Robust Face Tracking using Color

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

Abstract

In this paper we discuss a new robust tracking technique applied to histograms of intensity normalized color. This technique supports a video codec based on orthonormal basis coding. Orthonormal basis coding can be very efficient when the images to be coded have been normalized in size and position. However, an imprecise tracking procedure can have a negative impact on the efficiency and the quality of reconstruction of this technique, since it may increase the size of the required basis space. The face tracking procedure described in this paper has certain advantages, such as greater stability, higher precision, and less jitter, over conventional tracking techniques using color histograms. In addition to those advantages, the features of the tracked object such as mean and variance are mathematically describable.

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 technique (OBC), in which the normalized face image is projected onto a space of basis images.

The focus of this paper, however, will be on the face tracking system, or, more precisely, on the (skin) color tracking module. The OBC components have been introduced in [1, 2], and we will give here a rather short coverage and refer the reader to forthcoming papers for newer results and developments.

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.

The subsequent 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 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.

2 Multi-modal 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 which 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.

Figure 1. Probability images for the Connected Components Algorithm (left) and the Center of Gravity Algorithm (right).

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 correlation mask for the eyes, and a color histogram of skin are initialized, either by hand (mouse-click), or by an eye blink detection module. 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.

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. Figure 1 shows comparative images for both, the thresholded probability image (connected components algorithm) and the robust estimator (center of gravity algorithm).

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 PID-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 [4]. Schiele and Waibel showed that for face detection, color RGB triples can be divided by the luminance to remove the effects of relative illumination direction [5]. In earlier work [3] we described an algorithm in which a histogram of normalized skin color was initialized 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 (see Figure 1, left image). Since that time, we have reformulated the skin detection and tracking process using an approach inspired by robust statistics.

3.1 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 of the same dimensions, $h_{total}(r,g)$, 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 $h_{total}(r,g)$

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

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

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

$$p(skin) \approx \frac{N_{skin}}{N_{total}} \tag{2}$$

The probability of observing a color vector is given by :

$$p(r,g) \approx \frac{1}{N_{total}} h_{total}(r,g) \tag{3}$$

Bayes rule states that the probability of skin given a color vector (r, g) is

$$p(skin|r,g) = \frac{p(r,g|skin) \cdot p(skin)}{p(r,g)} \tag{4}$$

This reduces to the ratio of the two histograms as shown in equation 5.

$$p(skin|r,g) \approx h_{ratio}(r,g) = \frac{h_{skin}(r,g)}{h_{total}(r,g)} \tag{5}$$

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 5 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 models.

In order to detect a skin color region we must group skin pixels into a region. 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).

$$P_{skin}(i,j) = p(skin|r(i,j), g(i,j)) \tag{6}$$

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

$$\vec{\mu} = \begin{bmatrix} \mu_i \\ \mu_j \end{bmatrix} \qquad \mathbf{C}(i,j) = \begin{bmatrix} \sigma_i^2 & \sigma_{ij} \\ \sigma_{ij} & \sigma_j^2 \end{bmatrix} \tag{7}$$

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 initialized, 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 covariance from the previous image is multiplied with the probability map as shown in equation 8 to give new estimates for the mean and covariance.

$$\begin{aligned}
\mu_i &= \frac{1}{S}\sum_{i,j} P_{skin}(i,j) \cdot i \cdot g(i,j,\vec{\mu},\mathbf{C}) \\
\mu_j &= \frac{1}{S}\sum_{i,j} P_{skin}(i,j) \cdot j \cdot g(i,j,\vec{\mu},\mathbf{C}) \\
\sigma_i^2 &= \frac{1}{S}\sum_{i,j} P_{skin}(i,j) \cdot (i-\mu_i)^2 \cdot g(i,j,\vec{\mu},\mathbf{C}) \\
\sigma_j^2 &= \frac{1}{S}\sum_{i,j} P_{skin}(i,j) \cdot (j-\mu_j)^2 \cdot g(i,j,\vec{\mu},\mathbf{C}) \\
\sigma_{ij} &= \frac{1}{S}\sum_{i,j} P_{skin}(i,j) \cdot (i-\mu_i)(j-\mu_j) \cdot g(\cdot)
\end{aligned} \tag{8}$$

where $S = \sum_{i,j} P_{skin}(i,j) \cdot g(i,j,\vec{\mu},\mathbf{C})$. 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.

