1-1994

Identifying faces using multiple retrievals

Jian Kang WU

Arcot Desai NARASIMHALU
Singapore Management University, desai@smu.edu.sg

DOI: https://doi.org/10.1109/93.311656

Published in IEEE MultiMedia, 1994 January, Volume 1, Issue 2, Pages 27-38
https://doi.org/10.1109/93.311656

Identifying Faces Using Multiple Retrievals

Jian Kang Wu and Arcot Desai Narasimhalu
National University of Singapore

During a police investigation, officers often have to sort through hundreds of photographs to identify a suspect. To aid this task, we at the Institute of Systems Science developed and implemented a flexible database system that can retrieve faces using personal information, fuzzy and free-text descriptors, and classification trees.

A mug shot database encounters many challenges in real-world use as a crime-fighting tool. Witnesses might not have had enough time to observe the facial features of a suspect, or the light might have been bad. Moreover, the suspect could have altered his appearance after the event, by shaving off a beard, for example. The set of reference photographs at the police station could be several years old, adding the problem of age-related changes the database doesn't include, like wrinkles. With these problems in mind, we developed the Computer-Aided Facial Image Inference and Retrieval System (CAFIIR), a database management system (DBMS) applied to mug shot identification. Designed for use by an investigating officer, it incorporates expertise obtained from experienced crime investigators.

Sometimes a witness observes some special trait of the suspect, such as a limp on the left side, that could aid in the identification of the suspect. Therefore, a mug shot application should also include general information about physical traits. Witnesses usually provide very subjective descriptions of suspects, which calls for fuzzy searching. For example, a witness might describe the suspect as having a long, thin nose and deepset eyes. We developed the CAFIIR system using only frontal photographs—the standard available to us—which do not help in processing descriptions such as "deepset eyes."

The above considerations translated into the following set of requirements:

1. The system should support at least the following facial features—hair, chin outline, nose, eyes, eyebrows, and lips.
2. The system should support retrieval based on standard attributes of a person, such as name, address, identification number, and occupation; a class hierarchy for fast browsing; fuzzy descriptors of facial features; and free-text descriptors of facial and other features.
3. Given that most descriptions would be incomplete, we also decided the system should look for both exact and "close enough" matches, employing similarity retrieval techniques to determine the latter.
4. Since descriptors are subjective, we built a context model that uses user models to normalize the input descriptions.
5. While the system captured some of the expertise of investigating officers, we couldn't model everything. For example, different investigating officers might assign different weights to facial features in the same situation. Hence, the retrieval process still requires an expert. Therefore, we implemented direct control and manipulation of the feature weights.
6. Since we expected the descriptions to be both inaccurate and incomplete from the beginning, we provided for visual relevance feedback for iteration through the retrieval process.

Before we describe the CAFIIR system proper, let's clarify our terms. For our uses, traditional database attributes are information such as the name and address of a person whose photo is in the database. Image feature space refers to the metrics space of the extracted facial feature measures (also called image features), such as height, breadth, or the measuring vector. Fuzzy feature space is a set of linguistic tokens that defines the descriptive values of a facial feature. We use a mapping function to translate a point in image feature space into the fuzzy feature space and vice versa.

System design

Our method of facial identification is comprehensive. It differs from other mug shot systems in the following ways.

- We carry out classification on individual features, instead of on the whole face.
- We use a minimum number of landmarks (17)

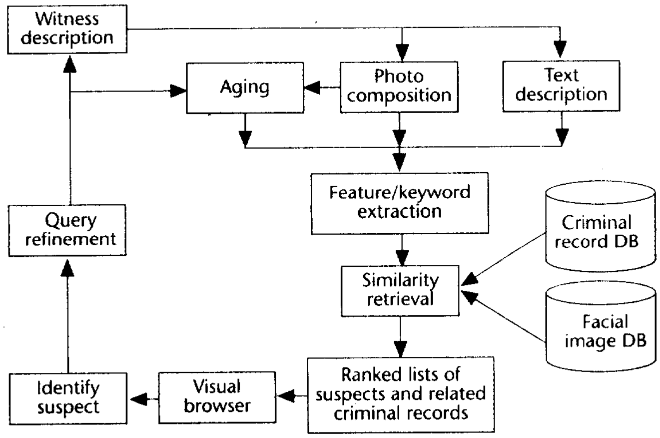


Figure 1. This flow chart shows how CAFIIR uses witness descriptions to perform a query.

for feature extraction. This is very important, because the database must store a large number of faces, and the registration time for each face has to be reasonably small. Once robust, automated, feature-extraction techniques develop, this will not matter.

- We allow a user to assign weights directly at the individual feature level.

- While most other systems can find a face only if it is in the database, CAFIIR lets you compose a nonexistent face as the input for search.

- We provide for a number of retrieval techniques, including classification-based browsing. For example, CAFIIR can retrieve a face based on descriptions of nonfacial characteristics.

- CAFIIR also supports visual relevance feedback, a feature most other systems do not support.

- Finally, our base data is still-image features. We map image features to fuzzy space during runtime, which allows you to build in a context model that can normalize a query to the image feature space.

We adopted a modular design approach. Figure 1 gives the overall process flow of the CAFIIR system, and Figure 2 shows the architecture of the system. We've implemented all the modules you see, except the aging module, which we are still working on. Let's look at the system in detail.

