Robust Tracking and Compression for Video Communication

James L. Crowley and Karl Schwerdt
Projet PRIMA, INRIA Rhône-Alpes,
655 ave. de l'Europe,
38330 Montbonnot St. Martin
France

Abstract

Principal components analysis has been studied by the computer vision community as a source of features for recognition of faces, objects and scenes [9], [13]. The use of the dominant principal components as "holistic" features for recognition has provided new insights into view invariant and illumination invariant recognition. Unfortunately, applications in object recognition generally require precise segmentation, and thus prove impractical. Nonetheless, under certain circumstances, principal components are optimal for reconstruction, and thus well suited for coding and compression of images. In such applications, precise tracking rather than segmentation, is required. Precise, stable tracking of faces renders Principal Components Analysis well suited for video coding for video communications.

In this paper we describe experiments with the use of principal components as a technique for coding and compression for video streams of talking heads. We describe a new robust tracking technique for normalizing the position and size of faces. We provide results of preliminary experiments with compression rates and image reconstruction quality using orthogonal basis coding for video communications. We show that a typical video sequence of a talking head can often be coded in less than 16 dimensions.

1. Introduction

In communication by video telephone or video electronic mail, the desired images are generally restricted to a view of the head and shoulders of a speaker. Relevant variations are movements of the mouth, eyes and head. Precise coding of the background is unimportant or may even be undesirable. Such image sequences have properties which make possible high compression ratios. Movements of the face and eyes tend to be repetitive making it possible for a compression algorithm to exploit the limited range of movements and their repetitive nature.

In this paper we report on experiments with techniques which exploit the simplified nature of a talking-head scene to provide a very high compression rate. Our technique has two components: 1) A face tracking system which keeps a face centered in the image at a particular size, and 2) an orthogonal basis coding techniques in which the normalized face image is projected onto a space of basis images.

We employ a multi-modal face tracker which integrates eye blink detection, cross-correlation, and robust tracking of skin colored regions. An earlier version of this multi-modal tracker was reported in [3]. While that system provided robust tracking of a moving face under changing illumination, the color skin detection technique relied on detecting connected components of thresholded color regions. Grouping thresholded pixels led to an unacceptable amount of jitter in the tracked images. We have recently developed a new technique which replaces thresholding and connected components with the moments of color pixels weighted by a Gaussian density function.

Our compression technique relies on estimating a basis of orthogonal images onto which the talking-head images are projected. We present the overall approach and then present preliminary experimental results with the off-line version of this algorithm. In this algorithm, a static fixed basis space is computed using principal components analysis based on a "representative" sample of images. Such an algorithm is well suited to off-line coding for applications such as video electronic mail and talking heads on web pages. We describe an algorithm for selecting the representative images from an image sequence. We then compare the image quality of the reconstructed images for different numbers of basis images.

2. Multimodal tracking of faces

Tracking greatly reduces the required bandwidth while providing the speaker with the freedom to move about while communicating. Our system uses a multi-modal face tracker to drive a motorized camera to normalize the face in position and size. The face tracker automatically detects a face, keeps track of its position, and steers a camera to keep the face in the center of the image. The modules of the face tracker are described in [3]. For completeness, we review the function of each module.

A face is represented as an image position, vertical and horizontal extent and a confidence factor. All measurements are accompanied by a covariance matrix, enabling them to be combined by a recursive estimator based on a zeroth order Kalman filter. A face is initially detecting as a pair of blinking eyes from the responses of

tuned spatio-temporal filters [4]. A correlation mask for the eyes, and a color histogram of skin are initialised based on the position detected by blinking. The position of the eyes and mouth are also useful in biasing the orthogonal basis coding algorithm described below to provide more coding bits for the mouth and eyes.

The eye-blink detection process is used for a quick initialization or re-initialization of the face tracker. This allows the system to continually adapt to changes in ambient illumination. Cross-correlation provides a fast but fragile means to follow facial features. We chose a rectangular area between the eyes of about 20 x 20 square pixels containing parts of the eyebrows as correlation mask. We limit the area to be searched to a "region of interest," which is roughly identical to a rectangle framing the face.

We initially built a color skin detection process which uses a connected components algorithm to group skin colored pixels. The connected components algorithm has been found to be overly sensitive to pixel noise, causing an unacceptable amount of jitter. In the following section we describe a new robust grouping algorithm which greatly enhances stability.

