VQ-Faces – Unsupervised Multi-View Face Recognition by Pairwise Clustering

Bisser Raytchev*
b.raytchev@aist.go.jp
ISI, AIST, Japan

Hiroshi Murase
murase@is.nagoya-u.ac.jp
Nagoya University, Japan

Takahito Kawanishi
kawanisi@eye.brl.ntt.co.jp
NTT CS Labs, Japan

ABSTRACT

In this paper we propose a new, unsupervised approach to multi-view face recognition, which can be formulated mathematically as a problem of partitioning of pairwise proximity data, obtained from multi-view face image sequences. The proposed approach is implemented in two steps: in the first step, the so-called VQ-faces are calculated incrementally as prototype vectors of local areas in image-space, coding for different face-views (i.e. a "view code-book" is generated). In the second step the different face sequences are clustered into identity categories by a combinatorial optimization-based partitioning of the proximity matrix of pairwise relations between the sequences, generated with respect to the VQ-faces. The general approach proposed here has been tested experimentally on a data set of multi-view face sequences gathered over a period of several months in real-world conditions and encouraging results were obtained.

1. INTRODUCTION

Although in recent years face recognition is attracting a lot of attention, the research effort in this area has largely been concentrated on *supervised* methods. *Unsupervised* face recognition (see [1,2] for some earlier work) can be considered important not only from theoretical point of view, but also because of the numerous potential applications which it can find like for online identification in video surveillance systems and man-machine interfaces, for content-based image retrieval in multimedia applications, among many others, when supervised strategies might be either impossible (category information is simply not available and category patterns have to be discovered, or "self-organized" from the input data stream) or impractical (when the manual segmentation/labeling of huge datasets into category groups can be overwhelming and costly).

Another difficult problem in face recognition, is the problem of *multi-view* face recognition – recognizing faces across different views. Achieving multi-view recognition in practical applications is difficult, because different people's faces observed in the same conditions (illumination, view angle, size and so on) look more similar to each other than the same person's face observed in different conditions – in frontal and side view, under extreme illumination conditions, occluded, and so on. Different approaches have been proposed to solve this problem. While initially most of the research work was concentrated predominantly on *static* face images like for example, the modular eigenspaces of [3], elastic graph matching [4], or learning the correspondence transformations between views [5], just to mention a few representative approaches, more recently, the fact that the information contained in *dynamic* video sequences with continuously changing face views can be used advantageously, has started to attract more research effort [6-8].

Here we further expand the dynamic approach by considering an unsupervised alternative to the predominantly supervised algorithms for multi-view face recognition from video sequences. In a previous paper [9], we have proposed *VQ-faces* (explained below) as an unsupervised approach to multi-view face recognition. However, in [9] the partitioning into face clusters was based on exhaustive combinatorial search, rendering it inefficient if a large number of sequences has to be considered. Here, in section 2.2, we propose a new and much more efficient framework for solving this problem, which is now formulated mathematically as a problem of clustering of *proximity data*, obtained from the VQ-faces. We also introduce one novel pairwise-clustering method (section 2.3), able to find the global solution of the combinatorial optimization problem above.

2. MULTI-VIEW RECOGNITION WITH VQ-FACES

The final aim of the algorithms introduced in this paper is to group a set of unlabelled face image sequences into their corresponding identity categories in unsupervised manner, without using any category information provided in advance. Here, first we will briefly describe the main points which determine our clustering strategy. Rather than making any assumptions on a global level about the form of the multi-view face clusters (which is unlikely to be of some well-defined distribution), we make the following assump-

*This research was performed while the first two authors were at NTT CS Laboratories, Atsugi, Japan.

tion about the local distribution of face samples in image space. If face-image space is divided into local areas represented by an *area prototype* (calculated as the centroid of the samples in the area) and all face-samples within a radius R from that prototype; then if a suitable value for R is chosen, the predominant part of the local areas will contain face-samples coming from different sequences, but corresponding to a similar face-view of the same category (same person), plus an insignificant amount of "noise samples", consisting of faces coming from different views of the same category, or similar (to the prototype) views of different category. Based on the above assumption, the *VQ-faces* approach can be implemented by the following two steps:

