Face-tracking and coding for video compression

William E. Vieux[1]*, Karl Schwerdt[2]**, and James L. Crowley[2]***

[1] Department of Electrical and
Computer Engineering
University of Oklahoma
202 West Boyd, Room 219
Norman, OK 73019 USA

[2] Project PRIMA, Lab. GRAVIR - IMAG
INRIA Rhone-Alpes
655, ave. de l'Europe
38330 Montbonnot St. Martin
France

Abstract. While computing power and transmission bandwidth have both been steadily increasing over the last few years, bandwidth rather than processing power remains the primary bottleneck for many complex multimedia applications involving communication. Current video coding algorithms use intelligent encoding to yield higher compression ratios at the cost of additional computing requirements for encoding and decoding. The use of techniques from the fields of computer vision and robotics such as object recognition, scene interpretation, and tracking can further improve compression ratios as well as provide additional information about the video sequence being transmitted. We used a new face tracking system developed in the robotics area to normalize a video sequence to centered images of the face. The face-tracking allowed us to implement a compression scheme based on Principal Component Analysis (PCA), which we call Orthonormal Basis Coding (OBC). We designed and implemented the face tracker and video codecs entirely in software. Our current implementation of OBC operates on recorded video sequences, making it appropriate for applications such as video email.

Key words: video coding, face tracking, computer vision, principal component analysis

1 Introduction

Availability of high-performance computers at reasonable price levels has led to the development of complex multimedia applications for teleconferencing, video telephony, visual data exchange, and other video communications. As these new applications are developed, the bottleneck of limited transmission bandwidths and storage space becomes a critical problem. In general, researchers must make the encoding and decoding of video data more complex in order to lower bandwidth and space requirements. A

* `billv@ou.edu`

** `Karl.Schwerdt@imag.fr`

*** `Jim.Crowley@imag.fr`

common approach today is to develop systems and algorithms using computer vision techniques, often for modeling [1], but also for tracking [2] and combinations of both.

Previous research in our group has been devoted to exploring new ways for Man-Machine Interaction [3], [4] and the interpretation of scenes in images or video sequences. This research has produced results that are easily applied to communication applications. While their theoretical feasibility has been demonstrated [5], these techniques yet have to be integrated into a system comparable to current standards (H.323/H.263, MPEG) and applications (PitureTel [2]) to prove their practical use for video communication. We have recently started a project to tackle the task of building a video communication system in order to compare our algorithms with other standards and developments. This video communication system integrates a face tracker and a standard video codec, as well as an Orthonormal Basis Coding (OBC) compression scheme for face-tracked video sequences. The standard video codec was implemented according to the ITU-T (Telecommunication Sector of the International Telecommunication Union) recommendation H.263 [6].

Actual video coding standards from the ITU-T and ISO (International Organization for Standardization) rely mainly on the statistical relation between images and pixels. Intuitively, the use of additional a priori information, if wisely applied, should further increase compression ratios. Today, face trackers are commonly used in model-based applications. That is, they directly interact with the video codec. We are not targeting a model-based algorithm. Instead, we present in this paper a new, low-complexity face tracker, which has been combined with a standard video codec, as well as a OBC compression scheme. The OBC compression scheme currently operates on face-tracked sequences and is appropriate for applications such as video email. The face tracker is based on techniques that have been used in the robotics area, and has a fast, robust, and modular structure. The robustness comes from parallel use of partially redundant modules.

We will discuss the face tracker in more detail in section 2. Section 3 then discusses the integration and setup of the entire video processing system. Performance evaluation and results are presented in section 4, and finally, section 5 concludes the paper, giving an outlook on further research.

2 Face Tracking

The goal of visual tracking is to always keep a moving target centered in the field of view of one or more cameras. This is useful in application areas such as robotics (assembly, navigation) and surveillance. Using tracking as a pre-processing step for video image encoding provides the coder with images normalized to the object being tracked, e.g., a human head. This reduces the diversity of the most important image content: the face. In addition, it gives us the position and size of the face. Our system uses this information in two ways. Firstly, by adapting the quantizer step-size and maintaining motion vectors, the standard video codec (H.263) can be measurably improved. Secondly, a video sequence with the head normalized in the center of the image can be generated for use with the OBC compression scheme.