Registration

In the first phase, a photograph is scanned into the common workspace managed by the common workspace management module, shown in Figure 2. The human operator registers selected landmarks on the scanned image. Then the system extracts the facial features, based on these landmarks. The system calculates metrics such as length and width for each feature, then enhances the feature edges. What remains of the face after feature extraction is the facial outline. All these

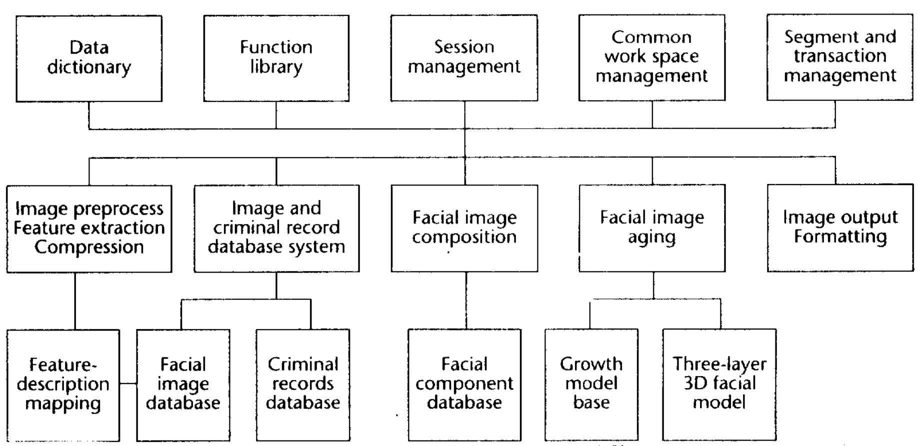


Figure 2. A block diagram outlines the CAFIIR architecture.

functions are stored in the function library shown in Figure 2 and invoked by the image preprocessing, feature extraction, and compression modules.

The criminal records database is traditional. We extended the registration module to handle free-text descriptions as well as images. The extracted facial features are stored in the facial image database. The feature description mapping module handles the mapping between the image feature space and the fuzzy feature space. It also normalizes the fuzzy descriptors based on a user model.

The image and criminal record database system is the key module of the system. It relies heavily on the data dictionary for the locations of different features and data input it needs.

Once a sufficient number of photographs have been registered, a principal component analysis reduces the dimensionality of the feature set. Then a clustering algorithm groups them into classes. A neural network-based classification creates indexing trees.

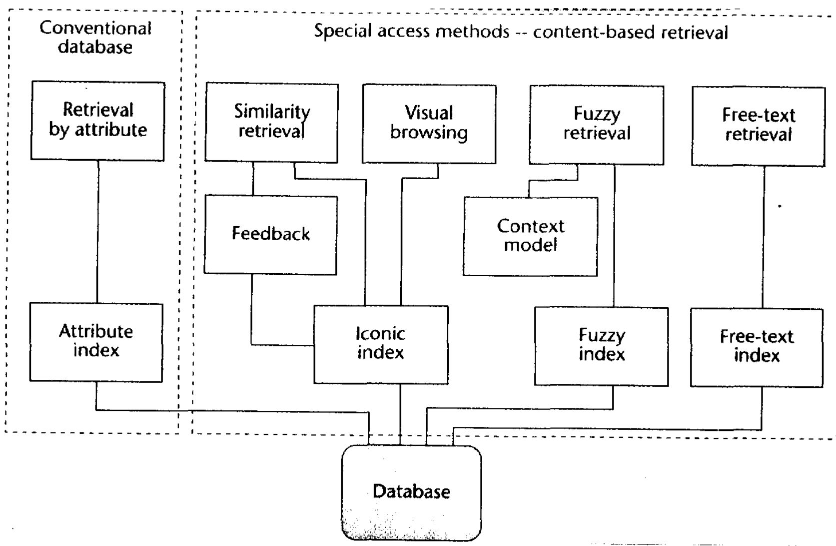


Figure 3. CAFIIR's database retrieval section contains both a conventional DBMS and a special area designed to perform content-based retrieval.

Retrieval

Retrieving a facial photo can start at different points, as shown in Figure 3. Similarity retrieval uses the facial image composition module to prepare an image query. Once the aging module is complete, the user will be able to apply it to an image query in the same way as facial image composition. The photo composition module lets a user compose the face of a suspect using sample facial features in the facial component database.

The facial image aging module will simulate aging of a photograph before submitting it for a query. This is very important in cases where the photograph of a suspect in the CAFIIR database was taken several years ago. This has also proved to be the most difficult module to construct. To build the aging module, we are using the growth model base, which holds growth information for different cross-sections of people. The three-layer 3D facial model will map the landmarks from a photograph onto the 3D model, carry out the aging on the 3D model, then map the results back onto the photograph.

The session, segment, and transaction management modules of the system perform the basic housekeeping functions. The image output formatting module invokes the proper user interface for different types of retrieval. Figure 4 shows the interface for facial retrieval using traditional database attributes.

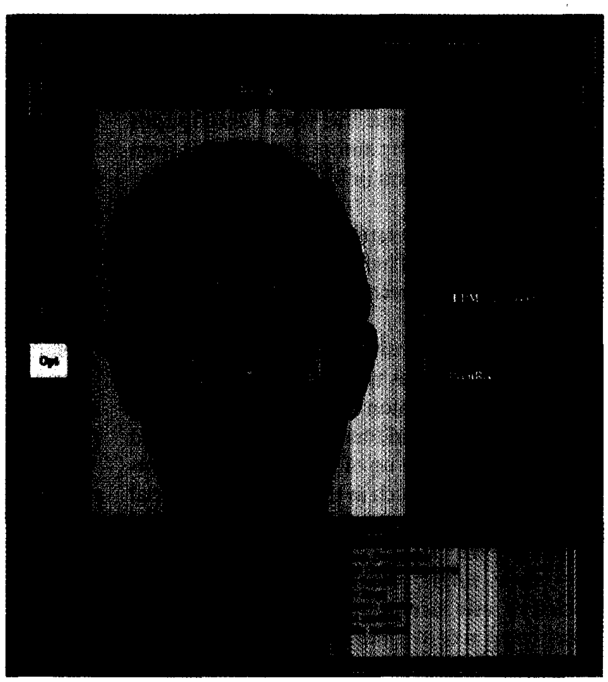


Figure 4. The database management interface panel displays the face record to be inserted or retrieved, stored in the work space. The user can edit the text and process the image using the workspace entry.