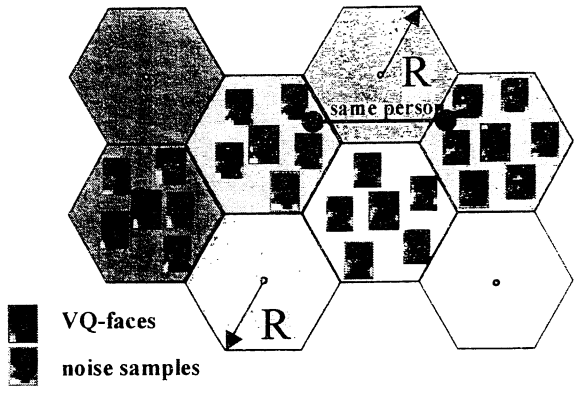

Figure1. Tesselation of face-space results in the generation of a "view-codebook". The VQ-faces are with blue frames and noise samples in red frames.

(1) divide image space into local areas of radius R (see Fig. 1);

(2) use as a temporal constraint the fact that each sequence belongs to a single category only (that is, different people's faces are being tracked separately and don't appear in the same sequence), in order to group the sequences by a combinatorial optimization process operating on the proximity matrix of pairwise relations between the sequences, generated with respect to the prototype areas.

2.1. Tesselation of input image space

As different unlabelled face image sequences become available from the face tracking module (only time-stamps being attached to each sequence at this stage, with no category-specific information provided), the following algorithm is used to divide input image space into polyhedral regions, each region represented by prototype face vectors, which we call *VQ-faces*.

In our algorithm summarized below, x and y are face vectors (face images represented in a vector form), N is the number of currently allocated prototype vectors (VQ-faces), $\zeta^{(m)}$ and $\mu^{(m)}$ stand for the VQ-face vector and the number of face samples in area m respectively, and $dist(x, y)$ calculates the distance between vector-faces x and y. Procedure $remove(y_{\max}, m)$ removes face $y_{\max}$ from area m and puts it into an area k, distance to whose prototype $\zeta^{(k)}$ is minimal compared to all prototypes, distance to which is less than R. If such area doesn't exist, a new area $N := N+1$ is formed with prototype $\zeta^{(N)} := y_{\max}$.

VQ-faces : face-space tessellation algorithm

INITIALIZE

R; $N = 1$; $\zeta^{(1)}$ = first face in first sequence; $\mu^{(1)} = 0$.

WHILE $\exists$ unprocessed image sequences

FOR each face-image vector x in current sequence

IF $\min_m\{dist(\zeta^{(m)}, x)\} > R \qquad (m = 1,\ldots, N)$

THEN

$N := N + 1$; $\zeta^{(N)} := x$; $\mu^{(N)} := 1$;

ELSE

$m := \arg\min_n\{dist(\zeta^{(n)}, x)\}$; $\mu^{(m)} := \mu^{(m)} + 1$;

$$\zeta^{(m)} := \zeta^{(m)} + \frac{x - \zeta^{(m)}}{\mu^{(m)}};$$

WHILE $\exists\ y_{\max} = \arg\max_{y \in m}\{dist(\zeta^{(m)}, y) > R\}$

$remove(y_{\max}, m)$;

$\mu^{(m)} := \mu^{(m)} - 1$;

$$\zeta^{(m)} := \zeta^{(m)} + \frac{\zeta^{(m)} - y}{\mu^{(m)}}.$$

As a result of the algorithm above, input face-image space is divided into N areas, each area being represented by its prototype (VQ-face). Several examples of such VQ-faces, together with some of the sample-faces in their area used to calculate them are shown in Fig. 1. The tessellation algorithm will build a "view-book" (similar to the codebook in the VQ algorithm) from the input sequences, with different prototypes coding for faces of different view angles.