Every observation is accompanied by a numerical confidence factor, computed statistically by comparing the observed parameters to an average parameter vector and normalizing by an observed covariance. This gives a form of Mahalanobis distance which is used as the power for an exponential function, giving a value of 1 for a typical parameter vector and tending towards zero for unlikely vectors. Confidence factors allow the system to detect which processes are functioning reliably and to reinitialize the individual processes dynamically. The estimated position and size of the face is fed into a camera control unit. This unit calculates the distance between the actual position of a face and the center of the image. A PD-controller then directs the camera to pan, tilt, and zoom so as to maintain the face at a standard size and position in the image.

3 Robust tracking of faces using color

Detecting pixels with the color of skin provides a reliable method for detecting and tracking faces. The statistics of the color of skin can be recovered from a sample of a known face region and then used in successive images to detect skin colored regions. Swain and Ballard have shown how a histogram of color vectors can be back-projected to detect the pixels which belong to an object [12]. Schiele and Waibul showed that for face detection, color RGB triples can be divided by the luminance to remove the effects of relative illumination direction [10]. In earlier work [3] we described an algorithm in which a histogram of normalized skin color was initialised by blink detection and then used to determine the possibility that a pixel represents skin. In that work we thresholded skin possibilities and then performed a connected components algorithm on the resulting binary images. Since that time, we have reformulated the skin detection and tracking process using an approach inspired by robust statistics.

3.1 The probability of skin

The reflectance function of human skin may be modeled as a sum of a Lambertian and a specular reflectance function. In most cases the Lambertian component dominates. For a Lambertian surface, the intensity of reflected light varies as a function of the cosine of the angle between the surface normal and illumination. Because the face is a highly curved surface, the observed intensity of a face exhibits strong variations. These variations may be removed by dividing the three components of a color pixel, (R, G, B) by the intensity. This gives an intensity-normalized color vector, with two components, (r, g).

$$r = \frac{R}{R+G+B} \qquad g = \frac{G}{R+G+B}$$

The intensity-normalized pixels from a region of an image known to contain skin can be used to define a two dimensional histogram, $h_{skin}(r, g)$, of skin color. The effects of digitizing noise can be minimized by smoothing this histogram with a small filter. A second histogram, $h_{tot}(r, g)$ can be made from all of the pixels of the same image. This second histogram should also be smoothed by the same filter. These two histograms make it possible to apply Bayes rule to each pixel of an image to obtain the probability that a given pixel is skin.

Application of Bayes rule requires the following terms:

$h_{skin}(r, g)$: Histogram of intensity normalized colors from a region of an image known to represent skin.

N_{skin} : Sum over r and g of $h_{skin}(r, g)$.

$h_{total}(r, g)$: Histogram of intensity normalized colors from the entire image.

N_{total} : Sum over r and g of $h_{tot}(r, g)$.

The probability of a color vector, (r, g) given skin is approximated by

$$p(r, g \mid skin) \approx \frac{1}{N_{skin}} h_{skin}(r, g)$$

The probability of obtaining a skin pixel in the image is approximated by the fraction of observed pixels known to be skin.

$$p(skin) \approx \frac{N_{skin}}{N_{total}}$$

The probability of observing a color vector is given by :

$$p(r, g) \approx \frac{1}{N_{total}} h_{total}(r, g)$$

Bayes rule states that the probability of skin given a

color vector (r, g) is

$$p(skin \mid r, g) = \frac{p(r, g \mid skin) \cdot p(skin)}{p(r, g)}$$

This reduces to the ratio of the two histograms as shown in equation 1.

$$p(skin \mid r, g) \approx h_{ratio}(r, g) = \frac{h_{skin}(r, g)}{h_{total}(r, g)} \qquad (1)$$

The ratio of these two histograms gives a table which directly converts an intensity-normalized pixel (r, g) into the probability that the pixel is skin, *p(skin | r, g)* by table lookup. A default value of 0 may be placed in this table for all pixels for which $h_{total}(r, g)$ is zero. Strictly speaking, equation 1 is only valid for the image from which the skin sample was obtained. In practice, we have found that the technique will work well for subsequent images provided that the color of the scene illumination does not change. This table is trivial to build and may be renewed whenever an independent source has detected the face in the image.

A number of authors have indicated a preference for using Gaussian mixture model in place of the two histograms. Our experience is that such a model provides a very slight improvement in the probability image, at a very great cost in computation whenever the histogram must be renewed, making frequent update of the histogram ratio impractical. For a real-time system, the robustness obtained by frequently renewing the histogram ratio table greatly exceeds the slight improvement observed with a static mixture of Gaussian model.

