THREE-DIMENSIONAL FACIAL MOTION AND STRUCTURE ESTIMATION IN VIDEO CODING

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL AND ELECTRONICS ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCE
OF BİLKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Gözde Bozdağı
January 1994

THREE-DIMENSIONAL FACIAL MOTION AND STRUCTURE ESTIMATION IN VIDEO CODING

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BİLKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Gözde Bozdağı

21 January 1994

Gözde Bozdağı.

tarafından bağışlanmıştır

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Levent Onural, Ph. D. (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Erdal Arıkan, Ph. D.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

A. Enis Çetin, Ph. D.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Bülent Özgüç, Ph. D.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Bülent Sankur, Ph. D.

Approved for the Institute of Engineering and Science:

Mehmet Baray, Ph. D.
Director of Institute of Engineering and Science

Dedicated to the memory of my father....

Abstract

THREE-DIMENSIONAL FACIAL MOTION AND STRUCTURE ESTIMATION IN VIDEO CODING

Gözde Bozdağı

Ph. D. in Electrical and Electronics Engineering

Supervisor:

Assoc. Prof. Dr.Levent Onural

21 January 1994

We propose a novel formulation where 3-D global and local motion estimation and the adaptation of a generic wire-frame model to a particular speaker are considered simultaneously within an optical flow based framework including the photometric effects of the motion. We use a flexible wire-frame model whose local structure is characterized by the normal vectors of the patches which are related to the coordinates of the nodes. Geometric constraints that describe the propagation of the movement of the nodes are introduced, which are then efficiently utilized to reduce the number of independent structure parameters. A stochastic relaxation algorithm has been used to determine optimum global motion estimates and the parameters describing the structure of the wire-frame model. For the initialization of the motion and structure parameters, a modified feature based algorithm is used whose performance has also been compared with the existing methods. Results with both simulated and real facial image sequences are provided.

Keywords: Image sequence coding, object-based coding methods, 3-D motion and structure estimation, stochastic relaxation, videophone, very low bit rate coding, object shape analysis, object motion analysis.

Özet

GÖRÜNTÜ DİZİSİ KODLAMADA YÜZE AİT ÜÇ BOYUTLU HAREKET VE YAPI KESTİRİMİ

Gözde Bozdağı
Elektrik ve Elektronik Mühendisliği Doktora

Tez Yöneticisi:
Doç. Dr. Levent Onural
21 Ocak 1994

Tipik bir konuşmacı için geliştirilmiş bir tel çerçeve modeline dayalı üç boyutlu hareket kestirimi ve yüzdeki derinlik bilgisinin eldeki konuşmacıya uyarlanımı için yeni bir metod önerilmiştir. Kullanılan algoritma optik akı metoduna dayanmakta ve harekete bağlı fotometrik etkileri de kullanmaktadır. Kullanılan tel çerçeve modeli birbirine bağlı üçgenlerden oluşmakta ve bu üçgenlerin konumları normal vektörleri ile gösterilmektedir. Üçgenlerin birbirine bağlı olma özelliği algoritmada kullanılan bağımsız değişken sayısını azaltmaktadır. Bilinmeyen hareket ve yapı bilgilerinin bulunması için bir olasılıklı gevşeme metodu, başlangıç noktasının bulunması için de bir nokta eşleştirilme metodu gerçekleştirilmiştir. Hem gerçek hem de benzetilmiş yüz görüntü dizileri kullanılarak elde edilen sonuçlar sunulmuştur.

Anahtar Sözcükler: Görüntü dizisi kodlama, cisim modeline dayalı kodlama, 3-boyutlu hareket ve yapı kestirimi, olasılıklı gevşeme, görüntülü telefon, çok düşük hızlarda iletim için kodlama, nesne şekli analizi, nesne hareketi analizi.

Acknowledgment

I would like to express my deepest gratitude to Levent Onural for his supervision and encouragement in all steps of the development of this work. I would also like to thank A. Murat Tekalp for his collaboration, guidance and invaluable advices. It was an extraordinary chance to work with him throughout this study.

I would like to thank Assoc. Prof. Dr. Enis Çetin, Assoc. Prof. Dr. Erdal Arıkan, Prof. Bülent Özgüç, and Prof. Bülent Sankur, the members of my jury, for their motivating and directive comments on my research.

I like to acknowledge the financial supports of TÜBİTAK through COST 211ter Project and Programme of Support for International Meetings, and the financial support of IEEE Turkey Section, for presentation of this work in the national and international conferences.

During a Ph.D. study one has to face with so many difficulties. Thanks to all of my friends who have been with me when I need them and for sharing good and bad times with me. Especially, it is my pleasure to express my thanks to Ogan for his valuable discussions especially when I am stuck during midnights; to Engin, Levent, Fatih, Cem, Bengi, Bilge and Mustafa for their moral support and friendship, and to Nail for listening to my endless complaints with patience, for making me smile even if I feel so desperate and for his valuable critiques during my thesis work.

Finally, my sincere thanks are due to my family for their love, patiance and continuous moral support throughout my graduate study. It is their unhesitating self-sacrifice which has enabled me to achieve my goals in my life.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Recent years have brought forward significant progress in the research and development activities in the field of digital image processing. Image processing is closely related to human vision which is probably the most important means of perception. As a result, image processing has a large number of applications such as remote sensing via satellites, image transmission and storage, medical image processing, robotic vision, automated inspection of industrial parts, etc., that play important roles in our daily life [1]-[6]. In most of these applications, we deal with image sequences instead of single images. These sequences are obtained by sampling and quantizing analog scenes into brightness levels which are represented by integer values. The amount of data represented by these sequences are extremely large so that without a substantial reduction, their transmission, storage and processing can be very expensive. For example, let us consider the transmission of 512x512x8 bits/pixel x 3-color video image over the telephone lines. Using a 9600 baud modem, the transmission would take approximately 11 minutes for just a single frame, which is unacceptable for most applications. Similarly, single color component of one frame of a Super 35 format motion picture may be digitized to a 3112 lines by 4096 pels, 10 bits/pel. Assuming three color components, 1 sec. of the movie takes approximately 1 GBytes [6]. If we consider long distance transmission of this movie, the cost is non-trivial. Therefore, image coding (or compression) is necessary to more

efficiently and economically utilize the channel bandwidth and storage space. Although bandwidth is becoming larger and storage is becoming cheaper in many applications, compression still remains of interest. The reason is that, people will continue to use relatively low capacity links such as low cost low rate modems, satellite communication links, and mobile communication. In addition, although there are high capacity links such as fiber optic links, the capacity of them may not be enough due to the growing amount of information that users wish to communicate. Also, in applications such as multichannel HDTV, these links may not be sufficient. Since the net bit rate generated by uncompressed HDTV is approximately $1Gbit/s$, the transmission of several such information will exceed the available capacity of fiber optic links. In the case of storage, we again need good compression algorithms since usually we have a bulk of information that is needed to be stored and quickly accessed like in medical data archiving.

In the following section, we will summarize various techniques which reduces the information content of an image. This reduction is possible since the data represented by an image is often highly redundant. The fundamental goal of image coding is to minimize the number of bits to represent an image using this redundancy and ideally not to introduce visual quality degradation. In most practical cases, slight degradation in the output may be allowed to achieve a lower bit-rate. To what extent the data can be compressed without significant degradation depends upon the redundancy in the data, i.e., higher redundancy results in larger compression. In general, we speak of two kinds of redundancy: Statistical redundancy which can be both spatial and temporal, and subjective redundancy [5]. Subjective redundancy has to do with data characteristics that can be removed without noticeable degradation by a human observer. On the other hand, statistical redundancy is due to the similarities, correlation and predictability of the data. For example, within an image frame, it is very likely that the neighboring pixels are statistically dependent to each other. If this dependency can be exploited, we can represent a picture element in terms of M previous elements where M depends on the degree of dependency. This formulation can be extended to video since the statistical dependency also exists in the temporal domain. All of these redundancies can be eliminated without significant loss of information and therefore picture quality.

1.1 Review of Video Coding Techniques

The information-theoretic foundations of image compression date back to the work of Shannon [7]. He stated that the ultimate limit to lossless compression is determined by the source entropy, i.e. the source can be coded with zero error if the encoder uses a transmission rate equal or greater than the entropy defined as

$$Entropy = -\sum_{i=0}^{L-1} p_i log_2 p_i \quad bits/symbol, \tag{1.1}$$

for a source with L possible independent symbols with probabilities p_i. A similar kind of argument can also be carried out for lossy coding where the original pixel intensities cannot be perfectly recovered. In this case, Shannon's Rate Distortion Theorem [7] states that for a given distortion D the least rate in bits per source outcome that any coder can achieve is given by the rate-distortion function. Although lossless codes are required for legal reasons in many applications such as medical image compression, for image transmission, lossy coding is much more suitable. For lossy video compression, several techniques are found in the literature that treat image frames either pixel by pixel or blocks of pixels or high level structural forms [1]-[10]. These techniques can also be explained in terms of a source model where the simplest source model is the pel (picture element) itself. The aim is to describe the image signal by parameters of the model and to encode the model parameters instead of the image signal. The efficiency of a source model can be measured with the data rate required for encoding the model parameters and how good the model represents the input image.

The simplest coding algorithm uses the pel itself as the source model and encodes only the amplitude of the pels. This coding algorithm is called pulse code modulation (PCM). Using PCM, acceptable quality pictures can be obtained with 3 bits/pel. Higher compression ratios cannot be achieved since in this technique, each pixel is processed independently, ignoring the inter pixel dependencies, i.e., the correlation among pixel intensities is not exploited.

One way to exploit some of the correlation is Differential Pulse Code Modulation

(DPCM) where the source model is statistically dependent pels. The concept is based on the fact that the current pel can be predicted from the neighboring pels. The difference between the current and predicted pel values is then quantized and coded. DPCM is relatively simple to implement, however its redundancy reduction capability is not as good as other techniques such as transform coding which also uses statistically dependent pels as the source model. The fundamental concept of transform coding is to convert a sequence of statistically dependent pixels into an array of less dependent and information-compacted transform coefficients via an orthogonal transform. Because of the positive correlation existing in most video frame pixels, their transform coefficients almost always have a higher energy in the low frequency region but very low energy in the high frequency region. Therefore, those coefficients can be efficiently quantized and relatively easily coded, i.e the image can be represented by fewer bits. Many orthogonal transforms such as Fourier Transform (FT), Discrete Cosine Transform (DCT), Karhuenen-Loeve Transform (KLT), Hadamard Transform have been applied to compress video images [4]. Among these, the KLT decorrelates the pixels, and therefore it has the best energy compaction and optimal for a given stationary model among the other pixel-wise linear transforms. But if one considers non-stationarity which is indeed the nature of the image signal then the wavelet transform which is also well localized in the frequency and time domains, becomes optimal. Although there exists optimal transforms, DCT is the most widely used transform in image coding. The advantages of DCT are that, it is close to the optimal transform KLT, does not depend on signal statistics and does not suffer from computational complexity. Due to these advantages, DCT has been incorporated in standardized video coding algorithms which will be discussed in Section 1.2.

Another coding method that is used for low-bit-rate applications is the vector quantization (VQ) [11, 12] where the source model is again statistically dependent pels. VQ can be used instead of the scalar quantization in both pixel based and transform based coding algorithms. VQ treats a small block of images as a vector and finds the best match from a present codebook according to some distance measure. The index of the best match is then sent to the receiver, where the reconstruction is simply a table look-up process.

These coding techniques can be applied to moving pictures as well as single images, by incorporating motion detection techniques [6]. In this case, the source model becomes statistically dependent moving pels. To estimate the motion, relative displacement (motion vector) is computed so that the data in the current frame best matches the data in the previous frame. Then the best match motion vector and the difference between the current and the motion compensated previous frame, i.e. the prediction error, are sent to the decoder. To code the prediction error, one of the methods that are described above can be used. Among them, DCT has been widely used as a world standard.

The processing described in the previous paragraphs has been on a pixel-by-pixel or block-by-block basis. When the source model becomes more complicated, higher compression and improved picture quality can be achieved. Efforts in this direction lead to new coding methods which are entitled as "second generation coding techniques" [14]. These techniques are based on the fact that the destination of almost every image processing system is the human eye so if we can understand the structure of human visual system model and incorporate it into image coding, high compression is inevitable. The human visual system is first used in the field of image coding in quantization of the transform coefficients. For example, in case of DCT coefficients, since human eye is more sensitive to the lower spatial frequencies, finer quantization must be done for the DCT coefficients corresponding to these lower spatial frequencies. Later, the structure of the human visual system is incorporated into image coding. The human visual system consists of the eyes that transform light to brain signals, and the brain cortex that processes these neural signals. The lens of the eye focuses the light on the photo-receptive cells of the retina, and the retina transforms the incoming light into electrical signals that are transmitted to the visual cortex through the optical nerve. The retina consists of several types of cells with different sensitivity to shapes and luminance. Similarly, the cells in the visual cortex introduce different processing for different orientations. So, in general the human visual system can be represented as a bank of directional filters which forms the basis for second generation coding techniques. Second-generation coding techniques can be grouped into two classes. The first class

consists of the local operator based techniques such as pyramidal coding. The other class contains the contour-texture oriented techniques which attempt to describe an image in terms of contour and texture. The methods in the first group can be classified as hybrid methods since they also make use of predictive and transform coding techniques. They are classified as 'second-generation' methods because they use functions close to those of the human visual system. For example, in pyramidal image coding the image is represented as a series of bandpass images each sampled at successively lower rates [14]. If instead of pyramidal structure, we use parallel bandpass filters, the algorithm is called subband coding [15]. The reason behind using subband techniques for coding is that subsignals are more easily encoded than the original signal. Also, they resemble the direction sensitive cells in the human visual system. The contour-texture oriented techniques attempt to segment the image into textured regions surrounded by contours such that the contours correspond, as much as possible, to those of the objects in the image, contour and texture informations are coded separately [13, 14].

At this point it is worthwhile to mention the fractal coding. The basic idea behind fractal image coding is to represent an image scene by a number of transformations that generate it. The complexity of the description of the transformations should be lower than that of the original image to achieve compression. The main problems with fractal coding are the difficulty of finding suitable transformations and the computational complexity.

Although the above-mentioned techniques yield higher compression than transform and predictive coding techniques, in some cases such as very low bit-rate coding we must exploit much more of the redundancy in an image sequence than what is being exploited at present. Recently a new coding technique which is related to both image analysis and computer graphics, called object based coding (OBC), has been developed [16]. An essential difference between conventional coding methods and these new approaches is the image model they assume and the major advantage is that they describe image content in a structural way. In this approach, each object is described by three sets of parameters, namely, the shape, the motion and the color (luminance and chrominance)

Source Model	*Motion Information*	*Shape Information*	*Color Information*
Rigid 2-D object, 3-D motion	600	1300	15000
Flexible 2-D object, 2-D motion	1100	900	4000
Rigid 3-D object, 3-D motion	200	1640	4000

Table 1.1: Expected bit rates in bits/(386×288 image) for different source models [28].

parameters. The goal is to omit the transmission of color parameters in an image area which is as large as possible and to do the synthesis by using only shape and motion parameters. Object based coding algorithms can also be thought of as an extension to contour-texture oriented techniques by incorporating the motion information into the source model. If we restrict the source model to be known objects, we can increase the compression ratio further. This coding algorithm which we named as 3-D object based coding, uses an explicit model of the object beforehand. By this way, we can further decrease the information to be coded by limiting the information needed to code the shape parameter. When there is no explicit object model, the unknown objects can be treated as [17]-[19]: 1) 2-D objects (rigid or flexible) with 2-D motion, 2) 2-D objects with 3-D motion, 3) 3-D objects (rigid or flexible) with 3-D motion. The average bit rates in bit/CIF(386×288) frame is given in Table 1.1 for these object models [27, 28].

Using explicit models for the object has also been addressed by many researchers [20]-[26]. Since dealing with unknown objects is an extremely difficult problem, mostly head and shoulders type scenes are used for application of 3-D object based coding algorithms. These schemes are expected to open up new applications in image coding techniques which cannot be obtained by conventional waveform coding. For instance,

1.2 Video Coding Standards

Various organizations have been involved in the development or promotion of the standardization of data compression algorithms. Among these organizations, mainly ITU-TS (International Telecommunications Union-Telecommunication Standardization Sector) which is formed after the reorganization of CCITT (International Telegraph and Telephone Consultive Committee) and CCIR (International Consultive Committee of Broadcasting), and ISO (International Organization for Standardization) deal with the standardization of video coding algorithms. To develop the standards ITU-TS and ISO committees solicit algorithm recommendations from a large number of companies, universities and research laboratories. The best of those submitted are selected on the basis of image quality and compression performance. Among the standards developed by these committees, we will mostly concentrate on H.261 and MPEG because of their relation to videotelephony.

1.2.1 CCITT H.261 standard

CCITT Study Group XV formed a Specialist Group in 1984 toward a coding standard for visual telephony. Efforts from this group has resulted in a standard CCITT Recommendation H.261 [30] approved in December 1990 [30]. H.261 represents the state of the art in picture coding for low and medium bit rates. It is primarily intended for videophone and teleconferencing using ISDN channels at $p \times 64$ kbps, $p = 1, 2, ..., 30$ for combined video and audio. CCITT H.261 has been demonstrated to be effective in providing videoconferencing application where the backgrounds rarely change. The quality depends on the value of p and it has been shown that for $p = 6$ the quality is satisfactory.

CCITT H.261 uses a CIF (Common Intermediate Format) or QCIF (Quarter CIF) image format, a DCT based coding algorithm and a selectible frame rate ranging from 30 frames/s to 7.5 frames/s.