The face tracker has been built with a modular structure. It automatically detects a face, keeps track of its position, and steers a camera to keep the face in the center of the image. This offers a user the possibility to freely move in front of the camera while the video image will always be normalized to his face. Figure 1 contains the block diagram of the face tracking system as it is implemented today.

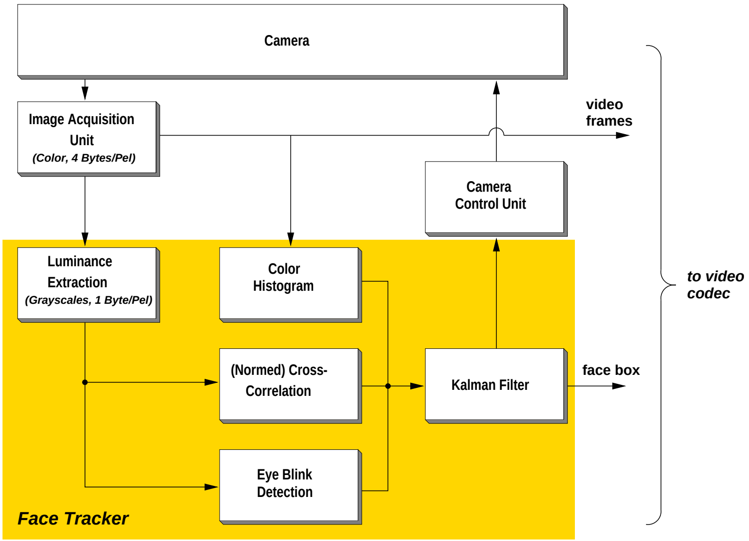

Fig. 1. Block diagram of the face tracker

We follow a face by maintaining a rectangle which frames the face, and call this rectangle the face box. There are three modules that actually do the tracking: Eye Blink Detection, Cross-Correlation, and Color Histogram. They have similar functions and are thus partially redundant. This redundancy guarantees that the face tracker is robust [5]. Since the modules of the face tracker have already been well-documented [5], we do not need to go into detail about the techniques involved. In the following subsection we rather briefly discuss the modules' features.

2.1 Eye Detection

The Eye or Blink Detection module is used for a quick initialization or recovery (re-initialization) of the face tracker. While appearance of different people can be extremely different (skin color, beard, hair, etc.), they all have to blink every ten seconds or so to keep their eyes moist. Blink duration is relatively short, well under a second. We have found that it is possible to detect eyes by taking the difference between two images with a time lag of approximately 200 milliseconds. This has shown to give good and reliable results.

Eye blinking does not occur frequently enough for use for permanent face tracking at a rate of 10 to 25 images per second. We therefore use the cross-correlation and color histogram modules for permanent tracking, and the eye detection module for the initialization of those modules.

2.2 Normalized Cross-Correlation

An intuitive way for object tracking is to cross-correlate two areas of subsequent images. However, this can turn out to be costly if 1) the areas to be correlated are relatively large and/or 2) the area to be searched is relatively large. Hence, it appears natural to use correlation only for the detection of small movements.

Our correlation algorithm needs to be initialized with a correlation mask, and this correlation mask should represent a characteristic area of the image, and, for practical purposes, should lie within the detected (or to be detected) face. We chose a quadratic area between the eyes of about 20 x 20 square pixels containing parts of the eyebrows as correlation mask. Moreover, we limit the area to be searched to a region of interest, which is roughly identical to the face box. We use a (normalized) cross-correlation algorithm to find a best match between the correlation mask and an area of the same size in the region of interest.

2.3 Skin Detection by Color Histogram

Another way to detect or follow objects within an image is to use their color. The color of human skin is usually well distinguishable in an environment like an office. It has been shown [7], that faces can be detected in a stable and reliable manner using a joint histogram of the color components of a pixel normalized by the pixel luminance. The color histogram module searches the entire image and returns the biggest area of skin. Once properly initialized, it can recover the face-box quickly and reliable.

