3] P. Comon, "Independent Component Analysis - A New Concept?," *Signal Processing*, vol. 36, pp. 287-314, 1994.

[4] M. A. Kramer, "Nonlinear Principal Component Analysis Using Autoassociative Neural Networks," *Am. Inst. Chemical Eng. Journal*, vol. 32, no. 2, pp. 1-10, 1991.

[5] M. E. Tipping and C. M. Bishop, "Probabilistic principal componet analysis," *Journal of the Royal Statistical Society*, vol. 21, no. 3, pp. 611-622, 1999.

[6] M. Isard and A. Blake, "CONDENSATION-Conditional Density Propagation for Visual Tracking," *International Journal of Computer Vision*, 1998.

[7] K.-C. Lee, J. Ho, M-H. Yang, and D. Kriegman, "Video-Based Face Recognition Using Probabilistic Appearance," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1 8, 2003.

[8] S. Zhou, V. Krueger, and R. Chellappa, "Probabilistic Recognition of Human Faces from Video," *Computer Vision and Understanding*, vol. 91, pp. 214-245, 2003.

[9] V. Kruger and S. Zhou, "Exemplar-based Face Recognition form Video," *Proc. IEEE Int'l Conf. on Automatic Face and Gesture Recognition*,pp. 175-180, 2002.