Very low bit rate wireless video telephony using face coding

Mikael Persson

August 2003

Supervisors:
Henrik Storm, Summus, Inc. (USA).
Dr. Fredrik Kahl, Mathematical Imaging Group, Centre for Mathematical Sciences, Lund Institute of Technology.
Abstract
In this thesis a novel video telephony compression scheme is proposed, implemented and discussed. The scheme generates a talking head sequence from a head and shoulder video telephony sequence. The generated talking head mimics the facial expressions of the individual depicted in the head and shoulder input sequence. The scheme is based on model based coding and more specifically based on an eigenspace approach. The model which is used to represent the objects to be encoded is statistically derived as the principal components of a training sequence depicting the individual performing a wide range of facial expressions. The thesis introduces the concept of eigenfeatures as used in video compression and a method for encoding the facial expressions of the talking head as a number of coefficients defining a linear combination of the eigenfeatures. Using the proposed scheme acceptable video telephony can be achieved at data rates as low as 3-4 kBit/s.
Acknowledgements
First of all, my thanks go to my supervisors, Henrik Storm, for inspiration, ideas and support and Fredrik Kahl for ideas and support especially regarding the mathematically intensive parts. Also thanks to Johan Strömbeck and Karl-Anders Johansson for their endless stream of mostly good ideas.

Especially thanks to Summus, Inc. (USA), Raleigh, North Carolina, for funding this project. Finally, thanks to all my colleagues at TAT Movide AB where most of my work has been performed.
Contents

1 INTRODUCTION ................................................................................................................................. 1
  1.1 Motivation .................................................................................................................................. 1
  1.2 Background ............................................................................................................................. 1
  1.3 Limitations ............................................................................................................................... 2
  1.4 Problem formulation ............................................................................................................... 3
  1.5 Overview of the thesis .............................................................................................................. 3

2 METHOD ........................................................................................................................................... 4
  2.1 Outline ..................................................................................................................................... 4
    2.1.1 Model based coding .......................................................................................................... 4
    2.1.2 Encoder .......................................................................................................................... 5
    2.1.3 Decoder ......................................................................................................................... 6
  2.2 Eigenspace based compression ................................................................................................. 7
    2.2.1 Principal component analysis ......................................................................................... 9
    2.2.2 Estimating the correlation matrix ................................................................................. 11
    2.2.3 Singular Value Decomposition ................................................................................. 12
    2.2.4 Representing arbitrary images in eigenspace ............................................................. 15
    2.2.5 Eigenfaces and Eigenfeatures ..................................................................................... 18
    2.2.6 Summary ...................................................................................................................... 21
  2.3 Face detection and tracking ..................................................................................................... 22
    2.3.1 Overview ...................................................................................................................... 22
    2.3.2 Skin color detection ...................................................................................................... 23
    2.3.3 Integral image ................................................................................................................. 27
    2.3.4 Fast facial region estimation .......................................................................................... 28
    2.3.5 Facial region tracking ................................................................................................... 32
    2.3.6 Summary ...................................................................................................................... 33
  2.4 Eigenfeature alignment ............................................................................................................. 34
    2.4.1 Motivation ....................................................................................................................... 34
2.4.2 Eigenfeature importance region ................................................................. 35
2.4.3 Candidate displacements estimation ...................................................... 36
2.4.4 Reconstruction error minimization ......................................................... 39
2.4.5 Candidate mouth displacements ........................................................... 41
2.4.6 Candidate displacements feedback ....................................................... 42
2.4.7 Summary ............................................................................................... 43

2.5 Coefficient encoding .............................................................. ........................................ 44
2.5.1 Differential Pulse Code Modulation .................................................. 44
2.5.2 Optimal predictor .................................................................................. 45
2.5.3 Efficient quantization ........................................................................... 46
2.5.4 Summary ............................................................................................... 48

3 RESULTS AND DISCUSSION ......................................................... 49

3.1 Implementation ......................................................................................... 49

3.2 Results ................................................................................................. 51
3.2.1 Reconstruction quality .......................................................................... 52
3.2.2 Data rate .............................................................................................. 55

3.3 Related work ........................................................................................ 56
3.3.1 Facial animation in MPEG-4 ............................................................... 56
3.3.2 Model based coding with CANDIDE .................................................. 57
3.3.3 Appearance-based video compression ............................................... 58

4 CONCLUSIONS .............................................................. 60

4.1 Contributions of the thesis ................................................................. 61

4.2 Future work ......................................................................................... 62

References .................................................................................................. 63

Table of figures ........................................................................................... 64

Color plates .................................................................................................. 67
Chapter 1 Introduction

1 Introduction

1.1 Motivation

Video communication is an increasingly important service in current fixed and wireless networks. Transmitting video usually requires high data rates. For instance, the raw data rate of the PAL TV signal requires approximately 230 MBit/s, which is compressed to about 4 MBit/s for digital broadcast TV [1].

Although broadband channels are increasing in availability, new narrow bandwidth channels such as mobile telephony are surfacing. Even though the third generation mobile telephony is reaching data rates of 384 kBit/s, the use of very low data rate applications still exists. First of all, if a large number of simultaneous video communication channels are to be supported the bandwidth is easily consumed. Secondly, if the video communication service is to be used as a complement to for instance a game application, the bandwidth needs to be shared among the services.

Evidently there exist a number of uses for narrow bandwidth video communication. This thesis will focus on video telephony where the face of the user covers the main portion of the image. The reason for this is simply the fact that this is the most common case discussed in the area of video communications. The goal of this thesis is to investigate the possibility of transmitting video telephony sessions with as low data rates as 4 kBit/s using an eigenspace based compression scheme. A system achieving this is proposed, implemented and discussed.

1.2 Background

The efficient coding and representation of human faces is an important problem in the area of image compression for video telephony. Lately, methods have been presented [2] that allow the coding of face images down to 300 bits (approximately 40 bytes) using eigenspace techniques. In order to enable video telephony over narrow bandwidth channels specialized encoding schemes, also called model based coding schemes, targeted towards compression of human faces such as the eigenface method can be used.
This thesis is based on previous work in the area of compression using eigenspace methods specifically the work of Karl Schwerdt, *Appearance-based video compression*, [4]. In his thesis he describes the use of an orthonormal basis spanning the space of face images, where each image is interpreted as a column vector that is to be coded. The complete sequence to be coded is used to build the orthonormal basis, when this is performed each frame of the sequence can be described as a point in the face space. If the basis has fewer dimensions than an individual frame of the sequence compression is performed. Even though Schwerdt includes thoughts on creating an online system, the described compression scheme is basically for offline use.

The work of Torres et. al. *A proposal for high compression of faces in video sequences using adaptive eigenspaces* [2] describes a video compression scheme where the orthonormal basis used to code individual frames is continuously updated to enable online communication. Even though the system displays acceptable results the computational cost of continuously computing the face basis is too high for implementation on constrained environments such as mobile phones.

While Schwerdt described an offline and Torres et. al. a semi online video compression scheme, this thesis describes an online compression scheme using the concept of eigenspaces of face and face feature images. The basic concept is to map the frame to be coded into a previously built eigenspace with as small reconstruction error as possible.

### 1.3 Limitations

The main limitation of the proposed video compression scheme is the fact that only a specific type of objects can be encoded. In order to allow efficient compression the eigenspace needs to be localized to span only the space of a certain type of objects, namely images of faces. More specifically only images of faces of a certain individual, the user transmitting over the video telephony system, can be encoded.

The system described in this thesis is a frontal face system. This implies that the video telephony sequences that can be encoded contain frames where the main part of the image is occupied by the frontal face of the user. Expressions and facial motions will be mapped on a talking head representing the user.

Furthermore audio compression is outside of the scope for this thesis.
1.4 Problem formulation

The project consists of designing, implementing, evaluating and discussing a video telephony system, targeted at narrow bandwidth channels, using the eigenspace technique. The project focuses on developing techniques from the point of view of usability in next generation mobile units. The use of a profile where user related information such as eigenspaces and skin color distributions can be stored will be assumed. This enables optimization of the video telephony system with respect to the actual user. The project definition can be broken down into the following sub problems:

- Deriving a profile containing a model of the user transmitting over the video telephony system and user specific parameters to enable efficient and robust face tracking.
- Tracking the face of the user in the video telephony sequence.
- Deriving a set of coefficients describing the facial expressions of the user.
- Encoding and transmitting the coefficients.
- Reconstruction of the facial expression of the user using the model contained in the user profile.

1.5 Overview of the thesis

The report is divided into three main sections, introduction, method and finally results and discussion. The first section has covered an introduction to the problem as well as a problem formulation. The method section covers a detailed technical description of the sub problems stated above. Finally, the results and discussion section will present some results from an implementation of a system as described in the method section followed by a discussion of the advantages and disadvantages of an eigenspace based system comparison with other model based compression schemes.
2 Method

2.1 Outline

This section will give an outline of the proposed video telephony system. First a more thorough introduction to the concept model based coding is given. Finally, the encoder and decoder schemes proposed in this thesis are outlined. The sections below are intended to ensure that the context of the mathematical and technical descriptions of the remaining sections is clear.

2.1.1 Model based coding

Model based coding implies that a model of the objects to be encoded is derived and is known a priori at both the receiving and transmitting end. The model will contain a number of parameters that can be configured such that the model resembles the object to be encoded. In the context of image compression, if the number of parameters required to fully describe the image is smaller than the image raw data size, compression is achieved. Optimally, the model should be derived such that with as few parameters as possible the space of objects to be encoded is covered. In this context a model of a face (namely the face of the user) is to be built such that the model covers the possible facial expressions of the user.

There are a number of methods to build the face model. Many model based coding systems are based on a three-dimensional model of the object to be encoded where the parameters can be so called action units [1] used to describe how to move the vertices of the three-dimensional model in order for it to resemble the object to be encoded. These models are primarily built empirically. In this thesis the proposed model will be partially built statistically from a training set of images of the user performing the facial expressions to be covered. As the model is based on a set of images of the user it will be tailored to the actual user of the video telephony system. Figure 1 illustrates the described scheme.

