Proceedings of ICIAP'99 (Int'l Conf. on Image Analysis and Processing), pp. 266–271, 1999.

An Efficient Implementation and Evaluation of Robust Face Sequence Matching

Shin'ichi Satoh and Norio Katayama
NACSIS
3-29-1 Otsuka, Bunkyo-ku, Tokyo 112-8640, Japan
{satoh, katayama}@rd.nacsis.ac.jp

Abstract

This paper presents a robust and efficient matching method for face sequences obtained from videos. Face information is quite important especially for news programs, dramas, and movies. Face sequence matching for such videos enables many multimedia applications including content-based face retrieval, automated face annotation, automated video authoring, etc. However, face sequences in videos are subject to variation in lighting condition, pose, face expression, etc., which cause difficulty in face matching. These problems are tackled to achieve robust face sequence matching applicable to real video domains, and its efficient implementation is presented. The paper proves the proposed method achieves good performance in actual video domains. In addition, by the combination with the high-dimensional index structure, the algorithm achieves practical computational time, as well as scalability against increase of the number of faces.

1 Introduction

Face information is quite important in videos, especially in news programs, dramas, and movies. By extracting face sequences from such videos, multimedia applications having content-based access to face information might become realizable. However, we may at least need to be able to tell which face sequences may correspond to the same person in order to realize multimedia applications such as content-based face retrieval, face annotation, video authoring, etc., which are desired for the next generation video processing.

Face images in videos are subject to variation in lighting condition, pose, facial expression, etc., which cause difficulty in face matching. There have been many research efforts of face matching, including face feature-based methods [2], image-based methods [1, 6, 10], combination of feature- and image-based methods [11], etc. Most of them are, however, concentrated only on still images rather than faces in videos. Moreover, most of them can handle very limited variation, i.e., developed for high-quality experimental images taken under strictly controlled lighting condition with fixed pose and facial expression. In addition, to make a face sequence matching method useful in real multimedia applications, the method should be scalable to the size of video corpus, namely, the number of faces to be matched.

In this paper, a robust and efficient face sequence matching method is explored which enables content-based access of face information for multimedia applications. The paper proves the proposed method achieves good performance in face sequence matching for actual video domains. In addition, by the combination with the high-dimensional index structure, the matching algorithm achieves practical computational time, as well as high scalability against increase of the number of faces.

2 Preparation

2.1 Face Sequence Extraction

As basic data elements for face sequence matching, face sequences are extracted from given videos. We used an equivalent method described in [9]. In this method, face detection is applied first to every frame within a certain interval of frames. In our experiment, we apply the face detector at intervals of 10 frames. The system uses the neural network-based face detector [8], which detects mostly frontal faces of various sizes and at various locations. The face detector can also detect eyes. To ensure that the faces are frontal and close-up, we use only faces in which eyes are successfully detected.

Once a face is detected, the system extracts the skin color model. We assume that human skin color has the Gaussian distribution in (R, G, B) space. The system calculates the mean and covariance matrix of $[R\ G\ B]^T$ vectors of pixels in the detected face region, and uses them as the skin color model in such a way that a pixel having smaller Mahalanobis distance than a certain threshold is a skin color pixel. This model is used to extract skin candidate pixels in the subsequent frames. Based on skin candidate pixels, skin regions are composed using binary image processing methods. The overlap between each of these regions and each of the face regions of the previous frame is evaluated to decide whether one of the skin candidate regions is the succeeding face region. Face region tracking is continued until no succeeding face region is found, or until a scene change is encountered.

The face detector incorporates the linear function-based lighting condition compensation and histogram equalization, which compensate lighting condition variation of face images [8]. In addition, even for a face sequence having large variation in pose and facial expression, since the system applies the detector to every frame within a certain interval, some faces composing the face sequence are expected to be detected successfully even though the others are not detected due to the variation. Thus the system can detect such face sequences with large variation. The method also can extract multiple face sequences appear simulatneously. Finally, the system outputs face sequences S_i, composing N detected faces $F_{i,k}$ where $k = 1, \ldots, N$.

2.2 Face Sequence Distance

The face sequence matching problem can be defined as a test whether two face sequences are the same person. Assume that we have a set of face sequences. Using a proper face sequence matching method, we can select face sequences among this set which are the same person to the given face sequence.