2.2. Generation of pairwise proximities

Now, the temporal-constraint can be used to group the unlabelled L face sequences into categories. From the tessellation obtained with the algorithm in the previous section, all areas (together with their prototypes) which contain samples from a single image sequence only are eliminated, thus the number of areas is reduced from N to M (where $M<N$). The following matrix $\mathbf{A}=\{a_{lm}\}$, is calculated from the remaining M areas:

$$a_{lm} = \sum_i \exp(-\alpha \left\| \zeta^{(m)} - x_{li}^{(m)} \right\|^2) \qquad (1)$$

$$\alpha = -\frac{\ln C}{R^2}; \qquad (2)$$

where C is a constant (e.g. it can be chosen C=0.01, meaning that that much the weights will decay at the boundary for radius R), and $x_{li}^{(m)}$ are all vector-faces of sequence l in area m.

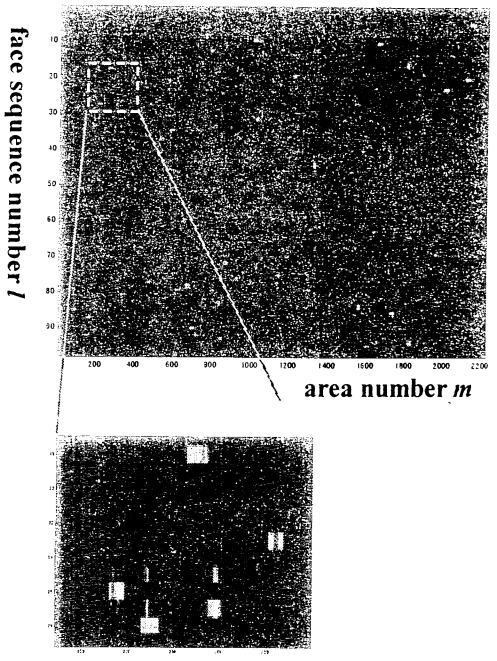

Figure 2. Matrix **A** generated from the data for experiment 1 in section 3.

From matrix **A**, the proximity (dissimilarity) matrix **P** can be calculated as

$$p_{ij} = (\mathbf{A}^{(i)} - \mathbf{A}^{(j)})^T \boldsymbol{\Sigma}^{-1} (\mathbf{A}^{(i)} - \mathbf{A}^{(j)}) \quad (i,j : 1..L) \quad (3)$$

where $\boldsymbol{\Sigma} = L^{-1}\mathbf{A}^T\mathbf{H}\mathbf{A}$, and $\mathbf{H} = \mathbf{I} - L^{-1}\mathbf{1}\mathbf{1}^T$, and using **P**, the available data can be partitioned into clusters by pairwise clustering. Usually first **P** is modified into an *affinity* (or *similarity*) matrix $\mathbf{S}_{L\times L} = \{s_{ij}\}$ defined as

$$s_{ij} = \begin{cases} \exp(-\dfrac{p_{ij}^2}{2\sigma^2}) & \text{if } \exp(-\dfrac{p_{ij}^2}{2\sigma^2}) > r \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

where σ is a free parameter reflecting some reasonable local scale, and values less than a certain small constant r are set to zero. If the face sequences are thought of as nodes in a graph, interacting with each other, the i-th row in **S** will describe the strength of interaction (or *affinity*) between the i-th node (sequence) and all other nodes (sequences), which will always be positive or zero.