The model consists of an empirically derived part, the face synthesis procedure, as well as a statistically derived part, the eigenfeatures. The eigenfeatures are derived from a set of training images using a method called principal component analysis, further described in Section 2.2 eigenspace based compression.
Chapter 2 Method

The user face model proposed in this thesis is composed of a number of so called eigenfeatures, a concept based on the more widely known concept of eigenfaces. The face is divided, as described in more detail in Section 2.2 eigenspace based compression, into three parts each containing one of the main face features, left eye, right eye and mouth. These eigenfeatures, as described later, can be linearly combined using a set of coefficients into a large number of facial expressions. The coefficients used to describe these expressions are the parameters of the model. Thus, in this thesis the model comprises the statistically derived eigenfeatures as well as the procedure of combining these using a set of coefficients into a specific facial expression. The face synthesis procedure and a detailed description of the eigenfeature concept are given in Section 2.2 eigenspace based compression.

2.1.2 Encoder

The main purpose of the encoder is to derive and encode the model parameters. This is performed in a number of steps, as illustrated in the encoder scheme in Figure 2. Firstly, the input video telephony frame (left in Figure 2) often not only contains the face of the user but it also contains background and probably some part of the shoulders. The face detection and tracking identifies the region of the images containing the face. This will simplify the next step in the encoder scheme namely to derive the model parameters from the facial region in the input frame.
Chapter 2 Method

Figure 2. The encoder scheme. The input frame is fed into the face tracker, when the face has been detected the parameters of the model can be derived. Finally, the parameters are encoded and transmitted to the receiver.

Finally when the model parameters have been derived, they have to be encoded into symbols that can be transmitted over the network to the decoder. The face detection and tracking is described in Section 2.3 face detection and tracking. The model parameter derivation is described in Section 2.4 eigenfeature alignment. Finally the model parameter encoding is described in Section 2.5 coefficient encoding.

2.1.3 Decoder

The main purpose of the decoder is to decode the model parameters and use them in conjunction with the prior known model in order to synthesize the facial expressions of the user. The decoder scheme is illustrated in Figure 3. The face synthesis procedure is described in Section 2.2 eignespace based compression and the model parameter decoding is described in Section 2.5 coefficient encoding.

Figure 3. The decoder scheme. The encoded model parameters are decoded. The decoded parameters are used with the prior known model to synthesize an image of the facial expression.
2.2 Eigenspace based compression

The most popular image compression schemes currently used are different types of transform coding. Basically, transform coding transforms the signal into a new base where large portions of the signal energy are focused on fewer elements. If only the elements with high energy are kept the signal can still be restored, by an inverse transform, to a certain level and requires less storage space, thus compression is achieved.

A common requirement of the transform coding approach is that the bases are signal-independent, i.e. they are not targeted for a specific class of images but should rather give acceptable performance on most images. When compressing a clearly defined class of objects, such as images of faces, the signal-independent approach is not the optimal choice. If the optimal transformation for a certain class of objects, in our case images of face, could be found it would result in greater compression capabilities. An optimal linear transform in a statistical sense will be presented and discussed.

If an image is represented as a vector where each element is the intensity of a specific pixel in the image, i.e. the image is column-stacked into a vector, all images of a certain dimension could be represented as points in a high-dimensional space. For instance an image of size 20x30 will be regarded as a point in $\mathbb{R}^{600}$. The space is called the image space. The class of all possible face images, or the “cloud” of face images in image space creates a data manifold called the face space. Figure 4 illustrates the concept of image and face space.

![Possible face space](image.png)

*Figure 4. An illustration of a possible image space, here illustrated as the space of two-pixel images, and the face space as a data manifold of the parent image space.*
The actual image space is of high dimension, since all possible images are covered, but the face space occupies only a small region of the full image space. If a number of base vectors in image space could be derived that effectively span the face space, less information is needed to describe the possible face images. In [4] a set of basis functions are created that are dependent on the input class, in that case images of faces. A number of face images are used as a training set in order to build the basis. The derived basis functions, called eigenfaces or the eigenspace of face images, can be combined linearly in order to create new face images within the space spanned by the basis. Thus if the basis functions are available only the coefficients are needed to represent a given face image.

In this thesis the type of object to be compressed is even more limited than the class of face images. In order to achieve high compression rates and better reconstruction the class of objects to be compressed is face images of a particular individual. Ideally, this implies that the class of objects will contain all facial expressions and motions performable by the individual. The set of training images used to build the basis functions will thus depict the user performing a number of facial expressions and motions representing the different sound building blocks of the language and emotions. Figure 5 illustrates a number of possible training images.

When all training images are collected these will be used to find the optimal basis functions spanning the face space. This basis can be found by eigenvalue analysis and it is
thus sometimes called the eigenspace. A popular method for finding the basis vectors is the principal component decomposition.

### 2.2.1 Principal component analysis

Principal component analysis (PCA) is a method for decomposing correlated data into uncorrelated data. Basically it is an automatic method for finding the optimal linear functions of the data with the property that the resulting data functions are uncorrelated. This transformation is performed such that a resulting base $B$ is orthogonal and thus the data $y$ is uncorrelated. The method involves finding the base $B$, an $N$ by $N$ matrix where $N$ is the dimension of the data, such that the correlated data $x$ is transformed into uncorrelated data $y$ according to:

$$x = \bar{x} + By$$

where $\bar{x}$ is the mean of the data $x$. The process to compute $B$ can be thought of as finding an orthogonal basis that covers the space of the data $x$ where the first basis vector accounts for the largest variation of the data. The second basis vector represents the second largest variation and is orthogonal with the first basis vector. Finally the last basis vector of $B$ accounts for the least variation of the data $x$ and is orthogonal with all previous basis vectors.

According to [5], the first step of computing the principal components of the data $x$ is to build the correlation matrix:

$$\Sigma = E\{(x - \bar{x})(x - \bar{x})^T\} \quad x \in \mathbb{R}^N.$$

Where $E\{}$ denotes the expected value of the argument and $\bar{x}$ is the mean value of $x$. In order to find the principal components of the vector $x$, the correlation matrix (or covariance matrix in case of a random variable, in this context these terms is interchangeable) needs to be diagonalised such that:

$$\Sigma = P\Lambda P^T,$$
where $\Lambda$ is a diagonal matrix containing the eigenvalues of the correlation matrix and $P$ is an orthogonal matrix containing the eigenvectors. Let the diagonal of $\Lambda$, i.e. the eigenvalues, be:

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N \geq 0.$$ 

The rows and columns of the diagonal matrix containing the eigenvalues can be rearranged such that the largest eigenvalue resides in the first row. Since covariance matrices are symmetric and positive-semidefinite, such a diagonalisation always exists, with all eigenvalues non-negative. The $k$:th principal component is then expressed $P_k^T$, where $P_k$ is the eigenvector in the orthogonal matrix $P$ corresponding to the largest eigenvalue. The variance of a principal component can be computed according to [5]:

$$V\{P_k^T x\} = P_k^T (PAP^T)P_k = \lambda_k.$$ 

Also the fact that $P$ is an orthogonal matrix implies that the principal components are uncorrelated:

$$C(P_j^T x, P_k^T x) = 0 \quad \forall j \neq k.$$ 

Above no distinction is made between random variables and observations of random variables. The transformation onto the principal component axes can be performed according to below:

$$y = P(x - \bar{x}).$$ 

And thus the original data $x$ can be restored using the expression below:

$$x = \bar{x} + P^T y.$$ 

The fact that the principal components are ordered of descending variation as described above will be used to enable compression expressed as:
\begin{equation}
x \approx \bar{x} + \hat{P}_k^T \hat{y} \quad \hat{y} \in R^K,
\end{equation}

where $\hat{P}_k^T$ contains the first $K$ eigenvectors of $P$ and $\hat{y}$ contains the $K$ coefficients, i.e. the parameters of the model as described in Section 2.1.1 \textit{model based coding}. In order to extract the principal components using eigenvalue analysis as described, the correlation matrix must be estimated.

\subsection*{2.2.2 Estimating the correlation matrix}

Given $n$ observations of an $N$-dimensional variable the correlation matrix (or covariance matrix in case of a random variable) can be estimated. According to \cite{5} the maximum likelihood estimation, assuming a normal distribution, can be computed as expressed below, by first introducing:

\begin{equation}
\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i, \quad y_i = x_i - \bar{x}.
\end{equation}

The final estimate is computed as follows:

\begin{equation}
\Sigma = \sum_{i=1}^{n} y_i y_i^T.
\end{equation}

If the data is packed into an $n$ by $N$ matrix:

\begin{equation}
Y = \begin{bmatrix}
y_{1,1} & \cdots & y_{n,1} \\
\vdots & \ddots & \vdots \\
y_{1,N} & \cdots & y_{n,N}
\end{bmatrix},
\end{equation}

the correlation matrix can be expressed as:

\begin{equation}
\Sigma = YY^T.
\end{equation}
2.2 Eigenspace based compression

The covariance matrix is of the dimensions $N$ by $N$ where $N$ is the dimension of a single observation. In our case an observation is a column-stacked image, and thus the image resolution decides the dimensions of the correlation matrix. At normal video telephony resolutions the resulting correlation matrix is simply too large and thus very memory and computationally consuming. Since only a limited number of principal components are of interest, computing the complete set wastes precious processing power.

### 2.2.3 Singular Value Decomposition

The connection between principal component analysis and singular value decomposition can be exploited to reduce computational complexity. This is a fact since there exists efficient methods for computing the eigenvectors using singular value decomposition without the need to compute the correlation matrix \[12\].

All observations can be composed into a matrix where each column represents one observation with the average observation subtracted. This was also performed in the previous section where the correlation matrix was expressed as:

\[
\Sigma = YY^T.
\]

As explained above the principal components, referred to as $P$, can be found by computing the eigenvectors of the correlation matrix for the observations according to below where $\Lambda$ contains the eigenvalues:

\[
\Sigma = P \Lambda P^T \quad \Leftrightarrow \quad \Sigma P = \Lambda P.
\]

Since the correlation matrix for the training image sequence is expensive to compute both regarding processing power and memory and since only a limited number of principal components will be used, the singular value decomposition is favorable. The singular value decomposition will decompose a matrix $X$ into the $U$, $S$ and $V$ matrices according to below, where $U$ and $V$ are orthogonal and unary i.e. $U^T U = I$ and $V^T V = I$.

\[
X = USV^T.
\]

The $U$ matrix is in fact the eigenvectors of the correlation matrix of $X$ as shown below:
Chapter 2 Method

$$XX^T = (USV^T)(USV^T)^T = USV^TVS^TU^T = USS^TU^T.$$ Thus,

$$XX^TU = USS^TU^TU = USS^T,$$

in other words the $U$ matrix contains eigenvectors of the correlation matrix and $SS^T$ contains eigenvalues. Another important fact is that the $U$ matrix contains orthogonal principal components, thus uncorrelated, and ordered after descending energy. If this procedure is performed using the correlation matrix of the observations $Y$ the following is obtained:

$$\Sigma = USV^T \Rightarrow \Sigma U = U\Lambda \Rightarrow U = P.$$ Singular value decomposition can thus efficiently be used to build the eigenspace basis. Figure 6, presents the first 16 base vectors spanning the eigenspace generated using singular value decomposition and the training sequence from Figure 5. In order to completely represent any frame from the original training sequence all basis vectors, i.e. eigenvectors of the correlation matrix, are required. However this results in the same amount of basis vectors as dimensions of a frame and thus does not result in any compression. The fact that the base vectors are ordered after descending variance is useful, since only a limited number of basis vectors are required to reconstruct a large number of different face images reasonably well.
Chapter 2 Method

Figure 6. The first 16 basis images spanning the eigenspace of face images. The basis vectors are generated using singular value decomposition.