2.4 Kalman Filtering

All three modules above output a face box, but the coordinates of the face boxes are generated differently. Therefore we added a zeroth order Kalman filter in order to smooth out jerky face box transitions and to keep the face-box at approximately the same size.

2.5 Confidence Factors

While all three modules return a result once they are run, those results have to be qualified in order to react properly to what is going on in front of the camera. One possibility is the use of a confidence factor. Confidence factors are a normed quality measure between zero and one. An empirically determined threshold is then applied to decide if the face box found by a particular module is good.

2.6 Camera Control

The detected and filtered face box is eventually fed into the camera control unit. This unit calculates the distance between the actual position of a face and the center of the image. Then, a PD- controller has the camera pan, tilt, and/or zoom accordingly. We use a PD-controller because it dynamically reacts to the speed of head movement.

2.7 Performance

The face tracking system has been improved over the version of [5] to work faster and more stable at a frequency of about 12 Hz with, and 18 Hz without automatic camera control. The video sequences recorded for use with the PCA compression are recorded at approximately 5 to 6 Hz. This is due to the use of the Silicon Graphics Digital Media Library for writing out the file containing the video sequence. Code improvements, such as saving the images in memory or other techniques should allow recording at a much higher rate.

3 Adding face tracking to video coding

4 Overview

Today, there are four video coding standards being used for commercial systems: The ITU-T recommendations: H.261 [8] and H.263 [6], plus the ISO standards 11172 [9] (MPEG-1) and 13818 [10] (MPEG-2). H.261 is intended for low-motion video communication over p x 64 kbit/s ISDN lines, H.263 for low bit rate communication, MPEG-1 for full-motion video coding at up to 1.5 Mbit/s, MPEG-2 for up to 10 Mbit/s. We chose the H.263 recommendation as the standard video codec for our system since it comes closest to our purpose.

Our second codec approach, Orthonormal Basis Coding, was developed with a video email application in mind. A face tracked video sequence is cropped in order to provide a sequence of images with the face normalized and centered in each image. Then selected frames from the sequence are used to create a basis space into which new images can be mapped. Each mapped image can then be represented as a vector of coefficients; the number of coefficients is equal to the number of images in the original ”basis space.” By only storing and transmitting the vectors, extremely high compression rates can be achieved, especially for long sequences.

4.1 Integrating face tracking and video coding

The purpose of using a face tracker is to normalize the video stream images to an object, in our case a face. In other words, our face tracker can be seen as a camera control device, without direct interaction with any part of a video codec. It is a pre-processing stage, and its optional use under real-time constraints solely depends on the computer it is to be run on. This is very practical from a system integration point of view, because we do not need to alter the video codec and thus stay compatible.

Figure 2 shows the block diagram of our video processing system with face tracker and video codecs as separate modules. Both video codecs consist of three parts, a control unit, an encoder and a decoder. The only interaction between the face tracker and the codec is that the face tracker is providing the coding control unit with the face box.

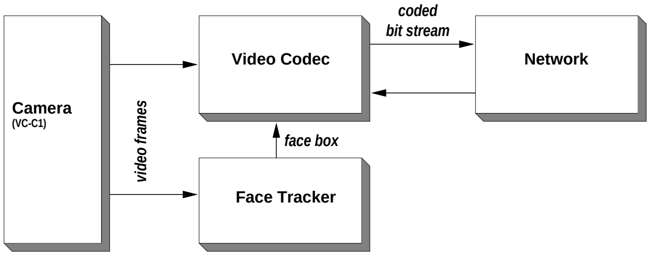

Fig. 2. Block diagram of the modules of the video processing system

MPEG-1 Basic algorithm of MPEG-1, as well as H.261, H.263, and MPEG-2, is the three step compression scheme: 1) energy compaction by Discrete Cosine Transform (DCT), 2) entropy reduction by Differential Pulse Code Modulation (DPCM), and 3) redundancy reduction by Variable Length Coding (VLC). Depending on their intended use, the different standards enhance this compression scheme by forward prediction, backward prediction, motion compensation, and other additional features.