1.2.2 MPEG phases

MPEG is a acronym for Moving Picture Expert Group which is under ISO-IEC/JTCI/SC29/WG11 and started its activity in 1988 [31]. It conducts liaison exchanges with ITU-TS and other relevant standards agencies. The first phase of MPEG, MPEG-1, is a standardization of coding for storage . Its activities are based on the premise that video and its associated audio can be stored and retrieved at about $1.5 Mbits/s$ at satisfactory quality. It has also been shown that when MPEG-1 algorithm is applied to CIF (Common Intermediate Format) image sequences at $30 frames/s$, we can get a quality similar to that of VHS tape at about $1.2 Mbits/s$ video rate. The draft of MPEG-1 has been finalized in June 1992. Its envisioned areas of application include electronic publishing, video games, entertainment, videophone, videomail, videoconferencing and education.

The second phase of audiovisual standardization, MPEG-2 is intended for higher data rates than MPEG-1. It is also a generic standard which is intended to serve a wide range of applications. The image quality is optimized in ranges from about 2 to $15 Mbits/s$ over cable, satellite, and other broadcast channels, as well as for Digital Storage Media (DSM) and other communications applications and various video formats (both progressive and interlaced) can be supported. The development of MPEG-2 was begun in November 1991 and aimed to be completed by the end of November 1993. MPEG is working jointly with the CCITT SGXV "Experts Group on ATM Video Coding" in this new phase of work.

In 1992, work is directed towards coding at very low bit rates, several tens of Kbits/s. The first studies on very low bit rate video coding concentrated on modification such as reducing the image size, frame rate etc., to existing standards. After completion of this short term objective, several groups moved towards more novel coding approaches within MPEG-4 which is initiated by ISO (MPEG-3 is incorporated in MPEG-2). This work has begun officially in 1993 and scheduled to result in a draft specification in 1997. This work mainly requires the development of fundamentally new algorithmic techniques such as object based approach in the very low bit rate coding area.

1.2.3 COST211

It is worthwhile to mention about the COST211 project at this point because of its contributions to the existing coding standards [32]. COST211 is one of the projects within the telecommunication activities of COST (European Cooperation in the Field of Scientific and Technical Research), where the major research concern is the redundancy reduction techniques for coding of video signals. It was initiated in 1977 with the participation of seven European countries. The first phase of the project was completed in 1982 resulting in a specification of a 2Mbit/s codec for videoconference signals which was indeed the basis of CCITT Recommendation H.221. After the completion of the first phase COST211bis was initiated in the same year with the objective of examining the possibility of applying redundancy reduction techniques to the digital transmission of visual teleconferencing signals and of broadcast quality TV signals. The most notable achievement of this project was the contributions made to the CCITT Recommendation H.261 for $p \times 64kbps$ video coding. The bulk of this video telephony standard resulted directly from the work undertaken by COST211bis. The last activity of COST211bis, before being completed in 1990, was the first studies of videocoding for Broadband-ISDN (B-ISDN) using an Asynchronous Transfer Mode (ATM), allowing variable bit rates. This item was further studied in the project COST211ter which was initiated in 1990 and dealt with redundancy reduction techniques for coding of video signals in multimedia services. In 1991, due to the growing interest in the developments in digital mobile networks, COST211ter members extended the scope of the project to cover the field of very low bit rate (8-32 kbps) coding of moving images. Together with modifications to existing standards such as H.261, COST211ter also considers novel techniques such as object based coding, with the aim of very low bit rate video coding [33]. If not extended, this project will be completed in 1995 with a proposal for a new standard for very low bit rate video coding.

1.2.4 Hardware implementation

Once the coding standards have been established several companies deal with the VLSI implementation of them. For H.261, we can mention GEC Plassey, LSI Logic, SGS-Thompson, GPT CLI. C^3 has developed an MPEG-1 decoder chip under the name CL 450 and in the near future they will produce a JPEG/H.261/MPEG1 codec.

It is worth mentioning some of the available videophones which can operate over the existing telephone lines, at this point although they do not yet use the novel techniques considered in MPEG4 [34]. Up to now, AT&T, British Telecom/Marconi, COMTECH Labs and ShareVision produced videophones. AT&T Videophone 2500 is working with 16.8 and 19.2 Kbit/sec modems. It uses motion compensated DCT for video compression. British Telecom/Marconi Rel 2000 Videophone is working with 9.6 and 14.4 Kbit/sec modem. It uses H.261 flavor motion compensated DCT video compression. COMTECH Labs STU-3 Secure Videophone's data rate is 9.6 Kbit/sec and it uses motion compensated DCT video compression as ShareVision does.

1.3 Scope and Outline of the Thesis

Due to growing interest in very low bit rate digital video (about 10 kbps), a significant amount of research focused on object based video compression [16]-[28] as stated in the previous sections. Engineers became interested in object based coding because the quality of digital video obtained by hybrid coding techniques, such as CCITT Rec. H.261 [30], is deemed unsatisfactory at these very low bit rates. Studies in object based coding employ object models ranging from general purpose 2-D or 3-D models [17, 19, 27] to application specific wire-frame models [20]-[26]. One of the main applications of object based coding has been the videophone, where scenes are generally restricted to head and shoulder type images. In many proposed videophone applications, the head and shoulders of the speaker are represented by a specific wire-frame model which is present at both the receiver and the transmitter. Then, 3-D motion and structure

estimation techniques are employed at the transmitter to track the motion of the wire-frame model and the changes in its structure from frame to frame. The estimated motion and structure (depth) parameters along with changing texture information are sent and used to synthesize the next frame in the receiver side.

Traditionally, the adaptation (fitting) of a generic wire-frame model to the actual speaker and motion estimation have been handled separately. Many of the existing methods consider fitting a generic wire-frame to the actual speaker using only the initial frame of the sequence [20, 35]. Thus, the modification in the z-direction (depth) is necessarily approximate. For subsequent frames, first the 3-D global motion of the head is estimated under rigid body assumption, using either point correspondences [20, 24, 36] or optical flow based formulations [25, 29]. Then, local motion (due to facial expressions) is estimated making use of Action Units (AU) described by Facial Action Coding System (FACS) [25]. Recently, Li *et al.* [26] proposed a method, to estimate both the local and global motion parameters from the spatio-temporal derivatives of the image. However, his method also requires *a priori* knowledge of the AU's and initial fitting of the wire-frame to the actual speaker.

In this dissertation, we propose a novel formulation where 3-D global and local motion estimation and the adaptation of the wire-frame model are considered simultaneously within an optical flow based framework including the photometric effects (changes in the shading due to 3-D rotations) of motion. Although, the utility of photometric cues in 3-D motion and structure estimation has recently been discussed [37]-[38], photometric information was not used in the context of motion estimation for videophone applications beforehand. The main contributions of this study are: (i) a flexible 3-D wire-frame model has been used where the X, Y and Z coordinates of the nodes of the wire-frame model are allowed to vary from frame to frame so as to minimize the error in the optical flow equation, and (ii) photometric effects are included in the optical flow equation. The proposed adaptation of the wire-frame model serves for two purposes that cannot be separated: to reduce the misfit of the wire-frame model to the speaker in frame $k-1$, and to account for the local motion deformations from frame $k-1$ to frame k

without using any *a priori* information about the AU's. The simultaneous estimation formulation is motivated by the fact that estimation of the global motion, local motion and adaptation of the wire-frame model including the depth values are mutually related; thus a combined optimization approach is necessary to obtain the best results. Because an optical flow based criterion function is utilized, computation of the synthesis error is not necessary from iteration to iteration; thus, resulting in an efficient implementation. The synthesis error at the conclusion of the iterations is used to validate the estimated parameters, and to decide whether a texture update is necessary.

In Chapter 2, we review the 3-D object based coding scheme together with the problems encountered at each step. In Chapter 3, we give an overview of the 3-D motion estimation methods in the field of object based image coding. In addition we propose a new feature based motion estimation algorithm and make comparison with the existing ones. In Chapter 4 the formulation of simultaneous motion estimation and wire-frame adaptation problem including the photometric effects of motion are given. In that chapter, we also discuss the problem of the illumination direction estimation and give an efficient algorithm for the proposed simultaneous estimation method. Experimental results on simulated and real video sequences are presented in Chapter 4 to demonstrate the effectiveness of the proposed methods. Finally future directions and conclusions are given in Chapter 5.

Chapter 2

3-D OBJECT BASED CODING

As stated in Chapter 1, coding schemes based on modeling the 3-D scene yield higher compression ratios compared to other techniques. Due to this advantage, 3-D object based coding methods have received much attention and could well form the basis of the next generation visual communication services. Research on 3-D object based coding has been going on since the early 1980's. Several similar projects are currently being pursued by various image coding groups [16]-[28]. In 3-D object based coding, both the encoder and decoder contain either a special 3-D model or special knowledge of the object to be coded. In general, describing a scene is a complicated task which is widely investigated in the field of computer vision. Therefore most research on 3-D object based coding has concentrated on restricted scenes such as head-and-shoulder scenes which are typical to video-phone applications.

A block diagram of a 3-D object based coding scheme for facial images is shown in Fig. 2.1. At the transmitting side, images are analyzed under the assumption that they show the head and shoulders of a person. Basic properties such as geometric properties of the head, surface color and texture are extracted and transmitted initially. As the head moves, motion parameters described by the global motion of the head and the local motion due to the facial expressions are detected and transmitted. At the receiving side, the image is synthesized using these estimated motion parameters assuming that both

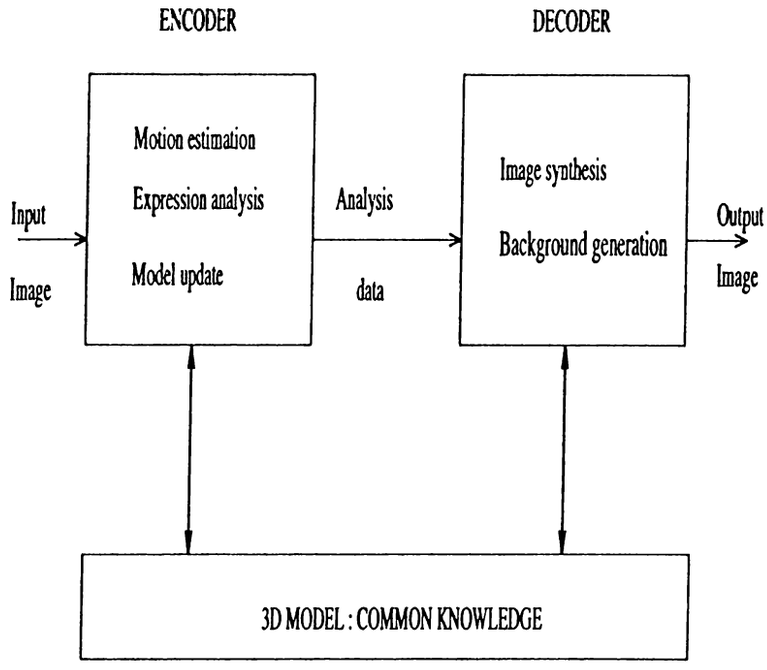

Figure 2.1: Main blocks in 3-D object based coding

the transmitter and the receiver possess the same 3-D facial model at the beginning. The system is indeed composed of three stages: construction and adaptation of a 3-D model of the face, analysis of the input image to extract the motion and structure parameters, and a synthesis of the image at the receiving side.

2.1 Model Construction and Adaptation

2.1.1 3-D model construction

The 3D modeling is first used in computer graphics for facial animation. The majority of the work in this field involved modeling the surface of a face with polygons and then rendering the surface with continuous shading. Parke [39] was the first to propose that a parameterized model of a face could be used for a form of videotelephony. He models the face by using connected networks of polygons where the vertex position

values of the polygons are determined by photogrammetrically measuring the surfaces of real faces. The depth map of a face can also be obtained by scanning the head using collimated laser light [40] or sound captors [41]. After obtaining the depth map, the 3-D polygonal representation is obtained mostly through a triangulization procedure where small triangles are put in high curvature areas and larger ones at low curvature areas. This triangular mesh, which is called the wire-frame, is put into computer memory as a set of linked arrays. One set of arrays give the X, Y, Z coordinates of each triangle vertex and another set gives the addresses of the vertices forming each triangle. There are several wire-frame models used by different research groups [23], [42]-[53]. For example, Terzopoulos uses a non-uniform mesh of polyhedral elements whose size depend on the curvature of the neutral face and muscular contractions [44, 45]. Adaptive division of the wire-frame such as division according to luminance deviations [48] or according to the semantic characteristics of a specific speaker's face [49], is also possible. In this study, we use the modified CANDIDE model [53] developed by Welsh (Fig. 2.2). This model contains a full description of the face with enough number of triangles [23].

2.1.2 Wire-frame adaptation

As stated previously, in 3-D object based coding, both the transmitter and the receiver have the 3-D wire-frame model of a generic face as a common knowledge. The image is synthesized at the receiver by modifying the wire-frame using the transmitted parameters obtained by analysis and recognition procedures carried out at the transmitting side. The main parameters that are transmitted are the motion vectors due to global and local changes of the head and face. The accuracy of tracking motion of the wire-frame model from frame to frame strongly depends on how well the wire-frame model matches the actual speaker in the scene. Since size and shape of the head and position of the eyes, mouth, nose vary from person to person, it is necessary to modify the 3D model according to the particular features of a person's face in an input image sequence. Thus, one of the challenging problems in 3-D object based coding of facial image sequences is to adapt a generic wire-frame model developed for an average speaker to fit the actual

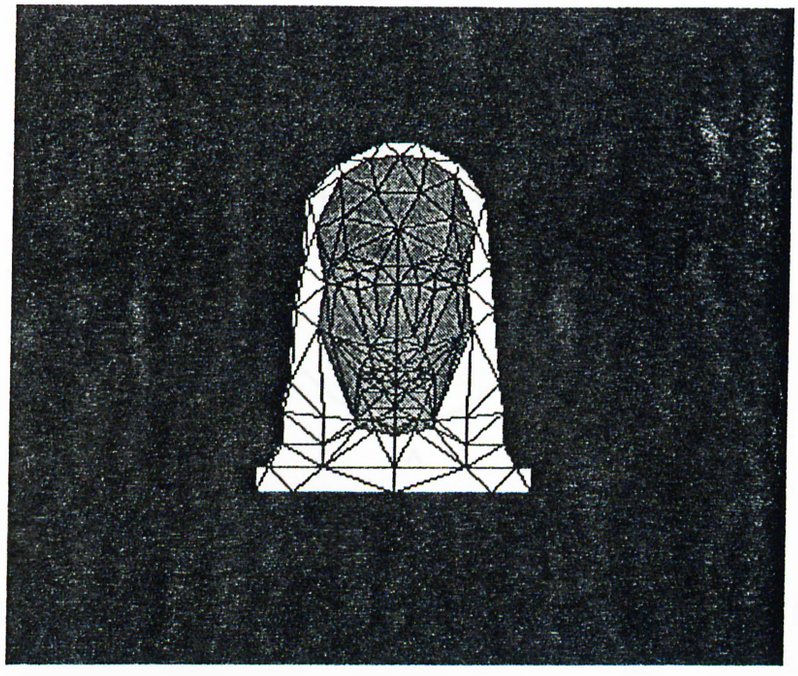

Figure 2.2: Wire-frame model of a typical head-and-shoulder scene where the gray region refers to the face.

speaker.

Initial studies on 3-D object based coding have fit the wire-frame model to the actual speaker manually. Aizawa *et al.* [20, 50] use 3-D affine transformation to match the frontal view of a particular face and its four feature points (tip of the chin, temples, a point midway between the left and right eyebrows) to the model. The four feature points are interactively specified (Fig. 2.3). Then the position of each vertex of the wire-frame model forming a contour along the cheek to chin is adjusted precisely to match the frontal view of the face to the wire-frame model. The positions of other vertices are adjusted proportionally to the shift of vertices on the contours. The depth of the feature points are estimated using the scale parameters (in x and y directions) of the wire-frame model and the depth of other vertices are adjusted proportionally in the direction of the center of the head.

Kaneko *et al.* [24] also use an interactive marking of the feature points. They use seven points; top of the head, tip of the chin, left and right cheeks-upper and lower

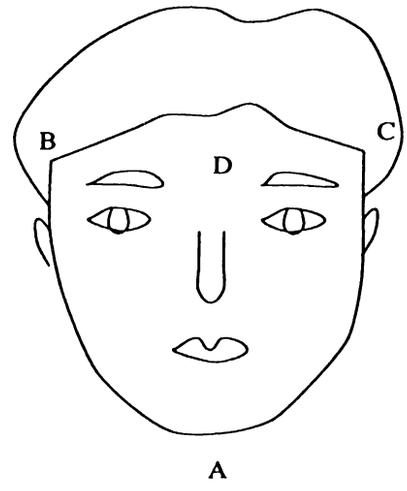

Figure 2.3: Feature points to adjust the wire-frame

positions, and a point midway between the right and left eyes, in modifying size and shape of the model. The affine transform, $x' = ax+by+c, y' = dx+ey+f$, transforms the point (x, y) to the point (x', y'). After finding the unknown coefficients by using the feature points, the affine transform is applied to the coordinates of each vertex constituting the model. The depth is modified by using the scaling factor $\sqrt{(a^2 + e^2)/2}$. Huang *et al.* [21] use spatial and temporal gradients of the image to estimate the length and width of the face and scale the wire-frame approximately. Then an interactive procedure specifies the location of the feature points on the face and translates the wire-frame vertices according to these points. Recently, Huang *et al.* [54] propose an automatic feature point extractor using some assumptions about the input image such as the user's face must appear at about the center of the input image and must be at least one-sixteenth the size of the input image. This method has not been applied to 3-D object based coding yet.