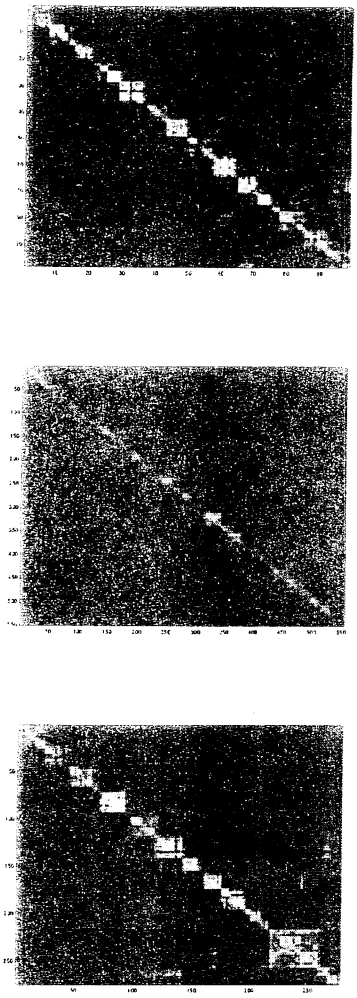

Figure 3. In matrices **W**, each entry represents the weighted similarity (5) between the i-th and j-th face sequences. For visualization purposes only, here the face sequences/nodes have been ordered into their corresponding correct groups, that is nodes belonging to the same cluster have been put next to each other. Of course, such ordering information is not available a priori (neither used by the grouping algorithm), but has to be found by the clustering itself. **(a)** data for experiment I (98 sequences, both frontal and multi-view faces); **(b)** data for experiment II (552 sequences, both frontal and multi-view faces); **(c)** data for experiment III (275 sequences, frontal faces). Details about the experiments are given in section 3.

However, rather than zeroing all interactions between nodes further away from each other than a certain distance, we propose instead of (4) above to use the following transformation on **P** to form matrix **W**, which permits both positive and negative interactions between nodes:

$$w_{ij} = \begin{cases} \exp(-\frac{p_{ij}^2}{2\sigma^2}) - \exp(-\frac{(p_{ij}-2\sigma)^2}{2\sigma^2}) & if \ \ p_{ij} < 2\sigma \\ -1 & if \ \ p_{ij} \geq 2\sigma \end{cases} \tag{5}$$

Several examples of matrix $\mathbf{W}$ (obtained for data used in the experiments described in section 3) can be seen in Fig. 3.

In this way we define two types of interactions between nodes (the face sequences): nodes for which w_{ij} is positive are said to *attract* each other with strength proportional to w_{ij}, and nodes for which w_{ij} is negative are said to *repel* each other with strength proportional to $|w_{ij}|$.

Using repulsion (rather than discarding the negative values in $\mathbf{W}$ as unnecessary information) together with attraction to guide the grouping process permits much fuller exploitation of the structural information hidden in the relation values of the proximity matrices, which is especially important in the case when real-world data with lots of noise have to be processed. As can be seen from the data in Fig. 3, the diagonal blocks, that is the clusters to be found, contain a lot of noise (caused, especially in our case, by factors like the non-uniform ranges of face-view change among sequences, problems due to illumination changes, imperfect preprocessing and so on), they are not clearly and unambiguously separated from each other and the off-diagonal blocks, but the nuances and gradation of the "noise" can be utilized both to distinguish between structures which otherwise might be considered homogeneous (leading to undersegmentation) or to find similarities between structures which otherwise might be taken apart (leading to oversegmentation).

2.3. Clustering with Attraction and Repulsion

We define a symmetric Boolean matrix $\mathbf{X} = \{x_{ij}\}$ of size $N \times N$, called an *indicator matrix*, for which $x_{ij} = 1$ if nodes (i, j) belong to the same partition (cluster), and $x_{ij} = 0$ otherwise (note that x_{ii} should always be 1). The i-th row of $\mathbf{X}$ is the *indicator vector* $\mathbf{x}_i^t$ (i=1,...,N), whose non-zero entries' indexes show which other nodes are grouped in the same cluster as node i. According to our definition, a permutation of $\mathbf{X}$ in which all nodes belonging to the same cluster are ordered next to each other would consist of diagonal blocks of all '1's, and off-diagonal blocks of all '0's. The aim of the algorithm then will be to determine $\mathbf{X}$ by optimizing a certain global criterion which is a suitable function of the matrices $\mathbf{X}$ (the sought partition) and $\mathbf{W}$ (the input data on which the attraction/repulsion transformation (5) has been applied). We consider the minimization of the following criterion (cost) function:

$$\begin{aligned} C(\mathbf{X}) &= \min\{-\sum_{i=1}^{N}\sum_{j=1}^{N} x_{ij} w_{ij} + \sum_{i=1}^{N}\sum_{j=1}^{N} (1 - x_{ij}) w_{ij}\} \\ &= \min\{\sum_{i=1}^{N}\sum_{j=1}^{N} (1 - 2x_{ij}) w_{ij}\} \end{aligned} \tag{6}$$

