

**PRIMITIVE-BASED IMAGE CODING TECHNIQUE
FOR STILL PICTURES**

**Adnan M. Alattar
and
Sarah A. Rajala**

**Center for Communications and Signal Processing
Department of Electrical & Computer Engineering
Box 7914
NCSU
Raleigh, NC 27695-7914**

CCSP-TR-89/4

February 1989

PRIMITIVE-BASED IMAGE CODING TECHNIQUE FOR STILL PICTURES

Adnan M. Alattar and Sarah A. Rajala

Department of Electrical and Computer Engineering
and
Center for Communications and Signal Processing
P.O. Box 7914, North Carolina State University
Raleigh, NC 27695-7914

ABSTRACT

A new image coding technique for achieving an extremely high compression ratio while producing good image quality is described. The idea is analogous to that used in forensic applications when constructing a composite image of a criminal from a collection of images of facial features. Simply, the input image is segmented into complicated primitives that are coded using a previously constructed database of primitives. A duplicate database is used at the receiver to decode the primitives. The decoded primitives are assembled as in computer animation to produce a faithful reconstruction of the original image. The technique has been applied to head and shoulders images and compression ratios over 1000:1 have been achieved.

1. INTRODUCTION

Most traditional image coding techniques follow one of two basic coding strategies: transform or predictive [1]. Both techniques concentrate on coding messages that are algebraically calculated from the pixel values of the image. Hence, these techniques may be classified as algebraic techniques. Recently, a new class of techniques known as second generation techniques [2,3] have utilized image segments of uniform properties as messages, rather than numerical messages [2,3]. This utilization has allowed a compression ratio of 100:1 to be achieved. This ratio is amazingly high compared to those obtained from traditional techniques (maximum 20:1).

An extremely high compression ratio can be achieved by extending second generation techniques to include segments of non-uniform properties and complicated primitives. The idea used is similar to that used in composing pictures from pieces of other pictures as in animation [4], computer graphics [5] and forensic [6] applications. Simply, any still picture, such as a head and shoulders photograph, is segmented into complicated primitives that are coded using a previously constructed database of primitives. A priori knowledge is used in constructing a set of filters and a dynamic model of the head and shoulders images, which allow fast and accurate segmentation of the head in the image into its primitives.

The paper is organized into five sections, including the introduction. Section (2) discusses the general methodology of the primitive-based image coding technique. Section (3) applies the primitive-based technique to head and shoulders images. Section (4) discusses the implementation and simulation results. The last section contains a conclusion and final remarks.

2. PRIMITIVE-BASED IMAGE CODING: THE GENERAL APPROACH

For each class of images, there is a universal set \mathbb{S} of primitives of which the set \mathbb{P}_f of primitives in any image f in the class, is a subset. If $\mathbb{P}_f = \{p_1^{(j_1)}, p_2^{(j_2)}, \dots, p_{n-1}^{(j_{n-1})}, p_n^{(j_n)}\}$, where the superscript j_i indicates the state of the i^{th} primitive p_i , then the image f can be represented as the union of its primitives; i.e.,

$$f = \bigcup_{i=1}^n p_i^{(j_i)} \quad (1)$$

If the universal set \mathbb{S} is known and the subset \mathbb{P}_f for an image f is identified from \mathbb{S} , then the image f can be easily reconstructed from the primitives of set \mathbb{S} and some information about the sizes, orientations, and locations of the original primitives in f .

Since the type of primitives that might appear in a given class of images is finite, the set \mathbb{S} is finite, and therefore can be represented by a database. This

database can be used to code any image in the given class. The coding process starts by using image analysis techniques to extract the primitives P_f of the input image. Each of the extracted primitives is normalized with respect to size and orientation, and is then matched to the database, which is assumed to be available before coding. When the best match is found, its order in the database, along with the primitive's normalization factors and location in the original image, is coded and transmitted. If a good match for a primitive cannot be found, the primitive itself is transmitted and the databases at both the transmitter and the receiver are updated to include the new primitive.

The receiver uses the transmitted information and a duplicate database to construct a faithful replica of the original image. The idea is very similar to that used in computer animation [4]. First, each primitive is decoded from the database using its order. It is then re-scaled and re-oriented using its normalization factors so that it assumes its original size and orientation. Finally, each primitive is projected onto its original place in the image and any artifacts that might appear in the transition region between primitives are removed with a smoothing filter.