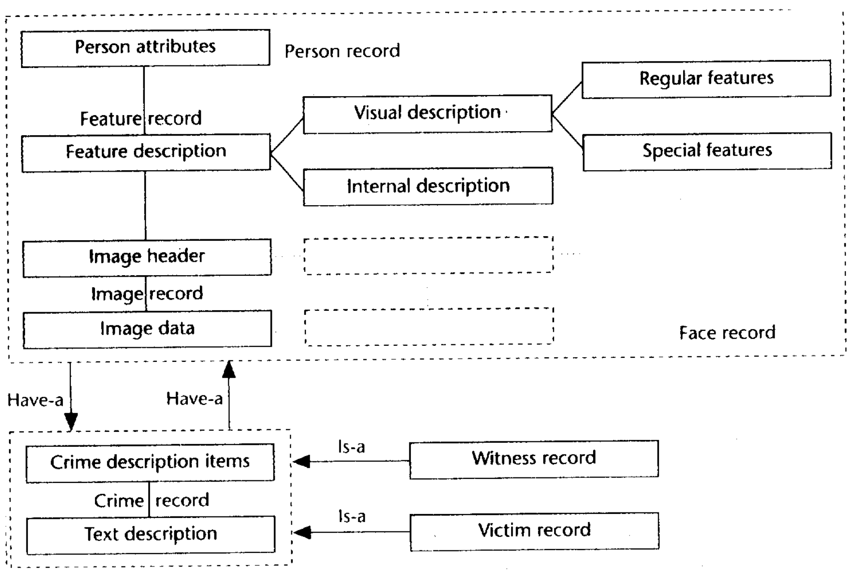


Figure 5. The data model combines information from witness and victim records to formulate the crime record. The crime records have multi-to-multi links with face records stored in the facial image database.

Data model

The CAFIIR data model has a face record for each person who has a facial image stored in the system. A face record consists of three types of records:

1. Person record, which stores the identification number, name, date of birth, address, and other traditional database information.

2. Feature record, consisting of two structured feature descriptions: the internal description obtained from image feature extraction, usually numerical, and the visual descriptions generated either by automatic mapping from internal description or by manual text input.

3. Image record, consisting of image data and the image header. (There can be several image records in a face record.)

A crime record consists of a routine description of the crime, such as time or type, and a more detailed text description of the crime. Of course, one crime record could have several witness and victim records made up of items to identify witnesses and victims, such as their names and social security numbers.

One person can commit several crimes, and one crime might involve several criminals. Therefore, one face record could correspond to many crime records, and vice versa. In Figure 5, we use Have-A to indicate the relationship between face and crime records.

Previous research suggests that the object-oriented model is the optimal model for multimedia database systems.[1] We extended the object-oriented model for CAFIIR to include vector, array, and text paragraph as elementary data types. In CAFIIR, a vector represents an internal feature measure, an unsigned character array represents a digital image, and a paragraph of text describes a crime or special features of a face. Although you could argue that elementary data types such as char, integer, real, and string suffice in this case, we use image, vector, and text paragraph as elementary data types because they show strong integrity. CAFIIR can map one image to another by image processing functions or to descriptions by feature extraction functions. The system can perform an indexing process on text paragraphs. These mapping and processing functions occur in terms of whole image, feature vector, or text paragraph, treated as discrete entities at the database level.

When using the object-oriented model, you should define a set of functions on each object class. We extended the model to include functions that perform mapping across object classes. For example, the feature extraction function first generates a set of internal feature measures from a facial image, then maps them to visual descriptions.

Feature extraction

The image preprocessing and feature extraction module provides various input and processing functions on the face and crime records, including image digitization, enhancement, and segmentation; feature extraction; image pyramid structure (hierarchical multiresolution) construction; compression; text input and editing; viewing feature and description; image display; facial image classification; and facial classification tables production.

The first key issue for feature-based image retrieval is feature extraction and mapping from the internal feature measures to the visual description of facial images. The difficulties of facial feature extraction and mapping arise because

- We don't know how human beings recognize faces, and what features we use to distinguish between faces. Therefore, cognitive science cannot yet provide practical hints for computer recognition of facial images.

- Although good work in facial feature extraction exists (see "Facial identification sources" sidebar), none can provide a robust solution. For example, the global principal component analysis[2] is affected easily by the background, while it is nearly impossible to optimally fit the deformed template[3] to a blurred facial image.

- Subjective descriptions of facial characteristics are fuzzy. The mapping between subjective descriptions and feature measures should allow image access from descriptions. This mapping requirement makes feature extraction even more complicated.

The feature extraction method in CAFIIR is application-oriented (for criminal identification), although you could extend it to other applications. By working with domain experts, we defined a visual description form for facial images. It contains six descriptive aspects: facial outline, hair, eyes, eyebrows, nose, and mouth. Each aspect contains items and possible descriptive values (such as large, medium, or small) for those items. To be consistent with this visual description form, CAFIIR performs image feature extraction on whole images as well as facial components to generate features on these six aspects.

Because of user requirements, the facial images stored in the system are all frontal images. To make full use of the information, we investigated several possible feature extraction methods and tried to integrate them to produce more reliable measures. Currently, we use face landmarks and selected principal component analysis coefficients of face images to generate feature measures. To make sure the first few principal components accurately reflect the main visual features of the facial aspects, three preprocessing steps take place before feature extraction:

1. A normalization process reduces the effects of image variations by normalizing grayscale, orientation, position, and size of the face inside the image. The normalization of face position, orientation, and scale is based on the anatomical observation that the position of landmarks Sella Turcica, a bony structure above the ears and behind the eyes, is invariant. The system geometrically tranforms all images so that these two landmarks occupy a standard position inside the images.

2. Defining the region of interest with the help of landmarks eliminates the effect of nonrelevant content. The principal component analysis is then performed on the facial feature aspect only.

3. Selectively enhancing the information essential to the facial features (for example, eye contours) ensures the presence of important elements in the large principal components.

We also examined other model-based feature analysis methods, such as using a deformed template for facial outline approximation. We will further test and integrate these methods, but face landmarks must suffice in the current implementation.

Iconic index of facial images

We cannot implement visual, fuzzy, and similarity queries using conventional indexing techniques such as B-trees and inverted files because conventional indexing techniques are based on individual keys, which are definite and do not provide any visual views of the database. For the purpose of defining visual, fuzzy, and similarity indexing, we extended the concept of indexing using abstraction and classification.[4] We used a spatial self-organization neural network model to generate an iconic index tree. You can readily apply this indexing technique to fuzzy indexing on multivariate fuzzy membership functions, as you will see in the section "Spatial self-organization neural networks."