The criterion function in (6) maximizes the within-cluster attraction and the between-cluster repulsion at the same time. In order to solve the combinatorial optimization problem (6) we use a stochastic optimization strategy in which state space is stochastically sampled by a Markov process, and new solutions are accepted or rejected according to the Metropolis algorithm with the following transition probabilities

$$P(\mathbf{X}^{old} \rightarrow \mathbf{X}^{new}) = \begin{cases} 1 & if \ \Delta C(\mathbf{X}) \leq 0 \\ \exp(-\Delta C(\mathbf{X})/T) & otherwise \end{cases} \tag{7}$$

where $\Delta C(\mathbf{X}) \equiv C(\mathbf{X}^{new}) - C(\mathbf{X}^{old})$. State changes corresponding to decreases in the cost function are always accepted, while in addition, to avoid local minima, state changes corresponding to an increased cost are accepted with an exponentially weighted probability, determined by the temperature parameter T. The temperature is gradually reduced during the stochastic search process according to a predetermined cooling schedule [10].

Another important detail is to determine the way in which the configuration $\mathbf{X}^{old}$ has to be perturbed in order to obtain the new configuration $\mathbf{X}^{new}$, or the definition of a "move" between two configurations. Here we would like a move between two configurations to implement the idea of merges/splits of nodes to/from clusters. For this purpose, we initially set $\mathbf{X}$ to be equal to the unit matrix $\mathbf{I}$, that is the N available nodes form N singleton clusters, and each move randomly selects some x_{ij} $(i \neq j)$ whose binary value is flipped as

$$x_{ij}^{new} := \begin{cases} 1 & if \ \ x_{ij}^{old} = 0 \\ 0 & if \ \ x_{ij}^{old} = 1 \end{cases}; \tag{8}$$

which can be interpreted in the following way:

(1) If x_{ij} is set to '1', this means that the j-th node will be added (merged) to the cluster who has node i among its members, after being removed (split) from its current cluster. In addition to that, the following update of $\mathbf{X}$ is necessary:

$$x_{kj}^{new} := 1 \ \ for \ \forall k \ for \ which \ x_{ik}^{old} = 1; \tag{9}$$

$$x_{kj}^{new} := 0 \ \ for \ \forall k \ for \ which \ x_{jk}^{old} = 1 \ (and \ k \neq j); \tag{10}$$

The updates (9) and (10) performed on $\mathbf{X}$ complete the split of the j-th node from its previous cluster and its merge to the new cluster.

(2) If x_{ij} is set to '0', this means that the j-th node will be removed (split) from its current cluster (which has also node i among its members). Also the following update of $\mathbf{X}$

$$x_{kj}^{new} := 0 \quad for\ \forall k\ for\ which\ x_{ik}^{old} = 1\ (and\ k \neq j); \quad (11)$$

is necessary in order to complete the split of the j-th node from its previous cluster to form a singleton (which can be merged again to another cluster at some future move).

Our clustering algorithm does not need to know the number of identity categories/clusters K in advance (although it can be easily modified to find exactly K clusters, if such information is available, by using an $N \times K$ matrix $\mathbf{X}$, and slightly modifying expressions (8)-(11) above), and the grouping process is guided by both attraction and repulsion.

3. EXPERIMENTS

In order to evaluate and compare the performance of the proposed method, several experiments were conducted using a data set of about 600 face image sequences obtained over a period of several months from 33 different subjects. Illumination conditions were very demanding and varied significantly with the time of the day during which the samples were taken. The video sequences' length varied between 30-300 frames, depending on the speed at which the subjects walked in front of the camera, in the range between slow walking with occasional stops, and running. For each one of 17 of the subjects were gathered between 10 and 50 sequences, while less than 3 sequences were available for the remaining 16 subjects ("rare visitors").