Orthonormal Basis Coding The Orthonormal Basis Coding scheme operates as follows: 1) a limited set of images is chosen from the sequence to form the basis, 2) a Karhunen-Loeve expansion is performed to generate an orthonormal basis space from the images, 3) each image in the sequence is mapped into this basis space resulting in a small set of coefficients, 4) the images used to create the basis space and the sets of coefficients are stored in a file for later decoding [11] [12]. An image mapped into the basis space will produce a number of coefficients equal to the number of images used to create the basis space. We have obtained good results using only fifteen basis images for a 400-frame video sequence. Thus, each frame was represented by only fifteen coefficients.

Due to processing constraints, a full Principal Component Analysis cannot be done in a reasonable time. That is to say, a basis space cannot be generated using every image in the sequence and keeping only the most representative eigenvectors for a basis. Thus, we explored two algorithms for choosing the images for our basis. The threshold method assumes that similar frames are likely to be located sequentially in the video sequence. This is not necessarily the case when each image contains only a face talking. The most-representative method attempts to find similar images anywhere in the sequence.

The threshold method has a complexity of O(n) and works as follows. The normalized cross correlation is computed between image zero and subsequent images until it drops below a certain threshold. At that point in the sequence, the current image is added to the basis and subsequent images are cross correlated with that image until the threshold is crossed again. The most-representative method has a best case complexity of O(n) and a worst case of O(n2) although neither are very likely. It takes image zero and cross correlates it with all the other images in the sequence, all of the images that are very similar to image zero are put in set A, the others are put in a ”to do” set. The first image of the ”to do” set is cross correlated with all the others in the ”to do” set and the most similar are put in set B. The rest stay in the ”to do” set. This continues until all similar images are grouped in sets. One image from each of the biggest sets is taken to form the basis. In general, the most-representative method produced superior results at the cost of slightly more computing time. This is due to the fact that while the subject is talking, the mouth and facial features return to common positions at different times in the sequence.

5 Performance Evaluation

The system described above was developed and run on a Silicon Graphics INDY workstation in a common configuration and under normal working conditions. That is, the computer was used in a network with several users logged in. As evaluation criteria for image quality, we use the well known Peak-Signal-to-Noise-Ratio (PSNR). The PSNR of the k-th frame is defined as

$$PSNR(k) = 10 \cdot \log_{10} \left[\frac{255^2}{\frac{1}{MN} \sum_{m,n} \left[f_k(m,n) - \hat{f}_k(m,n) \right]^2} \right],$$

where M and N are the width and height of the image, m and n the pixel indices, $f()$ the original pixel values, $\hat{f}()$ the decoded pixel values. Note that the term in the denominator is the mean squared error.

5.1 Reconstruction

Various sequences were compressed and reconstructed using the *threshold*, *most-representative*, and MPEG methods. *The most-representative* method produced better reconstructions than the threshold method in every case. In fact, it always averaged over 2 dB higher PSNR. See Figure 3.

In Figure 4, there is a noticeable improvement in the reconstruction quality as the number of basis frames is increased. Images included in the original basis space have no error in their reconstruction, thus PSNR has no meaning and is ignored in the following graphs.

Using the standard compression options with the Berkeley mpeg_encode program [10], we found an average PSNR of approximately 27 dB for the BillS2 sequence. The MPEG reconstruction errors were due to artifacts in the images, while the reconstructed images