In order to detect a skin color region we must group skin pixels into a region. We have recently developed a novel robust tracking method for such grouping. Let $P_{skin}(i, j)$ represent the probability map of skin for each color pixel (r(i, j), g(i, j)) at position (i, j).

$p_{skin}(i, j) = p(skin \mid r(i, j), g(i, j))$

The center of gravity or first moment of the probability map gives the position and spatial extent of the skin colored region.

$$\vec{\mu} = \begin{pmatrix} \mu_i \\ \mu_j \end{pmatrix} \qquad \mathbf{C} = \begin{pmatrix} \sigma_i^2 & \sigma_{ij} \\ \sigma_{ij} & \sigma_j^2 \end{pmatrix} \qquad (2)$$

Unfortunately, skin color pixels in any other part of the image will contribute to these two moments. This effect can be minimized by weighting the probability image with a Gaussian function placed at the location where the face is expected. The initial estimate of the covariance of this Gaussian should be the size of the expected face. Once initialised, the covariance is estimated recursively from the previous image.

For each new image, a two dimensional Gaussian function, $g(i, j; \vec{\mu}, \mathbf{C})$, using the mean and covatiance from the previous image is multiplied with the probability map as shown in equation 3 to give new estimates for the mean and covariance.

$$\vec{\mu} = \begin{pmatrix} \mu_i \\ \mu_j \end{pmatrix} \qquad \mathbf{C} = 2 \begin{pmatrix} \sigma_i^2 & \sigma_{ij} \\ \sigma_{ij} & \sigma_j^2 \end{pmatrix} \qquad (3)$$

where:

$$S = \sum_{i\,j} P_{skin}(i, j))$$

$$\mu i = \frac{1}{S} \sum_{ij} P_{skin}(i, j) \cdot i \cdot g(i, j; \vec{\mu}, \boldsymbol{C})$$

$$\mu j = \frac{1}{S} \sum_{ij} P_{skin}(i, j) \cdot j \cdot g(i, j; \vec{\mu}, \boldsymbol{C})$$

$$\sigma_i^2 = \frac{1}{S} \sum_{ij} P_{skin}(i, j) \cdot (i - \mu_i)^2 \cdot g(i, j; \vec{\mu}, \boldsymbol{C})$$

$$\sigma_j^2 = \frac{1}{S} \sum_{ij} P_{skin}(i, j) \cdot (j - \mu_j)^2 \cdot g(i, j; \vec{\mu}, \mathbf{C})$$

$$\sigma_{ij} = \frac{1}{S} \sum_{ij} P_{skin}(i, j) \cdot (i - \mu_i) \cdot (j - \mu_j) \cdot g(i, j; \vec{\mu}, \mathbf{C})$$

The effect of multiplying new images with the Gaussian function is that other objects of the same color in the image (hands, arms, or another face) do not disturb the estimated position of the region being tracked. The factor of 2 in equation 5 offsets the tendency of the Gaussian to shrink.

3.2 Performance evaluation

Our robust tracking algorithm carries a somewhat higher computational cost than connected components of a thresholded image. This is illustrated with the computing times shown in figure 1. This figure shows the execution time for a full sized image on an SGI-02 for the robust tracker "COG" (center of gravity), connected components "CCO" and connected components assisted by a zeroth order Kalman filter. Average execution times are around 25 milliseconds per image for the connected components and 70 milliseconds for the robust algorithm.

Jitter is the number of pixels that the estimated position moves when the target is stationary. Jitter is the result of interference with illumination, electrical noise, shot noise, and digitizer noise. Algorithms which employ a threshold are especially sensitive to such noise. Table 1 illustrates the reduction in jitter for the robust tracker when compared to connected components.