3.2 Behavior discussion

The use of a Gaussian weighting function for new input data actually can lead to a problem, if the object being tracked moves above a certain speed. Let us therefore consider the dynamic behavior of the color tracker. Since the two-dimensional case is largely more complex, we present here the one-dimensional case which actually suffices to justify our compensation measures.

3.2.1 One-dimensional case

Let $g(x,\mu_1,\sigma_1)$ be the *probability density function (pdf)* of our weighting function, and $g(x,\mu_2,\sigma_2)$ the *pdf* of the new input data (image). The resulting, i.e., effectively detected *pdf* is then product of both functions:

$$g(x,\mu_1,\sigma_1) \cdot g(x,\mu_2,\sigma_2) = \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{1}{2}\left(\frac{x-\mu_1}{\sigma_1}\right)^2\right]$$

$$\cdot \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{1}{2}\left(\frac{x-\mu_2}{\sigma_2}\right)^2\right] = \frac{A}{\sqrt{2\pi}\sigma_{new}} \exp -\frac{1}{2}\left(\frac{x-\mu_2}{\sigma_{new}}\right)^2, \quad (9)$$

where

$$A = \frac{\sigma_1\sigma_2}{\sqrt{2\pi}(\sigma_1^2+\sigma_2^2)} \exp\left[-\frac{1}{2}\frac{(\mu_1-\mu_2)^2}{(\sigma_1^2+\sigma_2^2)}\right] \quad (10)$$

$$\mu_{new} = \frac{\mu_1\sigma_2^2+\mu_2\sigma_1^2}{(\sigma_1^2+\sigma_2^2)} \quad (11)$$

$$\sigma_{new}^2 = \frac{\sigma_2^2\sigma_1^2}{\sigma_1^2+\sigma_2^2} \quad (12)$$

Equation 9 is a function with the shape of a Gaussian function which integrates from $-\infty$ to $+\infty$ to A.

Let $\sigma_1 = \sigma_2 = \sigma_0$, then

$$\sigma_{new}^2 = \frac{\sigma_2^2\sigma_1^2}{\sigma_1^2+\sigma_2^2} = \frac{\sigma_0^2\sigma_0^2}{2\sigma_0^2} = \frac{\sigma_0^2}{2} \quad (13)$$

That is, if we assume the distribution function of the tracked object – even if it does not move – to be approximately the same for subsequent images, the combined *pdf* of weighting and distribution function will shrink with each cycle, if no measure of compensation is taken. Equation 13 suggests a compensation factor of 2.

3.2.2 Motion compensation

If the tracked object moves, then the center of the combined *pdf* will lie between the center of the weighting function and the center of the *pdf* of the new input data. Equation 11 shows that the relation σ_1/σ_2 determines if the new center is closer to the center of the weighting *pdf* or to the center of the new input *pdf*. For $\sigma_1 \approx \sigma_2 = \sigma_0$ (a likely case for a sufficiently high frequency), the combined center will lie in the middle of both distribution functions.

Depending on the speed of the tracked object, equations 10 and 11 can cause the combined *pdf* to vanish, and – since we use quantized values for the probability map of our tracked object – the tracker to break.