Image compression is achieved in this technique through transmission of a small number of parameters rather than transmission of the primitive itself. These parameters include the order of the primitive in the database, its normalization factors, and its location in the original image.

3. PRIMITIVE-BASED IMAGE CODING: APPLICATION TO HEAD AND SHOULDERS IMAGES

In forensic applications, thousands of head and shoulders images of previously apprehended criminals are collected in mug-shot files. In spite of the recent advances in image compression techniques and electronic storage technology, the tremendous amount of data forming these files is a major obstacle in the way of realizing a fully computerized archiving system with storage and transmission capabilities. However, since users are often more concerned with the general view of a picture than with its details, these files can now be efficiently coded by using the new primitive-based image coding technique.

Applying the primitive-based image coding technique to a given class of images requires: (1) an accurate definition of the primitives of the images in that class, (2) a fast and accurate primitive extraction algorithm, (3) the construction of a representative database of reasonable size, and (4) a way to compare primitives.

3.1 Primitives of Head and Shoulders Images

The primitives of head and shoulders images are well-defined; they correspond to image segments containing facial features of the human face. Therefore, image segments which contain the eyes, nose, mouth, ears, chin, mustache, eyebrows, or hair sections are considered to be the primitives of head and shoulders images. Each of these primitives may have more than one state. For instance, the mouth may be closed, slightly opened, opened, or widely opened.

Location of the facial features in a given input image is based on a simple model in which the shape of the human head in a photograph is approximated by an ellipse. The geometry of the ellipse and knowledge from the art of drawing the human head are combined to derive simple mathematical relations to locate the facial features in a quick and accurate way. These relations and a complete feature-location algorithm are detailed in [7]. The orientation of the ellipse and the calculated positions of the facial features are used with a set of filters such as those depicted in Fig.1 to extract the facial features. Facial feature extraction is described in the next subsection.

3.2 Facial Feature Extraction

The facial features can be extracted from the input image by using a set of filters similar to those used by Wiederhold [8]. Some possible shapes for these filters are shown in Fig. 1. Each filter can be thought of as a mask of zeros and ones. The pixels corresponding to the shaded areas in Fig. 1 have the value of 1, and those outside the shaded area have the value of 0. When a filter is multiplied by the input image, the values of the image pixels that correspond to the shaded area in the filter do not change, since they are multiplied by 1. However, the

values of the other pixels become zero. This is the basis for primitive extraction from the input image.

Before it can be used to accurately extract a given primitive of the input image, the appropriate filter's size, orientation, and center of operation must be determined by applying the algorithm for facial feature location which was discussed in the preceding subsection.

3.3 The Database

The database that represents the universal set \mathbb{S} for head and shoulders images can be easily constructed from a collection of primitives extracted from prototype photographs. These photographs must be selected to be highly-representative of the class of head and shoulders images. A representative database of reasonable size would contain about the same number of primitives as the *Identikit* and *Photofit* kits used in forensic applications [6], which contain around 150 hair sections, 60 ears, 100 eyes and eyebrows, 50 noses, 100 mouths, and 70 chins.

The primitives must be normalized with respect to size and orientation and grouped in the database under their respective feature types. Each group would contain primitives representing the various shapes and states that may be assumed by its corresponding facial feature. For instance, the eye group would contain picture segments of eyes of different people. These eyes would be in various states, e.g., closed, slightly opened, and fully opened. The database should also contain a group of average pictures to be used as base images in the reconstruction process.

Singular-value-decomposition [9] can be used to construct a compact database. Since facial features of the same type look alike, it is possible to store the primitives of each group in terms of a single primitive from that group. First, a prototype primitive is selected to represent each group. Then, the singular-value-decomposition of the selected primitive is calculated. The basis vectors of this primitive are used to transform the other primitives in the group, and only the

differences between the transformation coefficients of these primitives and those of the prototype primitive are stored in the database.

3.4 Feature Matching

Feature matching is a major obstacle in realizing the primitive-based image coding technique. The difficulty lies in the fact that feature matching is more subjective than mathematical. However, mathematical error measures such as the mean-square-error are usually adequate for finding the best match when comparing a given image to others in a set of images. Therefore, the mean-square-error measure is used to find the best match of an extracted feature to the features that have been pre-stored in the database. Since there may be a slight mis-alignment between the extracted feature and the features in the database, it is necessary to displace the feature one or two pixels in the vertical and horizontal directions during the comparison. Then the feature from the database which produces the least mean-square-error is taken as the best match, provided that this mean-square-error is less than a specified threshold.