3.1. Data sets

We prepared two different data sets to be used in the following experiments:

(a) **Data set A**: in this data set, the subjects were just walking forward toward the camera. As a result, this data set contained predominantly frontal faces, with only a few side-view faces included at the end of the sequences, when the subjects passed beside the camera.

(b) **Data set B**: in this data set, the subjects were told to look to the left and right, up and down, as they moved towards the camera. Both frontal and side-view faces were represented.

Samples with and without glasses were included for all subjects (except for the "rare visitors"), and hairstyles changed with time. Resolution of the original images was 320x240 pixels, and 18x22 pixels for the size-normalized face-only images. Because of the large volume of data, we were unable to manually inspect all the face sequences output from the face tracking module, but from the few inspected ones it was obvious that the dataset on which the system had to perform contained many instances of noisy data, in the form of erroneous face croppings and misalignments, large variations in illumination with face shadows, and so on, as would be expected in a real-world situation.

3.2. Evaluation of the clustering

We propose the following formula to calculate the self-organization (recognition) rate ρ of the final clusters:

$$\rho = (1.0 - \frac{E_{AB} + E_O}{N}) \times 100\%, \quad (12)$$

where N is the total number of sequences to be grouped, E_{AB} is the number of sequences which are mistakenly grouped into cluster for certain category A, although in reality they come from category B, and E_O is the number of samples gathered in clusters in which no single category occupies more than 50% of the nodes inside them. While the meaning of E_{AB} above is obvious, analysis of many different partitions obtained for different data sets revealed the following interpretation for E_O. A small E_O (in comparison to E_{AB}) usually signals the relatively harmless presence of some small clusters of outliers, which have happened to be very near to each other, while a large value of E_O most probably is a sign of bad partitioning (undersegmentation), and most likely occurs when the clustering algorithm has been unable to discriminate between the members belonging to several different identity categories, effectively mixing them together in some huge cluster(s).

It should be noted, however, that although the recognition rate proposed above can evaluate the error of misclassification contained in the final partitioning precisely, it provides only partial information about the structural quality of the obtained partitioning. That is, it can detect (from the value of E_O above) when a certain clustering leads to an undersegmentation of the data, but it can be fooled by a clustering leading to oversegmentation, which might produce very high ρ, although the resulting partitioning might be practically useless (as the data is split into too many clusters). When the true partitioning is known (as in the case of the experiments reported below), a more objective judgement about the partitioning quality can be obtained by the combined information provided by (a) the number of the *true* partitions P (that is the number of identity categories); (b) the number of the *obtained* partitions K; (c) the number of detected *singleton-outliers* S; and (d) the recognition rate ρ, above. All these have been provided in the experimental results summarized in Table 1 below.

3.3. Experiments

The following 3 experiments were conducted.

Experiment I This experiment used a data set containing face sequences randomly selected from both data sets A and B, that is both frontal and multi-view sequences were in-

cluded. The samples included a total of 98 sequences, 7 sequences being available for each of 14 different subjects. The purpose here is to test on a data set which is (a) easy to be visualized, as the samples are distributed evenly among the categories; (b) relatively clean, as the small size of the data sets permitted the results of the face extractor to be inspected and "cleaned" from obviously wrong face croppings; (c) relatively small, which might render this experiment relatively easier (although definitely not trivial), but useful, in conjunction with the other two experiments below, to show how the performance of the algorithm changes with change in the size of the input data set. Additionally, there are many practical cases when algorithms have to perform on small data sets, possibly because of insufficient data.

Experiment II This experiment used all available data, that is all data in sets A and B put together. Both frontal and side-view faces were represented in this relatively large data set, which included 552 face sequences from 33 subjects.