In order with the discussed theory, Figure 7 displays the variances of the 64 first basis vectors generated from the training sequence of Figure 5. As can be seen the variances drops quickly and thus reasonable reconstruction can be performed with as little as about 20 base vectors as shown in Section 2.2.1 principal component analysis.
2.2.4 Representing arbitrary images in eigenspace

As previously mentioned, new face images can be constructed by linearly combining the eigenspace base vectors, i.e. the eigenfaces. An arbitrary image within the eigenspace can be expressed as follows:

\[ I = \sum_{i=1}^{n} c_i \phi_i. \]

Where \( I \) is the expressed image, \( c_i \) is the coefficient for basis vector \( i \) and \( \phi_i \) is the \( i \):th basis vector, here the average observation \( \bar{x} \) is included in the matrix \( \phi \) as the first eigenvector. Depending on the number of eigenfaces used, \( n \), more or less different face images can be expressed. The linear combination of eigenfaces can be expressed in matrix form as follows:

\[ I = \phi c, \]

where \( \phi \) is a matrix with the eigenfaces as column vectors and \( c \) is a row vector with the coefficients for each eigenface respectively. If a given original face image \( I_o \) is to be

\[ \text{Figure 7. The variances of the 64 first basis vectors. Generated from the training sequence of Figure 5.} \]
represented as a linear combination of eigenfaces the coefficients, \( c \), that minimize the following expression must be found:

\[
\min_c \| I_o - \phi c \|^2 = \min_c \left[ (I_o - \phi c)^T (I_o - \phi c) \right]
\]

in other words the mean square error between the original face image and the reconstruction. Expanding and differentiating the expression above with regards to \( c \) one can obtain the minimizing parameters:

\[
\frac{\partial}{\partial c} (I_o - \phi c)^T (I_o - \phi c) = \\
\frac{\partial}{\partial c} \left[ I_o^T I_o - I_o^T \phi c - c^T \phi^T I_o + c^T c \right] = \\
- I_o^T \phi - \phi^T I_o + 2c.
\]

Equating the derivative to zero results in a linear equation system that when solved results in:

\[
c = \phi^T I_o.
\]

Thus if an arbitrary image is to be encoded as a number of coefficients that image, interpreted as a vector, must be projected onto the base vectors of the eigenspace. More specific a coefficient, \( c_i \), can be derived as stated below:

\[
c_i = \sum_j I_j \phi_{i,j},
\]

where \( I_j \) is the \( j \):th pixel intensity element of the image \( I \) and \( \phi_{i,j} \) is the \( j \):th element of the \( i \):th base vector. Thus an arbitrary image can be encoded into a number of coefficients and reconstructed. Of course only images inside the eigenspace is encoded and reconstructed with acceptable quality. Figure 8 illustrates a number of images, not belonging to the training sequence, projected into the eigenspace and reconstructed. Notice the face with closed eyes, since none of the training images contained a face with closed eyes that property is not part of the eigenspace and thus such faces cannot be reconstructed.
The eigenfaces built from the training sequence and presented in Figure 5 is used to reconstruct the images. Thus an image can be encoded into coefficients using $n$ multiplications and $(n+1)$ additions for each eigenface, $n$ being the dimension of the image, interpreted as a vector, to be encoded. The image can be reconstructed again using $n$ additional multiplications per eigenface.
2.2.5 Eigenfaces and Eigenfeatures

The training images used to build the eigenspace should, as previously described, estimate the extent of the face manifold. This manifold or the true face manifold will not be convex, this is easy to see; Let $A$ be a face image represented as a vector with its eyes open and let $B$ be a face with its eyes closed, both $A$ and $B$ should be contained within the face space. Now if the face space were convex, all convex combinations, $C$, would be inside the face space:

$$C = \alpha A + (1 - \alpha)B \quad \alpha \in [0,1]$$

This is however not true since $\alpha = 0.5$ results in a face with transparent eyelids, a combination of a face with its eyes closed and with its eyes open. Hence, the face space is not convex. This implies that using a set of orthogonal basis functions and reconstructing faces by linear combinations of these, as described in the previous section, one will be able to represent faces not part of the face space [6]. Figure 9 illustrates the concept of convex combinations of face images and the fact that the face space is not convex.

![Figure 9](image)

Figure 9. Left, an image from the face space with eyes closed. Right, an image from the face space with its eyes open. Middle, a convex combination of the left and right image resulting in a face with transparent eyelids not within face space.

It is reasonable to assume that the singular value decomposition will produce better results for convex manifolds, since all points within the manifold is to be represented as linear combinations of the resulting basis vectors.

Another quirk is the fact that the amount of local spatial variance, in the context of principal component analysis, is based upon variation in pixel intensities in that region. This implies that motion in a large area, such as the region around the mouth, results in high variance but subtle motions such as the eyebrows and eyes results in low variance.
Chapter 2 Method

Figure 10 illustrates this by presenting the variance at each pixel in the images from the training sequence. It is obvious that the area surrounding the mouth represents the largest variance and that the region around the eyes has noticeable lower variance.

![Figure 10. Left, the variances at each pixel in the images from the training sequence color coded. Right, the variances at each pixel in the images from the training sequence as a three-dimensional plot. See Color plate 1.](image)

The behavior and prioritizing of the principal component analysis is of course correct from a mathematical and statistical point of view, but not to a human observer. The regions around the eyes are vital for correctly representing facial expressions. Simply being able to communicate with the mouth is actually not feasible. Thus with the experiences illustrated in Figure 10 the motions of the mouth will be clearly represented in the first principal components as these will capture the largest variances, but in order to represent subtle eye movements a large number of principal components are required. And even so the mouth will receive much more attention than the eye regions.

In order to improve the issues of face space convexity and of pixel variance distribution, and thus improve performance of the compression scheme the term *eigenfeature* is introduced. These are simply eigenspaces for each face feature, left eye, right eye and mouth. Thus the face is divided into three regions, each with focus on one face feature, as illustrated in Figure 11.
In each region, the dominant variance is located at the targeted face feature, thus the large variance of the mouth will not interfere when generating the eigenspaces for the eyes. Each face feature will require its own eigenspace, also called eigenfeatures. This will result in the three eigenfeatures $\phi_{re}$, $\phi_{le}$ and $\phi_{m}$, for right eye, left eye and mouth respectively. Thus each of the eigenfeatures only has support in one face region.

Thus face images can be reconstructed from three distinct sets of coefficients, $c_{re}$, $c_{le}$ and $c_{m}$, as expressed below:

$$I_r = \sum_j c_{re,j}\phi_{re,j} + \sum_j c_{le,j}\phi_{le,j} + \sum_j c_{m,j}\phi_{m,j}$$

where $I_r$ is the reconstructed image. The work presented in this thesis has empirically shown that a higher objective reconstruction quality can be achieved by separating the face image into face feature regions. The work done by Ström et. al. [6] also presents similar results, but in their report the face is divided into a grid of 32 by 32 pixel regions. The purpose of the described method is to empirically divide the face into regions based on the estimated pixel variances of Figure 10.

Also as motivated above the fact that the high dimensional face space is divided into three face feature spaces with lower dimension, each more likely to occupy a smaller manifold than the complete face space increases the effectiveness of the principal component analysis. In other words the produced face feature principal components are compared to using the entire face less likely to cover large regions of non-relevant space.
2.2.6 Summary

A motivation of why and a description of how to derive a transform coding method tailored for a specific class of objects, in this context images of faces, have been presented. The transform is derived based on the statistical properties of the class of objects, namely by analyzing a set of training objects and deriving the basis for the transform through principal component analysis.

Since the principal component analysis required the computation of a very large correlation matrix, and thus consuming a lot of memory and processing power to work with, the use of a more efficient algorithm such as singular value decomposition is motivated. The relationship between principal component analysis and singular value decomposition was shown.

Finally, it was empirically shown that high variance of pixel intensity of face frames from a typical video telephony sequence is spatially concentrated around the mouth region. This implies that a large number of principal components are required to represent the subtle eye movement required to fully reconstruct many facial expressions. A solution to the problem that consisted of dividing the face into three regions based on the information of pixel intensity variance in the face images and computing eigenspaces for each region separately was proposed.

This section has primarily covered the model used to describe facial expressions of the user of the video telephony system as described in Section 2.1 outline. The principal component analysis, and the more practically efficient singular value decomposition, is used to derive the statistical part of the model, i.e. the eigenfeatures. The empirical part, i.e. the face synthesis procedure, was described in Section 2.2 eigenfaces and eigenfeatures.
2.3 Face detection and tracking

2.3.1 Overview

The previous section, eigenspace based compression, covered a technique for representing face images as a vector in a derived low-dimensional eigenspace to perform the compression. However, this process only works well if the face image is completely aligned with the basis functions. In order to use this novel compression scheme with a video telephony sequence the facial region must be detected, tracked and aligned with the basis functions.

The face detection and tracking performs the first step in the procedure. Its purpose is to efficiently find the approximate position and extent of the facial region, drastically minimizing the work for the more complex and computationally demanding basis alignment methods, described in Section 2.4 eigenfeature alignment. In the case of video telephony the image is assumed to only contain one face, roughly centered. If the user is moving or perhaps transmitting from a crowded place the face tracker must minimize the influence of other facial regions in the image that might appear and disappear.

The face detection scheme is illustrated in Figure 12. The purpose of the face detection scheme is to estimate the facial region in the video telephony frame. The basic assumption of the detection scheme is that the facial area is mainly covered with skin color and that the skin color distribution of the user is known a priori. The video telephony frame is fed into the skin color probability map computation step described below. Using the computed skin color probabilities the area most likely to represent the face is computed through the skin color probability integral image and the facial region estimation steps.