Another method of feature matching utilizes a dedicated neural network. Stonham [10] developed a self-adapting machine called *Wisard* as a general purpose pattern recognition system. He also described its application to human face recognition. The central component of *Wisard* is a neural network that adapts to a particular training set during a learning process. The network is then able to recognize patterns that are very similar to those of the training set. The *Wisard* machine is of parallel nature and is easily fabricated using the currently available VLSI technology. Therefore, such a network could be used to match an image primitive of a given type to a similar primitive in the database.

4. IMPLEMENTATION OF THE PRIMITIVE-BASED CODER AND SIMULATION RESULTS

Several steps were taken to implement the primitive-based image coder. First, the general shape of each filter in the filter set of Fig. 1 was defined and

stored in a separate file for each filter. The size of each filter was normalized to fit within a square of size 1 by 1. This square was centered at the origin. Second, the centers of the facial features in the input image were determined by applying the facial feature location algorithm [7]. Third, the located features were extracted using the defined filters. The center of each filter was first shifted to coincide with the center of the corresponding feature in the image, then it was re-scaled and re-oriented according to the size and orientation of the head in the input image. The filter was then multiplied with the input image to extract the corresponding feature. The shifting, scaling, and orienting factors used in adjusting the filter were all provided by the facial feature location algorithm.

A real database has not yet been constructed, so a hypothetical one was used in the coding/decoding process. Simply, the extracted features were manually compared with the features of other images and the best matches were selected subjectively. The selected features were projected onto an average face that was constructed from another image of the same person in the input image. For example, the average face of Fig. 3 was constructed from the image of Fig. 2. This was done by blurring the head in the image of Fig. 2 with an 11 by 11 average filter, and then rotating and shifting it so that its center and orientation coincided with those of the input image of Fig. 4. When the selected primitives were projected onto this average face the image of Fig. 5 was obtained. The previously described filter set was used to extract the primitives from the original image and from the other images. It also was used to project the primitives of best match onto the base image. Finally, the boundaries of the primitives were smoothed with a smoothing filter that replaced the value of each pixel on the boundary with the average of the values of the pixels in a 3 by 3 window around the boundary pixel.

It should be noted that only the five parameters of the ellipse and nine database indices corresponding to the nine facial features extracted from the input image would be needed to reconstruct the image of Fig. 5. If 16 bits were allocated to the orientation parameter of the ellipse, and 8 bits to each of the four other parameters and the nine indices, then a total of 120 bits would be required to transmit the image of Fig. 4 and obtain the image of Fig. 5. Since each pixel in the input image is represented by 8 bits, and the size of the image is 128 by 128

pixels, a compression ratio of 1092:1 is achievable using this technique. However, with full implementation of the primitive-based coder, this compression ratio is expected to be somewhat less, because when a best match can not be found in the database, the original primitive itself has to be transmitted. In any case, the expected compression ratio is extremely high compared to those which were achievable by the earlier image coding techniques (maximum of 100:1 by second generation techniques[6]).

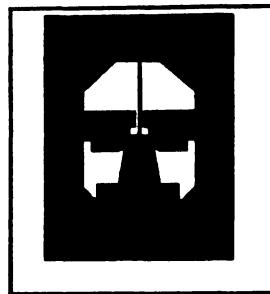
Replacing the original primitives in a head and shoulders image with similar ones from a database introduces image degradation. This degradation, however, can be reduced with careful selection of the entries of the database. With such a selection, the recognizability of the face can be maintained. In fact, Haig [11] performed a set of experiments that produced strong evidence to indicate that some facial features can be replaced with similar ones without affecting face recognizability. Similar studies show that the general shape of the head, including hair style, is the most salient feature, and that the eyes are the second most salient feature. Other facial features such as the nose and the mouth are much less significant in face recognition.

5. CONCLUSION

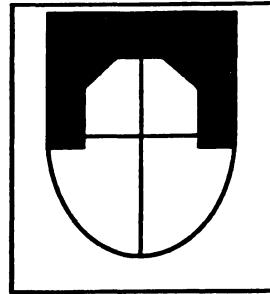
Primitive-based image coding can achieve an extremely high compression ratio by using a priori knowledge about an input image and a highly-representative database of primitives. The new technique is well-suited to coding of head and shoulders images of the type commonly found in mug-shot files and "face-to-face" telecommunications. Other classes of images can also be coded using this technique. This requires well-defined primitives, primitive location and extraction algorithms, and a representative database for the given class of images. It is also possible to extend the new technique to sequences of images, such as those encountered in *Picture Phone* and teleconferencing. The achievable compression ratio in these cases is expected to be high enough to allow real-time transmission of image sequences through local telephone networks at 64 Kbits/s.