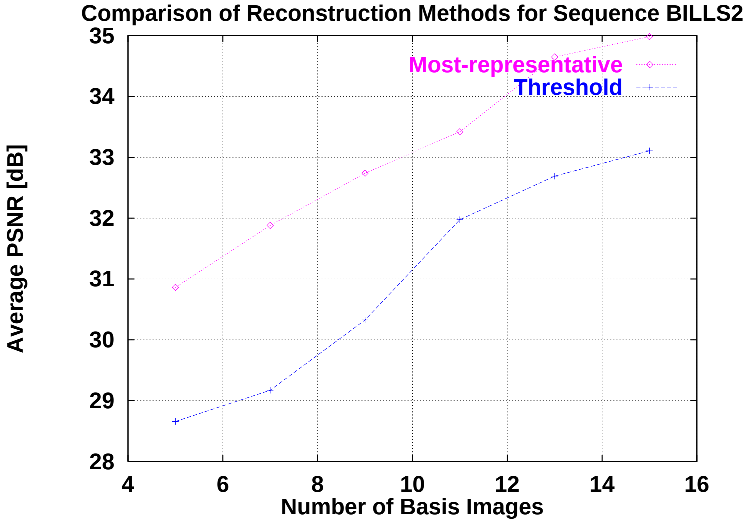

Fig. 3. PSNR vs. Number of basis images for both methods

from the OBC codec were slightly blurred, as can be seen in Figure 5. The closed mouth in Figure 5b is due to the fact that there were no images with a fully opened mouth among the fifteen basis images. This is a problem that could be rectified by better selection of the basis images, as discussed in the Conclusions and Future Work section.

5.2 Compression

The BillS2 video clip contains 418 frames and lasts 69 seconds (6 FPS). The various file sizes are shown in Table 1. It is important to note however, that the basis images are stored in the OBC file in raw YCrCb format. We used the GZIP utility [7] on the OBC to do simple compression (redundancy elimination) on the file. As explained in the Conclusions and Future Work section, specific compression (e.g., JPEG) of these images would significantly reduce file size. Each additional frame for the 15-basis-frame reconstruction would have added 60 bytes to the OBC file size. Additional frames for the 5-basis-frame reconstruction would have added only 20 bytes, while additional frames for the MPEG would have added significantly more.

6 Conclusions and Future Work

There are several areas that would benefit from additional research. The current system stores the original basis images in the file to transmit along with the coefficients for each frame. The coefficients are relatively small byte-wise compared to the images, and somewhat benefit from a variable length compression (VLC) scheme. The images are

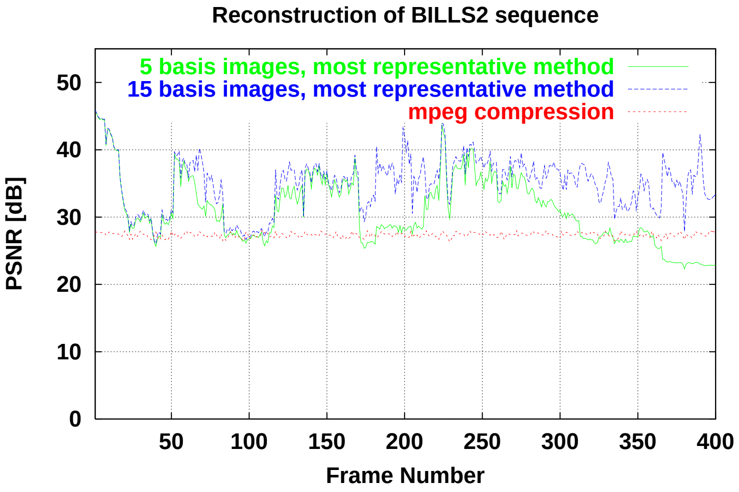

Fig. 4. PSNR for each frame of BillS2

Table 1. File Size Comparison (kB) for BillS2

Video Stream	File Size [kB]
Original Video (Uncompressed)	12550
MPEG	72
OBC (5 basis frames)	71
OBC (5 basis frames) with GZIP	58
OBC (15 basis frames)	217
OBC (15 basis frames) with GZIP	178

stored in raw YCrCb format, and we can further exploit a priori information about the images; they are all images of a normalized face. Either through JPEG compression of each basis image or an MPEG-like compression of the sequence, the file size could be significantly reduced. The impact of information loss (due to the DCT and quantization in the JPEG/MPEG standards) on the image reconstruction is yet to be determined. Even a simple differencing of the images and VLC compression would likely reduce the file size significantly.