Finally, a face tracking scheme exploiting the fact that the facial region of a certain frame is largely correlated with previous frames is proposed. This scheme significantly reduced the overall computational complexity of the face detection and tracking scheme.
2.3.2 Skin color detection

A very popular face detecting method is skin color segmentation, which is especially useful if prior information of the user and scenario is known. Therefore it is especially applicable in this thesis since information needed to enable skin color detection can be stored with the user’s profile along with the eigenfeatures, i.e. the model, described in Section 2.2 eigenspace based compression. Furthermore it is assumed that all video telephony sessions actually contain a face. It is the purpose of the skin color detection algorithm to provide an initial guess of the position of the face in order to facilitate more complex detection algorithms.

It is empirically shown that human skin color, fairly independent of lighting, occupy a limited region in some color spaces [7], [8]. To date, various color spaces have been proposed for face detection, e.g. the $C_bC_r$, I-Q and $r$-$g$ models. In this thesis two color spaces are examined, $C_bC_r$ and $r$-$g$. The $R$, $G$ and $B$ color components of each pixel in the video telephony frame are converted into $YC_bC_r$ and normalized $r$, $g$, $b$ as expressed below:

$$
\begin{bmatrix}
Y \\
C_b \\
C_r
\end{bmatrix} =
\begin{bmatrix}
0.299 & 0.587 & 0.115 \\
-0.1688 & -0.3312 & 0.5 \\
0.5 & -0.4187 & -0.0816
\end{bmatrix}
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix},
$$
and into normalized $r$, $g$, $b$ as follows:

\[
    r = \frac{R}{R + G + B},
    \quad g = \frac{G}{R + G + B},
    \quad b = \frac{B}{R + G + B}.
\]

In order to model the skin color distribution in the two color spaces a number of sample images of faces were used to gather pixels of skin color and pixels not belonging to the skin color regions. These pixels were converted into the described color spaces and plotted as presented in Figure 13.

![Figure 13. Left, Pixels in Cb-Cr space, blue (dark) dots represent face pixels and red (light) dots represent non-face pixels. Right, Pixels in r-g space, blue (dark) dots represent face pixels and red (light) dots represent non-face pixels. See Color plate 2.](image)

Most articles [7], [8] assume that the skin color distribution can be modeled using a normal distribution. This seems to be the case for some color spaces, for instance the $C_bC_r$ color space in Figure 13 but not as much for the normalized $r$-$g$ color space.

Some articles [9] use a more effective system both regarding speed and robustness. These systems compute the skin probability for a given color by application of Bayes rule.
Application of Bayes rule requires the following terms:

- \( h_{\text{skin}}(u,v) \): Histogram of colors in the selected color space from a region of an image known to represent skin.
- \( N_{\text{skin}} \): Number of pixels known to represent skin used to build the histogram.
- \( h_{\text{total}}(u,v) \): Histogram of colors in the selected color space from the entire image.
- \( N_{\text{total}} \): Total number of pixels used to build the histogram.

The probability of a color vector in some color space \((u,v)\) given it represents skin is approximated by:

\[
p(r, g \mid \text{skin}) \approx \frac{1}{N_{\text{skin}}} h_{\text{skin}}(u,v).
\]

The probability of obtaining a skin pixel in the image is approximated by:

\[
p(\text{skin}) \approx \frac{N_{\text{skin}}}{N_{\text{total}}}.
\]

The probability of a certain color is given by:

\[
p(u,v) \approx \frac{1}{N_{\text{total}}} h_{\text{skin}}(u,v).
\]

Using Bayes rule the probability for skin given a certain color can be expressed by:

\[
p(\text{skin} \mid u,v) = \frac{p(u,v \mid \text{skin})p(\text{skin})}{p(u,v)} = \frac{h_{\text{skin}}(u,v)}{h_{\text{total}}(u,v)}.
\]

Using the expression above, a lookup table can be derived translating colors expressed in a certain color system to the probability of the color belonging to a skin region. Using the lookup table the probability of a certain pixel representing human skin can be computed. In Figure 14 and Figure 15 the probability lookup table and skin pixel probabilities for a number of images using both \(r-g\) and \(C_bC_r\) color spaces respectively is presented.
In the skin probability images below, the mean value, i.e. center of gravity, and the horizontal and vertical variances are computed. These values could be used as an estimate position and scale for the facial region in the image but for reasons covered later is not accurate enough.

Figure 14. Left, skin probability map for the r-g color space. Right, skin probability computed for each pixel in a number of images. The blue cross represents the mean of the pixel probabilities and the lines represent the horizontal and vertical variances. Color plate 3.

Figure 15. Left, skin probability map for the Cr-Cb color space. Right, skin probability computed for each pixel in a number of images. See Color plate 4.
If one studies the skin probabilities in the example images above using the two different color spaces one finds that some pixels are classified as being skin that obviously are not part of the face, for instance skin colored background features. It is also apparent that when using different color spaces different background features are misidentified. This feature can be exploited as described in [9], by using a number of color spaces and a fusion of the resulting skin probability images a large number of misidentification can be eliminated.

### 2.3.3 Integral image

The skin color probability map, computed as described in Section 2.3.2 skin color detection, will form a base for estimation of the facial region. As proposed in the previous section a possible approach would be to compute the mean and co-variances of the skin color probability map and interpret these as position and scale of the facial region. These computations are however extremely sensitive to noise and skin colored background elements. Thus a more sophisticated method is needed.

An empirical approach easily conveyed by studying the skin probability maps is to find the facial region by searching for a region of a certain aspect ratio with high amounts of skin probability. However, this approach needs an efficient method for evaluating a certain facial region according to the stated rules. One efficient method is to use an intermediate image representation called the integral image [10].

Using the integral image, described in [10], the content of rectangular areas of different dimensions can be summed rapidly. This is a useful method for searching for rectangular feature areas in both the skin color probability map and the original intensity image. The integral image is defined as:

$$ ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') $$

where $ii$ is the integral image and $i$ is the source image, in our case the skin color probability map. The integral image can be computed in linear time using the following expression:
Chapter. 2 Method

\[ s(x, y) = s(x, y - 1) + i(x, y) \]
\[ ii(x, y) = ii(x - 1, y) + s(x, y) \]

where \( s(x,y) \) is the cumulative row sum, \( s(x,-1) = 0 \) and \( ii(-1,y) = 0 \). Using the integral image a rectangle sum can be computed using four reference operations. This enables a great number of rectangles to be investigated at very low computational complexity. The sum of a certain rectangle can actually be computed with only four array references as illustrated in Figure 16. The value of the integral image at local 1 is the sum of the contents of rectangle A. The value at 2 is the sum of A and B, the value at 3 is the sum of A and C and finally the value at 4 is the sum of A, B, C and D. The sum of D can thus be computed as \( 4 + 1 - (2 + 3) \).

![Figure 16. The sum of the rectangle D can be computed with four array references. The value of the integral image at local 1 is the sum of the contents of rectangle A. The value at 2 is the sum of A and B, the value at 3 is the sum of A and C and finally the value at 4 is the sum of A, B, C and D. The sum of D can thus be computed as 4 + 1 – (2 + 3).](image)

### 2.3.4 Fast facial region estimation

The previously described integral image technique can be used to efficiently compute the sum of the content in a given rectangle. In order to concentrate the more computationally complex basis alignment search to a limited segment of the video telephony frame the facial region can be estimated. Computing an integral image of the skin color probability map, which enables the skin color content of a given rectangle to be efficiently computed, performs this.
Chapter. 2 Method

This operation is performed for all rectangles within the skin probability map of a given size and aspect ratio interval \([m, n]\), where \(m\) and \(n\) represent the dimensions of the rectangle under the given aspect ratio and size constraint. All rectangles with an average skin color probability, \(\bar{p}_{\text{skin}}\), over a given threshold is further processed. The average skin color probability of a given region is computed as expressed below:

\[
\bar{p}_{\text{skin}} = \frac{1}{mn} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f_{mn}(x,y),
\]

where \(f_{mn}(x,y)\) is the sum of the content of a rectangle of dimensions \(m\) and \(n\) at position \((x, y)\) in the skin color probability map:

\[
f_{mn}(x,y) = ii(x + m, y + n) + ii(x, y) - [ii(x + m, y) + ii(x, y + n)]
\]

Figure 17 illustrates the rectangles with average skin color probability over the given threshold for an example image.

![Figure 17](image-url)

*Figure 17. A frame from a possible video telephony session with a number of rectangles with an average skin probability over the defined threshold marked.*

Finally, overlapping rectangles must be merged to form facial region candidates. The merging is performed iteratively while evaluating the rectangles, as soon as the first
rectangle with average skin color probability over the threshold is found it is denoted the root of cluster $c_0$. When an a new rectangle is found, a check is performed such that:

- If the rectangle overlaps any of the existing clusters $c_{0,n}$, where $n$ is the current amount of clusters, the intersected cluster is expanded to include the new rectangle.

- If the rectangle intersects more than one cluster, these clusters are merged to form a cluster covering the original small clusters and the new rectangle.

- If the rectangle does not overlap any of the existing clusters a new cluster $c_{n+1}$ is formed containing the new rectangle.

When the complete frame has been covered the largest cluster closest to the center of the screen is assumed to be the facial region of the user. Figure 18 illustrates some results from the described facial region estimation.
Figure 18. Example results of the face region estimation algorithm. Left, the skin color probability map and the estimated facial region. Right, the original image and the estimated facial region. See Color plate 5.
2.3.5 Facial region tracking

Even though the facial region estimating is greatly optimized with the use of the integral image the fact that the facial region is largely correlated with facial regions of previous frames cannot be ignored. The connection to the facial region estimation is illustrated in Figure 19.

The estimated facial region for the previous frame is used to limit the search for a new facial region. The possible displacements and scaling of the facial region from the previous frame is illustrated in Figure 20. The actual displacement amounts are empirically derived and largely depend on the frame rate of the video telephony sequence.
Chapter. 2 Method

The largest region with the highest average skin probability over the given threshold is selected as the new facial region estimation. This implies that if the face has moved closer to the camera since the last frame a larger facial region with average skin probability over the given threshold can be found and will thus be selected, and vice versa.