We show that a proper definition of face sequence distance $d(S_i, S_j)$ can be used for face sequence matching. By using a threshold θ, we can infer that face sequences S_i and S_j are matched if $d(S_i, S_j) < \theta$. In addition, assume that we have a set of face sequences U_s. Face sequence selection with a given face sequence S_0 can be defined as follows;

$$\sigma(U_s; S_0) \quad \stackrel{\mathrm{def}}{=} \quad \{S | S \in U_s \wedge d(S_0, S) < \theta\}.$$

To evaluate face sequence distance, we can see whether face sequence selection result $\sigma(U_s; S_0)$ is close enough to ideal selection result $\sigma^*(U_s; S_0)$; i.e., a set of all face sequences which are actually the same person to the sequence S_0. ROC curves should be helpful for evaluation.

2.3 Face Similarity Evaluation

To calculate face sequence distance, the system may need to evaluate similarity of faces. We employ the eigenface-based method [10] to evaluate face similarity. Although many other face similarity research efforts have been reported and the eigenface-based method does not necessarily achieve the best performance, we choose this method because it is less restrictive to input face images (it does not require face features detection, e.g., eyes, nose, mouth corners, etc.). Each of detected faces is normalized into a 64×64 image using the eye positions, applied lighting condition compensation and histogram equalization [8], then converted to a point in the 16-dimensional eigenface space. Face similarity can be evaluated as the face distance, i.e., the Euclidean distance between two corresponding points in the eigenface space. Since eigenface method is quite sensitive to variation in pose, facial expression, etc., we need to incorporate some techniques to suppress effect of these variation. In the following sections, $f_{i,k}$ is assumed to be a converted vector in the eigenface space of the face $F_{i,k}$.

3 Face Sequence Matching

In this section, face sequence distance between two face sequences is defined to realize face sequence matching. As described before, face images in videos are subject to variation in pose, facial expression, etc., yet at the same time, they are of course subject to variation by personal identity. Ideal face sequence matching should enhance variation by personal identity, while suppress other variation. We propose two types of face sequence distance: face sequence distance using the best frontal view selection method and using the closest pair method. The former is more efficient, but less robust than the latter. Meanwhile, the latter is more robust to variation but much less efficient than the former. We describe overviews of these two methods, and evaluate them in terms of precision and recall in face sequence retrieval. Acceleration of face sequence distance calculation using the closest pair method will be given in the next section as a remedy for inefficiency.

3.1 The Best Frontal View Selection Method

Since each face sequence is composed of several faces, variation not by personal identity can be compensated more or less by selecting and using a face from each face sequence which has less variation. Based on this idea, we developed the best frontal view selection from a face sequence (this method is used in the system, which automatically associates faces and names in news videos, described in [9]).

To choose the best frontal view of face, a face skin region clustering method is first applied. For each detected face, cheek regions, which are sure to have the skin color, are located by using the eye locations. Using the cheek regions as initial samples, region growing in the (R, G, B, x, y) space is applied to obtain the face skin region. We assume the Gaussian distribution in (R, G, B, x, y) space; (R, G, B) contributes by making the region have skin color, and (x, y) contributes by keeping the region almost circular. Then, the center of gravity (x_f, y_f) of the face skin region is calculated. Let the locations of the right and left eyes of the face be $(x_r, y_r), (x_l, y_l)$, respectively. We assume that the most frontal face has the smallest difference between x_f and

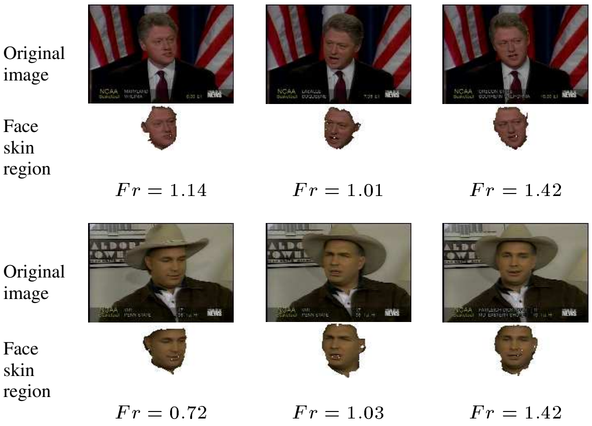


Figure 1. Frontal Face Selection

$(x_l + x_r)/2$, and the smallest difference between y_l and y_r. To evaluate these conditions, we calculate the frontal factor Fr for every detected face;

$$\begin{aligned} w_f &= \frac{5}{3}(x_l - x_r) \\ Fr &= (1 - \frac{|2x_f - x_r - x_l|}{w_f}) + \frac{1}{2}(1 - \frac{|y_l - y_r|}{w_f}) \end{aligned}$$

where w_f is the normalized face region size. The factor for an ideal frontal face is 1.5. The system chooses the face having the largest Fr as the most frontal face of the face sequence. Figure 1 shows example faces, extracted face skin regions, and frontal factors.