First experiments with compensation measures suggest a Kalman filter update function

$$C_{new} = C + \Delta T^2 \cdot \begin{bmatrix} v_{i,obj}^2 & 0 \\ 0 & v_{j,obj}^2 \end{bmatrix}, \quad (14)$$

where ΔT is the time elapsed since the last frame was processed. Obviously, $v_{i,obj} = \Delta\mu_i \cdot \Delta T$ and $v_{j,obj} = \Delta\mu_j \cdot \Delta T$, with $\Delta\mu_i = \mu_{i,1} - \mu_{i,2}$ and $\Delta\mu_j = \mu_{j,1} - \mu_{j,2}$. Equation 14 compensates uncertainty due to accelerations and object movements. Combining the results of equations 13 and 14 we get as the covariance matrix of the Gaussian weighting function for new incoming data:

$$\mathbf{C}'(i,j) = \begin{bmatrix} 2(\sigma_i^2+\Delta\mu_i^2) & \sigma_{ij} \\ \sigma_{ij} & 2(\sigma_j^2+\Delta\mu_j^2) \end{bmatrix} \quad (15)$$

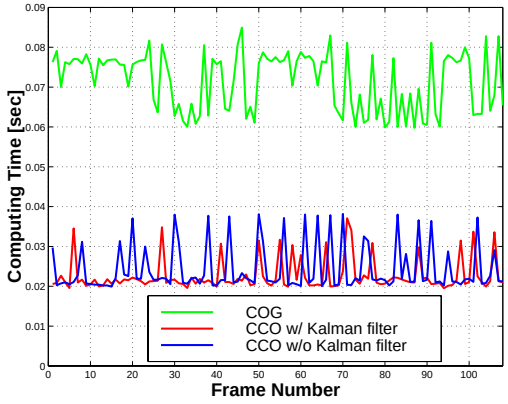

Figure 2. Computing time per image for the robust estimator (COG or Center of Gravity), Connected Components Algorithm without (CCO w/o Kalman) and with (CCO w/ Kalman) assistance from a Kalman filter recursive estimator.

3.3 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 2. This figure shows the execution time for a QCIF [6] sized image on a SGI 02 workstation 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 3 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 Kalman filter. The Kalman filter eliminates position jitter but reduces precision of global position estimation.

	COG	CCO w/o KF	CCO w/ KF
Jitter Energy	29	308	151

CCO : Connected Components Algorithm
COG : Robust Algorithm
KF : Kalman Filter

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

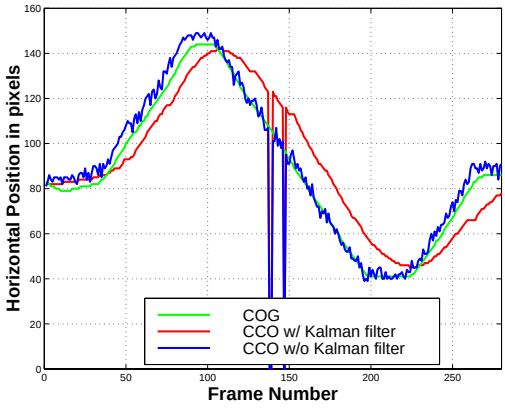

Figure 3. Comparing tracking precision of a moving object.

4 Orthogonal basis coding for video communications

For a more thorough discussion of this module we refer the reader to previous publications [1, 2]. Forthcoming publications will include newer developments and results such as an incremental version of the Orthonormal Basis Codec (OBC).

There are currently four video coding standards used for commercial systems, the ITU-T recommendations H.261 [6] and H.263 [7], plus the ISO standards 11172 [8] (MPEG-1) and 13818 [9] (MPEG-2). 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 [10], we are investigating an approach based on projecting images into an orthogonal basis space.

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.