2.3.6 Summary

A motivation and overview of the face detection and tracking system based on skin color segmentation has been covered. A common yet efficient skin color segmentation algorithm is presented resulting in a skin color probability map, i.e. a skin color probability for each pixel of the video telephony frame.

In order to estimate the facial region based on the skin color probability map, a large number of rectangle sums needs to be computed. To perform this efficiently the integral image approach is presented. The integral image is used to estimate the facial region of the user by merging smaller rectangles with an average skin probability over a certain threshold.

The fact that the facial regions in different frames are temporally correlated is exploited in the face tracking. The estimated facial region of previous frames is used to limit the search for a new facial region greatly reducing the computational complexity of the face detection and tracking.

The described face detection and tracking scheme is used in the encoder scheme described in Section 2.1 outline. Estimating the facial region will, as described in the next section, greatly reduce the complexity of the model parameter derivation process.
2.4 Eigenfeature alignment

2.4.1 Motivation
The purpose of the video telephony compression scheme described in this thesis is given a frame of the user, to extract a number of coefficients describing each of the eigenfeatures in their eigenspace, i.e. the model parameter derivation as described in Section 2.1 outline. With these coefficients the facial expressions of the user can be reconstructed using the eigenfeatures as described in Section 2.2 eigenspace based compression. In order to achieve good reconstruction quality the eigenfeatures must be aligned with the corresponding facial feature of the video telephony frame before the coefficients are computed by projection. One approach is to find the displacements of each of the eigenfeatures such that the reconstruction error is minimized. However computing the reconstruction error for all possible displacements of the eigenfeatures are too computationally demanding, hence this section is dedicated to describing the scheme used to obtain the eigenfeature coefficients using reconstruction error minimization in an efficient manner. The scheme is outlined in Figure 21.

The face detection and tracking step results in a facial region estimation that will be used to limit the search for the eigenfeature displacement. An initial candidate displacement estimation will be performed using an efficient statistical pattern recognition algorithm reducing the number of likely displacements. The eigenfeature displacement identified by

![Figure 21. The eigenfeature alignment scheme. The facial region estimation above only gives a rough estimate of the position and extent of the facial region.](image-url)
the reconstruction error minimization for the previous frame will be used in the candidate displacements feedback to generate a number of likely displacements candidates.

2.4.2 Eigenfeature importance region

There are two major problems with attempting to minimize the reconstruction error of the eigenfeatures. One being the fact that the eigenfeatures are of rather large dimensions resulting in computationally demanding reconstruction error computations. The second is the fact that the background and secondary facial features such as the haircut in the eigenfeatures is most likely not compatible with the current background and haircut in the video telephony session. It is therefore reasonable to focus on minimizing the reconstruction error for a limited region of each of the three eigenfeatures, this region containing the most important features required to represent the vast majority of facial expressions as illustrated in Figure 22.

Thus the reconstruction error minimization will be limited to these regions. The complete eigenfeatures each representing one third of the facial region as described in eigenspace based compression are still used as basis for reconstruction. From these eigenfeatures small regions, $U_{re}$, $U_{le}$, and $U_m$ are extracted. The reconstruction error to be minimized will be computed for these regions only, thus omitting the possible background and haircut mismatch. The coefficients extracted from these regions will be used with the complete basis to reconstruct the face at the receiving end, as described in eigenspace based compression. Thus the displacement $(x,y)$ that minimizes the reconstruction error at the eigenfeature importance region has to be found. This can be achieved by minimizing:
where \( I_{x,y} \) is a region from the video telephony frame at displacement \((x, y)\) of the same size as the basis \( U_{le} \). The procedure for finding the coefficients \( c \) is almost identical to the one outlined in eigenspace based compression. The major difference is the fact that the basis \( U_{le} \) is extracted from an orthonormal basis, the eigenfeature, and thus is not itself guaranteed to be orthonormal. The differentiation of the coefficients thus results in,

\[
\frac{d}{dc} \left[ (I_{x,y} - U_{le} c)^T (I_{x,y} - U_{le} c) \right] = \\
-I_{x,y}^T U_{le} - U_{le}^T I_{x,y} + 2c U_{le}^T U_{le}.
\]

Equating the derivative to zero results in an equation system which has the following solution:

\[
c = \left[ (U_{le}^T U_{le})^{-1} U_{le}^T \right] I_{x,y} = U_{le}^T U_{le} I_{x,y},
\]

where \( U_{le}^T \) is the Penrose inverse of \( U_{le} \). The derived coefficients can, as previously mentioned, be used to reconstruct an image of the face as described in eigenspace based compression. The procedure is identical for each of the three eigenfeatures. The remainder of this section will be dedicated to estimating the displacements \((x, y)\) for each eigenfeature using the smaller eigenfeature importance regions \( U_{re}, U_{le} \), and \( U_m \).

### 2.4.3 Candidate displacements estimation

As a preprocessing step prior to the more computationally complex eigenfeature alignment phase rough candidate displacements of the eigenfeatures are estimated. This is performed using a statistical pattern recognition method influence by a method described by Viola et. al. [10]. The purpose of the method is to classify a certain image region, interpreted as a vector \( x \), as either being a candidate for an eigenfeature displacement, in this case an image of an eye, or not. If the image regions possibly containing an eye can be narrowed down it would greatly improve the performance of the eigenfeature alignment. As in most statistical pattern recognition methods initially a data
reduction step is performed, so that a vector \( z \) can be derived from \( x \) such that \( z \) is of much lower dimension than the initial image region \( x \). A common data reduction method is the previously described principal component analysis but in this case a method inspired by [10] is applied.

The common appearance of the eye, as illustrated in Figure 23, is modeled using a set of rectangular features. The work of Viola et. al. [10] describes a machine learning approach where a number of features are derived that can be used with a hypothesis function to evaluate the probability of a certain image region being a face. A feature is a collection of differences between adjacent rectangle sums within an intensity (i.e. grayscale) image. The use of the integral image, described in Section 2.3.3 integral image allows for very fast evaluation of a large number of features at different scales.

Although a lot simpler the approach used in this thesis to estimate eye candidates has similarities with the method described in [10]. As opposed to [10] these features are empirically derived from the appearance of eye image regions as presented in Figure 23.

![Figure 23. Left, four sample 16 by 16 regions of eyes from the training sequence. Left, two empirically derived features. The first modeling the fact that the center of the eye is very dark compared to the surrounding region. The second modeling the fact that the eyebrows often shadows the eyes.](image)

By computing the value of these features for a certain \( n \) by \( m \) image region the dimension of the data vector can be lowered from \( nm \) to the number of features used, in this case two. The features are chosen such that they represent the common appearance of eyes in this case the fact that the center of the eye is very dark compared to the surrounding region and the fact that the eyebrows often shadows the eyes. The value of a certain feature is computed by subtracting the sum of the intensity content of the highlighted rectangle from the sum of the intensity content of the white area, as illustrated in Figure
23. The feature can be efficiently evaluated with only eight array references using the integral image technique, described in Section 2.3.3 integral image.

The properties of the feature can be illustrated by examining a number of training regions some positives, i.e. images of eyes, and other negatives, i.e. random regions taken from the training sequence. The investigated regions have been limited to the facial area of the training sequence since this area has been identified for further processing by the fast facial region estimation step previously described. The values of the two features for the regions have been plotted in a so-called feature space, see Figure 24, where dots represent non-eye observations and crosses represent eye observations.

The area representing eye candidate regions can be delimited using linear discriminants. As illustrated in Figure 24 two very simple discriminants are chosen such that feature values residing in the upper right quadrant are determined as eye candidates. It is important to choose discriminants rather such that too many positives are detected than such that the true displacements are missed. The results from the procedure for an arbitrary frame are presented in Figure 24 as well. The described technique is far from acceptable as a genuine object detection method, although it has been shown [10] that with a large number of features and a robust machine learning technique the described approach can yield very high detection rates. In this thesis the described method is only used as a basis to further enhance the robust object detection methods.
2.4.4 Reconstruction error minimization

The above statistical pattern recognition step results in a number of possible displacements of the eigenfeatures. Left to find out, are the true displacements by reconstruction error minimization. The obvious approach is to compute the reconstruction error for all candidate displacements. There is however one additional optimization that can be performed. The reconstruction error can be expressed, as previously described, according to:

$$mse_{x,y} = \| I_{x,y} - cU \|^2$$

where U can be any of $U_{re}$, $U_{ic}$, and $U_{m}$ depending on the eigenfeature. By expanding the reconstruction the following expression is obtained:

$$mse_{x,y} = \| I_{x,y} - c_1 U_1 - c_2 U_2 - \ldots - c_n U_n \|^2$$

As previously mentioned the first basis vector $U_1$ represents the largest variations of the eigenfeature, it is thus reasonable to assume that the reconstruction error at the correct displacement $(x, y)$ will drop primarily during the first reconstructions. By studying Figure 25, one can assume this seems indeed to be the case.

![Figure 25](image)

*Figure 25. Left, a number of possible displacements in proximity of the true region, the true region being the bottom right of the four (D). Right, the mean square reconstruction error for each of the displacements and with increasing number of basis projections.*
In fact the reconstruction error of the true displacement, i.e. the one with the least reconstruction error after \( n \) base function projections, is the lowest of the four after only the first base function projection. This is however not always the case as illustrated in Figure 26.

![Image](image_url)

*Figure 26. Left, four displacements each one pixel from the true displacement, i.e. the displacement minimizing the reconstruction error with a given set of bases, except the one at bottom right which is the true displacement (D). Right, the mean square reconstruction error for each of the displacements with increasing number of basis projections.*

In Figure 26 as opposed to in the case of Figure 25 the true displacement, i.e. the displacement minimizing the reconstruction error with a given set of bases, cannot be determined until all basis projections have been performed, although the candidates can be narrowed down to two after only six basis projections. The process of narrowing down candidates can be exploited to further optimize the reconstruction error minimization procedure. The algorithm can be outlined as follows:

1. Let \( D \) be the set of possible displacements identified either by the *candidate displacements estimation* or the *candidate displacements feedback* described later.

2. For each displacement in \( D \) the first coefficient is computed by basis projection. Thereafter the reconstruction error is computed using the one coefficient as outlined above.
3. When the reconstruction errors using one basis is computed for each displacement in $D$ the top 40% (i.e. with the lowest reconstruction error) is selected the rest is removed from $D$.

4. Finally the procedure is repeated with start at step 2. This time the second coefficient is included and thus the reconstruction error when using the two first coefficients is computed.