Another way of adaptation of the wire-frame is to use snakes or ellipses to find the face borders. Recently, Reinders *et al.* [35] consider automated global and local modification of the 2-D projection of the wire-frame model in the x and y directions. They segment the image into background-foreground, face, eyes and mouth, and approximate the contours of the face, eyes and mouth with ellipses in order to get an estimate for control features necessary for global transformation of 3-D wire-frame. Then local

transformations are performed using elastic matching techniques. However, they have applied an approximate scaling in the z-direction (depth) since they use only a single frame. Waite and Welsh use snakes to find the boundary of the head which is found to be a fairly robust method [55]. However, they do not consider the modification of the depth values, either.

2.2 Image Sequence Analysis

Once the estimation of the pose of the face has been achieved, an analysis of the facial image can take place. In the analysis of facial image sequences both the head motion parameters (global motion) and the facial expression parameters (local motion) must be estimated. The head motion parameters are due to 3-D motion of the whole head or change in viewpoint (global motion), and facial expression parameters are due to the motion of elements such as mouth, eyebrows, eyes caused by the changes in their shapes (local deformations).

A general overview of 3-D rigid body motion and structure estimation methods can be found in [56]. In Chapter 3, we will further concentrate on global motion estimation techniques in the context of 3-D object based coding. For facial expression parameter estimation (local motion), there has been extensive research based on Facial Action Coding system (FACS) [57]. FACS starts out from visual changes in the facial expressions which are specified in terms of Action Units (AUs) being single muscles or clusters of muscles. According to FACS, a human facial expression can be divided into approximately 44 basic AUs and all facial expressions can be produced by the combination of these AUs. In 3-D object based coding AUs are also widely used [25, 46]. Once the displacements of control points related to each AU are detected, the wire-frame model can be deformed according to this knowledge. Several algorithms have been proposed to do this facial analysis. Aizawa [47] used a tree structure for the efficient classification of AUs. The characteristic changes for each AU are investigated and the most characteristic AU is classified from the detected displacements of the positions of

the feature points. Displacements of the classified AU are removed from the detected ones and the secondary characteristic AU is classified. This process is continued until all the detected displacements vanish. Forchheimer [16, 58] used the residual error field after correcting the global motion, as the displacement vector. He uses an estimate of AUs through the relationship

$$\Delta d = Aa$$

where a is a vector of AU parameters, Δd is the set of displacement vectors and A is the matrix describing the effect of AUs. Kaneko [24, 59] extracted the shape of the mouth, the eyes etc. using a thresholding operation within a rectangular area. From this result, he marked several distinctive points to represent the changes in the shape of characteristic features. Choi [60] formulate an AU as a vector whose components are the deforming velocities of the wire-frame nodes. He again used the constraint between the velocity and spatio-temporal gradient of the brightness of two consecutive frames to estimate the AU intensities.

Deformable contour models are also used in the field of 3-D object based coding to track the non-rigid motions of facial features in the image. The most significant work is done by Terzopoulos [61]-[63]. His model parameters use three layered deformable lattice structures for facial tissue. The three layers correspond to the skin, the subcutaneous fatty tissue, and the muscles. His method is only capable of tracking features when the motion is very small. Sferedis [64] uses an unsupervised tracking of the facial features. His method is a combination of morphological edge detector and a matching technique. The method is strongly dependent on the quality of the edge detection algorithm. Huang *et al.* [21] use splines to track features, i.e. eyes, eyebrows, nose and the lips. When the features are not visible, they use a database of vectors, called action vectors, each corresponding to the maximum possible motion of one of the control points. The feature and hence the control points are tracked across the image sequence by using the information in the database. Yuille *et al.* use deformable templates for detecting and describing features of faces [65]. The template consists of a collection of parameterized curves which, taken together, describe the expected shape of the feature to be detected in the

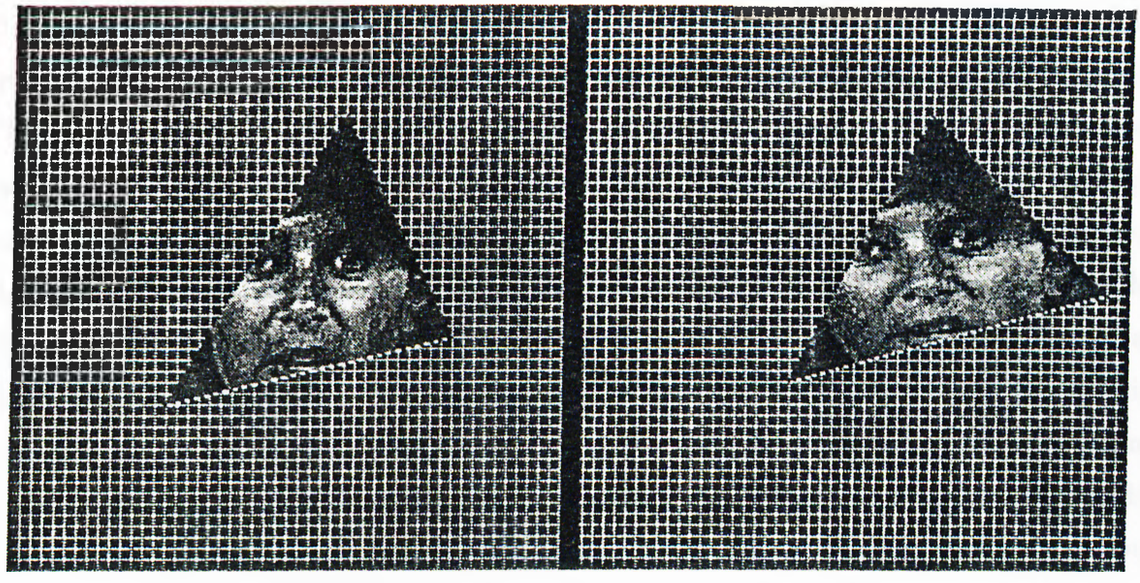

Figure 2.4: Texture mapping before and after processing. The background is partitioned into squares in order to show how the nodes of the triangle change after processing.

image. The template interacts dynamically with the image by altering its parametric values to minimize the energy function. Later, Welsh [66] modified this method by considering the geometric configuration. He normalized the image before tracking the feature in such a way that the feature in the image attains a standard shape.

In all the methods described above, global and local motion estimation problems are treated separately, which in reality cannot be separated. Recently, Li *et al.* [26] proposed a method to recover both the local and global motion parameters together from the spatio-temporal derivatives of the image. However, his method also requires *a priori* knowledge of the AU's.

2.3 Image Sequence Synthesis

Procedures of synthesizing the facial images at the receiving side consist of deforming the wire-frame model through the global and local motion parameters and mapping the

texture of the first frame onto the surface of the deformed wire-frame model. Texture mapping is an important task in order to get natural and realistic facial images [67]-[69]. This topic is widely investigated in the field of computer graphics since it is an easier way to create the appearance of complex surface details without having to go through modeling and rendering every 3-D detail of a surface. In the context of 3-D object based coding texture mapping involves the projection of the 2-D facial image onto the triangles forming the 3-D wire-frame model, i.e. the values of pixels inside a triangular area are taken from the original image and assigned to the corresponding triangle in the 3-D shape model. Fig. 2.4.a shows one of the triangles constituting the wire-frame model superimposed on the array of pixels in the initial frame and Fig. 2.4.b shows the corresponding triangle and pixels in the output image after being processed by rotation, translation and deformation. In order for texture mapping to be independent of the size and position of the triangles, each side of a triangle is first divided into equal segments and the position of a pixel inside a triangle is represented in terms of its relative position inside the triangle.

2.4 Problems of 3-D Object Based Coding

Although 3-D object based coding opens up the possibility of image transmission at extremely low bit-rates, several problems such as generality and analysis errors limit its practical usage.

Modeling objects is one of the important issues in 3-D object based coding. The assumption that the input images always consist of a moving head and shoulder is not appropriate for practical use. However, dealing with unknown objects is an extremely difficult problem. The second problem is the presence of analysis and synthesis errors. These errors are due to mismatch of the wire-frame, inaccurate motion estimation and rapidly changing texture information and can cause serious artifacts in the decoded images.

To cope with these problems a practical solution is to use a hybrid coding system which is a combination of 3-D object based coding and conventional waveform coding. A general description of a hybrid coding system is given in Fig. 2.5 [46]. The transmitter includes a local decoder which enables the system to detect the regions where the model does not fit. At the transmitter, image synthesis is performed using the analysis parameters extracted at the analysis part. The differences between the synthesized images and the input images are coded by the conventional waveform coder. The information extracted at the analysis part can also be used to control the waveform coder to avoid unnecessary waveform information being transmitted. If there is a complete misfit between the model and the input image, then one can again use the conventional waveform coding to code the entire image instead of 3-D object based coding. It is shown in [46] that incorporation of 3-D object based coding into conventional waveform coding improves the signal to noise ratio (SNR) at very low transmission rates such as 16 kbps, especially when the face of the person in the input image sequence widely moves.

Another way to cope with the analysis and synthesis errors is to improve the algorithms of image analysis so that the motion and structure estimations can be done with the highest possible accuracy. In Chapter 4, we will give a new formulation to achieve this goal. The formulation proposed takes into account the errors due to misfit of the wire-frame to the actual speaker, global and local motion estimation errors.

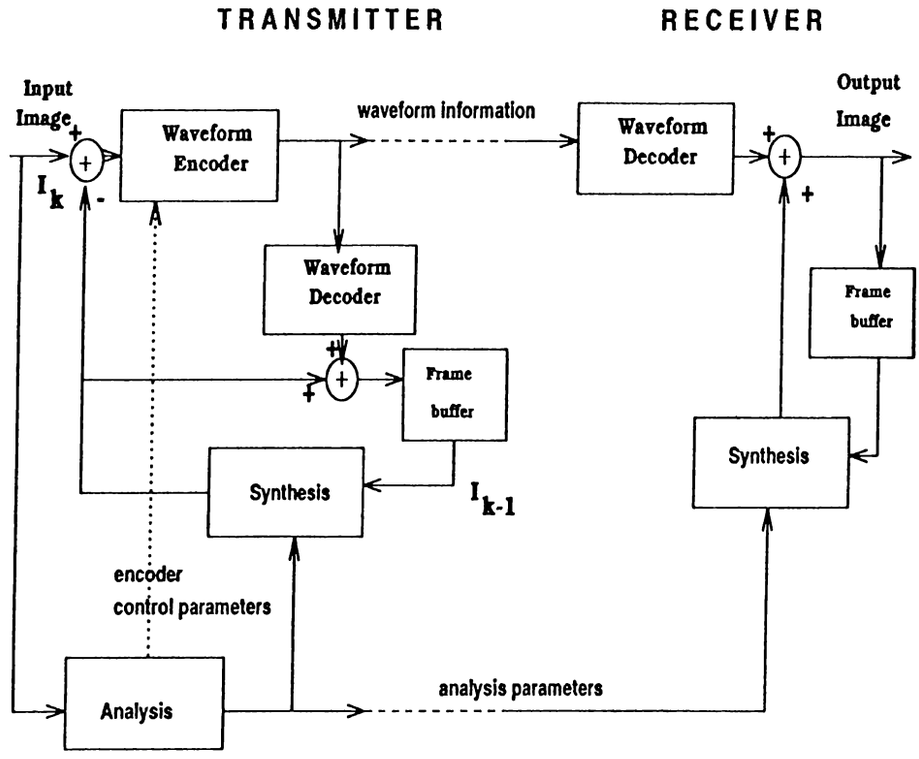

Figure 2.5: Block diagram of a hybrid coding system.

Chapter 3

GLOBAL MOTION ESTIMATION

Estimating the motion of objects in the field of view from the image sequences captured by a television camera is one of the important problems in computer vision and image processing. An understanding of the three dimensional (3-D) motion makes it possible to predict the future locations and configurations of the moving objects which can be of great importance in image coding, remote sensing and military applications. Although the objects around us are 3-D and perform 3-D motion, TV cameras can only capture their two dimensional (2-D) projections. Therefore, the nature and parameters of 3-D motion must be estimated from these 2-D projections. In this Chapter, we will review the 3-D motion estimation methods in the field of 3-D object based image coding, propose an improved feature based motion estimation algorithm and make comparison with the existing ones.

3.1 Motion in the Image Plane

In the literature, two projection models of image formation have been widely used: perspective projection and orthographic projection [70]. Perspective projection is the most familiar projection technique, since the images formed by eyes and by lenses on

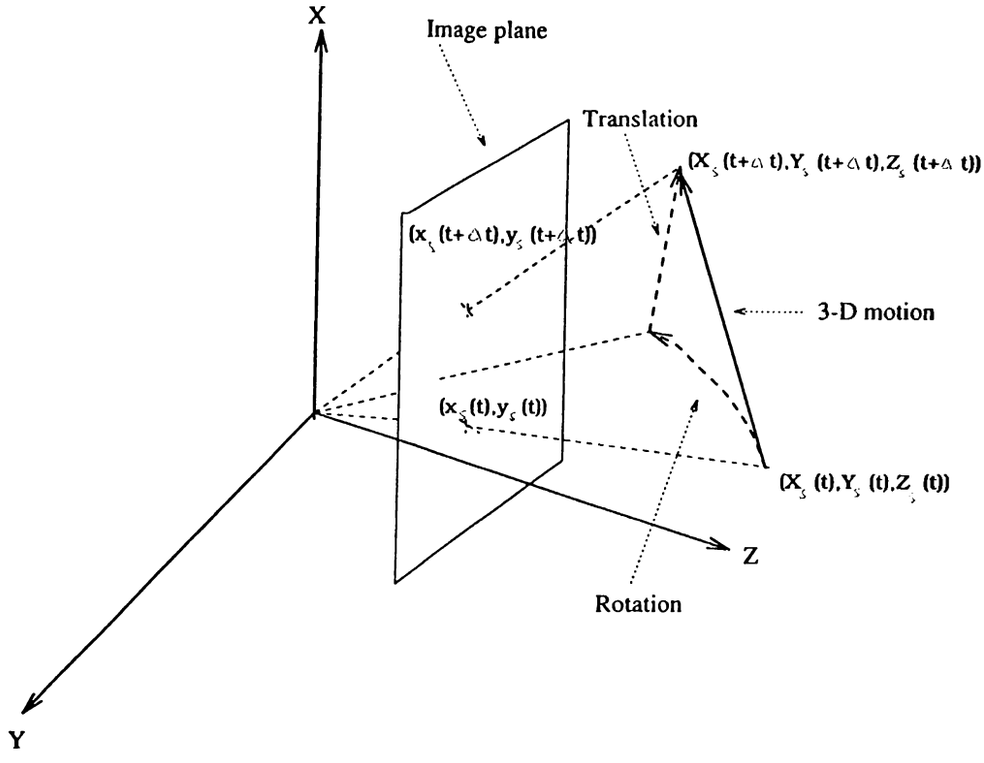

Figure 3.1: Camera model for perspective projection.

intensity sensitive media are perspective projections. The perspective projection conveys depth information by making distant objects smaller than the near ones. On the other hand, orthographic projection shows only the correct or true x and y sizes of an object.

The motion estimation problem has been investigated mainly for perspective projection [71]-[73] with some work on orthographic projection [74]. However, the effect of perspective projection decreases when the object size or the variation of the surface depth is small with respect to the distance to the camera. In 3-D object based coding, the imaging process can also be considered as orthographic projection assuming that the camera is far enough away that perspective effects should not make any great contributions. Throughout this work we will also concentrate on orthographic projection because of the above reason and the ease of formulations.

Fig. 3.1 shows how the changes in 3-D appear in 2-D (image plane) due to the projection of the object point onto the image plane. This is a commonly used version

of projection where the camera is oriented along the positive z-axis, i.e. the normal of the image plane is parallel to the z-axis. In the figure, the image plane is at the focal plane of the camera, f. $X_s(t), Y_s(t), Z_s(t)$ are the coordinates of a point s on the 3-D object and $x_s(t), y_s(t)$ are the coordinates of its projection onto the image plane. For perspective projection,

$$\begin{aligned} x_s(t) &= f\frac{X_s(t)}{f + Z_s(t)} \\ y_s(t) &= f\frac{Y_s(t)}{f + Z_s(t)} \end{aligned} \tag{3.1}$$

and for orthographic projection where f is assumed to be large with respect to the depth z,

$$\begin{aligned} x_s(t) &= X_s(t) \\ y_s(t) &= Y_s(t). \end{aligned} \tag{3.2}$$

3.2 Three Dimensional Rotation and Translation

In order to estimate the motion in 3-D we have to identify how motion changes the structure of the scene. Let $[X_s(t) \quad Y_s(t) \quad Z_s(t)]^T$ be the vector of the coordinates of a particular point s of a moving object at time t and S refers to the object which is the set of all such points. If we assume that the object is rigid and subject to small rotation, we can express the position of s at time $t + \Delta t$ given its position at time t as,

$$\begin{bmatrix} X_s(t+\Delta t) \\ Y_s(t+\Delta t) \\ Z_s(t+\Delta t) \end{bmatrix} = \begin{bmatrix} 1 & \omega_Z & -\omega_Y \\ -\omega_Z & 1 & \omega_X \\ \omega_Y & -\omega_X & 1 \end{bmatrix} \begin{bmatrix} X_s(t) \\ Y_s(t) \\ Z_s(t) \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}, \qquad \forall s \in S \tag{3.3}$$

where ω_X, ω_Y, and ω_Z are the rotational displacements around the X, Y and Z axes, respectively, and T_X, T_Y, and T_Z are the translational displacements along the X, Y and Z axes, respectively. Under orthographic projection along the z-direction, Eq. 3.3

becomes,

$$\begin{aligned} x_s(t+\Delta t) &= x_s(t) + \omega_Z y_s(t) - \omega_Y Z_s(t) + T_X \\ y_s(t+\Delta t) &= -\omega_Z x_s(t) + y_s(t) + \omega_X Z_s(t) + T_Y, \end{aligned} \tag{3.4}$$

$$\forall s \in S.$$

As the only information we can obtain from the 2-D images are the projections of the 3-D objects around us, we have to estimate the rotational and translational displacements from Eq. 3.4.