Among faces $F_{i,k}$ composing a face sequence S_i, the face having the largest frontal factor is defined as $F_i^{frontal}$. Similarly, let $f_i^{frontal}$ be the corresponding vector in the eigenface space. Then the face distance using the best frontal view selection is defined as follows:

$$d_{frontal}(S_i, S_j) \stackrel{\text{def}}{=} |f_i^{frontal} - f_j^{frontal}|.$$

3.2 Closest Pair Method

As shown in Figure 1, the best frontal view face might be successfully selected. However, this method has apparent problems; The method cannot distinguish "nodded" faces. Moreover, it cannot handle variation in facial expression.

Meanwhile, in comparing two face sequences, it is not necessary to use the best frontal view face. Instead, a pair of faces having the same variation can be used, whichever variation they have. Therefore, we propose new face sequence distance based on distance between a closest pair of faces. This method depends on the presumption that when two face sequences correspond to the same person, the closest pair of them corresponds to faces having the similar pose, facial expression, etc. The new face sequence is defined as follows:

$$d_{closest}(S_i, S_j) \stackrel{\text{def}}{=} \min_{k,l}(|f_{i,k} - f_{j,l}|). \qquad (1)$$

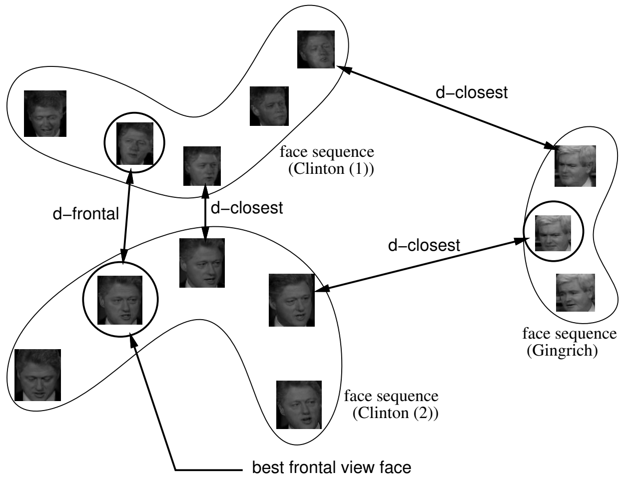


Figure 2. Face Sequence Distance

This method is supposed to work fine, because in most cases

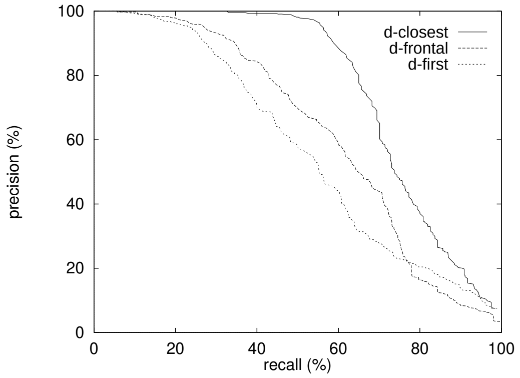
- A face sequence surely corresponds to a particular person (no variation by personal identity),
- Each face sequence has sufficient variation in pose, facial expression, etc., not by personal identity, so that there can be a closest pair having the same variation.

Figure 2 shows difference between distance by the best frontal view face method and distance by the closest pair method.