The algorithm continues until either all coefficients are used in which case the best remaining reconstruction of $D$ is chosen, or $D$ only consist of one remaining displacement. The algorithm has much in common with breadth first based search schemes. Both the argument that the principal component analysis orders the basis vectors after preceding variance and the empirically shown cases of Figure 25 and Figure 26 supports the use of a breadth first approach.

### 2.4.5 Candidate mouth displacements

The sections above have primarily covered the two eigenfeatures representing the left and right eye of the user. The reason for this is simply that images of eyes have good statistical properties making them suitable for pattern recognition. When the position of the eyes have been found, i.e. after the reconstruction error minimization step previously described, the candidate displacements of the mouth can be estimated. Figure 27 below illustrates the manner in which the candidate displacements of the mouth are derived from the known positions of the eyes.

![Figure 27. Illustrates the manner in which the candidate displacements of the mouth are derived from the known positions of the eyes.](image)
Chapter 2 Method

The end point of the line perpendicular to the line passing through the left and right eye, A, extended from the center between the eyes B a certain given length will become the center of the region of candidate mouth displacements. The given length of the line along the nose and the extent of the region of candidate mouth displacements can be set beforehand.

2.4.6 Candidate displacements feedback

Section 2.4.3 candidate displacement estimation above described a method for estimating a number of possible displacements based on no prior information about the position of the eyes or mouth, i.e. eigenfeatures. It is reasonable to assume that the displacements of the eigenfeatures are greatly correlated with the displacements of the previous frame as used in facial region tracking. This can be used to define new candidate displacements based on the displacements that minimized the reconstruction error of the previous frame.

![Candidate displacement regions are based on true displacements of the previous frame.](image)

Figure 28. Candidate displacement regions are based on the displacements that minimized the reconstruction error of the previous frame.

These regions are fed back as candidate displacements to the reconstruction error minimization. With constant intervals the candidate displacements estimation is computed to ensure that no drifting occurs.
2.4.7 Summary

This chapter has covered a search method for obtaining the coefficients encoding the facial region as described in Section 2.2 eigenspace based compression. I.e. this section has covered the model parameter derivation procedure as described in Section 2.1 outline.

It is described how the coefficients can be obtained by projecting only a limited region of the eigenfeatures, thus reducing the computational complexity of the projection and reducing influence of mismatching background features. The obtained coefficients are the one that minimizes the reconstruction error.

The described method is based on a breadth first search scheme where unlikely displacements are removed at an early stage so that the computational demanding operations are performed on fewer candidates. The first part of the scheme involves a basic statistical pattern recognition approach identifying displacement candidates for the eigenfeatures representing the left and right eye.

The second part covers a method for finding the displacements that minimizes the reconstruction error without having to perform a full basis projection and reconstruction on all candidate displacements.

After the eigenfeatures representing the eyes are fixed the candidate displacements of the mouth can be estimated. Finally a scheme where the eigenfeature displacements of the previous frame are used to derive candidate displacements for the current frame is outlined.
2.5 Coefficient encoding

2.5.1 Differential Pulse Code Modulation

The technique used to encode the coefficients is called Differential Pulse Code Modulation, or DPCM. The aim of DPCM is to exploit the fact the variable to be encoded is temporally correlated, i.e. the value at a certain time is largely dependant on previous values. A brief introduction of DPCM follows, for a more detailed description see [11]. The coding scheme is depicted in Figure 29.

![Figure 29. The encoding scheme of the Differential Pulse Code Modulation.](image)

A coefficient, $f_n$, at a given time $n$ is to be encoded. Differential Pulse Code Modulation is a predictive coding method, which implies that the value of a coefficient at time $n$ is estimated using a linear combination of the $m$ previously transmitted values according to below:

$$\hat{f}_n = \sum_{1 \leq k \leq m} a_k f_{n-k},$$

where $\hat{f}_n$ is the predicted value. The predicted value is subtracted from the value to be encoded resulting in the difference, $e_n$. The difference is quantized resulting in $\hat{e}_n$ which is encoded into a given symbol and transmitted to the receiving end. Finally the quantized difference is added to the prediction as follows:
where \( \hat{f}_n \) will be used to derive the prediction \( \hat{f}_{n+1} \), this closed loop configuration ensures that no error build-up will occur at the receiving end. The decoder follows, as depicted in Figure 30.

\[
\hat{f}_n = \hat{f}_n + \hat{e}_n
\]

The reconstructed value \( \hat{f}_n \) is computed as described above. The encoding quality is greatly dependent of the predictor, thus a good method for automatically deriving the predictor from training data is required.

### 2.5.2 Optimal predictor

The optimal predictor is found by minimizing the squared predicted error as expressed below:

\[
E\{e_n^2\} = E\left[ f_n - \hat{f}_n \right]^2
\]

under the assumption that the quantization error can be neglected:

\[
f_n = \hat{f}_n + e_n \approx \hat{f}_n + \hat{e}_n = \hat{f}_n
\]

which of course is not always suitable. It is important not to apply too rough quantization since that will affect the prediction as well. Finally, substituting \( \hat{f}_n \) in the expression to be minimized, the following is obtained:
\[ E\{e_n^2\} = E\left( f_n - \sum_{1 \leq k \leq m} a_k f_{n-k} \right)^2. \]

If the expression above is differentiated with respect to \( a_k \), the derivatives equated to zero and the resulting system of equations is solved under the assumption that \( f \) has zero mean the following expression is obtained:

\[ a = R^{-1}r, \]

where \( R \) is the \( m \) by \( m \) correlation matrix:

\[ R = \begin{bmatrix}
E\{f_{n-1}f_{n-1}\} & \cdots & E\{f_{n-1}f_{n-m}\} \\
\vdots & \ddots & \vdots \\
E\{f_{n-m}f_{n-1}\} & \cdots & E\{f_{n-m}f_{n-m}\}
\end{bmatrix}, \]

and \( r \) is a vector with the correlation of the first and all previous values:

\[ r = \begin{bmatrix}
E\{f_nf_{n-1}\} \\
\vdots \\
E\{f_nf_{n-m}\}
\end{bmatrix}. \]

Thus an optimal predictor can be derived for each of the coefficients using a set of training data.

### 2.5.3 Efficient quantization

The prediction error, \( e_n \), must be quantized into, \( \hat{e}_n \), in order to be converted into a symbol and transmitted to the receiver. The quantizer is a simple function that maps the input signal \( s \) onto reconstruction levels \( t \). Each reconstruction level represents a reconstruction region that is encoded with a specific symbol. The described quantization procedure is illustrated in Figure 31.
In order to achieve efficient quantization the distribution of the variable to be encoded must be studied. Variables of a certain distribution can be quantized with great success, and this is the key of the DPCM method. Figure 32 presents the distribution of a coefficient to be encoded as well as the distribution of the prediction error of the very same coefficient subjected to DPCM. Clearly, since the prediction error is concentrated around zero the quantization must utilize this in order to be efficient.

Using the distribution of the variable to be quantized efficient reconstruction levels can be derived. By computing the integral of the distribution function the reconstruction regions can be obtained by dividing it into the number of regions required, as illustrated in Figure 33.
The efficient quantizer is used in conjunction with the optimal predictor to form a communication solution for each coefficient, since the solution will be tailored based on the statistical properties of the signal. The solution is dependant on the statistical properties of the signal in order to be efficient, thus the solution must be computed for each individual base vector.

### 2.5.4 Summary

In order to transfer the encoded images, i.e. eigenfeature coefficients or the model parameters as described in Section 2.1 outline, in an efficient manner a tailored communication solution must be derived. This section covered the method used in this thesis, Differential Pulse Code Modulation. The method is based on the fact that a signal is temporally correlated such that the signal value at a specific time can be predicted using a linear combination of previous values. Thus the prediction error at each time can be transmitted. It is described how to derive an optimal predictor, in a mean square error sense, and an efficient quantizer based on the statistical properties of the prediction error.
3 Results and discussion

This section contains a description of an implementation of a system as described in the method chapter, followed by a presentation of some results regarding primarily reconstruction quality and data rate. Finally a section shortly presenting related work on model based coding, with a comparison with the proposed system, is given.

3.1 Implementation

The implementation consists of two main parts. Firstly, a reference implementation of the complete video telephony scheme in MATLAB. Secondly, an implementation of the decoder scheme in C/C++ on a Nokia 7650 in order to show the feasibility of implementation on memory and processing power constrained mobile devices will be performed at Summus Inc, NC. Details regarding the implementation will be included as an appendix to this report.

Firstly, the statistical part of the model, as described in eigenspace based compression, must be derived. This is performed by capturing a sequence of the face where the user performs a wide range of facial expressions. These expressions must include the vital sound building blocks and the basic emotions, sorrow, anger, fear, surprise, joy and disgust. The training sequence will determine which facial expressions that can be represented by the model and therefore should be as complete as possible.

A MATLAB function that loads the training sequence and lets the user specify the approximate center of each of the eigenfeatures, left eye, right eye and mouth is provided. These centers also define the importance region, as described in Section 2.4 eigenfeature alignment and illustrated in Figure 34. The complete training sequence is used to build the three eigenfeatures. It is also used to sample skin colored pixels in order to build the skin color histogram as described in Section 2.3 face detection and tracking. It is important not to base the histogram computation solely on the training sequence since the resulting classification system will be sensitive to lighting changes. The histogram contains of a number of fixed skin color pixels from other images although the skin color pixels of the training sequence will represent the majority of the samples.
Chapter. 3 Results and discussion

When the user profile, i.e. the model and the skin color histogram has been built, the video telephony scheme can be tested. A reference implementation of both the encoder and decoder scheme is presented in Figure 35.

The different steps of the encoding and decoding scheme are represented by an image in the screenshot of Figure 35. The original frame is followed by the skin color probability image with the estimated facial region. Thereafter the result from the eigenfeature alignment is presented followed by some of the encoded coefficients. Finally the eigenfeature importance regions resulting in the lowest reconstruction error are presented followed by the complete face reconstruction as described in Section 2.2 eigenspace based compression.
3.2 Results

A specific training sequence and test sequence has been captured for evaluation of the coding scheme in this thesis. A number of frames from the two sequences are presented in Figure 36. The training sequence covers a number of facial expressions but far from all. In order to give a solid statistical foundation for all facial expressions a comprehensive training sequence is required, but the concept is still demonstrated. The test sequence is also far from a realistic case. The camera is static which will most certainly never be the case when a mobile device such as a cell phone is used. Another fact is that the sequence is captured slightly after the training sequence is captured which implies that the lighting conditions are very similar. Finally the variation of scale and...
rotation of the face is very limited. In summary the two sequences are derived in order to prove the concept video telephony system proposed in this thesis. The training sequence contains 184 frames that have been used to derive 16 eigenfeature bases.