In some of the image reconstructions, the eye and lip movements were slightly blurred. This could be improved by applying a weighted mask over the eyes and lips when calculating which basis images to use by the most-representative method. A greater variety of eye and lip configurations would then be placed in the basis space allowing for better reconstruction of important facial expressions. Various techniques from computer vision have been used to create a fast and robust face tracking system, which in turn was used to improve the compression ratio of a standard video codec and

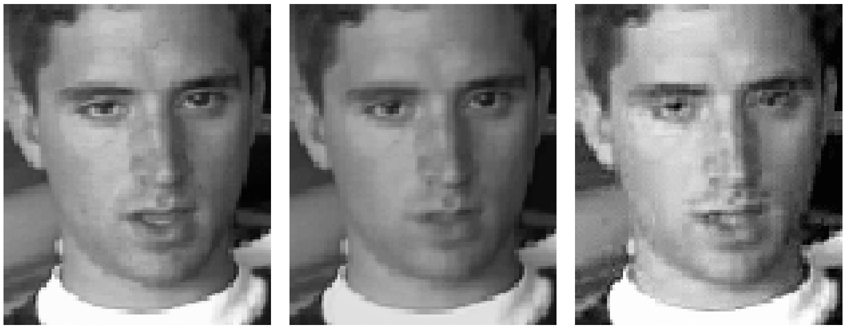

Fig. 5. Frame 136 of BillS2 a) Original Image: no error b) Image from a sequence reconstructed using 15 basis images: note slight blur, closed mouth c) Image from an MPEG reconstruction: note artifacts

our OBC compression scheme. The face tracker also enhances the usability of the entire video communication system by allowing the user to freely move in front of the camera while communicating. It is crucial however, that the face-tracking system be stable and accurate in order to provide the best results for OBC compression. An important question when enhancing any video coding system is, if the results in terms of image quality and compression ratio make up for the added complexity. The system described in this paper gives provides a positive outlook on further development of low-bandwidth video communication.

References

1. T. S. Huang and R. Lopez, "Computer vision in next generation image and video coding," *Lecture Notes in Computer Science*, vol. 0, no. 1035, pp. 13–22, 1996.
2. "La visioconfrence sur IP selon PictureTel et Intel," *01 Reseaux*, pp. 64–65, January 1998.
3. J. L. Crowley, "Integration and control of reactive visual processes," *Robotics and Autonomous Systems*, vol. 16, no. 1, pp. 17–28, 1995.
4. J. L. Crowley, "Vision for man-machine interaction," *Robotics and Autonomous Systems*, vol. 19, no. 3-4, pp. 347–358, 1997.
5. J. L. Crowley, F. Berard, and J. Coutaz, "Multi-modal tracking of faces for video communications," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 640–645, June 1997.
6. ITU-T Study Group XV, "*Recommendation H.263*: Video coding for low bit rate communication," tech. rep., Telecommunication Standardization Sector of the International Telecommunication Union, Geneva, Switzerland, "http://www.itu.ch", 1996.
7. "The GZIP home page," *http://www.gzip.org*, May 1998.
8. ITU-T Study Group XV, "*Recommendation H.261*: Video codec for audiovisual services at px64 kbit/s," tech. rep., Telecommunication Standardization Sector of the International Telecommunication Union, Geneva, Switzerland, "http://www.itu.ch", 1993.
9. ISO/IEC 11172-2, "Information technology – coding of moving pictures and associated audio for digital storage media at up to about 1,5 mbit/s – part 2: Video," tech. rep., International Organization of Standardization, Geneva, Switzerland, "http://www.iso.ch/cate/d22411.html", 1993.

10. “Berkeley MPEG research,” *http://bmrc.berkeley.edu/projects/mpeg*, 1997.
11. M. J. Turk and A. Pentland, “Eigenfaces for recognition,” in *IFIP Working Conference on Engineering for Human-Computer Interaction*, vol. 3, pp. 71–86, 1991.
12. M. Kirby and L. Sirovich, “Application of the karhunen-loeve procedure for the characterization of human faces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 103–108, 1990.