3.3 Methodologies for Motion Estimation

A general overview of 3-D motion and structure estimation methods can be found in [56]. In the context of 3-D object based coding, we can divide the methods developed for the computation of motion from image sequences into two categories: feature based and optical flow based motion estimation. The first of these is based on extracting a set of 2-D features in the images, establishing inter-frame correspondences between these features and computing the 3-D motion parameters from the displacements of these 2-D image features. Aizawa and Harashima [20, 25] estimate the 3-D motion parameters and depth information of the head by this approach which will be given in the next section in detail. In order to extract the 3-D motion they use the rigid body assumption, i.e. they do not take into account the local deformations. Welsh also gives a least-squares method to estimate only the global motion parameters [22]. The drawback of these methods is that, extracting and establishing feature correspondences is a difficult task due to hidden and false features. However, feature-based methods are widely used in 3-D object based coding due to their low computational complexity.

The other approach is based on computing the optical flow [75], the 2-D field of instantaneous velocities of gray levels in the image plane. In this approach, there is no need to define correspondences between features of successive images. However, most

of the methods reported in the literature for 3-D motion estimation based on optical flow consider only rigid body motion where no deformation of the body is allowed as a function of time [29]. Recently, Li *et al.* [26] proposed a method, to recover both the local and global motion parameters from the spatio-temporal derivatives of the image. In the following sections we will review the abovementioned methods together with various improvements.

3.4 Feature Based Motion Estimation

Among the methods in the literature about feature based motion estimation, MBASIC, recently proposed by Aizawa *et al.* [20], is a simple and effective iterative algorithm for 3-D motion and depth estimation under the orthographic projection. MBASIC algorithm, reviewed in the following section, requires a set of initial depth estimates which are usually obtained from a generic wire-frame model. Although the performance of MBASIC is very good when the initial depth parameters contain about 10% error or less, it degrades with the increasing amount of error in the initial depth estimates. But in practical applications the initial depth estimates may contain 30% or more error due to problems in scaling the generic wire-frame model to a particular speaker. Thus, in Section 3.4.1 we propose a modification to the MBASIC algorithm which makes it more robust to errors in the initial depth estimates with a small increase in its computational load, thus making it more useful in practical applications. We also discuss the computational complexity of the improved algorithm; compare the performance of the MBASIC algorithm and the improved algorithm in the presence of various degrees of inaccuracy in the initial depth estimates, and show that the improved algorithm converges to the true motion and depth parameters even in the presence of 50% error in the initial depth estimates.

3.4.1 MBASIC algorithm for motion estimation

Each iteration of the algorithm is composed of two steps: 1) Determination of motion parameters given the depth estimates from the previous iteration, and 2) update of depth estimates using the new motion parameters.

In Eq. 3.4, there are five unknown global motion parameters w_X, w_Y, w_Z, T_X and T_Y, and an unknown depth parameter $Z_s(t)$ per given point correspondence $(x_s(t), y_s(t))$ and $(x_s(t+\Delta t), y_s(t+\Delta t))$. The equation has a bilinear nature, since $Z_s(t)$ multiplies the motion parameters. It is thus proposed to solve for the unknowns in two steps:

Step 1. Given at least three corresponding coordinate pairs $(x_s(t), y_s(t))$ and $(x_s(t+\Delta t), y_s(t+\Delta t)$ and their depth parameters $Z_s(t)$, $s = 1, \ldots, N$, $N \geq 3$, we can rearrange Eq. 3.4 to lead to $2N$ equations in 5 unknowns:

$$\begin{bmatrix} x_s(t+\Delta t) - x_s(t) \\ y_s(t+\Delta t) - y_s(t) \end{bmatrix} = \begin{bmatrix} 0 & -Z_s(t) & y_s(t) & 1 & 0 \\ Z_s(t) & 0 & -x_s(t) & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_X \\ \omega_Y \\ \omega_Z \\ T_X \\ T_Y \end{bmatrix}. \tag{3.5}$$

Hence, the motion parameters can be solved from Eq. 3.5 using the least squares method.

Step 2. Once the motion parameters are found, we can estimate the new Z_i values using

$$\begin{bmatrix} x_s(t+\Delta t) - x_s(t) - \omega_Z y_s(t) - T_X \\ y_s(t+\Delta t) - y_s(t) + \omega_Z x_s(t) - T_Y \end{bmatrix} = \begin{bmatrix} -\omega_Y \\ \omega_X \end{bmatrix} \begin{bmatrix} Z_s(t) \end{bmatrix}. \tag{3.6}$$

which is again obtained from Eq. 3.4. Here, we have one equation pair, per given point correspondence, which can be solved for $Z_s(t)$ in the least squares sense.

The procedure consists of repeating steps 1 and 2 until the estimates no longer change from iteration to iteration. However, it has been observed that unless we have reasonably good initial estimates for $Z_s(t)$, $s = 1, \ldots, N$, the two-step iteration may converge to a local but not global minimum. In the next section, we propose a solution to this problem.

3.4.2 Improved motion and depth estimation by random perturbation

In the MBASIC algorithm there is a strong correlation between the error in the motion parameters and the error in the depth parameters. This can be seen from Eqs. 3.5 and 3.6, as the errors in the depth parameters are fed back on the motion parameters and vice versa, iteratively. To circumvent this problem, we define an error criterion (see Eq. 3.7 below), and update $Z_s(t)$ in the direction of the gradient of the error with a proper step size (instead of computing them from Eq. 3.6) at each iteration. To facilitate convergence of the estimates to their correct values, we also perturb the depth estimates in some random fashion after each update. The motion parameters are still computed from Eq. 3.5 after each update/perturbation of the depth estimates. The principle used here to update the depth parameters is similar to stochastic relaxation, where each iteration consists of perturbing the state of the system in some random fashion before computing the next state, with the ultimate goal of convergence to the global optimum [76]. The update in the gradient direction increases the rate of convergence as compared with totally random perturbations of $Z_s(t)$. In our experiments, the random perturbations are generated as samples of uniform or Gaussian distributed numbers. The magnitude of perturbations decreases with the number of iterations, so that convergence should result. The proposed algorithm with improved convergence characteristics is as follows:

1. Set the iteration counter $m = 0$.

2. Given at least three corresponding coordinate pairs $(x_s(t), y_s(t))$ and $(x_s(t + \Delta t), y_s(t + \Delta t))$ and their depth parameters $Z_s(t)$, $s = 1, \ldots, N$, $N \geq 3$, determine the motion parameters from Eq. 3.5.

3. Compute $(x_{s_{(m)}}(t + \Delta t), y_{s_{(m)}}(t + \Delta t))$, the coordinates of the matching points that are predicted by the present estimates of the motion and depth parameters, using

Eq. 3.4. Compute the model prediction error

$$E_m = \frac{1}{N}\sum_{s=0}^{N} e_s \tag{3.7}$$

where

$$e_s = (x_s(t+\Delta t) - x_{s_{(m)}}(t+\Delta t))^2 + (y_s(t+\Delta t) - y_{s_{(m)}}(t+\Delta t))^2. \tag{3.8}$$

Here $(x_s(t+\Delta t), y_s(t+\Delta t))$ are the actual coordinates of the matching points which are given.

4. If $E_m < \epsilon$, stop the iteration,

   Else, set $m = m+1$, and perturb the depth parameters as

$$\hat{Z}_{s_{(m)}}(t) \leftarrow \hat{Z}_{s_{(m-1)}}(t) - \beta g(Z_s(t)) + \alpha^m \Delta_s, \tag{3.9}$$

   where $g(Z_s(t))$ is the gradient of e_s with respect to $Z_s(t)$ (which can be analytically computed from Eq. 3.4), and, α and β are constants.

   For Gaussian distributed perturbations, $\Delta_s = N_s(0, \sigma^2_{s_{(m)}})$, i.e., zero mean Gaussian with variance $\sigma^2_{s_{(m)}}$, where $\sigma^2_{s_{(m)}} = e_s$.

   For uniformly distributed perturbations, $\Delta_s = U_s(\hat{Z}_{s_{(m-1)}}(t) \pm a_{s_{(m)}})$, i.e., uniformly distributed in an interval of length $2a_s^{(m)}$ about $\hat{Z}_{s_{(m-1)}}(t)$ where U_s denotes uniformly distributed random numbers. To make reasonable comparisons with the case of Gaussian perturbations, $a_{s_{(m)}}$ is chosen such that

$$\frac{a^2_{s_{(m)}}}{3} = \sigma^2_{s_{(m)}} = e_s. \tag{3.10}$$

5. Go to step (2).

The difference in computational complexity between the two algorithms originates from the estimation of the depth (Z) parameters. The MBASIC algorithm treats this as another least squares estimation problem which requires seven multiplications and eight additions per point pair, per iteration. Our method is based on perturbation of the depth

parameters and requires sixteen multiplications and twelve additions per point pair, per iteration. Experimental results presented in the next section show that the MBASIC algorithm usually converges to a result in about 5-10 iterations. Our algorithm generally provides superior results after about 15-20 iterations (see Figures). Considering that we work with 5-10 point pairs, the computational complexity of the improved algorithm is just slightly higher.

3.4.3 Comparisons

In this section, we compare the performance of the proposed improved algorithm with that of the MBASIC algorithm in the presence of various degrees of inaccuracy in the initial depth estimates, and for different number of point correspondences. The comparative analysis has been performed by means of a number of numerical simulations as well as an experiment with a typical videophone scene, *Claire*. The wire-frame model (CANDIDE) [53] consisting of 100 triangles was used in the experiment with the Claire sequence.

The simulations were carried out by using 5, 7 and 10 point correspondences, respectively, with 10%, 30% and 50% error in the initial depth estimates in each case. The data for the simulations were generated as follows: A set of 5 to 10 points, $(x_s(t), y_s(t))$ with the respective depth parameters $Z_s(t)$, in the range 0 and 1, were arbitrarily chosen. The coordinates $(x_s(t+\Delta t), y_s(t+\Delta t))$ of the matching points in the next frame were generated from $(x_s(t), y_s(t))$ using the transformation (3.3) with the "true" 3-D motion parameters listed in Table 3.1 . The computed coordinates $(x_s(t+\Delta t), y_s(t+\Delta t))$ are then truncated to the nearest integer. This truncation approximately corresponds to adding 40 dB noise to the matching point coordinates. Then, $\pm 10\%$, $\pm 30\%$ or $\pm 50\%$ error is added to each depth parameter $Z_s(t)$, for the respective simulations. The signs of the error (+ or −) were chosen randomly. At each iteration of the algorithm, first the motion parameters are estimated as the least squares solution of Eq. 3.5 using the present depth parameters. (This step is the same as in the MBASIC algorithm.) Then,

the depth parameters are updated as given by Eq.3.9. We set $\alpha = 0.95$ and $\beta = 0.3$ to obtain the reported results. We iterate between Eqs. 3.5 and 3.9 until E_m given by Eq. 3.7 is less than an acceptable level. In order to minimize the effect of random choices in the evaluation of the results, the results are repeated three times using three different seed values for the random number generator. The results shown in Table 3.1 and Figures 3.2, 3.3 and 3.4 are the average of these three sets.

Table 3.1 provides a comparison of the motion parameter estimates obtained by the MBASIC algorithm and the proposed method using uniform and Gaussian distributed random perturbations at the conclusion of the iterations (in this case after 500 iterations). Table 3.1 shows the results only for the 10-point correspondence case. The 5-point and 7-point results are similar. The comparison of the results of the depth parameter estimation is shown in Figures 3.2-3.4. In these figures the average estimation error in the depth parameters vs. iteration number is plotted, where the average error is defined as

$$Error = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\frac{(Z_s(t) - \hat{Z}_s(t))^2}{Z_s^2(t)}}. \tag{3.11}$$

where N is the number of point correspondences; $Z_s(t)$ and $\hat{Z}_s(t)$ are the "true" and estimated depth parameters, respectively. Note that the scale of the vertical axis is not the same in each case.

In the MBASIC algorithm, the errors in the depth estimation directly affect the accuracy of the motion estimation and vice versa, since the algorithm iterates between Eqs. 3.5 and 3.6. This can be seen from Tables 3.1 and 3.2, where the error in the initial depth estimates mainly affects the accuracy of ω_x and ω_y which are directly multiplied by Z in both equations. Thus, in the MBASIC algorithm, the error in ω_x and ω_y estimates increases as we increase the error in the initial depth estimates (see Table 3.1). Further, in the MBASIC algorithm, the error in the depth estimates (at convergence) increases with increasing error in the initial depth parameters (see, e.g., Figs. 3.2.c, 3.3.c and 3.4.c). However, in the proposed algorithm, an update scheme given by Eq. 3.9 that is indirectly tied to the current estimates of the motion parameters is used. As a result, a smaller average error is obtained for depth parameter estimation in all cases. As can be

	True motion	MBASIC	Uniform	Gaussian
$\omega_x(rad.)$	0.01	0.009951	0.010181	0.010141
$\omega_y(rad.)$	0.02	0.0199901	0.020351	0.020255
$\omega_z(rad.)$	-0.01	-0.009994	-0.009998	-0.009995
$T_x(pixel)$	0.02	0.019933	0.020067	0.019939
$T_y(pixel)$	0.05	0.05004	0.049967	0.050031

(a)

	True motion	MBASIC	Uniform	Gaussian
$\omega_x(rad.)$	0.01	0.07856	0.010241	0.010779
$\omega_y(rad.)$	0.02	0.015712	0.020504	0.021464
$\omega_z(rad.)$	-0.01	-0.009994	-0.009996	-0.009984
$T_x(pixel)$	0.02	0.018079	0.019966	0.021038
$T_y(pixel)$	0.05	0.050961	0.050018	0.049481

(b)

	True motion	MBASIC	Uniform	Gaussian
$\omega_x(rad.)$	0.01	0.005040	0.010768	0.010441
$\omega_y(rad.)$	0.02	0.010084	0.021548	0.020982
$\omega_z(rad.)$	-0.01	-0.009545	-0.010002	-0.010018
$T_x(pixel)$	0.02	0.015438	0.020363	0.019958
$T_y(pixel)$	0.05	0.052281	0.049818	0.050018

(c)

Table 3.1: The true and estimated motion parameters for 10 point correspondences with (a) 10%, (b) 30% and (c) 50% initial error in the depth estimates.

seen from Figs. 3.2 -3.4, the depth estimates, using the proposed method, converge closer to the correct parameters even in the case of 50% error in the initial depth estimates. For example, in the case of estimation using 10 point correspondences with 50% error in the initial depth estimates, the proposed method results in about 10% error after 500 iterations whereas the MBASIC algorithm results in 45% error. In the 10% initial error case, the error at the end of the iterations is 3% in MBASIC algorithm and 0.5% in our algorithm.

The proposed method with uniform perturbations has also been applied to a typical

	MBASIC	Uniform	Gaussian
$\omega_x(rad.)$	2.4010e-09	3.2761e-08	1.9881e-08
$\omega_y(rad.)$	9.8010e-11	1.2320e-07	6.5025e-08
$\omega_z(rad.)$	3.6000e-11	4.0000e-12	2.5000e-11
$T_x(pixel)$	4.4890e-09	4.4890e-09	3.7210e-09
$T_y(pixel)$	1.6000e-09	1.0890e-09	9.6100e-10

(a)

	MBASIC	Uniform	Gaussian
$\omega_x(rad.)$	4.7005e-03	5.8081e-08	6.0684e-07
$\omega_y(rad.)$	1.8387e-05	2.5402e-07	2.1433e-06
$\omega_z(rad.)$	3.6000e-11	1.6000e-11	2.5600e-10
$T_x(pixel)$	3.6902e-06	1.1560e-09	1.0774e-06
$T_y(pixel)$	9.2352e-07	3.2400e-10	2.6936e-07

(b)

	MBASIC	Uniform	Gaussian
$\omega_x(rad.)$	2.4602e-05	5.8982e-07	1.9448e-07
$\omega_y(rad.)$	9.8327e-05	2.3963e-06	9.6432e-07
$\omega_z(rad.)$	2.0703e-07	4.0000e-12	3.2400e-10
$T_x(pixel)$	2.0812e-05	1.3177e-07	1.7640e-09
$T_y(pixel)$	5.2030e-06	3.3124e-08	3.2400e-10

(c)

Table 3.2: The mean square error in the estimated motion parameters for 10 point correspondences with (a) 10%, (b) 30% and (c) 50% initial error in the depth estimates.

videophone scene, *Claire*. Here, seven point pairs which are interactively specified, are used. The coordinates of the corresponding points are determined by the block matching technique where the block size is 8×8 and the search window is 10×10. Fig. 3.5.a depicts the original wire-frame model manually fitted to the first frame of the *Claire* sequence as in Aizawa et al. [20]. Fig. 3.5.b and Fig. 3.5.c show the projection of the modified wire-frame model onto the image plane for the seventh frame using the estimated depth and motion parameters with the MBASIC and the proposed algorithms, respectively. Inspection of the results indicates a much better fit in the case of the proposed algorithm.

3.5 Optical Flow Based 3-D Motion Estimation

In this section, we will review the equation describing the relation between the 3-D motion and structure and the corresponding 2-D velocity field (optical flow), which is related to the projection of the 3-D motion onto the image plane under certain assumptions [37]. By using additional constraints regarding the 3-D structure of the scene, it is possible to recover the parameters of the 3-D motion from the associated optical flow field [56].