3.2.1 Reconstruction quality

It is hard to talk about general compression ratio and reconstruction performance, such as signal-to-noise ratios, since the result of this compression scheme is very subjective. Therefore a number of sample frames from the sequence have been collected.

Figure 37 presents some frames with subjectively good quality. The facial expression of the reconstructed face is very similar to that of the input frame. The most common reason for achieving excellent reconstruction quality is the fact that the facial expression is well represented in the training sequence used to build the eigenfeatures. The fact that a single person has a fairly limited set of facial expressions which are reused implies that good eigenfeatures can be built and is in favor for the proposed compression scheme.

Figure 38 presents some of the frames with subjectively poor quality. The main reason for the poor results is failure of the eigenfeature alignment due to poor face normalization. This means that the face detection and tracking have not been able to normalize the face due to rotation or scaling since the eigenfeature alignment is scale and rotation dependant, it will fail. Another fact is that the training sequence contains fairly
poor samples of faces with eyes closed and almost closed which implies that these features are not enough represented in the eigenfeatures. This error is apparent in the left most example of Figure 38.

The most apparent problem with the compression scheme is not visible in the presented still images. If the user’s head is not absolutely static in the training sequence, i.e. the defined center of the eigenfeature is not precise enough, this will affect the face reconstruction. Since the three eigenfeatures are encoded and decoded separately small displacements can occur. If for instance the left eye eigenfeature is aligned with a frame from the training sequence where the face was slightly to the left, while the right eye is aligned with a frame where the face was slightly to the right, this will result in a face slightly wider than the original. This can be improved by for instance tracking the eigenfeature center or simply not moving the head at all when capturing the training sequence.

Another fact that became apparent when testing the final system is that it is rather lighting and environment dependent. The face detection and tracking works fairly well under various lighting conditions, it fails in dark places or when colored lighting is used. The skin color detection used is also very dependant on the color balance of the camera used, such that when changing camera the skin color histogram must be reestimated. However, the eigenfeature encoding fails if the environment and lighting, such as time of day, differs between the training sequence and the video telephony sequence. This is a major problem that needs to be improved if a robust system based on the proposed method is to be achieved.
Chapter 3 Results and discussion

Figure 37. Frames with subjectively good quality. See Color plate 9.

Figure 38. Frames with subjectively poor quality. See Color plate 10.
Chapter 3 Results and discussion

3.2.2 Data rate

In the previous section a number of resulting reconstructions have been presented showing that the concept of the face encoding is feasible. This section will present some of the impacts of the coefficient encoding. The coefficients are quantized and encoded with the DPCM scheme as expressed in Section 2.5 coefficients encoding. Figure 39 presents some reconstructions from an encoded frame with different number of bits allocated for the DPCM prediction error. The quantization artifacts do not appear as in other common compression schemes. In this scheme they result in accuracy problems when combining the eigenfeatures, which implies that the facial expression cannot be mimicked correctly.

However, the results of Figure 39 show that the proposed system can achieve reasonable quality at data rates of 3-4 kBit/s at 25 frames per second. Furthermore, the data rate is resolution independent since only the eigenfeatures are resolution dependent. However, in practice high resolution reveals more details which in other hand often require more eigenfeatures to describe, resulting in more coefficients that need to be encoded.

Figure 39. Top original frame, bottom from left 2 bit, 3 bit, 4 bit and 5 bit per coefficient 16 coefficient per eigenfeature with 25 frames per second results in 2.4 kBit/s, 3.6 kBit/s, 4.8 kBit/s and 6 kBit/s respectively.
3.3 Related work

3.3.1 Facial animation in MPEG-4

In 1999 facial animation became part of the MPEG-4 standard, called the MPEG-4 Face Animation Standard (FAS). The standard includes a scheme for decoding face parameters. These parameters consist of a number of feature points that represent vertices in a mesh describing the surface of the face. The MPEG-4 Face Animation Standard is not only intended for model based coding, but also from traditional computer graphics face synthesis with application to for instance cartoons.

The standard defines 84 feature points, as illustrated in Figure 40, that are considered critical for controlling the shape of the face to be animated. The shape and motion of the face is controlled by repositioning each feature point individually. The standard also
defines Facial Definition Parameters (FDP) which includes an initial shape and texture for the face, each feature point in the shape is also equipped with texture coordinates respectively. Finally, the standard includes Facial Animation Parameters (FAP) which includes both high-level and low-level motions of the feature points enabling the face to form sounds with the mouth and close and open the eyes. Another widely used facial animation system is the CANDIDE model described below.

### 3.3.2 Model based coding with CANDIDE

The first CANDIDE model was created by M. Rydefalk in 1987. It contained 75 vertices supporting a mesh of 100 triangles. However, the first model is rarely used instead it is replaced by the model proposed by M. Strömberg that contain 79 vertices, 108 surfaces and 11 so called action units. This version of CANDIDE has also become the most widely used version.

In order to support a higher grade of mimic and shape capabilities and further compatibility with the MPEG-4 Face Animation Standard a new model was proposed. In 2001 J. Ahlberg developed a new version of the CANDIDE model including 113 vertices, 184 surface, 65 Animation Units and 12 Shape Units. The model is presented in Figure 41.

![Figure 41. The 2001 CANDIDE model developed by J. Ahlberg.](image)

Of course there exist a great number of models with further complexity allowing more facial expressions to be represented. But the CANDIDE model has been renowned for its
ability to represent many facial expressions with few feature points, making it especially useful for model based coding schemes.

The Animation Units and the Shape Units closely correspond to the Facial Animation Parameters and Facial Definition Parameters of the MPEG-4 standard. Basically, the corresponding feature points of the model are tracked in the sequence of the user's face. When the points are tracked the face model, for instance CANDIDE, can be altered such that it approximates the shape of the user. Finally, the texture is sampled from the video sequence and mapped onto the model.

The major advantage of this form of model based coding is the fact that the feature points can be very efficiently encoded resulting in very low bit rate. However the reconstruction quality is limited by the complexity of the model, and as the complexity rises the number of feature points that need to be detected and tracked in the frame to be coded increases. This fact implies more computational requirements but also that the encoding is more error prone.

The reconstruction quality achieved by the proposed compression scheme exceeds most results shown with CANDIDE of the MPEG-4 FAS system. However, as previously described, the proposed system is tailored to a specific user and thus requires more setup, this is not the case for the above mentioned systems. Both CANDIDE and MPEG-4 FAS have also been able to reach very low bit rates while the proposed system has a somewhat higher bit rate. The CANDIDE based system have reached bit rates as low as 1 kBit/s while the in this thesis proposed system require bit rates in at least the range of 3 kBit/s. Although, when put into perspective with the increased reconstruction quality of the proposed system the lower bit rate is definitely not an obvious advantage of the CANDIDE and MPEG-4 FAS based systems.

3.3.3 Appearance-based video compression

Substantial work on appearance based video compression has been performed by Karl Schwerdt, and is presented in his thesis Appearance-based video compression, [4]. In his thesis he describes the use of an orthonormal basis spanning the space of face images, where each image is interpreted as a column vector that is to be coded. The complete sequence to be coded is used to build the orthonormal basis, when this is performed each frame of the sequence can be described as a point in the face space. If the basis has fewer dimensions than an individual frame of the sequence compression is performed. The system proposed by Schwerdt has much in common with the system proposed in this thesis. However, Schwerdt describes a system for offline use where the video sequence to
Chapter. 3 Results and discussion

be compressed is available and can be used to build the eigenspace used for compression. This results in high reconstruction quality but is not feasible for online or real-time use since the process of building the eigenspace is as described very computationally requiring. The result presented in his thesis is regarding reconstruction quality somewhat better than MPEG-2 but at equal compression ratio.

In this thesis a system for real-time use have been presented where the eigenspace used to encode the frames are built beforehand using training data from the user of the system. The system then attempts to encode the facial expression of the user and reconstruct it with the previously derived eigenspace. This is mentioned by Schwerdt but never exploited in his thesis. The system in this thesis is far from as robust as the one proposed by Schwerdt but on the other hand allow bit rates as low as 3 kBit/s and is feasible for real-time use.
4 Conclusions

In this thesis a novel video telephony compression scheme has been proposed and discussed. The scheme generates a talking head sequence from a head and shoulder video telephony sequence where the talking head mimics the facial expressions of the user. The scheme is based on model based coding and more specifically based on an eigenspace approach. The model used to represent the objects to be encoded, in this context the facial expressions of the talking head, is statistically derived as the principal components of a training sequence depicting the user performing a wide range of facial expressions. The thesis introduces the concept of eigenfeatures as used in video compression and a method for encoding the facial expressions of the talking head as a number of coefficients defining a linear combination of the eigenfeatures.

In order to encode the facial expressions of the talking head the face of the user must be detected and tracked in the head and shoulder sequence. A face detection and tracking scheme based on skin color segmentation and the concept of an integral image have been presented. The scheme efficiently estimates the facial region of the user.

A method for aligning the eigenfeatures with the face of the head and shoulder sequence and thus deriving the coefficients encoding the facial expression is presented. A number of optimization is introduced to traditional eigenspace search methods in order to enable the use of the proposed system in constrained mobile devices. Finally, a method for encoding the coefficients efficiently is presented.

The thesis has shown the feasibility of using the proposed system for video telephony on constrained devices where the otherwise commonly used model based coding schemes is not possible. However, proposed system also has applications on animated talking heads for computer-human interaction and facial expression detection. The described concepts can be extended in order to perform facial expression classification where, for instance, a computer system can detect if the user is frustrated or satisfied.
4.1 Contributions of the thesis

This thesis has three main contributions to the area of video compression. Firstly, the proposed face synthesis procedure based on three eigenfeatures empirically derived from the pixel variance of video sequences of different facial expression. This method significantly improves quality for eigenspace based compression schemes where the facial expression of the user in a head and shoulder sequence is to be mapped on a talking head model.

Secondly, the eigenfeature alignment method presented in this thesis where the property of descending energy of the eigenfeatures are exploited to enable a fast breadth first search based scheme for aligning the eigenfeatures with the head and shoulder sequence. This scheme significantly reduces computation complexity since the number of reconstruction error evaluations is limited.