Facial identification sources

Facial identification has been a topic of interest for some time. Some efforts, such as Pentland's work, treat the face as a whole.

A. Pentland, "Eigenface for Recognition," *J. Cognitive Neuroscience*, Vol. 3, No. 1, Winter 1991, pp. 59-70.

Bach, Paul, and Jain, among others, used landmarks to extract facial features.

J. Bach, S. Paul, and R. Jain, "A Visual Information Management System for The Interactive Retrieval of Faces," *IEEE Trans. on Knowledge and Data Eng.*, Vol. 5, No. 4, 1993, pp. 619-628.

Others, such as Ralescu and Iwamoto, used a large number of landmarks to accurately represent facial features, transform image features into fuzzy measures, and carry out their inferencing using fuzzy measures.

A. Ralescu and H. Iwamoto, "Reading Faces: A Fuzzy Logic Approach to Representation, Recognition, and Description of Facial Expressions, *Proc. 13th Int'l Conf. on Artificial Intelligence*, Chambery, France, 1993.

For a complete survey of related work, see Samal and Iyengar.

A. Samal and P.A. Iyengar, "Automatic Recognition and Analysis of Human Faces and Facial Expressions: A Survey," *Pattern Recognition*, Vol. 25, No. 1, Jan. 1992, pp. 65-77.

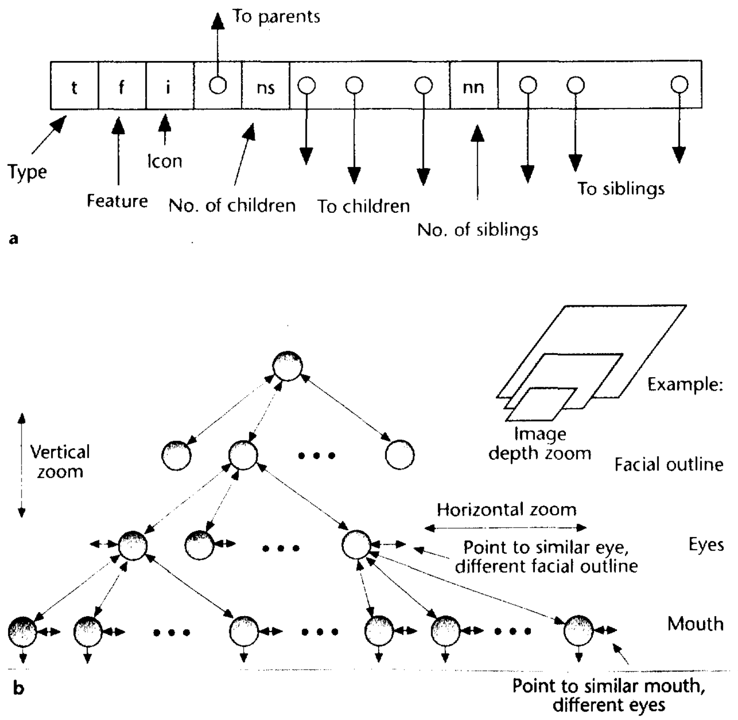


Figure 6. The iconic feature index tree follows a hierarchical self-organization neural network model. (a) The node structure locates the records within the facial image index tree. (b) The index tree facilitates visual browsing, similarity retrieval, and fuzzy descriptive queries using fuzzy concepts like "round face" and "large eyes."

Indexing by abstraction and classification

We extended the conventional indexing technique in three ways. First, we allowed an attribute to be an abstract data type. It can be a vector (in our case, a feature vector), a multidimensional array (feature vectors, image data, or image sequences), or a pointer to a data structure (in our data model, to the feature record).

As a result, the grouping criteria and grouping functions are by no means primitive logical expressions. Therefore, we generalized the grouping criteria to be a similarity measure. For a data structure like the face record, the grouping criteria can be a well-defined set of functions that extracts and measures the similarities between images. We needed automatic grouping to balance the population between nodes. Because the attributes are now multidimensional, we also needed spatial self-organization of the child nodes of an abstract data item node.

The third generalization allows different key attributes for the different levels of the index tree. For example, at the top level of the facial image index tree, the key attribute is the facial outline aspect, at the second it is hair aspect, and so on.

The algorithm to build an iconic index tree can be either top-down (split) or bottom-up (merge). Figure 6 shows the structure of the iconic index tree. A top-down algorithm works in the following four steps.

1. Select a feature aspect. Cluster the facial images into m classes by an extended self-organization neural network. The algorithm uses the weight vectors in the neural network as reference vectors of the selected feature aspect and constructs an iconic image for each node.

2. Repeat the first step until each node has at most m descendants.

3. Create the nodes and allocate fields among them.

4. Build horizontal links for the feature aspects already used for clustering at the levels above. For example, if the current level performs clustering according to eye features, and previous levels clustered according to face outline and mouth feature measures, the algorithm builds two kinds of links to link nodes in the same level having similar eye feature measures but differing in face outline and mouth only. This link provides users with multidimensional views of the facial images with respect to the feature aspect.

As defined previously, an intermediate node of the index tree is an abstract facial image, not an actual facial image in the database. It represents a set of facial images indicated by its descendant leaf nodes. Therefore, the system should construct an abstract facial image icon that is the abstraction of all those actual facial images.

An immediate way of constructing the abstract icon image is to use the mean facial image. Because at each level of the index tree the clustering took place according to one feature aspect, the descendants of a node are similar only with respect to this feature aspect. Other features can vary widely. Thus, averaging all the children images often creates a distorted icon image.

To avoid distortion, you can adopt one of two alternative methods. You could find a facial image closest to the template and use it as the icon image. Alternatively, you could average over the distorted region with respect to the feature aspects used for clustering so far, taking other regions of the icon image from the image closest to the template. We chose the first method for its simplicity.

Spatial self-organization neural networks

The index tree of CAFIIR needs category consistency between feature measures and descriptions. Feature measures are internal and serve as a criterion for index tree construction. Therefore, the similarity measures here perform a very important role. The neural network model LEP (Learning based on Experiences and Perspectives) suggests combining multifeature perspectives to achieve reliable learning.[5] We developed a spatial self-organization neural network based on LEP for index tree construction.