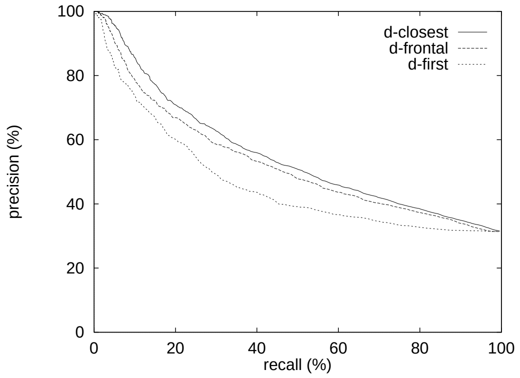
This problem can be regarded as colored nearest neighbor(NN)-search. In this analogy, face vectors $f_{i,k}$ are data, and they are assumed to be colored based on face sequences S_i they belong to; e.g., $f_{i,k}$ has color S_i. Assume that a query set $\{f_{q,k}\}$ is given (which has the color S_q). The problem now is to decide the color S_a of the point $f_{a,l}$ which is the nearest neighbor of any point in the query set.

3.3 Comparison

We evaluated two proposed face sequence matching methods in terms of precision and recall in face sequence retrieval. Two sets of videos were used for the experiments: 5 hours CNN Headline News, and 1.5 hours of drama video produced by Fuji Television Network, Japan. From the news video, the system extracted 556 face sequences composing 8134 detected faces. Therefore, each face sequence contains 15 faces on average. On the other hand, from the drama video, the system extracted 673 face sequences composing 3426 detected faces. In this case, each face sequence contains 5 faces on average. For a training set for eigenface calculation, we used the best frontal view faces each of which was selected from each face sequence. To evaluate precision



(a) results for the news video



(b) results for the drama video

The graphs represent precision and recall with changing the threshold θ. The face sequence distance using the closest pair method, the best frontal view method, and the first face of the sequence, are labeled "d-closest," "d-frontal," and "d-first," respectively.

Figure 3. Evaluation of Face Sequence Distance Functions

and recall, we manually named extracted face sequences. 288 face sequences and 507 face sequences were named from the news and drama respectively. We used these two set of face sequences S_i where $i = 1, \ldots, N$; $N = 288$ for the news and $N = 507$ for the drama.

We regarded pairs whose distance fell below a certain threshold ($d(S_i, S_j) < \theta$) as expected identical pairs. Each face sequence distance was evaluated by comparing plots of precision and recall while varying threshold θ. We also defined much simpler face sequence distance for comparison which measures the first face of each face sequence:

$$d_{first}(S_i, S_j) = |f_{i,1} - f_{j,1}|.$$

Figure 3 shows precision and recall graphs for each definition of face sequence distance. Figure 3(a), (b) are obtained using the news and drama video, respectively. The graphs clearly depict that the closest pair method achieves better precision while retaining better recall, thus it achieves the best performance in the three methods. We also see that the best frontal view method is still much better than the first face method. The best frontal view method and the first face method are equivalent in computational cost in face sequence retrieval, because they require only one comparison between a pair of faces in comparing two sequences. However, the closest pair method is computationally costly because it requires minimum operation comparing many combinations of face pairs (Eq.(1)).

We can say that the performance difference between the closest pair method and the others is much prominent in Figure 3(a) than in Figure 3(b). The main reason for this would be each face sequence is composed of much more faces in the news video than in the drama video as described in the first part of this subsection. News videos tend to include longer duration face shots in speech scenes, interviews, or in anchorperson scenes. On the other hand, in drama videos, scenes in which two persons, A and B, are talking, are typically used. In such a scene, very short shots showing A and B are shown alternatively. Thus very short face sequences were obtained from drama videos which make the closest pair method less effective. Special care should be taken for drama videos in the future research.

4 Efficient Implementation of the Closest Pair Method

4.1 Cost of the Closest Pair Method

As shown in the previous section, the closest pair method is advantageous in matching performance. However, its computational cost is larger than the best frontal view method. Comparison of CPU and I/O cost for the two methods clarifies this problem. The methods are implemented using linear scan method, i.e., feature vectors of face images in the file are scaned linearly. We measured average costs to find the most similar sequence to a given query sequence. Programs are written in C++ and costs are measured on a Sun Microsystems workstation, Ultra-60 (CPU: UltraSPARC-II 360 MHz, main memory: 512 Mbytes, OS: Solaris 2.6). The results are shown in Figure 4. The figure also shows the results using the SR-tree, which will be described later.