Let $I(x,y,t)$ represent the intensity at points on a path defined by $(x = x(t), y = y(t), t)$, in the 2-D image plane where t is the time. If we wish to know the rate at which the intensity changes with respect to t as we travel along the path, we have to evaluate the total derivative of I along that path with respect to t, assuming that $I(x,y,t)$ has continuous partial derivatives I_x, I_y and I_t and $x = x(t)$, $y = y(t)$ are differentiable functions of t (Eq. 3.12).

$$\frac{dI}{dt} = I_x \frac{dx(t)}{dt} + I_y \frac{dy(t)}{dt} + I_t. \tag{3.12}$$

We can interpret Eq. 3.12 as the rate at which I changes with respect to t as we move on any arbitrary path defined by $(x(t), y(t), t)$ with a velocity $(\frac{dx(t)}{dt}, \frac{dy(t)}{dt}, 1)$. Therefore, it is the directional derivative of I along the direction $(\frac{dx(t)}{dt}, \frac{dy(t)}{dt}, 1)$.

Now, let us define an object S in the 3-D object space. The intensity of a point s on S at time t can be represented by $I(x_s(t), y_s(t), t)$, $\forall s \in S$ and $\forall t$. If the object is subject to motion we can find the change of the intensity of the point s with respect to t using Eq. 3.12, along its motion trajectory $(x_s(t), y_s(t), t)$. Assuming that the intensity of the point s does not change in time, i.e. it is constant as the point moves along its trajectory, we can write the directional derivative of I as

$$\left.\frac{dI(x,y,t)}{dt}\right|_{\substack{x=x_s(t) \\ y=y_s(t)}} = 0. \tag{3.13}$$

Eq. 3.12 and 3.13 give us information about the 2-D velocity vectors between a discrete set of images. Since our aim is to compute the 3-D motion, we need to know

the relation between the 2-D and 3-D velocity vectors. Let us assume that the object S is under small rotation. Therefore, we can approximate the 3-D velocity of a particular point s of a moving object using Eq. 3.3 as

$$\begin{bmatrix} \dot{X}_s(t) \\ \dot{Y}_s(t) \\ \dot{Z}_s(t) \end{bmatrix} = \begin{bmatrix} 0 & \omega_Z & -\omega_Y \\ -\omega_Z & 0 & \omega_X \\ \omega_Y & -\omega_X & 0 \end{bmatrix} \begin{bmatrix} X_s(t) \\ Y_s(t) \\ Z_s(t) \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}, \tag{3.14}$$

where $\dot{X}_s(t) = X_s(t+\Delta t) - X_s(t)$, $\dot{Y}_s(t) = Y_s(t+\Delta t) - Y_s(t)$, $\dot{Z}_s(t) = Z_s(t+\Delta t) - Z_s(t)$. Under orthographic projection along the z-direction, we can represent the 2-D velocity field associated with the projection of the point s as

$$\begin{aligned} \dot{x}_s(t) &= \omega_Z y_s(t) - \omega_Y Z_s(t) + T_X \\ \dot{y}_s(t) &= -\omega_Z x_s(t) + \omega_X Z_s(t) + T_Y \end{aligned} \tag{3.15}$$

where $x_s(t) \triangleq X_s(t)$, $y_s(t) \triangleq Y_s(t)$.

Combining Eq.3.15, 3.12 and 3.13, we get

$$I_x(\omega_Z y - \omega_Y Z + T_X) + I_y(-\omega_Z x + \omega_X Z + T_Y) + I_t = 0. \tag{3.16}$$

Eq. 3.16 is a constraint that relates the spatio-temporal image gradients to the 3-D motion ω_x, ω_y, ω_z, T_x, T_y and the structure (Z) parameters under the assumption that the variation in image intensity pattern is solely due to the 3-D motion of the underlying scene. Eq. 3.16 alone is not sufficient to determine the 3-D motion and the structure parameters if they are allowed to change freely and independently at every point. Several approaches exist in the literature to compute the motion and structure parameters from Eq. 3.16 under piecewise rigid scene assumption (constraining the variation of the 3-D motion parameters) and with certain surface structure models (constraining the variation of Z with (X, Y)) [56, 74]. Another approach is to compute the 2-D velocity vectors using Eq. 3.12 under the assumption of smoothness of optical flow, and then to compute the 3-D motion parameters from Eq. 3.15.

In order to compare the results of feature based methods with that of optical flow based methods, we give the results obtained by one of the first methods that uses optical

flow based formulation for 3-D motion and structure estimation in the context of image coding ([29]). The method is based on the assumption that the moving objects exhibit a smooth motion due to inertia and elasticity. The 3-D motion and structure parameters are found by minimizing the energy function given by Eq. 3.17 using Newton-Raphson method.

$$E(\Delta\omega, \Delta T, \Delta z) = \sum_{i=1}^{n}((u_x^i - \hat{u}_x^i)^2 + (u_y^i - \hat{u}_y^i)^2) + \alpha||\Delta\omega||^2 + \beta||\Delta T||^2 + \zeta \sum_{i=1}^{n}(\Delta z^i)^2. \quad (3.17)$$

In Eq. 3.17 α, ζ and β are scale parameters, n is the number of points considered in the computation, u_x^i and u_y^i are the image plane motions found by Eq. 3.15 using Horn and Schunk's method, $\hat{u}_x^i$ and $\hat{u}_y^i$ are the motions found by Eq. 3.16.

Using the smoothness assumption (Horn and Schunk's method), the computed optical flow from the second and the fifth frames of the *Claire* sequence is shown in Fig. 3.6. The 3-D motion estimated by using Eq. 3.17 is shown in Fig. 3.7. In this simulation, 10 iterations have been done in minimization of the energy function which is enough to drop the energy below a certain limit with the given parameters (3.17). In addition, a block matching algorithm is used to find the image plane motions at the beginning of the iteration. α, ζ and β are chosen as 100, 3 and 1 respectively. The number of points used to compute optical flow is 251 which corresponds to the number of points on the edges.

Comparing the results, we see that in spite of the high computation cost, optical flow based methods give a more accurate estimation than the feature-based methods. So, there is a trade-off between accuracy and computational load. Another drawback of the methods based on optical flow is that, they do not consider the non-rigid objects, i.e., they assume that only a rigid body is under 3-D motion for the ease of computation. Also, these methods neglect the effects of changing shading due to the 3-D motion of the scene in order to simplify the optical flow equation. For example, in the case of rotational motion because the surface normals change, the shading of the objects varies even if the external illumination remains constant.

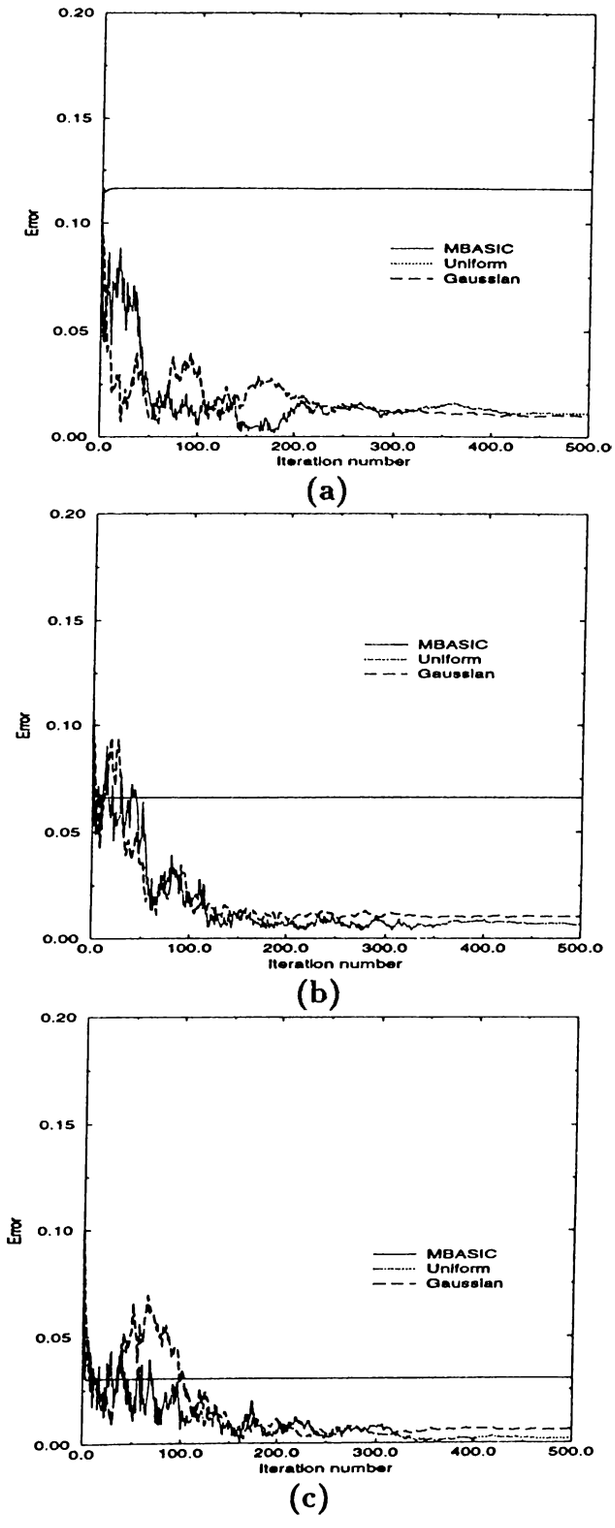

Figure 3.2: Average estimation error in the depth parameters with 10% error in the initial depth estimates for (a) 5, (b) 7, (c) 10 point correspondences.

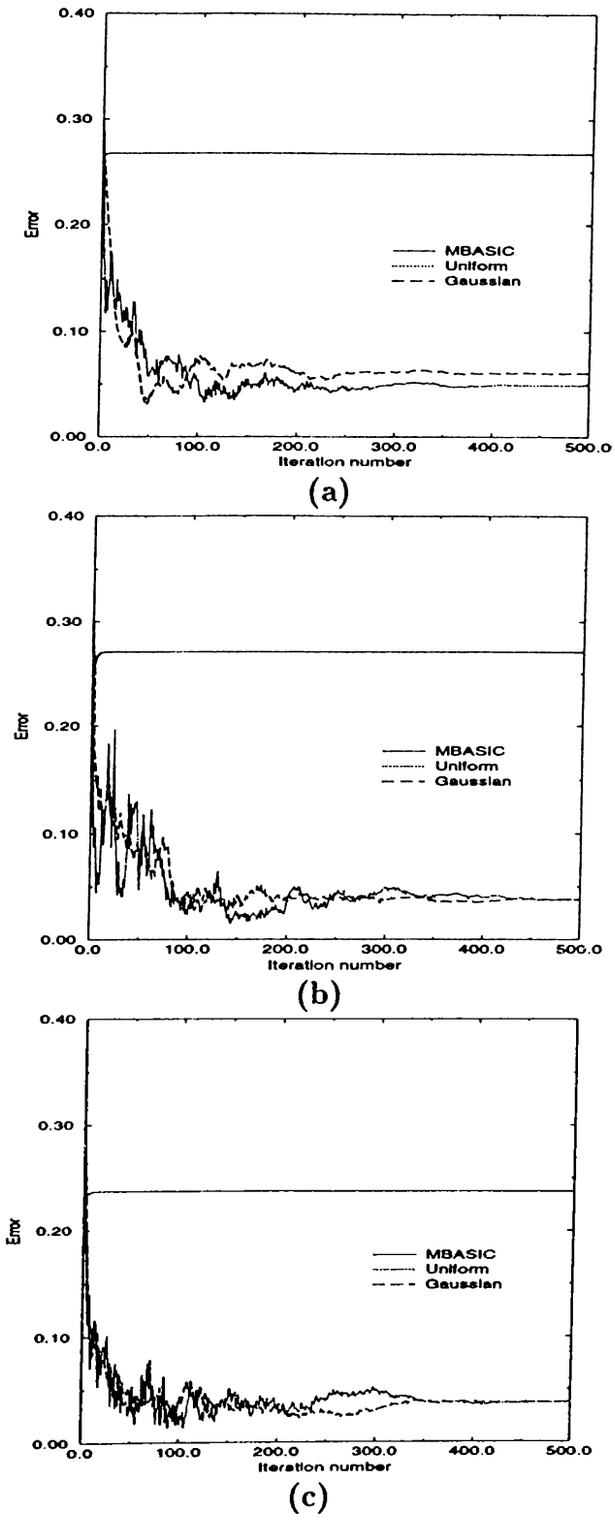

Figure 3.3: Average estimation error in the depth parameters with 30% error in the initial depth estimates for (a) 5, (b) 7, (c) 10 point correspondences.

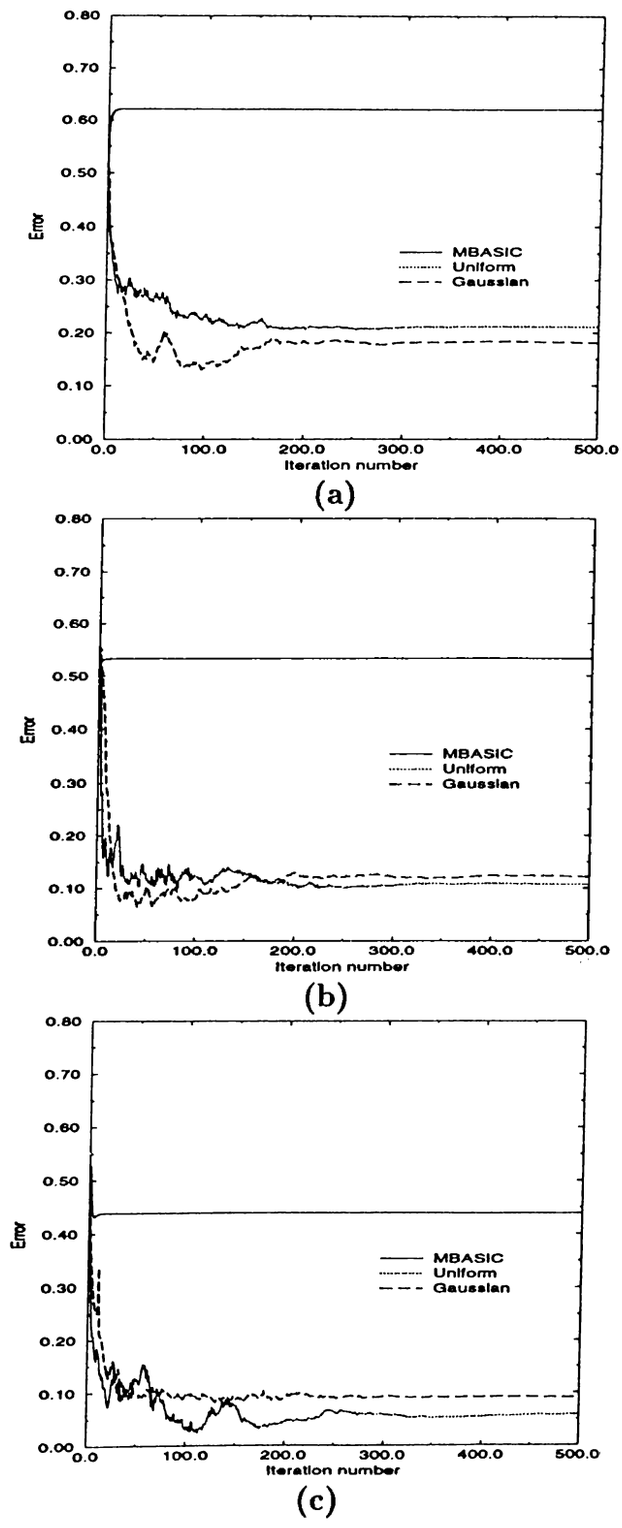

Figure 3.4: Average estimation error in the depth parameters with 50% error in the initial depth estimates for (a) 5, (b) 7, (c) 10 point correspondences.

Figure 3.5: Wire-frame model fitting for a typical video-phone sequence *Claire*. (a) Wire-frame model fitted to the first frame, (b) Modified wire-frame model for the seventh frame using the depth and motion parameters estimated by Aizawa's algorithm, (c) Modified wire-frame model for the seventh frame using the proposed algorithm with uniform perturbations.

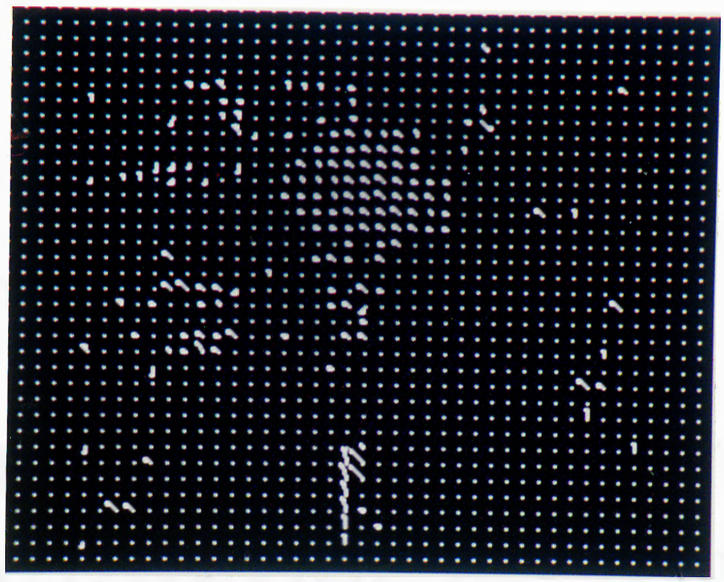

Figure 3.6: Optical flow computed using the smoothness of motion constraint from the first and fifth frames of *Claire* sequence.

Figure 3.7: The modified wire-frame using the parameters estimated by optical flow based formulation.