The network has two layers. The input layer receives input feature vectors, and the output units are arranged as a 2D array. Suppose there are M input units and N output units in the network. Each input unit connects to every output unit with a certain synaptic weight $\{w_{mn}, m = 1, 2, \ldots, M; n = 1, 2, \ldots, N\}$. For an output unit n, CAFIIR stores a template vector $\{p_{mn}, i = 1, 2, \ldots, M\}$ and a weighting vector for input feature vectors to define the relative importance of their elements $\{r_{mn}, m = 1, 2, \ldots, M\}$. The system will match the weighting and template vectors against the input vectors during learning. Let $\mathbf{x} = (x_1, x_2, \ldots, x_M)^T$ be the M-dimensional real input-feature vector presented to the input array at time $t = 1, 2, 3, \ldots$.

When presenting an input-feature vector to the network in the index creation phase, the output units begin to compete with each other through inhibitory links among themselves, within a certain neighborhood window. The system selects unit c as winner based on both a correlation and minimum distance basis:

$$a_c = \min_n a_n = \frac{\mathrm{dis}(\mathbf{x}, \mathbf{p}_n)}{\mathrm{cr}^k(\mathbf{x}, \mathbf{w}_n)}$$

$$\mathrm{dis}(\mathbf{x}, \mathbf{p}_n) = \left[\sum_m (x_m r_{mn} - p_{mn})\right]^{1/2}$$

$$\mathrm{cr}(\mathbf{x}, \mathbf{w}_n) = \frac{\mathrm{abs}\left(\sum_m x_m w_{mn}\right)}{\left[\sum_m x_m^2 \sum_m w_{mn}^2\right]^{1/2}} \quad (1)$$

where k is the parameter that adjusts the effect of normalized correlation to the whole similarity measure. For a detailed discussion of Equation 1 and the LEP neural network model, see Chapter 5 of Wu.[5]

As a result of competitive learning with a decreasing neighborhood window, the weight vectors (templates) tend to approximate the probability density function of the input vectors in a spatially ordered fashion. Figure 7 shows the self-organized map of facial images with the feature aspect "hair." The hair thickness descends from the upper left to the lower right.

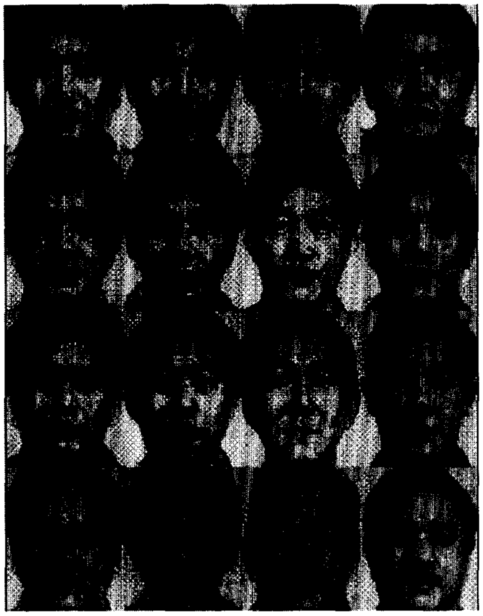


Figure 7. Facial image clusters spatially arranged according to the hair feature vector show good consistency with hair thickness: From the upper left to the lower right, the hair thickness is in descending order. An icon image represents each cluster. Because of the limited number of facial images in the experiment, the icon images show degradation in parts other than hair.

Implementation

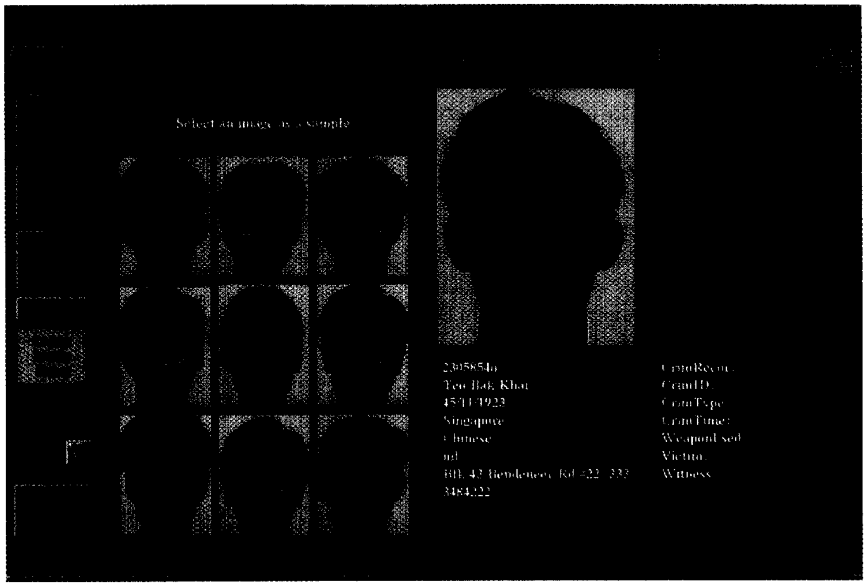
We implemented the CAFIIR system in C using X Windows and Motif interfaces. It can run on most Unix workstations. The conventional database shown in Figure 3 was implemented using a network model of DBMS. Free text and other retrieval engines were developed at the Institute of Systems Science.

Browsing and similarity retrieval

As we saw in the index tree construction process, the system uses similarity measures extensively on various feature aspects. The index tree traversal is therefore based on similarity measures, and similarity retrieval is its basic retrieval operation. Imagine the index tree as a decision tree: When presented with a sample facial image, similarity retrieval occurs in the same way as pattern classification happens using a decision tree. Retrieval follows the tree down to the leaf nodes. At each level, similarity measures determine the decision.