4.1 Integrating face tracking and video coding

Due to processing constraints, Principal Components Analysis cannot be computed using every frame from a sequence of several minutes of video in a reasonable time. An algorithm, called *Most-representative Algorithm* attempts to find similar images anywhere in the sequence to be encoded. The OBC compression scheme operates as follows:

1. choose a limited set of images from the sequence to form the basis (*Most-representative Algorithm*),
2. energy compaction by Karhunen-Loeve expansion to generate an orthonormal basis space from the images,
3. entropy reduction by mapping each image in the sequence into this basis space, resulting in a small set of coefficients,
4. redundancy reduction by LZW loss-less compression on the basis and, if sensible, the parameter vectors.

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.

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 representation in the selected sample set.

4.2 Performance evaluation

Using the standard compression options with the Berkeley mpeg_encode program [11], we found an average PSNR of approximately 27 dB for the BillS2 [1] sequence. The MPEG reconstruction errors, see Figure 4c, were due to the typical blocking artifacts of DCT/DPCM [1] codecs. The reconstructed images from the OBC codec were slightly blurred, as can be seen in Figure 4b, which is the typical artifact for codecs using eigenspace representations. The closed mouth in Figure 4b is due to the fact that there were no images with a fully opened mouth among the fifteen basis images.

The BillS2 video clip contains 418 frames and lasts 69 seconds (6 fps). The various file sizes are shown in Table 2.

[1] MPEG video compression algorithms use the Discrete Cosine Transform (DCT) for energy compaction, and Differential Pulse Code Modulation (DPCM) as entropy reduction step.

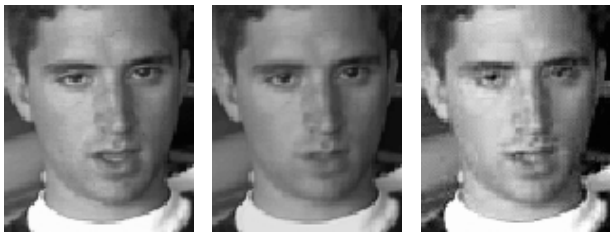

Figure 4. Example frame from test-sequence BillS2: a) Original Image, b) Reconstruction w/ 15 basis images, c) MPEG reconstruction.

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

Table 2. Comparison of file sizes (kB) for OBC and MPEG

We used the a LZW-based compression library on the OBC basis to do simple compression (redundancy elimination) on the file.

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.

5 Conclusions

Various techniques from computer vision have been used to create a fast and robust face tracking system, which in turn was used to build an efficient video codec based on orthonormal basis coding (OBC). The face tracker also enhances the usability of a 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 provides a positive outlook on further development of low-bandwidth video communication.

The reader is referred to future publications for more detailed information on those topics.

Acknowledgement

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

References

[1] W. E. Vieux, K. Schwerdt, and J. Crowley, "Face-tracking and coding for video-compression," in *First International Conference on Computer Vision Systems*, (Las Palmas, Spain), January 1999.

[2] J. L. Crowley and K. Schwerdt, "Robust tracking and compression for video communication," *IEEE Computer Society Conference on Computer Vision, Workshop on Face and Gesture Recognition*, September 1999.

[3] 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.

[4] M. J. Swain and D. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, 1991.

[5] B. Schiele and A. Waibel, "Gaze tracking based on face color," *International Workshop on Automatic Face- and Gesture-Recognition*, June 1995.

[6] ITU-T Study Group XV, "*Recommendation H.261*: Video codec for audiovisual services at px64 kbit/s," tech. rep., ITU-T, Geneva, Switzerland, "http://www.itu.ch", 1993.

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

[8] ISO/IEC 11172-2, "Coding of moving pictures and associated audio – part 2: Video," tech. rep., ISO, Geneva, Switzerland, "http://www.iso.ch/cate/d22411.html", 1993.

[9] ISO/IEC 13818-2, "Generic coding of moving pictures and associated audio information: Video," tech. rep., ISO, Geneva, Switzerland, "http://www.iso.ch/cate/d22990.html", 1996.

[10] 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.

[11] B. M. R. Center, "Berkeley mpeg tools," *http://bmrc.berkeley.edu/projects/mpeg*, 1997.