Chapter 4

ESTIMATION INCLUDING PHOTOMETRIC EFFECTS

In this section, we propose a novel formulation for adaptation of a generic wire-frame model to a particular speaker in a scene and for estimation of the 3-D global and local motion. The estimation and adaptation is done within an optical flow based framework including the photometric effects of the motion. In the formulation, 3-D global motion refers to the 3-D rotation and translation of the head as a whole, and local motion refers to the motion of the individual points on the face corresponding to the wire-frame nodes. We use a flexible wire-frame model whose local structure is characterized by the normal vectors of the patches which are related to the coordinates of the nodes. Geometric constraints that describe the propagation of the movement of the nodes are introduced, which are then efficiently utilized to reduce the number of independent structure parameters. A global random search algorithm has been used to determine optimum global motion estimates and the parameters describing the structure of the wire-frame model (local motion).

Results with both simulated and real facial image sequences are provided.

4.1 Photometric Model of Image Formation

The utility of photometric cues in 3-D motion and structure estimation has recently been discussed [38, 77], but photometric information was not used in the context of knowledge-based coding before. Recently, Pentland [77] has shown that photometric effects (changes in image intensity due to object motion) can be more important than the geometric effects (distortion of the projected surface shape due to motion) in structure estimation. Similar discussions can be found in [37, 38]. Here we briefly discuss a photometric model of image formation with the aim of incorporating photometric effects into the aforementioned optical flow based formulation.

Let us define an object S as a set of labeled points s forming a surface in 3-D space. Suppose that we observe the object S at a fixed time t. Since S is a surface, we can represent it as a depth function $Z(x,y)$. Now, let us represent the partial derivative of depth $Z(x,y)$ with respect to the image coordinates x and y by $p = \frac{\partial Z}{\partial x}$ and $q = \frac{\partial Z}{\partial y}$, respectively. Denoting $p_s(t)$ and $q_s(t)$ as the partial derivatives p and q at point s at time t, respectively, where s is a particular point of the object S, we can represent the image intensity associated with the point s through a reflectance map

$$I(x_s(t), y_s(t), t) = \mathcal{R}(p_s(t), q_s(t)) \tag{4.1}$$

where $\mathcal{R}$ denotes the reflectance map function. Under the assumptions of orthographic projection onto the image plane and a Lambertian surface with constant albedo, ρ, we can express the reflectance map as [70]

$$\mathcal{R}(p_s(t), q_s(t)) = \rho \vec{N_s}(t) \cdot \vec{L} \tag{4.2}$$

where $\vec{L} = (L_x, L_y, L_z)$ is the unit vector in the mean illuminant direction and $\vec{N_s}$ is the unit surface normal of the object at point s given by

$$\vec{N_s}(t) = (-p_s(t), -q_s(t), 1)/(p_s^2(t) + q_s^2(t) + 1)^{1/2}. \tag{4.3}$$

Note that the illuminant direction can also be expressed in terms of tilt and slant angles as

$$\vec{L} = (L_x, L_y, L_z) = (\cos\tau \sin\sigma, \sin\tau \sin\sigma, \cos\sigma) \tag{4.4}$$

where τ, the tilt angle of the illuminant, is the angle between $\vec{L}$ and the $X-Z$ plane, and σ, the slant angle, is the angle between $\vec{L}$ and the positive Z axis.

The model (4.2) is widely used in the computer vision literature for estimating the object shape and the illuminant direction from shading [77]-[81]. Note that the Lambertian reflection model used here is appropriate for diffuse reflection and will be sufficient to incorporate the photometric effects of rotational motion. There are models developed for specular reflection, too [38].

4.1.1 Estimation of illumination direction

In order to incorporate the photometric effects of motion into 3-D motion estimation and wire-frame adaptation, the illuminant direction $\vec{L}$ must be known or estimated from the available frames. In this study, we use the method proposed by Zheng *et al.* [81] to estimate the illuminant direction.

The method to estimate the tilt angle is based on the assumption that the surface points are umbilical points. An umbilical point is a point at which the surface is approximately spherical. An estimate of the tilt angle is given by

$$\tau = \arctan\left(\frac{E\{\hat{L}_x/\sqrt{\hat{L}_x^2+\hat{L}_y^2}\}}{E\{\hat{L}_y/\sqrt{\hat{L}_x^2+\hat{L}_y^2}\}}\right) \tag{4.5}$$

where $E\{.\}$ denotes the averages over the spatial variables. These values are computed by averaging the results obtained over 3×3 moving windows in our implementation. $\hat{L}_x$ and $\hat{L}_y$ are the x and y components of the local estimate of the tilt of the illuminant, respectively, computed as (using the notation in [81])

$$\begin{bmatrix} \hat{L}_x \\ \hat{L}_y \end{bmatrix} = (B^tB)^{-1}B^t \begin{bmatrix} \delta I_1 \\ \delta I_2 \\ \cdot \\ \cdot \\ \cdot \\ \delta I_N \end{bmatrix}, \quad \text{and} \quad B = \begin{bmatrix} \delta x_1 & \delta y_1 \\ \delta x_2 & \delta y_2 \\ \cdot & \\ \cdot & \\ \cdot & \cdot \\ \delta x_N & \delta y_N \end{bmatrix}.$$

Here, δI_i is the difference in image intensity of two neighboring pixels along a particular direction $(\delta x_i, \delta y_i)$, and N is the number of directions (in our implementation we have set N=8 for each 4 × 4 window).

The slant angle σ can be uniquely estimated from

$$\frac{E\{I\}}{E\{I^2\}} = f_3(\sigma) \tag{4.6}$$

since $f_3(\sigma)$ (defined in [81]) is a monotonically decreasing function of σ, where $E\{I\}$ and $E\{I^2\}$ are the averages of the image intensities and the square of the image intensities, respectively, over all pixels in the image area where the wire-frame model is fitted.

Finally, the surface albedo can be estimated from

$$\rho = \frac{E\{I\} \cdot f_1(\sigma) + \sqrt{E\{I^2\} \cdot f_2(\sigma)}}{f_1^2(\sigma) + f_2(\sigma)}, \tag{4.7}$$

where $f_1(\sigma)$ and $f_2(\sigma)$ are seventh order polynomials in $\cos\sigma$ as defined in [81].

4.2 Problem Formulation

In this section we present the proposed formulation for simultaneous estimation of 3-D global motion parameters and adaptation of the wire-frame model that also takes the presence of local motion into account where global and local motions are as defined at the beginning of this chapter.

4.2.1 Incorporation of the photometric effects

We represent the 3-D structure of the speaker by a wire-frame model where the surface of the wire-frame model is composed of planar patches. So, the variation in the intensity of a pixel due to photometric effects of motion will be related to a change in the normal vector of the patch that this pixel belongs to. This variation in the intensity can be found by using the image formation model. (see Eq. 4.1.) From Eq. 4.3 and Eq. 3.3, we

can write the change in the normal vector associated with the point s due to the 3-D motion as

$$\begin{aligned} d\vec{N_s}(t) &= \vec{N}(X_s(t+\Delta t), Y_s(t+\Delta t), Z_s(t+\Delta t)) - \vec{N_s}(X_s(t), Y_s(t), Z_s(t)) \\ &= \frac{(-p_s(t+\Delta t), q_s(t+\Delta t), 1)^T}{(p_s(t+\Delta t)^2 + q_s(t+\Delta t)^2 + 1)^{1/2}} - \frac{(-p_s(t), -q_s(t), 1)^T}{(p_s(t)^2 + q_s(t)^2 + 1)^{1/2}}, \end{aligned} \tag{4.8}$$

where,

$$\begin{aligned} p_s(t+\Delta t) &= \frac{-\omega_Y + p_s(t)}{1 + \omega_Y p_s(t)} \\ q_s(t+\Delta t) &= \frac{\omega_X + q_s(t)}{1 - \omega_X q_s(t)}, \end{aligned} \tag{4.9}$$

using orthographic projection. Assuming that the mean illuminant direction $\vec{L} = (L_x, L_y, L_z)$ remains constant, we can represent the change in intensity due to photometric effects of motion using Eq. 4.2 as

$$\frac{dI(x_s(t), y_s(t), t)}{dt} = \rho \vec{L} \cdot d\vec{N_s}(t); \tag{4.10}$$

where the derivative is taken along the path the point s travels. Following the discussion given in section 3.5, we can relate the spatio-temporal image gradients to the 3-D motion and structure parameters with the inclusion of photometric effects as

$$I_x(\omega_Z y - \omega_Y Z + T_X) + I_y(-\omega_Z x + \omega_x Z + T_Y) + I_t = \rho \vec{L} \cdot \left[\frac{(-p', -q', 1)^T}{\sqrt{p'^2 + q'^2 + 1}} - \frac{(-p, -q, 1)^T}{\sqrt{p^2 + q^2 + 1}} \right]. \tag{4.11}$$

where $p' = p(t+\Delta t)$, $q' = q(t+\Delta t)$, $p = p(t)$ and $q = q(t)$. The term on the right hand side of Eq. (4.11) may be significant especially if the change in the surface normal has components either toward or away from the illuminant direction [77].

4.2.2 Structure of the wire-frame model and problem statement

The wire-frame model is composed of triangular patches which are characterized by the (X, Y, Z) coordinates of their respective vertices. Given the (X, Y, Z) coordinates of the

vertices of a patch, we can write the equation of the plane containing this patch. Let $P_1^{(i)} = (X_1^{(i)}, Y_1^{(i)}, Z_1^{(i)})$, $P_2^{(i)} = (X_2^{(i)}, Y_2^{(i)}, Z_2^{(i)})$ and $P_3^{(i)} = (X_3^{(i)}, Y_3^{(i)}, Z_3^{(i)})$ denote the vertices of the i^{th} patch, and $P^{(i)} = (X^{(i)}, Y^{(i)}, Z^{(i)})$ be any point within this patch. Then,

$$\vec{P^{(i)}P_1^{(i)}} \cdot (\vec{P_1^{(i)}P_2^{(i)}} \times \vec{P_1^{(i)}P_3^{(i)}}) = 0$$

gives the equation of the plane containing $P_1^{(i)}$, $P_2^{(i)}$ and $P_3^{(i)}$, where $\vec{P^{(i)}P_1^{(i)}}$, $\vec{P_1^{(i)}P_2^{(i)}}$, and $\vec{P_1^{(i)}P_3^{(i)}}$ are the vectors from the former point to the latter, respectively. We can express this equation in the form

$$Z_i = p_i X_i + q_i Y_i + c_i, \tag{4.12}$$

where

$$p_i = -\frac{(Y_2^{(i)} - Y_1^{(i)})(Z_3^{(i)} - Z_1^{(i)}) - (Z_2^{(i)} - Z_1^{(i)})(Y_3^{(i)} - Y_1^{(i)})}{(X_2^{(i)} - X_1^{(i)})(Y_3^{(i)} - Y_1^{(i)}) - (Y_2^{(i)} - Y_1^{(i)})(X_3^{(i)} - X_1^{(i)})},$$

$$q_i = -\frac{(Z_2^{(i)} - Z_1^{(i)})(X_3^{(i)} - X_1^{(i)}) - (X_2^{(i)} - X_1^{(i)})(Z_3^{(i)} - Z_1^{(i)})}{(X_2^{(i)} - X_1^{(i)})(Y_3^{(i)} - Y_1^{(i)}) - (Y_2^{(i)} - Y_1^{(i)})(X_3^{(i)} - X_1^{(i)})},$$

and

$$c_i = Z_1^{(i)} + X_1^{(i)} \frac{(Y_2^{(i)} - Y_1^{(i)})(Z_3^{(i)} - Z_1^{(i)}) - (Z_2^{(i)} - Z_1^{(i)})(Y_3^{(i)} - Y_1^{(i)})}{(X_2^{(i)} - X_1^{(i)})(Y_3^{(i)} - Y_1^{(i)}) - (Y_2^{(i)} - Y_1^{(i)})(X_3^{(i)} - X_1^{(i)})} +$$

$$Y_1^{(i)} \frac{(Z_2^{(i)} - Z_1^{(i)})(X_3^{(i)} - X_1^{(i)}) - (X_2^{(i)} - X_1^{(i)})(Z_3^{(i)} - Z_1^{(i)})}{(X_2^{(i)} - X_1^{(i)})(Y_3^{(i)} - Y_1^{(i)}) - (Y_2^{(i)} - Y_1^{(i)})(X_3^{(i)} - X_1^{(i)})}.$$

Using Eq. 4.12, the Z coordinate of any point on the i^{th} patch can be expressed in terms of the parameters p_i, q_i and c_i and the X and Y coordinates of the point. Then, we can eliminate Z from Eq. 4.11 by substituting Eq. 4.12 into Eq. 4.11 with $X_i = x_i$, $Y_i = y_i$, where the patch index i is determined for each (x, y) according to the orthographic projection.

The problem of simultaneously estimating the 3-D global motion parameters ω_X, ω_Y, ω_Z, T_X, T_Y, describing the global motion of the head, and the parameters p_i, q_i, c_i, that describe the structure of the wire-frame model, can then be formulated

as to minimize the squared error in the optical flow equation (4.11) over all pixels in a frame given by

$$E = \sum_{i} \sum_{(x,y)\in i^{th}patch} e_i^2(x,y) \tag{4.13}$$

where

$$e_i(x,y) = I_x(\omega_Z y - \omega_Y(p_i x + q_i y + c_i) + T_X) + I_y(-\omega_Z x + \omega_X(p_i x + q_i y + c_i) + T_Y) + I_t -$$

$$\rho(L_x, L_y, L_z) \cdot \left(\frac{(-\frac{-\omega_y + p_i}{1+\omega_Y p_i}, -\frac{\omega_X + q_i}{1-\omega_X q_i}, 1)}{((\frac{-\omega_Y + p_i}{1+\omega_Y p_i})^2 + (\frac{\omega_X + q_i}{1-\omega_X q_i})^2 + 1)^{1/2}} - \frac{(-p_i, -q_i, 1)}{(p_i^2 + q_i^2 + 1)^{1/2}} \right), \tag{4.14}$$

with respect to the variables ω_X, ω_Y, ω_Z, T_X, T_Y, p_i, q_i, c_i and $i = 1, \cdots,$ number of patches.

It is important to note that the surface normals p_i, q_i and the translation (in the Z direction) parameters c_i of each planar patch of the wire-frame are not completely independent of each other. An efficient algorithm to reduce the number of independent unknowns is given in Section 4.3.

4.3 Optimization Method

In this section a global random search algorithm [82] to estimate the global motion parameters ω_x, ω_y, ω_z, T_x, T_y, and the structure parameters p_i, q_i, c_i is proposed. There is a variety of global optimization methods where each of them is suitable for different classes of problems. Among them, global random search methods occupy a peculiar place as sometimes they offer the unique way of solving complicated problems. The main popularity reasons of global random search methods among users are that they are rather insensitive to irregularity of the cost function behaviour as well as to the presence of noise in the cost function evaluations and also to the growth of dimensionality. Besides, it is easy to construct simple methods that guarantee global convergence. Due to these reasons, we also use a global random search algorithm to find the unknown parameters in our problem.

The simplest of the random search methods is to evaluate the cost function at points obtained by sampling from a uniform distribution on the search space. To generalize this algorithm, one can use a random independent sampling of points in the search space with some given nonuniform distribution which is a Gaussian distribution in our case. The usage of the previously obtained information through the search also improves the efficiency of the algorithm. A simpler manner of including adaptive elements into the global random search techniques consists of determining a distribution for the new solution as depending on the previous point and the cost function value at this point. In our experiments, the independent unknowns are perturbed at each iteration, where the perturbations are generated as samples of Gaussian distributed numbers with the variance of the distribution adjusted according to the value of the cost function E (given by Eq. 4.13). Further, the magnitude of perturbations is reduced with the number of iterations, so that convergence should result.

As stated in Section 4.2.2, not all of the parameters p_i, q_i, c_i are independent due to the geometrical constraints defining the structure of the wire-frame model. The dependent structure parameters are determined as follows: At the beginning, the x, y coordinates of the nodes of the wire-frame at the boundary of the facial region shown in Fig. 2.2 are fixed whereas the Z coordinates are allowed to move. This means that only the projections of the boundary nodes are kept fixed during the iterations. All three coordinates of a non-boundary node are free to change. At each iteration cycle, we visit the patches of the wire-frame model belonging to facial region in a sequential order. If, at the present iteration cycle, none of the neighboring patches of patch i has yet been visited for updating their structure parameters (e.g., the initial patch), then p_i, q_i, c_i are all independent and are perturbed. If only one of the neighboring patches, say patch j, has been visited (p_j, q_j, c_j have already been updated), then two of the parameters, say p_i and q_i are independent and perturbed. The dependent variable c_i is computed as

$$c_i = p_j x_{ij} + q_j y_{ij} + c_j - p_i x_{ij} - q_i y_{ij}, \tag{4.15}$$

using the line equation between the patches where (x_{ij}, y_{ij}) is one of the nodes common to both patches i and j and is known before the present iteration cycle: it is either on the

boundary or has been already updated. If two of the neighboring patches, say patches j and k, have already been visited, i.e., the variables p_j, q_j, c_j and p_k, q_k, c_k have been updated, then only one variable, say p_i, is independent and perturbed. Then, c_i is found from Eq. 4.15, and q_i is evaluated as

$$q_i = \frac{p_k x_{ik} + q_j y_{ik} + c_k - p_i x_{ik} - c_k}{y_{ik}}, \tag{4.16}$$

where (x_{ik}, y_{ik}) is one of the nodes common to both patches i and k and is known before the present iteration cycle, that is either on the boundary or has been already updated.