The CPU time of the closest pair method is much larger than that of the best frontal view method. This is mostly due to the number of comparison of feature vectors. Since the best frontal view selection method requires one face image per sequence, the number of comparison of feature vectors is equal to the number of face sequences. On the other hand, the closest pair method requires all face images contained in face sequences. Therefore, the number of comparison of feature vectors is equal to [the number of face images in the

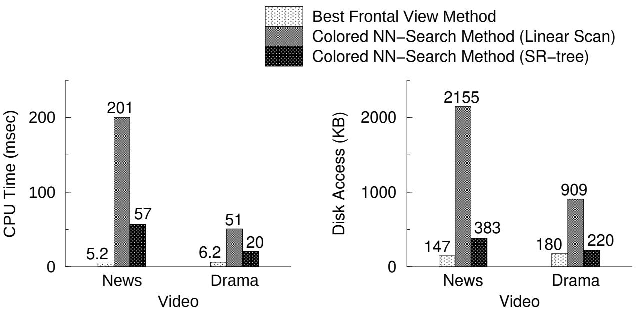


Figure 4. Comparison of Matching Cost

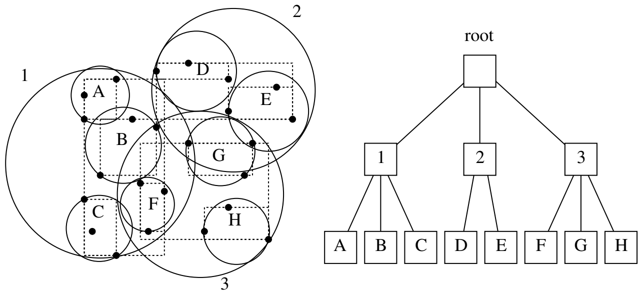


Figure 5. Structure of the SR-tree

face sequences] × [the number of face images in a query sequence].

As shown by these results, the linear scan implementation of the closest pair method is intractable. Therefore, we propose a better implementation for the closest pair method taking advantage of a high-dimensional index structure, the SR-tree [5].

4.2 The SR-tree (Sphere/Rectangle-Tree)

The SR-tree [5] is an index structure proposed for accelerating NN-search in high-dimensional space. It is a secondary-memory data structure and designed for indexing a large-scale data set. Its fundamental structure is derived from the R-tree [3]. Thus, its tree structure corresponds to the nested hierarchy of the space decomposition permitting overlap as shown in Figure 5.

The significant property of the SR-tree resides in its region shape. The shape is determined by the intersection of a bounding sphere and a bounding rectangle. In high dimensional space, a bounding rectangle and a bounding sphere are complementary to each other. A bounding rectangle is more suitable for minimizing volume, while a bounding sphere is better in minimizing diameter. Thus the SR-tree employs the intersection of them to reduce both volume and diameter of regions at the same time.

With this property, the SR-tree generates a region hierarchy whose regions have both short diameter and small volume. Thus similar feature vectors are clustered into the same region, and the NN-search performance is enhanced by reducing the search space to a small number of regions.

4.3 Algorithm of Closest Pair Method Using the SR-tree

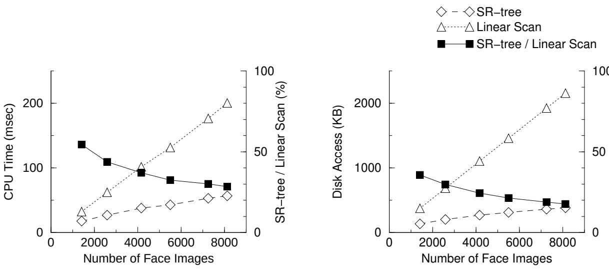
The SR-tree is originally proposed for non-colored NN-search. In order to achieve an efficient implementation of the closest pair method, we developed a new algorithm for colored NN-search using the SR-tree.

Colored NN-search is realized based on the non-colored NN-search algorithm [4, 7], with adaptation to the SR-tree structure. In the same way as the non-colored NN-search, the colored NN-search selects candidates of nearest neighbors by visiting nodes and leaves of the SR-tree in the ascending order of the distance from query points. To achieve colored NN-search, we made two modifications to the non-colored NN-search. Firstly, the distance calculation was modified. Since a query is given as a set of points, the distance is determined by the closest query point. Secondly, the way of candidate selection was modified. Since the result of a query is a set of colors, no two candidates can have the same color; if two candidates have the same color, the farther one is removed.