Using distance as the similarity measure, the index tree selects a node in the next level if $d(\mathbf{x}, \mathbf{t}^i) = \min_j d(\mathbf{x}, \mathbf{t}^j)$, where $\mathbf{x}$ is sample image and $\mathbf{t}^j$ is the template of the jth node. At the leaf node

Figure 8. This similarity retrieval example matched the sample image with the stored photo of the same man, despite a three-month gap in the images. The sample image is displayed on the right display panel. The nine most similar among the retrieved images appear in the left display panel. The user can place any one of them in the middle panel to compare with the sample image. The person and crime records associated with the similar image appear on the two text panels beneath it.



level, all leaf nodes similar to the sample image will be selected.

Figure 8 shows a typical similarity retrieval example. The similarity retrieval found the image of the same person, even though the sample image was taken three months later and contained noticeable differences, such as longer hair. The other eight retrieved images shown in the left display panel appear in order from most to least similar.

To gain flexibility, we created a parameter panel (bottom right of Figure 8) that provides two functions. The first adjusts the relative weights of six feature aspects, then reactivates the query. For example, if you expect someone to change his hair often, you can put a small weight on the hair feature aspect. The other function is query feedback. The user can choose one or more images from the retrieval results that most resemble the desired image, then activate the feedback process. The feedback function can then follow the user's input step by step to narrow the search until the correct image results.

The user can perform a visual query of the facial image database by visually browsing through the database via the interactive index tree traversal. The system presents the user with the root of the index tree by displaying the icons of its descendants. At a node of the index tree, the user chooses to browse up, down, left, or right. Going up employs a pointer to its parent node, while moving down involves selecting a specific descendant icon. The system considers the selected icon to be the current node and displays its children.

Horizontal links in the index tree provide more freedom for visual browsing. Imagine that, as you browse down the tree, you find one image very close to what you seek, except that the eyes should be a little bigger. However, you've already passed the eye selection level. In this case, with horizontal browsing, you can just select the feature aspect "eye," and the system will display images that differ only in eye size.

Zooming in and out allows viewing images at different scales and resolutions at the leaf-node level. Nonleaf nodes lack multiresolution icon image capabilities.

Fuzzy retrieval of facial images

Fuzzy retrieval of images is a common phenomenon in human memory. In conversation, you might say, "Oh, I recall the person you described." The problem occurs when we have thousands of facial images in a database and want to retrieve the faces with a rounded chin, big eyes, and thick hair. Human descriptions are neither exact nor objective. After seeing a person's face, five people would have five different descriptions of its features. The words used for descriptions are also fuzzy. To describe a chin, a witness might say, "It is rounded, but seems oval as well."

Fuzzy retrieval in CAFIIR consists of two parts: query preprocessing and processing. The preprocessing tries to recover fuzzy membership functions from user-defined fuzzy queries. The query processing then searches the database for the best

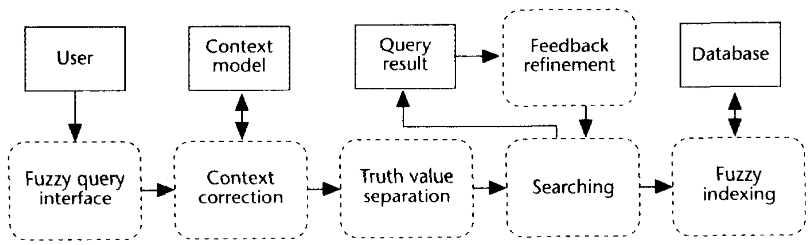


Figure 9. During fuzzy retrieval, preprocessing translates the fuzzy descriptions given by the user and corrects them with a context model. Query processing then locates the best match in the database for the described attributes.

matches based on the fuzzy similarity measures. A diagram of fuzzy query processing appears in Figure 9.

Fuzzy query definitions are subjective. That is, different people could have different perceptions about the same object. For example, an eye considered big from a Japanese point of view could be just of normal size to someone from India. To deal with this context-sensitive query definition, we used the context model to remove user dependency. The context model is a set of curves representing the user's viewpoint. When a user logs in, CAFIIR automatically loads an appropriate context model. The user can also select a context model when someone else forms a fuzzy query. The system considers fuzzy queries to be user-independent, once corrected by the context model.

In the fuzzy space, coordinates represent fuzzy subsets. The coordinate value represents the membership value. When the user defines fuzzy queries, the system combines the truth value of the definition and the membership function value for fuzzy subsets. Before going to similarity calculation, the system performs a process to extract truth value from the fuzzy definition.

The difficulty of fuzzy query processing lies not only in its fuzziness, but also in the incompleteness of the fuzzy query. When the user defines her query, she just specifies what she knows and leaves unknown terms blank. Therefore, blank terms are not zero, although sometimes zeros are used to fill those blanks. To deal with this incompleteness, let us now examine feature and fuzzy space.

Feature space and fuzzy space

After feature extraction, we have feature vectors written as $\mathbf{x}^i = (x_1^i, x_2^i, \ldots, x_{M_i}^i)^T$, where i stands for the ith feature aspect and M_i is the dimension of the ith feature vector, typically 16.

To compensate for the imprecision and vagueness of feature descriptions in facial images, we designed a number of fuzzy descriptions for each feature aspect. For example, we have nine fuzzy subsets conceptually representing chin types: tapered, oblong, short oval, rounded, long tapered, long oblong, short oblong, short rounded, and long rounded. We defined these fuzzy sets over the multidimensional universe $\mathbf{x}^0 = (x_1^0, x_2^0, \ldots, x_{M0}^0)$. Here the chin is the 0th feature aspect. The membership function for fuzzy set B_j^i, where i denotes the feature aspect and j denotes the fuzzy subset for a feature aspect, takes the following form:

$$m_{B_j^i}(\mathbf{x}) = e^{-(\mathbf{x}-u_j^i)^T \Sigma_j^i (\mathbf{x}-u_j^i)} \qquad (2)$$

where u is the central point of the membership function in the multidimensional feature space. There is a linguistic meaning for the fuzzy subset "approximately u". Σ is the covariance matrix of all data points falling into the fuzzy subset.

If we could convert a fuzzy description to the feature vector when processing a fuzzy query, we could invoke a similarity retrieval technique for direct query processing. Unfortunately, we can't. Fuzzy descriptions are fuzzy and incomplete, while feature vectors are multidimensional. With very limited information from these fuzzy and incomplete descriptions, we cannot localize a point in the high-dimensional feature space to represent the defined fuzzy query. The only other option is to convert from feature space to fuzzy space, a process called fuzzification.