The perturbation of the structure parameters p_i, q_i and c_i for each patch i results in a change in the coordinates of the nodes of the updated wire-frame. The new coordinates (X_n, Y_n, Z_n) of the node n can be computed given the updated structure parameters of three patches that intersect at node n. Let the patches i, j and k intersect at node n. Then, the relations

$$\begin{aligned} p_i X_n + q_i Y_n + c_i &= p_j X_n + q_j Y_n + c_j \\ p_i X_n + q_i Y_n + c_i &= p_k X_n + q_k Y_n + c_k \end{aligned} \tag{4.17}$$

specify X_n and Y_n. Thus, the new X and Y coordinates of the nodes are given by

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} p_i - p_j & p_j - p_k \\ q_i - q_j & q_j - q_k \end{bmatrix}^{-1} \begin{bmatrix} c_j - c_i \\ -c_j + c_k \end{bmatrix} \tag{4.18}$$

The new Z coordinate can be computed from Eq. 4.12 given the X and Y coordinates and the p_i, q_i, c_i for any patch passing through that node. It is this updating of the coordinates of the nodes that allows the adaptation of the wire-frame model to lower the misfit error and accommodate the presence of local motion, such as the motion of the eyes and the mouth.

We summarize the proposed algorithm as:

1. Estimate the illumination direction using Eq. 4.5 and Eq. 4.6, and the surface albedo using Eq. 4.7.

2. Initialize the coordinates of the nodes (X_n, Y_n, Z_n), for all n, using an approximately scaled initial wire-frame model, where the scaling is done by positioning four extreme points, interactively, according to the location and the size of the face. Three coordinates of a non-boundary node are free to change. Fix the x, y coordinates of the nodes at the boundary of the facial region (Fig. 2.2) and leave the Z coordinate free to move. Leave the coordinates of a non-boundary node free to change, too. Set the iteration counter $m = 0$.

3. Determine the initial motion parameters using the stochastic relaxation method described in Chapter 3 (or any other point correspondence method to compute the motion parameters using a set of selected nodes given their depth values).

4. Compute the value of the cost function E given by Eq. 4.13.

5. If $E < \epsilon$, stop.

   Else, set $m = m + 1$, and

   perturb the motion parameters $\omega = [\omega_x, \omega_y, \omega_z, T_x, T_y]^T$ as

$$\omega_{(m)} \longleftarrow \omega_{(m-1)} + \alpha^m \Delta, \tag{4.19}$$

   where $\Delta = N(0, \sigma^2_{(m)})$, i.e., zero mean Gaussian with variance $\sigma^2_{(m)}$, where $\sigma^2_{(m)} = E$, and

   the structure parameters p_i, q_i and c_i as

   `Define count_i as the number of neighboring patches to patch i whose structure parameters have been perturbed. Set count_i=0, for all patches i belonging to the facial region.`

   `Perturb p_1, q_1, c_1 as`

$$\begin{aligned} p_{1_{(m)}} &\longleftarrow p_{1_{(m-1)}} + \alpha^m \Delta_1, \\ q_{1_{(m)}} &\longleftarrow q_{1_{(m-1)}} + \alpha^m \Delta_1, \\ c_{1_{(m)}} &\longleftarrow c_{1_{(m-1)}} + \alpha^m \Delta_1, \end{aligned} \tag{4.20}$$

where $\Delta_i = N_i(0, \sigma^2_{i(m)})$, i.e., zero mean Gaussian with variance $\sigma^2_{i(m)}$, where $\sigma^2_{i(m)} = \sum_{(x,y)\in patch i} e_i^2(x,y)$.

```
increment count_j, for all j denoting neighbors of patch 1.

for( i=2 to number of patches)
  {
  if(count_i==1) {
              perturb p_i and q_i
       increment count_m, for all m denoting neighbors of patch i
       Compute c_i using Eq. 4.15 where the x and y coordinates are that
       of a fixed or a precomputed node on the intersection line
       between patch i and j
       }

  if(count_i==2) {
       perturb p_i
       increment count_m, for all m denoting neighbors of patch i
       Compute c_i using Eq. 4.15 and q_i using Eq. 4.16 where x_ij,y_ij
       and x_ik,y_ik are coordinates of a fixed or a precomputed node
       on the intersection line between patches i,j and i,k
       respectively.
       }
  If p_i, q_i, and c_i for at least three patches intersecting at a node are
  updated, then update the coordinates of the node by using Eq. 28.
  }
```

6. Go to step (4).

Experimental results will be presented in the next section to demonstrate the performance of the proposed method.

4.4 Simulation Results

We have demonstrated the proposed method with both real and synthetic image sequences [83], [84]. The real image sequences are "Miss America" and "Claire" where each frame consists of 256×256 and 352×288 pixels, respectively. The synthetic sequence is generated by moving and shading the wire-frame (Fig.2.2) which is an extension of the CANDIDE wire-frame [53] and composed of 217 triangles and 144 nodes [22]. The experimental results are summarized in the following subsections.

According to our model, the synthesis error in knowledge-based image synthesis originates from: (i) misfit of the wire-frame, (ii) error in global motion estimation, (iii) error in local motion estimation, and (iv) omission of the photometric effects of the motion. The following set of simulation experiments is intended to test each of these causes in a controlled manner. In all the experiments, the global motion refers to the 3-D rotation and translation of the wire-frame as a whole which is composed of 144 nodes, and local motion refers to the individual motions of the 83 nodes of the wire-frame corresponding to the facial region (Fig. 2.2).

4.4.1 Results with synthetic image sequences

SIMULATIONS WITH AN ARTIFICIALLY GENERATED IMAGE SEQUENCE

The first simulation tests the accuracy of the proposed method for global motion estimation by eliminating all other sources of error. To this effect, we generate an image frame by taking the orthographic projection of the wire-frame in Fig.2.2, and painting its patches by black and white, alternatingly. Since the image is obtained directly from the wire-frame itself, there is no fitting problem, and therefore, no misfit. The next frame is obtained by rotating and translating the wire-frame using a given set of motion parameters and computing the projection of the wire-frame in this new position. No local motion or shading effects are simulated. Table 4.1 shows the true and estimated motion parameters. "Initial point" indicates the motion parameters at the beginning

of the iterations, and "Result" shows the estimated parameters at convergence. The "Error" gives the absolute deviation of our results from the true values. The first and the second frames are shown in Fig. 4.2.a and 4.2.b. The initial wire-frame, and, the rotated wire-frame with the estimated motion parameters, put on the first and the second frames, are shown in Fig. 4.2.c and 4.2.d, respectively. The mean square synthesis error between the actual second frame $I_a(x,y)$ and the synthesized second frame $I_s(x,y)$ is computed according to

$$MSE = \left(\frac{1}{N \times M} \sum_{x,y} (I_a(x,y) - I_s(x,y))^2 \right)^{\frac{1}{2}}, \tag{4.21}$$

where N and M show the x and y extents of the image, respectively. The MSE in this case is 5.06 with $N = M = 256$. In case of "Miss America" sequence $N = M = 256$ and in case of "Claire" sequence $N = 352$, $M = 288$.

The next simulation tests the ability of our method to track the local motion deformations. For this purpose, in addition to the above global motion, the action units (AU) 17 and 46, corresponding to "chin raiser" and "winking", are also synthesized. Table 4.2 shows the true and estimated global motion parameters. The first and the second frames are shown in Fig. 4.3.a and 4.3.b. The initial wire-frame, and, the rotated wire-frame with the estimated motion parameters, put on the first and the second frames, are shown in Fig. 4.3.c and 4.3.d, respectively. The MSE is computed to be 5.58.

Simulations with the "Miss America" sequence

With no photometric effects

After getting successful results from the previous experiments we tested our algorithm, again for global and local motion tracking performance, using a textured 3-D model instead of the previous bare wire-frame model. The textured model is obtained by mapping a single frame from the "Miss America" sequence to the initial wire-frame model. The mapping is accomplished after scaling the wire-frame model approximately to the location and the size of the face by positioning four extreme points,

interactively. A second frame is obtained from the first one by rotating and translating it. Then we applied our algorithm to check its performance in finding these already known motion parameters. The results of global motion estimation with no local motion and no photometric effects are presented in Table 4.3. In addition, we have synthesized a new frame from the first frame using the estimated motion parameters and computed the difference between this synthesized frame and the second frame. The MSE is found to be 6.39.

We repeat the above experiment by including local motion specified by the AU2, AU17 and AU46 which correspond to "outer brow raiser", "chin raiser" and "winking". By this way, we are able to test how local deformations affect the 3-D global motion and structure parameter estimation and whether we can track these deformations. Table 4.4 shows the true and estimated global motion parameters. The first, second, and the synthesized frame in this case are shown in Fig. 4.4. The MSE between the second and the synthesized frames is now equal to 7.55.

With photometric effects

The purpose of this set of simulations is to see whether there is an improvement or not when the illumination effects are also considered. To simulate the photometric effects, we estimate the direction of the illuminant from the first frame. The estimated tilt and slant angles are 141.13 and 72.51 which correspond to $L = (-0.74, 0.60, 0.33)$. The 3-D object of the first frame is again rotated and translated (with or without local motion) as done in the previous experiment, but this time the object is shaded using Eq. 4.1 according to the change in the surface normal vectors due to rotation (assuming the direction of the illuminant remains the same). Tables 4.6 and 4.5 show the results of global motion estimation with and without local motion. In the case of local motion, we again use the AU 2, 17 and 46 which correspond to "outer brow raiser", "chin raiser" and "winking", respectively. The second and synthesized frames using estimated parameters are shown in Fig. 4.5 in the case with local motion. The MSE are found to be 7.10 and 5.84, for the experiments with and without local motion, respectively.

For the above two cases we also tested the performance of AU tracking, i.e. local motion estimation. Table 4.7 shows the original displacements of the nodes due to AU 2, 17, 46 and the estimated displacements with and without considering the photometric effects.

To see how the algorithm behaves in case of large motion, we increase ω_x and plot the percentage estimation error in ω_x versus ω_x with considering the photometric effects. We carried out the same test for ω_y and ω_z, too. The results are given in Fig. 4.1.

4.4.2 Results with real image sequences

EXPERIMENTS WITH TWO FRAMES OF THE "MISS AMERICA" SEQUENCE

Now, we test our algorithm using the first and the tenth frames from the"Miss America" sequence as the first and second frames in our algorithm. Here, there exists both global and local motion. Further, since the second frame is not artificially synthesized, there is also additional wire-frame misfit error. To see the importance of incorporating the photometric effects into the optical flow equation, we repeat the experiment once ignoring the photometric effects and once taking them into account. Tables 4.8 and 4.9 show the estimated global motion parameters with and without considering the photometric effects, respectively. The synthesized images using the estimated motion and structure parameters are depicted in Fig. 4.6 and Fig. 4.7, and the MSE are 5.82 and 6.23, for the experiments with and without the photometric effects, respectively. The MSE's are calculated only using the region of the image pasted on the wire-frame.

MOTION TRACKING EXPERIMENTS WITH THE "CLAIRE" SEQUENCE

Finally, we test our method on a longer sequence using the first 16 frames of the "Claire" sequence where we omit every other frame. The resulting 8 frames are shown in Fig. 4.8. In the beginning Claire looks straight into the camera. Later, she turns her head to the left. There are also some facial movements such as blinking and opening of the

mouth. A frame consists of 352×288 pixels (which is the CIF picture format). Only the luminance component is used in the experiments. Our task is to track the motion of the head and extract the facial expressions. Fig. 4.9 shows the synthesized image sequence using the estimated parameters. The MSE for the frames used in the experiment are $6.98, 7.17, 8.06, 8.27, 8.21, 8.77, 9.48$. Upon viewing the two sequences, the difference is hardly noticeable. From the results, we also see that the small motion assumption given in Chapter 2 is valid for the real image sequences, too.

4.5 Comparisons

As stated in Chapter 2, there are different algorithms for object based coding of head-and-shoulder type images. In most of these methods, only the analysis parameters and the color information of the first frame is transmitted to the receiver side. The comparison of the proposed algorithm with other existing methods on object based coding can be done in two steps: 1) The accuracy of the analysis parameters such as global and local motion should be compared since these are the only parameters that are transmitted during a session. 2) The mean bit-rate achieved by different methods should be compared and guaranteed to stay within the limits of very low bit rate coding (8-32 kbps).

The local and global motion estimation results based on the proposed algorithm are given in Section 4.4. We can compare these results with the work of Li [26] since this work also estimates global and local motion simultaneously within an optical flow based formulation. Although they use a simple linear transformation for the unknows, the algorithm needs some control points (AUs) to be known beforehand. Problems occur when AUs are tracked that have similar effects on the same feature points. Also the algorithm does not consider the illumination effects. Since our algorithm estimates the surface normals of the wire-frame instead of the control points, no preknown information is necessary. Also the proposed method combines motion and shape estimation with illumination estimation. Results given in Section 4.4 (Tables 4.3-4.6, Figs. 4.3-4.6) show that incorporation of the photometric effects to the formulation improves the

estimation results by a considerable amount. When we compare the global and local motion estimation results with that of Li [26] for the *Claire* sequence we see that we stay within the same limits for the synthesis error without the burden of the AUs.

We can also compare our method with other object based coding algorithms like the one developed within COST211 project. However, since these techniques are based on 2-D models, the transmitted analysis parameters are different. So, we can make the comparison in terms of bit-rate and signal-to-noise ratio (SNR). For the proposed algorithm, we compute the bit rate for the worst case where all the nodes (except for the boundary nodes) of the wire-frame move from frame to frame, i.e. the structure parameters of all the patches are due to change within the same limits. Indeed this is never the case due to the fact that although the nodes on the facial regions such as eyes, mouth, etc. heavily move from frame to frame, the relative motions of the nodes on the regions such as forehead and cheeks are extremely low. If we let the nodes of the wire-frame move within the ranges of the frame size by integer values, we can represent the displacement values by 9 bits since the frame size is 352×288. As we do not let the nodes move out of the frame, the displacement values can be represented by 9 bits with 1 pel accuracy. Since only the facial region (Fig. 2.2) of the wire-frame is used to do motion estimation, the displacement values corresponding to 83 nodes of the wire-frame must be transmitted. Assuming a picture frequency of $25/3Hz$, we will get a bit rate of $\frac{25}{3} \times 9 \times (5 + 3 \times 83)$ without any entropy coding. If we introduce entropy coding, we can further decrease the bit rate. Also we must notice that representing the displacement values by less number of bits (less than 5) will cause the iterations converge to a wrong result due to the limitation in the perturbations. If we consider the SNR which is given by

$$SNR = 20log\frac{255}{RMSE} \tag{4.22}$$

where $RMSE$ is the root-mean-square-error between the original and the synthesized images, the results are again encouraging. For the Claire sequence, peak SNR is $31.25dB$. In case of COST211 results, the bit rate is about $56kbps$ for a CIF size Claire sequence resulting in a peak SNR of $34dB$. As a result, we obtained the same quality images

without increasing the bit rate. This is also shown by the subjective tests permormed using the *Claire* sequence coded by COST211 method and the proposed algorithm.

Up to now, we only discussed the coding of the analysis parameters. However, to get the decoder properly working, a first frame of the sequence has to be sent to the decoder. How this frame is coded, i.e. either lossy or lossless, is a point of discussion. Assuming no coding of the first frame, the number of data for a 4:1:1 CIF size picture with 8 bit quantization is given by

$$Number\ of\ bits\ /\ frame\ =\ 12 \times 352 \times 288. \tag{4.23}$$

So, transmitting the first frame at $16kbps$ which is a reasonable number for very low bir rate coding requires a compression ratio of 1:76. This ratio can be achieved with the coding techniques given in Chapter 1, i.e. either transform coding techniques or second generation coding techniques. In this work, we assumed no coding of the first frame since our aim is to check the efficiency of the motion estimation algorithm.

	True motion	Initial point	Our method	Error
$\omega_x(rad)$	-0.1	-0.08894	-0.1046	0.0046
$\omega_y(rad)$	0.35	0.3368	0.3526	0.0026
$\omega_z(rad)$	-0.03	-0.0113	-0.030641	0.000641
$T_x(pixel)$	6	4.962	5.9860	0.014
$T_y(pixel)$	-3	-2.8999	-2.9791	0.0209

Table 4.1: Global motion estimation with the synthetic sequence.

	True motion	Initial point	Our method	Error
$\omega_x(rad)$	-0.1	-0.08894	-0.1072	0.0072
$\omega_y(rad)$	0.35	0.3368	0.3461	0.0039
$\omega_z(rad)$	-0.03	-0.0113	-0.02724	0.00276
$T_x(pixel)$	6	4.962	5.8697	0.1303
$T_y(pixel)$	-3	-2.8999	-2.9737	0.0263

Table 4.2: Global and local motion estimation with the synthetic sequence.

	True motion	Initial point	Our method	Error
$\omega_x(rad)$	-0.1	-0.08894	-0.1054	0.0054
$\omega_y(rad)$	0.35	0.3368	0.3369	0.0131
$\omega_z(rad)$	-0.03	-0.0113	-0.02717	0.00283
$T_x(pixel)$	6	4.962	5.7126	0.2874
$T_y(pixel)$	-3	-2.8999	-3.0796	0.0796

Table 4.3: Global motion estimation with the simulated Miss America sequence without the photometric effects.

	True motion	Initial point	Our method	Error
$\omega_x(rad)$	-0.1	-0.08894	-0.1083	0.0083
$\omega_y(rad)$	0.35	0.3368	0.33446	0.01554
$\omega_z(rad)$	-0.03	-0.0113	-0.02683	0.00317
$T_x(pixel)$	6	4.962	5.4719	0.5281
$T_y(pixel)$	-3	-2.8999	-2.7853	0.2147

Table 4.4: Global and local motion estimation with the simulated Miss America sequence without the photometric effects.

	True motion	Initial point	Our method	Error
$\omega_x(rad)$	-0.1	-0.08894	-0.1052	0.0052
$\omega_y(rad)$	0.35	0.3368	0.3482	0.0018
$\omega_z(rad)$	-0.03	-0.0113	-0.02801	0.00199
$T_x(pixel)$	6	4.962	6.2038	0.2038
$T_y(pixel)$	-3	-2.8999	-3.0702	0.0702

Table 4.5: Global motion estimation with the simulated Miss America sequence including the photometric effects.