6. REFERENCES

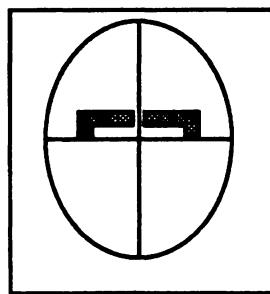
- [1] A.N. Netravali and J.O. Limb, "Picture Coding: A review," *Proc. IEEE*, vol. 68, pp. 366-406, March 1980.
- [2] M. Kunt, A. Konomopoulos and M. Kocher, "Second-Generation Image Coding," *Proc. IEEE*, vol. 73, pp. 549-574, April 1985.
- [3] S.A. Rajala, M.R. Civanlar and W.M. Lee, "A Second Generation Image Coding Technique Using Human Visual System Based Segmentation," *Proc. ICASSP'87*, pp. 1362-1365, April 1987.
- [4] N. Magnenat and D. Thalmann, *Computer Animation*, Springer-Verlag, Tokyo 1985.
- [5] M.L. Gillenson and B. Chandrasekaran, "A Heuristic Strategy for Developing Human Facial Images on a CRT," *Pattern Recognition*, vol. 7, pp. 187-196, 1975.
- [6] G. Davies, H. Ellis, and J. Shepherd, *Perceiving and Remembering Faces*, Academic Press, London 1981.
- [7] A.M. Alattar and S.A. Rajala, "A Model-based Algorithm for Facial Feature Extraction from A Photograph," *Computer Vision, Graphics, and Image Processing*, submitted for publication.
- [8] T.P. Wiederhold, Facial Image Generation By Computer, *Master's Thesis*, University of Houston, Dec. 1976.
- [9] H.C. Andrews and C.L. Petterson, "Singular Value Decomposition (SVD) Image Coding," *IEEE Trans. Commun.*, COM-24, 4, April 1976, 425-432.
- [10] T.J. Stonham, "Practical Face Recognition and Verification With Wisard," in *Aspects of Face Processing*, Edited by H.D. Ellis, M.A. Jeeves, F. Newcombe, and A. Young, Martinus Nijhoff 1986.
- [11] N.D. Haig, "Investigating Face Recognition With an Image Processing Computer," in *Aspects of Face Processing*, Edited by H.D. Ellis, M.A. Jeeves, F. Newcombe, and A. Young, Martinus Nijhoff 1986.



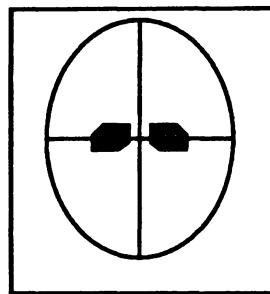
a. All Filters



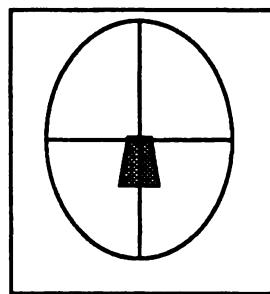
b. Hair Filter



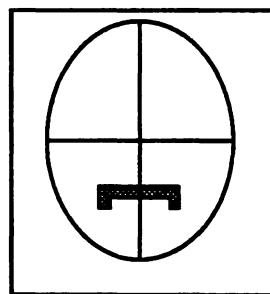
c. Eyebrows Filter



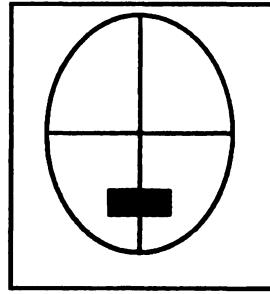
d. Eyes Filter



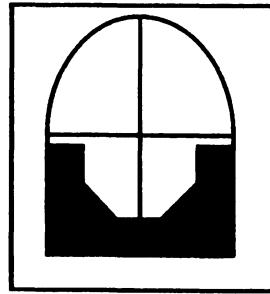
e. Nose Filter



f. Mustache Filter



g. Mouth Filter



h. Chin Filter

Fig. 1. Filter Set For Facial Feature Extraction



Fig.2. General Image For Constructing Average Face



Fig.3. Average Face Constructed From the Image in Fig. 2



Fig. 4. Input Image to Be Coded By Primitive-Based Coder