After fuzzification, we are in fuzzy space. In the multidimensional fuzzy space, a fuzzy query definition and feature description of an image are points. Using a fuzzy vector to represent a point in fuzzy space, the system can now compute the similarity between a query definition and image data so that similar images can be retrieved.

Unfortunately, since fuzzy space is not orthogonal, ordinary correlation and distance measures do not apply. Previous work on fuzzy similarity measures[6] and fuzzy retrieval[7] does not provide a solution. We proposed our own fuzzy similarity measure, which is the distance between fuzzy query vector Q_j, $j = 1, 2, \ldots, q$ and fuzzy image vector B_j, $j = 1, 2, \ldots, q$, where Q_j, B_j are fuzzy subsets in the same fuzzy space:

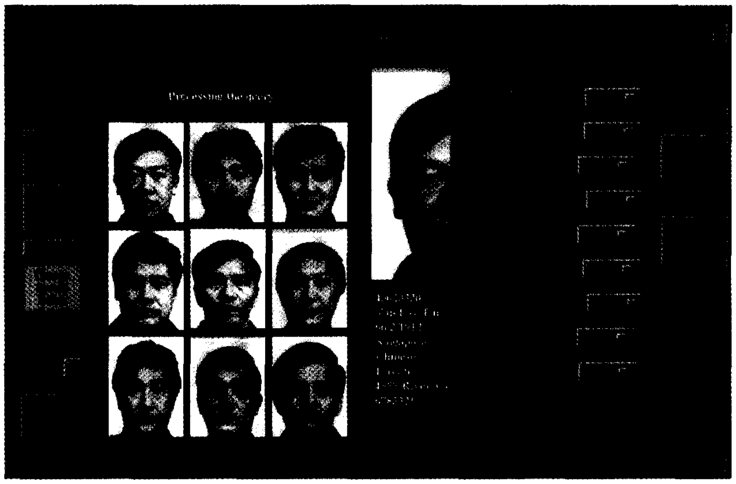


Figure 10. For fuzzy retrieval of images, the user assigns truth values, or measures of certainty, to the described features. This example searched for images with either a round chin (60 percent certainty) or a short round chin (40 percent), and normal hair (80 percent) or thin hair (60 percent).

$$\mathrm{dis}(Q,B) = \sum_j \left|m_{Q_j}(x) - m_{B_j}(x)\right| \sum_k -\mathrm{cor}(Q_j,B_k) \left|m_{Q_k}(x) - m_{B_k}(x)\right| \quad \text{if } Q_j \neq 0(3) \tag{3}$$

The correlation between the two fuzzy axes is $\mathrm{cor}(Q_j, B_k)$, which equals $\mathrm{card}(Q_j \cap B_k)$ divided by $\mathrm{card}(Q_j \cup B_k)$. The cardinality of a fuzzy set, card(Q), is

$$card(Q) = \int m_Q(x)dx$$

or

$$card(Q) = \sum_x m_Q(x) \tag{4}$$

Intuitively, if two coordinates are not orthogonal, the distance between a pair of points in the space will become shorter.

Fuzzy query processing produces a set of images that best fit the query definition, arranged in descending order with respect to their fuzzy similarity measures. You can use feedback here to refine the query. Figure 10 shows the fuzzy retrieval of images with "round, maybe short round chin" and "normal, maybe thin hair," arranged in order of hair thickness.

Free-text retrieval

Quite often, people are identified by special features, such as moles or scars. Special facial features are very difficult to describe in a structured way, and witnesses are usually not familiar with the terminologies or codes that police use to describe them. Our system lets the user describe special facial features in their own way, using one or more sentences.

To index the special features and the crime record based on the text description, we adopted the free-text retrieval technique developed at the Institute of Systems Science.[8] CAFIIR checks every word in the text against a stop word list and eliminates the word if it appears on that list. The stop word list consists of commonly used words such as "the" and "a." Notice here that words such as "on," "in," and "near" are essential to represent the location of special features, and therefore cannot be included in the stop word list. The system then removes the word variants with a stemming algorithm, reducing the words "face," "facial," and "facing" to the word face, for example. This reduces the total number of distinct terms in the index and increases the effectiveness of retrieval, because similar words generally have similar meanings identifiable by a single stemmed word. The system indexes stemmed words using an inverted file structure.

In the retrieval phase, the user submits a query in a free format, such as a sentence, a short paragraph, or a list of keywords and/or phrases. The system then applies the same processing used for indexing to the defined query. After an initial search, the system presents to the user a few face or crime records that best match the query. At this stage, the user can modify the query and submit it again, or just select a few facial images or crime records. The system repeats the feedback process until the user is satisfied with the results.

Feedback for query refinement

Through the feedback function, the user can select one or more images that most resemble the desired one from the query result. The feedback function then refines the query using information from the selected images. Our method assumes that users will select feedback images in order of similarity, based on the most similar features among the selected images. Once N images are selected, CAFIIR computes the feature vectors for feedback according to the following procedure:

1. With a predefined threshold, find similar feature aspects among N images.

2. For the rest of the feature aspects, check for similarity among $N - 1$, $N - 2$ selected images. If similarity is not present, use the first image as a representative of the feature aspect.

3. Find the center of the selected images with respect to those similar aspects.

4. Perform a similarity search using the computed feature vectors.

Compound query

In many cases, users want to retrieve images similar in several aspects, such as visual feature measures and text description, in one compound query. CAFIIR performs the compound query using the following equation:

$$S(\mathbf{t},\mathbf{p}) = \prod_i s_{\text{exact}}(\mathbf{t}_i,\mathbf{p}_i) \sum_j s_{\text{sim}}(\mathbf{t}_j,\mathbf{p}_i)^{w_j} \quad (5)$$

where s_{exact} is the exact match between template **t** and the stored pattern **p**. It takes the value of either 0 or 1. If there is any mismatch between these feature aspects, the system will reject the stored pattern. The similarity measure between feature aspects, s_{sim}, does not require an exact match. The overall similarity measure is the weighted summation of similarity measures of those individual feature aspects. We are currently working on combining selected retrieval methods to form a combined query.