Finally, the way in which an importance region can be extracted from the complete eigenfeature and used to derive coefficients compatible with the full basis. This ensures that background features does not interfere with the coefficient derivation, i.e. the model parameter derivation, and reduces computational complexity.
4.2 Future work

The proposed video telephony scheme of this thesis is far from complete. This thesis has only proved the concept of such a system; a complete system would definitely need some of the improvements stated below:

- The described system only encodes grey scale images. In order to encode color images, a set of compatible eigenfeatures for each color component must be built. It is important to realize that the data rate will not be affected when encoding color images, only the size of the eigenfeatures.

- The face detection and tracking scheme described in this thesis is effective but rather primitive. In order to support rotation and scaling of the head, a more robust scheme is required where the face is normalized such that the eigenfeature alignment works with face images with common scale and rotation.

- In order to avoid error, as described in Section 3 results and discussion, when reconstructing the talking head due to displacements of the eigenfeatures in relation to each other, eigenfeature tracking should be introduced. If the eigenfeatures are tracked in the training sequence, relative displacements during face reconstruction can be avoided as described in Section 3.2.1 reconstruction quality.
References


Table of figures

Figure 1. The model consists of two parts, an empirically derived face synthesis procedure and the statistically derived eigenfeatures. The eigenfeatures are derived from a set of training images using a method called principal component analysis. .............................................................................................. 5

Figure 2. The encoder scheme. The input frame is fed into the face tracker, when the face has been detected the parameters of the model can be derived. Finally, the parameters are encoded and transmitted to the receiver .................................................................................................................................. 6

Figure 3. The decoder scheme. The encoded model parameters are decoded. The decoded parameters are used with the prior known model to synthesize an image of the facial expression. ........................................................................... 6

Figure 4. An illustration of a possible image space, here illustrated as the space of two-pixel images, and the face space as a data manifold of the parent image space. ........................................................................... 7

Figure 5. Images from the training image sequence resembling facial expressions. From top left, the sounds AO, EI, M, F and P followed by the facial expressions for sorrow, anger, joy, fear, disgust and surprise. The training sequence originally contains about 100 frames. ............................................................................. 8

Figure 6. The first 16 basis images spanning the eigenspace of face images. The basis vectors are generated using singular value decomposition. ......................................................................................... 14

Figure 7. The variances of the 64 first basis vectors. Generated from the training sequence of Figure 5. .......................................................................... 15

Figure 8. A number of test cases. The left column consists of original face images not part of the training sequence of Figure 5. The middle column contains the reconstructed face respectively. The right column contains the difference between the original and reconstructed. The eigenspace presented in Figure 5 is used to reconstruct the images of the middle column. ...................................................................................... 17

Figure 9. Left, an image from the face space with eyes closed. Right, an image from the face space with its eyes open. Middle, a convex combination of the left and right image resulting in a face with transparent eyelids not within face space. ................................................................................................. 18

Figure 10. Left, the variances at each pixel in the images from the training sequence color coded. Right, the variances at each pixel in the images from the training sequence as a three-dimensional plot. See Color plate 1. ......................................................................................... 19

Figure 11. The face is divided into three regions each with focus on one face feature, left eye, right eye and mouth. ......................................................................................................................................... 20

Figure 12. Illustration of the flow of the face detection and tracking scheme. Bottom left, the input video telephony frame. Bottom right, the video telephony frame with estimated facial region. ........................................................................ 23

Figure 13. Left, Pixels in Cb-Cr space, blue (dark) dots represent face pixels and red (light) dots represent non-face pixels. Right, Pixels in r-g space, blue (dark) dots represent face pixels and red (light) dots represent non-face pixels. See Color plate 2. ......................................................................................... 24

Figure 14. Left, skin probability map for the r-g color space. Right, skin probability computed for each pixel in a number of images. The blue cross represents the mean of the pixel probabilities and the lines represent the horizontal and vertical variances. Color plate 3. ......................................................................................... 26
Table of figures

Figure 15. Left, skin probability map for the Cr-Cb color space. Right, skin probability computed for each pixel in a number of images. See Color plate 4. .............................................................. 26

Figure 16. The sum of the rectangle D can be computed with four array references. The value of the integral image at local 1 is the sum of the contents of rectangle A. The value at 2 is the sum of A and B, the value at 3 is the sum of A and C and finally the value at 4 is the sum of A, B, C and D. The sum of D can thus be computed as $4 + 1 - (2 + 3)$. ............................................................... 28

Figure 17. A frame from a possible video telephony session with a number of rectangles with an average skin probability over the defined threshold marked ................................................................. 29

Figure 18. Example results of the face region estimation algorithm. Left, the skin color probability map and the estimated facial region. Right, the original image and the estimated facial region. See Color plate 5. ............................................................................................................................. 31

Figure 19. The facial region tracking scheme. Using the fact that the facial region is largely correlated with the facial regions of previous frames the facial region is fed back to the facial region tracking. At regular intervals the facial region is re-estimated. ................................................................. 32

Figure 20. Illustrates the possible displacements and scaling of the facial region from the previous frame. 32

Figure 21. The eigenfeature alignment scheme. The facial region estimation above only gives a rough estimate of the position and extent of the facial region............................................................... 34

Figure 22. Frame from the training sequence with importance regions marked. The reconstruction error minimization will be limited to these regions. .............................................................. 35

Figure 23. Left, four sample 16 by 16 regions of eyes from the training sequence. Left, two empirically derived features. The first modeling the fact that the center of the eye is very dark compared to the surrounding region. The second modeling the fact that the eyebrows often shadows the eyes. ....... 37

Figure 24. Left, dots represent non-eye observations plotted in feature space and crosses represent eye observations. The linear discriminants define the region in feature space where observations are interpreted as eye candidates. Right, an image from the sample sequence where the centers of the 16 by 16 regions estimated as eye candidates are marked. See Color plate 6................................................. 38

Figure 25. Left, a number of possible displacements in proximity of the true region, the true region being the bottom right of the four (D). Right, the mean square reconstruction error for each of the displacements and with increasing number of basis projections............................................................ 39

Figure 26. Left, four displacements each one pixel from the true displacement, i.e. the displacement minimizing the reconstruction error with a given set of bases, except the one at bottom right which is the true displacement (D). Right, the mean square reconstruction error for each of the displacements with increasing number of basis projections............................................................ 40

Figure 27. Illustrates the manner in which the candidate displacements of the mouth are derived from the known positions of the eyes. ......................................................................................................................... 41

Figure 28. Candidate displacement regions are based on the displacements that minimized the reconstruction error of the previous frame................................................................. 42

Figure 29. The encoding scheme of the Differential Pulse Code Modulation................................. 44
Table of figures

Figure 30. The decoding scheme of the Differential Pulse Code Modulation.............................................. 45
Figure 31. Illustrates the stair function of the quantizer. Input values s is converted into reconstruction levels t. ........................................................................................................................................... 47
Figure 32. Left, the distribution of a coefficient. Right, the distribution of the prediction error to be encoded using DPCM. ........................................................................................................................................... 47
Figure 33. The integral of the distribution function of the prediction error can be used to derive the reconstruction levels and regions. The above function has eight regions i.e. three bits are required for each sample........................................................................................................................................... 48
Figure 34. A screenshot of the function enabling eigenfeature computation with MATLAB. See Color plate 7. ........................................................................................................................................... 50
Figure 35. Screenshot of complete encoding and decoding in MATLAB. Top left, original frame. Top middle, skin color probability and estimated facial region. Top right, greyscale original frame with aligned eigenfeatures. Bottom left, DPCM-encoding of the first coefficient of left eye, right eye and mouth eigenfeatures. Bottom middle, the reconstructed individual eigenfeature importance regions. Bottom right, the final reconstructed image. See Color plate 8. ........................................................................................................................................... 51
Figure 36. Top, a number of frames from the training sequence. All frames are recorded in front of a blue screen. Below, the test sequence........................................................................................................................................... 52
Figure 37. Frames with subjectively good quality. See Color plate 9. ........................................................................................................................................... 54
Figure 38. Frames with subjectively poor quality. See Color plate 10........................................................................................................................................... 54
Figure 39. Top original frame, bottom from left 2 bit, 3 bit, 4 bit and 5 bit per coefficient 16 coefficient per eigenfeature with 25 frames per second results in 2.4 kBit/s, 3.6 kBit/s, 4.8 kBit/s and 6 kBit/s respectively. ........................................................................................................................................... 55
Figure 40. The MPEG-4 Face Animation Standard feature points........................................................................................................................................... 56
Figure 41. The 2001 CANDIDE model developed by J. Ahlberg............................................................... 57
Color plate 1. Left, the variances at each pixel in the images from the training sequence color coded. Right, the variances at each pixel in the images from the training sequence as a three-dimensional plot.

Color plate 2. Left, Pixels in Cb-Cr space, blue dots represent face pixels and red dots represent non-face pixels. Right, Pixels in r-g space, blue dots represent face pixels and red dots represent non-face pixels.
Color plate 3. Left, skin probability map for the r-g color space. Right, skin probability computed for each pixel in a number of images. The blue cross represents the mean of the pixel probabilities and the lines represent the horizontal and vertical variances.

Color plate 4. Left, skin probability map for the Cr-Cb color space. Right, skin probability computed for each pixel in a number of images.
Color plate 5. Example results of the face region estimation algorithm. Left, the skin color probability map and the estimated facial region. Right, the original image and the estimated facial region.
Color plate 6. Left, dots represent non-eye observations plotted in feature space and crosses represent eye observations. The linear discriminants define the region in feature space where observations are interpreted as eye candidates. Right, an image from the sample sequence where the centers of the 16 by 16 estimated as eye candidates are marked.

Color plate 7. A screenshot of the function enabling eigenfeature computation with MATLAB.
Color plate 8. Screenshot of complete encoding and decoding in MATLAB. Top left, original frame. Top middle, skin color probability and estimated facial region. Top right, greyscale original frame with aligned eigenfeatures. Bottom left, DPCM-encoding of the first coefficient of left eye, right eye and mouth eigenfeatures. Bottom middle, the reconstructed individual eigenfeature importance regions. Bottom right, the final reconstructed image.
Color plates


Color plate 10. Frames with subjectively poor quality.
Color plate 11. A concept illustration of the proposed video telephony system. The face of the user is detected and tracked in the mobile unit, left. The face is encoded into the model parameters, i.e. coefficients derived using the eigenfeatures, middle. Finally, the face is reconstructed using the eigenfeatures and the image synthesis procedure, right.