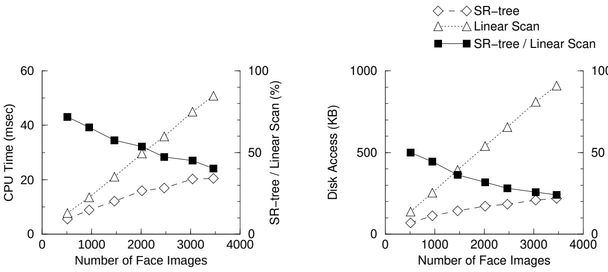
The search process consists of two steps. Firstly, it visits the leaf closest to the query points to find initial candidates of nearest neighbors. Secondly, it visits such leaves and nodes that are closer to the query points than the candidates. Every time it visits a leaf or node, the candidates are reselected and the search terminates when there is no more leaf or node that is closer to the query points than the candidates. The final candidates are the search result. Traversal of nodes and leaves will be done efficiently during the search taking advantage of the SR-tree's hierarchical structure.

4.4 Performance Evaluation on Acceleration and Scalability

To evaluate acceleration achieved by the proposed algorithm, we conducted the same experiment with the linear scan described in Section 4.1. The results are shown in Figure 4. The proposed algorithm reduces disk access significantly. The disk access of the proposed algorithm is 18% of that of the linear scan for the news video, and 24% for the drama video. These amounts are even comparable to the disk access of the best frontal view method. This proves that the proposed algorithm succeeds in reducing the search space with the SR-tree's hierarchical index structure. As for the CPU time, less improvement is obtained. This is mainly because the tree-traversal algorithm of the SR-tree



(a) News Video



(b) Drama Video

Figure 6. Scalability of the Acceleration of the SR-tree

needs more computation than the linear scan. Nevertheless, the CPU time decreases 72% and 60% for the news and the drama video, respectively. These results demonstrate the effectiveness of the algorithm in accelerating the closest pair method.

The most important advantage of the hierarchical index structure, such as the SR-tree, is scalability. The size of the search space is expected to be logarithmic to that of the entire data space. To evaluate scalability, performances of the linear scan and the proposed algorithm are measured with varying the size of the dataset. We composed subsets of the news and drama video by choosing some of the face sequences at random. The results are shown in Figure 6. The horizontal axis indicates the number of face images, i.e., the number of feature vectors, and the vertical axis indicates the CPU time and the amount of disk access. As Figure 6 shows, performance of the proposed algorithm is almost logarithmic to the number of face images and the cost ratio of the proposed algorithm to the linear scan is getting smaller as the number of face images increases. This proves the scalability of our implementation using the SR-tree.

5 Conclusions

As a key technology to achieve content-based face information handling for multimedia applications, robust and efficient face sequence matching methods are proposed. The proposed methods are robust against variation: variation in lighting condition, pose, face expression, and they are evaluated in terms of precision and recall using actual news and drama videos. The evaluation proved that the proposed closest pair method achieved quite good performance in face sequence matching, however, it was computationally costly. As a remedy for this, we incorporated the new algorithm using the SR-tree into the implementation of the method, and showed that improved performance was achieved with the experiments using real news and drama video data.

Acknowledgement

This material is based upon work supported in part by the Ministry of Education, Science, Sports and Culture of Japan, as the Grant-in-Aid for Creative Basic Research, No. 09NP1401, "Research on Multimedia Mediation Mechanism for Realization of Human-oriented Information Environments," and the Grant-in-Aid for Encouragement of Young Scientists, No. 10750296.

References

[1] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on PAMI*, 19(7):711–720, 1997.

[2] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Trans. on PAMI*, 15(10):1042–1052, 1993.

[3] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *Proc. ACM SIGMOD*, pages 47–57, 1984.

[4] G. Hjaltason and H. Samet. Ranking in spatial databases. In *4th International Symposium, SSD'95*, pages 83–95, 1995.

[5] N. Katayama and S. Satoh. The SR-tree: an index structure for high-dimensional nearest neighbor queries. In *Proc. of ACM SIGMOD*, pages 369-380, 1997.

[6] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. on PAMI*, 19(7):696–710, 1997.

[7] N. Roussopoulos, S. Kelley, and F. Vincent. Nearest neighbor queries. In *Proc. ACM SIGMOD*, pages 71–79, 1995.

[8] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. on PAMI*, 20(1):23–38, 1998.

[9] S. Satoh, Y. Nakamura, and T. Kanade. Name-It: Naming and detecting faces in news videos. *IEEE MultiMedia*, 6(1):22–35, January-March (Spring) 1999.

[10] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

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