Figure 2 compares the precision of tracking an object moving in the horizontal direction. All three trackers were applied to the same image sequence. The output of the color tracker using the connected components algorithm is shown with and without a Kalman filter. The Kalman filter eliminates position jitter but reduces global

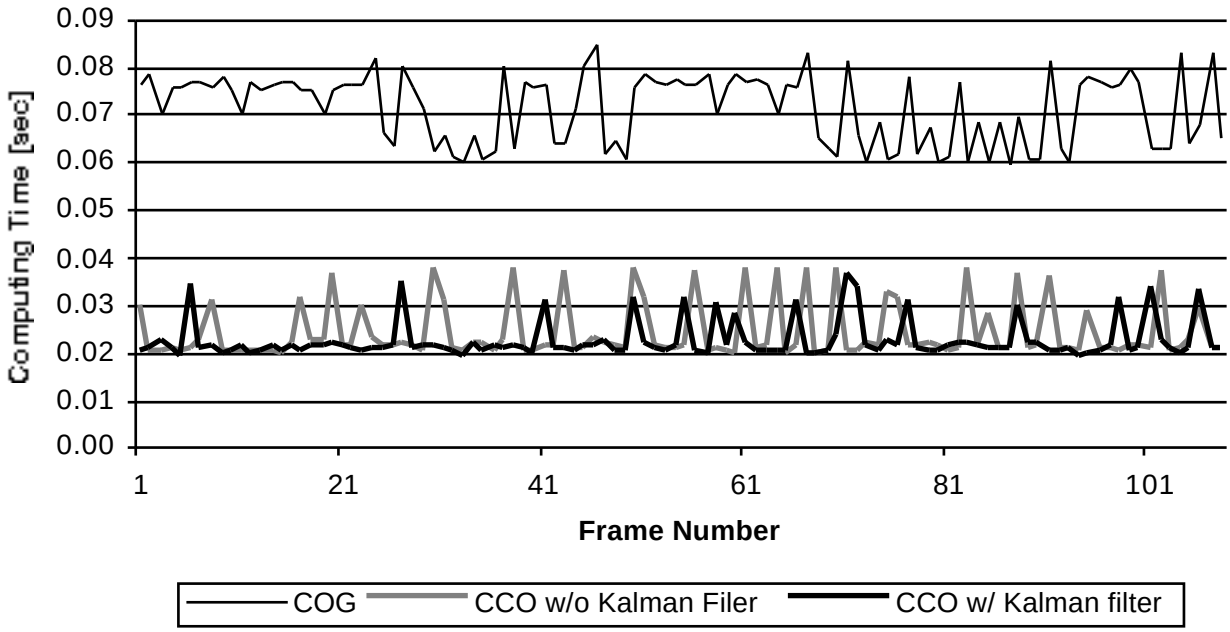

Figure 1 : computing speed of robust estimator (COG or center-of-gravity) , Connected Components (CCO w/o Kalman) and connected components algorithm assisted by a Kalman filter recursive estimator (CCO w Kalman)

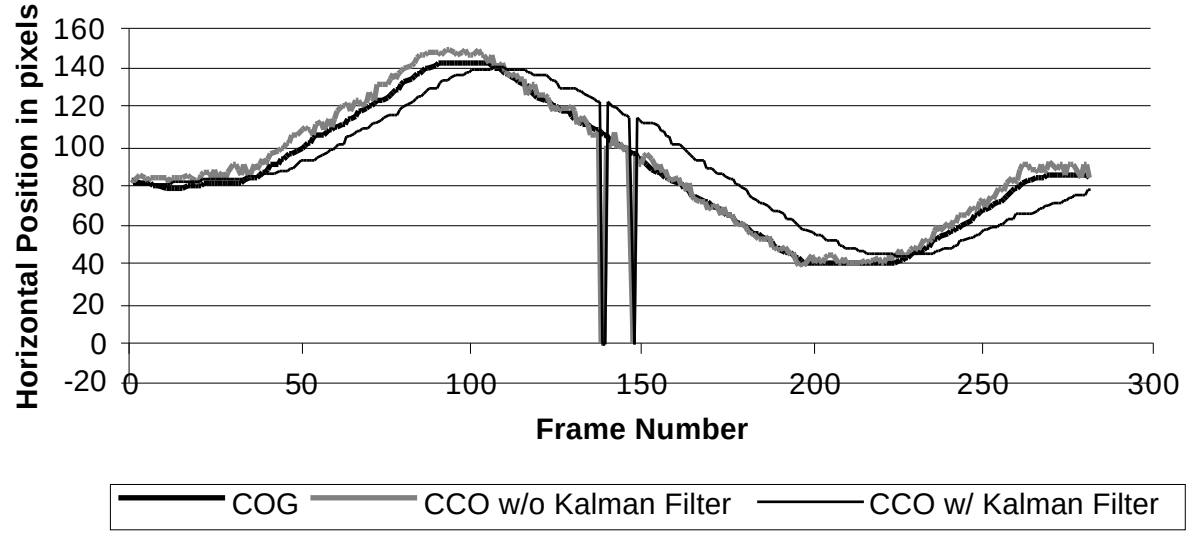

Figure 2 comparing tracking precision of a moving object

	Robust Algorithm	Connected Components without Kalman Filter	Connected Components with Kalman Filter
Jitter Energy	29	308	151