Experiment III Only data from data set A (frontal or near-frontal faces only) were used in this experiment. The purpose was to isolate the "multi-view" factor, and report results obtained for unsupervised frontal face recognition. The data set included 275 face sequences from 17 subjects (excluding the "rare visitors" as for those we didn't have frontal-face-only data).

Each of the three experiments above were conducted using the algorithm described in section 3.3, and the recognition results are summarized in Table 1.

experiment	P	K(S)	E_{AB}	E_O	ρ (%)
I(98)	14	17(27)	4	0	95.9
II(552)	33	58(64)	8	52	89.1
III(275)	17	30(47)	6	0	97.8

Table 1. Experimental results

As can be seen from the experimental results in Table 1, for all experiments an acceptable structural quality of the obtained partitions was achieved, the latter being judged by comparing the number of obtained clusters to the number of the original identity clusters (if those are not very different, it is unlikely to have an oversegmentation) and small values for E_O (which means there was no undersegmentation either). A certain difference between the number of obtained clusters and the number of the original identity clusters is acceptable and inevitable, having in mind that the illumination conditions were very demanding and the data was taken over a long period of time, while the simple distance measure between faces in the tessellation algorithm was not specifically chosen to be invariant under such conditions.

4. CONCLUSION

In this paper we have proposed a novel method for unsupervised face recognition from video sequences of time-varying face images obtained over an extended period of time in real-world conditions. The proposed method provides the following important advantages: (a) it allows all stages of the resulting system to be completely automated, avoiding the need for manual segmentation and labeling of the input stream; (b) it can be easily implemented as a pairwise clustering algorithm which is simple and robust to noise in the data; (c) both frontal and side view faces can be recognized by the method.

We reported results from several face recognition test experiments using both frontal and side-view face sequences obtained under demanding real-world conditions. The results seem encouraging, having in mind the difficulty of the task, the bottleneck of the preprocessor output and the suboptimal distance measures used. It is expected that the proposed method can find application in video surveillance systems, human-computer interfaces, for content-based information retrieval from video databases of multi-view objects (like faces), and generally in situations where manual segmentation and labeling of the input video stream into categories are considered impractical or impossible.

Acknowledgment

The authors are grateful to Dr. K. Ishii and Dr. N. Sugamura of NTT CS Labs for their help and encouragement.

REFERENCES

[1] H. Ando, S. Suzuki and T. Fujita, Unsupervised visual learning of three-dimensional objects using a modular network architecture, *Neural Networks*, vol. 12, pp.1037-1053, 1999.

[2] J. J. Weng and W. S. Hwang, Toward Automation of Learning: The State Self-Organization Problem for a Face Recognizer, *Proc. 3rd Int. Conf. on Automatic Face and Gesture Recognition*, pp.384-389, 1998.

[3] B. Moghaddam and A. Pentland, Face recognition using view-based and modular eigenspaces, In *Automatic Systems for the Identification and Inspection of Humans*, SPIE, vol. 2277, 1994.

[4] L. Wiskott, J. M. Fellous, N. Kruger and C. von der Malsburg, Face recognition by elastic bunch graph matching, *IEEE Trans. on PAMI*, 19 (7), pp. 775-779, 1997.

[5] D. Beymer and T. Poggio, Face recognition from one example view, In *Proc. ICCV*, pp. 500-507, 1995

[6] Y.Li, S. Gong and H. Liddell, Recognizing the dynamics of faces across multiple views, in *BMVC*, p. 242-251, 2000.

[7] S. Satoh, Comparative Evaluation of Face Sequence Matching for Content-based Video Access, In *Proc. 4th Int. Conf. on Automatic Face and Gesture Recognition*, pp.163-168, 2000.

[8] V. Krueger and S. Zhou, Exemplar-Based Face Recognition from Video, In *Proc. ECCV 2002, LNCS 2353*, pp. 732-746, 2002.

[9] B.Raytchev and H.Murase, VQ-faces – Unsupervised Face Recognition from Image Sequences, In *Proc. ICIP 2002*, II p. 809-812, 2002.

[10] P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, D. Reidel, Hingham, MA, 1987.