Project status

We have implemented all the modules of the CAFIIR system except the aging module. The present system works on faces scanned in from black-and-white photographs, and we've tested it using a few hundred photos.

CAFIIR still suffers from the lack of access to a large number of good-quality photographs. We will have to refine the classification and the clustering using traditional and neural network approaches when we do obtain a large photograph collection.

Besides a large number of photographs, an effective system should have a large set of photographs that are similar to each other. We could not lay our hands on such a collection, but we circumvented this problem by composing several similar faces using individual facial features stored in the facial component database. This allowed us to fine-tune our similarity retrieval algorithms.

The lack of a large collection of faces also affects the browsing module. We do not have enough samples across all categories of the classification tree. As a result, our present implementation has only three levels. We plan to increase the levels to six or more when we obtain a larger collection of photos.

One major problem was feature extraction. Our group spent a significant amount of time developing automatic feature extraction techniques. We modified this approach to incorporate a minimally interactive feature extraction method. We needed to minimize the number of landmarks the user has to place for semiautomatic feature extraction to succeed. We chose these landmarks with care, to ensure that different users could recognize them easily and register them with reasonable accuracy.

The aging module turned out to be more complex than we expected. Had we taken a simplistic, image-based approach to aging, we could have integrated something by now. But our optimum model based on biological features was too complex and had to be reengineered midway through development. As a result, the integration of the aging module into the system has been delayed.

The present implementation of the facial component database used for face composition is file-based. We need to make it part of the CAFIIR database for ease of maintenance.

We are extending the CAFIIR system in two directions. First, we're augmenting the face composition module with accessories such as spectacles and headgear, as well as features like moles and scars. The objective is to extend CAFIIR's ability to identify faces in the presence of these artifacts. We are also revising CAFIIR to handle color photographs. While experts disagree on whether color is important, the trend in photography indicates the general public prefers color photos.

Plans are underway to scale up the database size of the CAFIIR system. Immediate plans include testing the system with several thousand photographs, then scaling it up further by a factor of 100. The second scaling effort will be possible only in an operational environment.

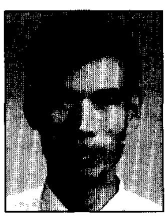
The system is generally easy to use. We've exhibited the system in several forums, receiving many positive comments. The user interface especially has been well received. We made a conscious decision to keep the design of the CAFIIR system modular, resulting in what we believe to be a general-purpose, multiretrieval, multimedia database engine. We are presently confirming this belief by redeploying the database engine for other image- and text-oriented applications such as STAR, the System for Trademark Archival and Registration, which allows trademark offices to compare applicants' submissions to previously trademarked materials. **MM**

Acknowledgments

We would like to thank our colleagues, Y.H. Ang, C.P. Lam, H.H. Loh, J.J. Lim, and K. Moorthy. The ideas in this article would be impossible without their effort.

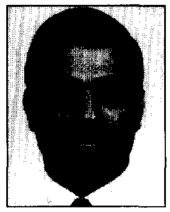
References

1. A. Gupta, T.E. Weymouth, and R. Jain, "An Extended Object-Oriented Data Model for Large Image Bases," *Proc. Second Symp. on Large Spatial Databases*, Springer-Verlag, Berlin, 1991, pp. 45-62.
2. A. Pentland, "Eigenface for Recognition," *J. Cognitive Neuroscience*, Vol. 3, No. 1, Winter 1991, pp. 59-70.
3. A.L. Yuile, "Deformable Templates for Face Recognition," *J. Cognitive Neuroscience*, Vol. 3, No. 1, Winter 1991, pp. 59-70.
4. W.I. Grosky and R. Mehrota, "Index-Based Object Recognition in Pictorial Data Management," *Computer Graphics, Vision and Image Processing*, Vol. 52, No. 3, Dec. 1990, pp. 416-436.
5. J.K. Wu, *Neural Networks and Simulation*, Marcel Dekker, New York, 1993.
6. L.A. Zadeh, "Similarity Relation and Fuzzy Ordering," *Information Science*, Vol. 3, No. 2, Apr. 1970, pp. 177-206.
7. E. Binaghi, I. Gagliardi, and R. Schettini, "Indexing and Fuzzy Logic-Based Retrieval of Color Images," *Proc. 2nd Working Conf. on Visual Database Systems*, Springer-Verlag, Budapest, 1991, pp. 84-97.
8. J.J. Lim, "Free Text Databases," Tech. Report, Inst. of Systems Science, Nat'l. Univ. of Singapore, 1992.



Jian Kang Wu is a member, research staff at the Institute of Systems Science, National University of Singapore, heading a group for multimedia database research. He received his Bachelor degree from the University of Science and Technology of China and a PhD from the University of Tokyo.

Wu was named a "Research Pioneer" by the National Committees of Science and of Education of China, and has won prizes from the Chinese Academy of Science. His research interests include multimedia database systems, image information systems, geographic information systems, computer vision, and neural networks.



Arcot Desai Narasimhalu is a program manager at the Institute of Systems Science, National University of Singapore, where he directs the archiving and retrieval of multimedia information program. Narasimhalu has a BEng degree in electronics and communications from Madras University, an MTech in electrical engineering (light current) from the Indian Institute of Technology, Madras, and a PhD in electrical engineering from the University of Hawaii.

Narasimhalu is a member of the editorial boards of Pergamon Press' *Expert Systems with Applications, Information Processing and Management*, British Computer Society's *Computer Journal*, ACM/Springer Verlag's *Journal on Multimedia Systems*, and Singapore Computer Society's *Journal on Information Technology*. He coedited Pergamon Press' book, *Operational Expert Systems in Far East Asia*. Narasimhalu's present interests include multimedia systems, multimedia archiving and data management, expert systems, heterogeneous database management, and decision support systems.

Readers can contact the authors at the Institute of Systems Science, National University of Singapore, Heng Mui Keng Terrace, Kent Ridge, Singapore 0511. Wu's e-mail address is jiankang@iss.nus.sg.