Table 1. Jitter energy measured for a stationary object by the robust estimator, and by connected components with and without a Kalman Filter

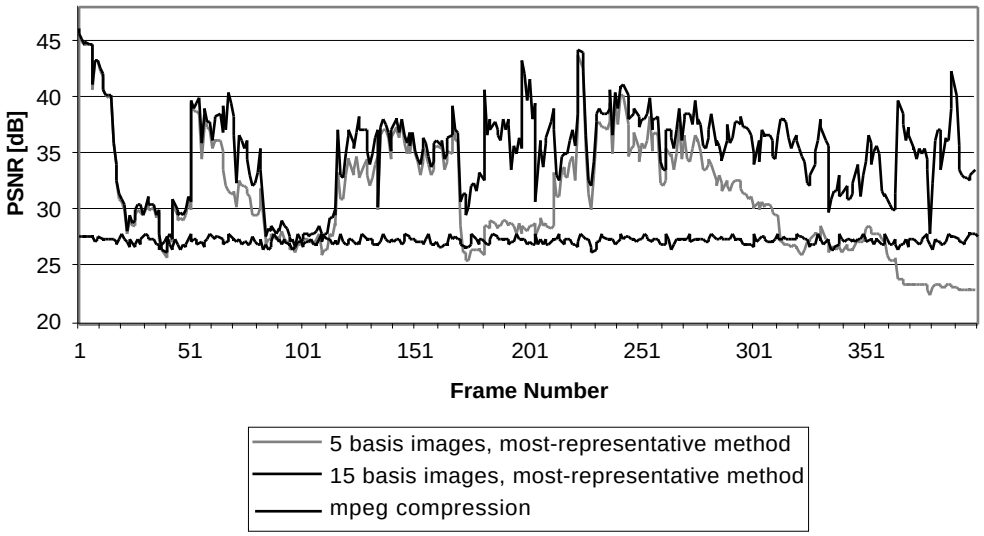

Figure 3 : PSNR vs. Number of basis images

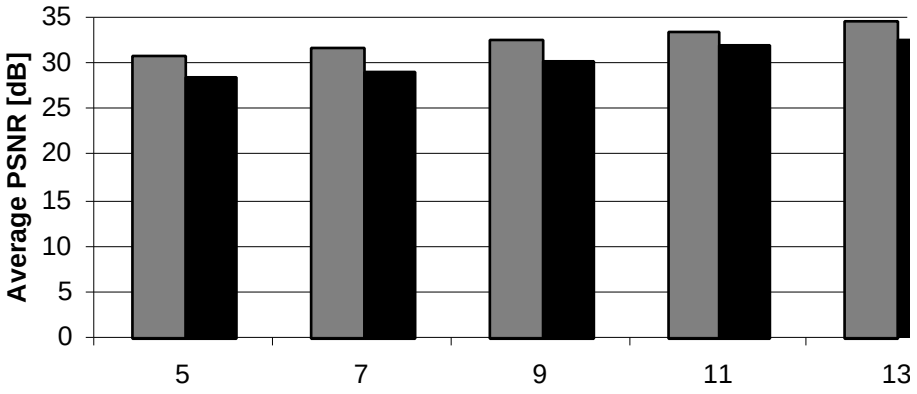

Figure 4 PSNR for each frame of BillS2

precision because of lag. The statistical color tracker has less jitter than the output of the Kalman filter and much less lag. The robust tracker also resisted noise that caused the connected components algorithm to fail at frames 138 and 147.

4 Orthogonal basis coding for video communications

4.1 Overview

There are currently four video coding standards used for commercial systems: The ITU-T recommendations: H.261 [7] and H.263 [8], plus the ISO standards 11172 [1] (MPEG-1) and 13818 [6] (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. More recent developments such as MPEG-4 and MPEG-7 target the integration of multimedia services and use MPEG-1 and MPEG-2 for video/audio-coding (MP3 is just the audio layer of MPEG-1). Rejecting a model-based approach for further research into video compression, we are investigating an approach based on projecting images into an orthogonal basis space.

Our algorithm for Orthonormal Basis Coding has been initially developed for off-line coding applications, such as video electronic mail. An incremental version is being evaluated for on-line coding. A stabilized video sequence is cropped in order to provide a sequence of images with the face normalized and centered in each image. Selected

frames from the sequence are used to create a *basis space* into which new images can be mapped. Each mapped image is 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.

4.2 Integrating face tracking and video coding

MPEG-1, H.261, H.263, and MPEG-2, are the three step compression schemes: 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 (OBC) 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. 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, Principal Component Analysis cannot be computed using every frame from a sequence of several minutes of video in a reasonable time. 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 sample selection 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 compares it by computing an inner product with all the other images in the sequence. All images that are 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 compared 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 likely 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.

The algorithm can be made sensitive to movements of the eyes and mouth by multiplying these regions by an extra weighting factor during the comparison of images with the to-do set. Thus variations in eye and mouth configurations receive a better represented in the selected sample set.

5 Performance evaluation

We use the Peak-Signal-to-Noise Ratio (PSNR) as an evaluation criteria for the reconstruction of images. The PSNR of the *k*-th frame is defined as

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

where *M* and *N* are the width and height of the image, *m* and *n* the pixel indices, *f()* the original pixel values, and $\hat{f}()$ the decoded pixel values.

5.1 Reconstruction quality

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 [1], 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 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 problem is rectified by weighting the eye and mouth during the sample selection process as discussed above.

Figure 5: Frame 237 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.

5.2 Compression

Video Stream	Kbytes
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

Table 2. Comparison of file sizes for OBC and MPEG

The *BillS2* video clip contains 418 frames and lasts 69 seconds (6 FPS). The various file sizes are shown in Table 2. 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 [11] on the OBC to do simple compression (redundancy elimination) on the file. Compression of the basis images using a common compression algorithm such as JPEG would significantly reduce file size and increase the compression rate.

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

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 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 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.

Acknowledgement

This work has been sponsored by the EC DG XII Human Capital and Mobility Network SMART II.

Bibliography

[1] “Berkeley MPEG Research”, http://bmrc.berkeley.edu/projects/mpeg, 1997

[2] Crowley, J.L., ”Integration and control of reactive visual processes”, in Robotics and Autonomous Systems, Vol. 19, No. 1, 1995, pp. 17-28

[3] Crowley, J.L., Berard, F., ”Multi-Modal Tracking of Faces for Video Communications”, IEEE: Computer Society Conference on Computer Vision and Pattern Recognition, Puerto Rico, June 17-19, 1997, ISBN 0-8186-7822-4, pp. 640-645

[4] Chomat, O. and J. L. Crowley, "Probabilistic Recognition of Activity using Local Appearance", 1999 IEEE: Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 99, Fort Collins, June 1999.

[5] Huang, T.S., Lopez, R., ”Computer Vision in Next Generation Image and Video Coding”, Lecture Notes in Computer Science, ISBN 0302-9743, No. 1035, 1996

[6] 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,” Geneva, http://www.iso.ch/cate/d22411.html, 1993

[7] ITU-T Study Group XV, *"Recommendation H.261:* Video codec for audiovisual services at p x 64 kbit/s", http://www.itu.ch, 03/1993

[8] ITU-T Study Group XV, *"Recommendation H.263:* Video coding for low bit rate communication", http://www.itu.ch, 03/1996.

[9] Kirby, M., Sirovich, L., "Application of the Karhunen-Loeve procedure for the characterization of human faces", 1990, *IEEE: Transactions on Pattern Analysis and Machine Intelligence* **12**(1), pp. 103-108.

[10] Schiele, B. and A. Waibul, "Gaze Tracking Based on Face Color", IWAGFR '95- International Workshop on Face and Gesture Recognition, Zurich. July 1995.

[11] Schwerdt, K., Crowley, J.L., "Contributions of computer vision to the coding of video sequences", *Proceedings of the 6th International Symposium on Intelligent Robotics Systems*, Edinburgh, U.K., July 21 - 23, 1998, pp. 281-289

[12] Swain, M. J. and D.H. Ballard, "Color Indexing", International Journal of Computer Vision, Vol 7, No 1, 1991.

[13] Turk, M., Pentland, A., "Eigenfaces for recognition", *Journal of Cognitive Neuroscience* **3**(1), 1991, pp. 71-86

[14] Vieux, W.E., Schwerdt, K., Crowley, J.L., "Face-tracking and coding for video compression", *ICVS* , Las Palmas (Gran Canaria), Spain, January , 1999.