	True motion	Initial point	Our method	Error
$\omega_x(rad)$	-0.1	-0.08894	-0.1079	0.0079
$\omega_y(rad)$	0.35	0.3368	0.34001	0.00999
$\omega_z(rad)$	-0.03	-0.0113	-0.0272	0.0028
$T_x(pixel)$	6	4.962	6.4660	0.466
$T_y(pixel)$	-3	-2.8999	-2.8852	0.01148

Table 4.6: Global and local motion estimation with the simulated Miss America sequence including the photometric effects.

AU	Vertex	True displacements		Estimated displacements without photometric effects		Estimated displacements with photometric effects	
		X	Y	X	Y	X	Y
2	16	0	14	0	15	1	16
	49	0	14	0	16	0	16
	18	0	14	0	12	0	13
	51	0	14	0	13	0	13
	15	2	7	2	6	2	5
	48	-2	7	-1	6	-3	7
	17	0	2	0	2	0	2
	50	0	2	0	3	0	2
17	9	0	-2	1	-1	0	-1
	8	0	-1	0	-2	0	-1
	7	0	-1	0	-2	0	-2
	40	0	-1	0	-2	0	-2
46	21	0	-5	2	-3	1	-5
	22	0	2	0	2	0	2
	54	0	-5	2	-4	2	-5
	55	0	2	-1	2	0	2

Table 4.7: Real and estimated displacements for the AUs corresponding to "outer brow raiser", "chin raiser" and "winking".

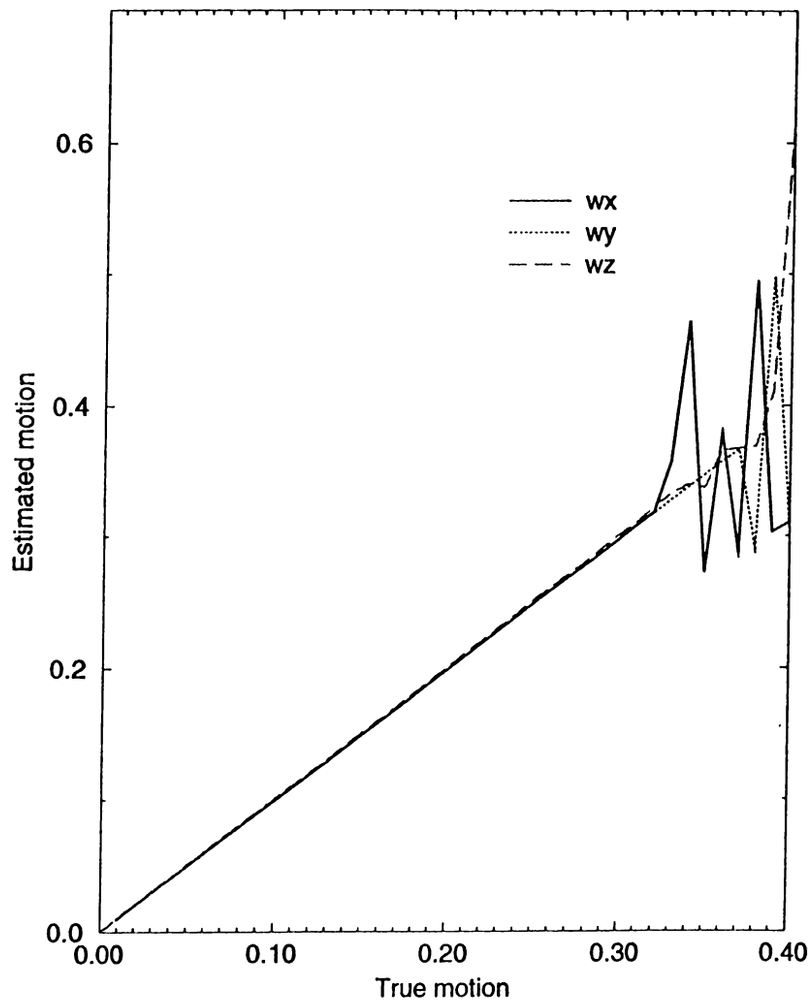

Figure 4.1: The behaviour of the estimated motion parameters with increasing motion.

	Initial point	Our method
$\omega_x(rad)$	-0.053	-0.095
$\omega_y(rad)$	0.114	0.235
$\omega_z(rad)$	0.035	0.024
$T_x(pixel)$	0.908	0.857
$T_y(pixel)$	1.4827	2.1505

Table 4.8: The estimated global motion parameters with the real Miss America sequence including the photometric effects.

	Initial point	Our method
$\omega_x(rad)$	-0.053	-0.1098
$\omega_y(rad)$	0.114	0.204
$\omega_z(rad)$	0.035	0.0226
$T_x(pixel)$	0.908	0.7648
$T_y(pixel)$	1.4827	2.6744

Table 4.9: The estimated global motion parameters with the real Miss America sequence without the photometric effects.

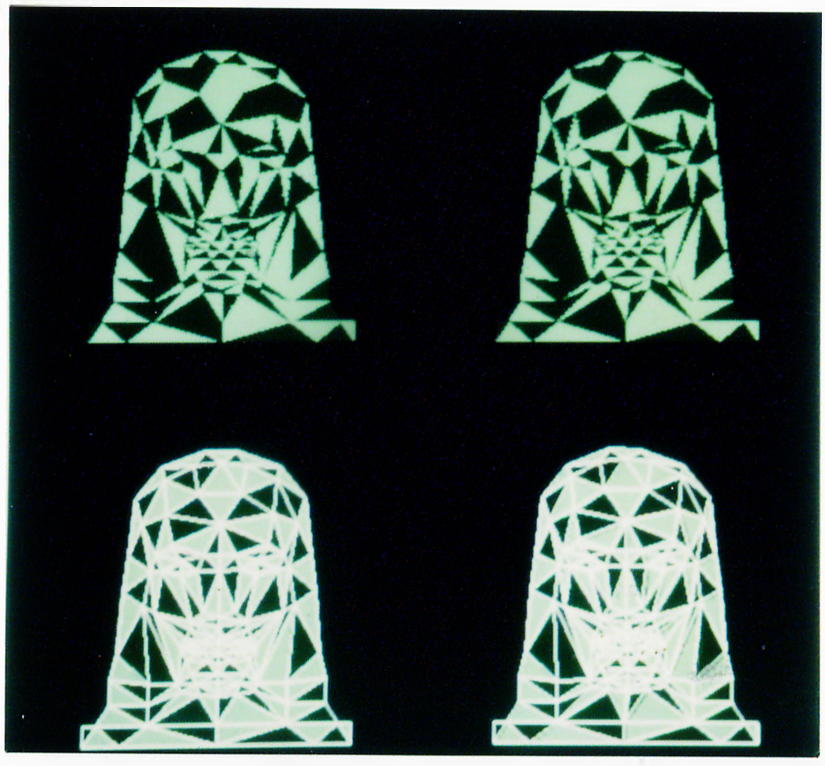

Figure 4.2: (a) The first and (b) the second frames of the synthetic image sequence with global motion; (c) the initial wire-frame and (d) the rotated wire-frame through the estimated motion parameters pasted on the first and the second frames, respectively.

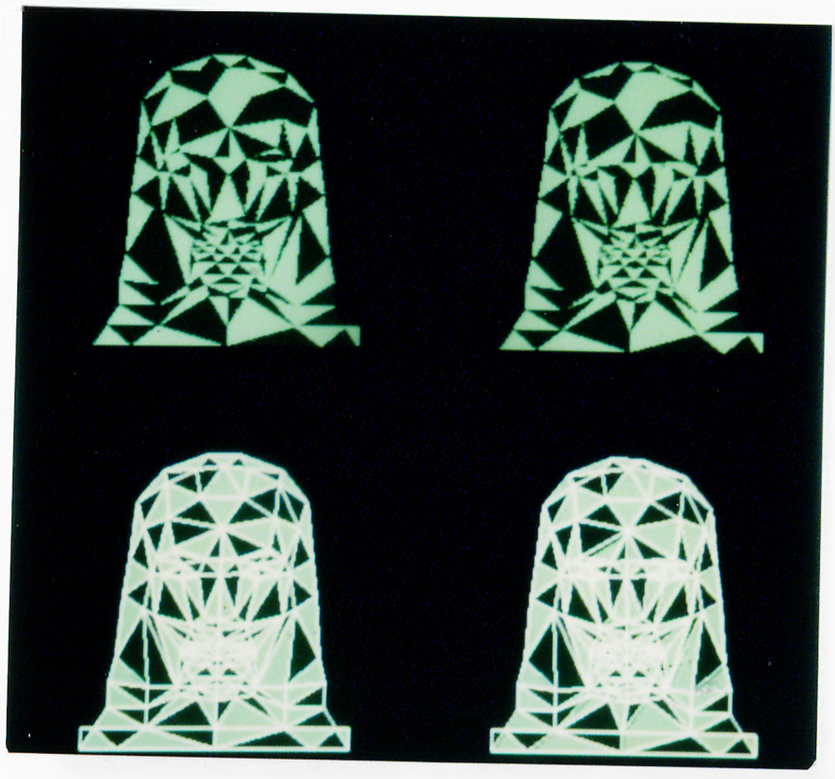

Figure 4.3: (a) The first and (b) the second frames of the synthetic image sequence with global and local motion; (c) The initial wire-frame and (d) the rotated wire-frame through the estimated motion parameters pasted on the first and the second frames, respectively.

Figure 4.4: (a) The first frame of "Miss America", (b) simulated second frame with global and local motion (without photometric effects); (c) synthesized second frame using the estimated motion and structure parameters; (d) absolute difference between the simulated and the synthesized second frames.

Figure 4.5: (a) The first frame of "Miss America", (b) simulated second frame with global and local motion,and the photometric effects; (c) synthesized second frame using the estimated motion and structure parameters; (d) absolute difference between the simulated and the synthesized second frames.

Figure 4.6: (a)The first, (b)tenth, and the (c)synthesized tenth frame of the real "Miss America" sequence including the photometric effects; (d) absolute difference between the real and the synthesized tenth frames.

Figure 4.7: (a)The first, (b)tenth, and the (c)synthesized tenth frame of the real "Miss America" sequence without the photometric effects; (d) absolute difference between the real and the synthesized tenth frames.

Figure 4.8: The 8 frames obtained by omitting every other frame of the first 16 frames of the original "Claire" image sequence.

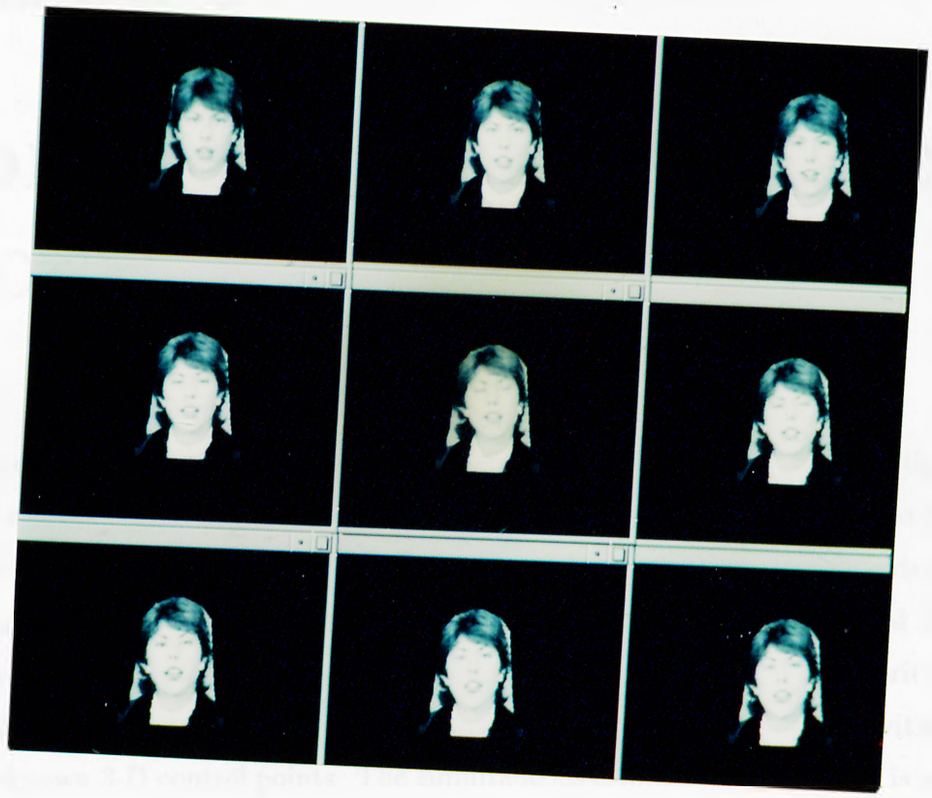

Figure 4.9: The synthesized "Claire" sequence using the estimated motion and structure parameters.

Chapter 5

CONCLUSION AND FUTURE WORK

In this dissertation, we address the problem of 3-D motion estimation in the context of 3-D object based coding of facial image sequences. The main contribution of our approach is that, it handles the global and local motion estimation and the adaptation of a generic wire-frame to a particular speaker simultaneously within an optical flow based framework including the photometric effects of motion. In addition, the algorithm tracks the motion without having to perform a synthesis step in each iteration and without using any preknown 3-D control points. The simultaneous estimation formulation is motivated by the fact that estimation of the global motion, local motion and adaptation of the wire-frame model are mutually related; thus a combined optimization approach is necessary. The estimation is done by minimizing the squared error in the optical flow equation (4.13) over all pixels in a frame corresponding to the facial region. The validity of the estimated 3-D motion and structure parameters are tested by controlled experiments in terms of the synthesis error at convergence where the synthesis error is due to (i) misfit of the wire-frame model to the actual speaker, (ii) 3-D global motion estimation error, (iii) 3-D local motion estimation error, (iv) error in the estimation of photometric effects of the motion. Our experiments show that the simultaneous estimation gives

more accurate results than the ones found in the literature [20],[26]-[28] as expected due to the mutuality of the estimated parameters. We also see that the small motion assumption given in Chapter 2 is justified in the experiments we have conducted using real sequences as seen from the results. The incorporation of the photometric effects to the formulation also improves the estimation results by a considerable amount (Tables 4.3-4.6, Figs. 4.3-4.6). When we compare the MSEs obtained from the simulated and real Miss America sequences, we see that the error decreases by about 7% when we incorporate the photometric effects (see Sec. 4.4.1 and 4.4.2).

We also consider the efficiency of the algorithm in terms of bit rate and the signal-to-noise ratio (SNR). Assuming a picture frequency of $25/3Hz$, and using floating point representation for the transmission of rotation and translation parameters, we will get a bit rate of approximately $20kbps$ without any entropy coding. If we introduce entropy coding, we can further decrease the bit rate. If we consider the SNR, the results are again encouraging. For the Claire sequence, peak SNR is $31.25dB$. If we compare these results with that of proposed object based methods in the context of COST211, we see that we get similar results. In case of COST211 results, the bit rate is about $56kbps$ for a CIF size Claire sequence resulting in a peak SNR of $34dB$. As a result, we obtained the same quality images without increasing the bit rate.

Future work at this point will include analysis of the quantization effects to the performance of the algorithm and implementation of the parameter coding. Also, to decrease the synthesis error further, the proposed method is aimed to be modified to take care of the change in texture as a result of motion. In the proposed algorithm, the frame has to be updated when the estimated motion parameters are incapable of synthesizing the actual frame. However, the error may also increase due to the change in texture, i.e. when the new frame cannot be synthesized using the texture information of the previous frame. This can be incorporated into the formulation by also minimizing the synthesis error between the actual second frame and its 3-D object based synthesis which would require a synthesis step after each perturbation. Another point that can be investigated is the representation of the global motion also in terms of the structure

parameters. This decreases the number of parameters that is perturbed during the iterations, but the bit-rate is expected to increase since the projections of the global motion should also be transmitted in this case.

In this study, we have also shown that since the 3-D motion equation is a nonlinear equation in terms of the motion and the depth parameters, finding a least-squares solution iteratively does not always give the correct results. The iteration may converge to a local minimum unless we have a good initial guess solution as shown in Chapter 3. To avoid this we use a random perturbation in one of the parameters (in our case depth) that causes the nonlinearity. It has been shown that the improved algorithm converges to the true motion and depth parameters even in the presence of 50% error in the initial depth estimates.

As a future research, another point that can be investigated is the application of this scheme to Asynchronous Transfer Mode (ATM) environment since ATM can provide a high degree of flexibility in video communications and take advantage of the inherent burstiness of video information [87]. Since no major studies of diverse traffic in ATM networks have been performed so far, the area of video traffic characterization is still open to further research.

As a result, in this thesis we have shown how to improve the accuracy of motion estimation in the context of 3-D object based coding. Since the bit rates achieved by 3-D object based coding are low enough, with the improved quality, this method has a chance to be the basis of the future videophones. Also this kind of approach is expected to open up new applications in image processing techniques such as graphical animation and automatic answering machines using video.

Bibliography

[1] A.N.Netravali, B.G.Haskell, Digital Pictures: Representation and Compression, Plenum Press, New York, 1988.

[2] A.Rosenfeld, A.C.Kak, Digital Picture Processing, Academic Press Inc., Orlando, 1982.

[3] M.Rabbani, P.W.Jones, Digital Image Compression, vol TT7, SPIE Optical Eng. Press, USA, 1991.

[4] A.K.Jain, Fundamentals of Digital Image Processing, Printice-Hall International, NJ, 1989.

[5] J. S. Lim, Two-dimensional Signal and Image Processing, Prentice-Hall International, Inc., NJ, 1990.

[6] N. D. Kenyon, "Audiovisual telecommunications," in "BT Telecommunication Series, Audiovisual Telecommunication," N. D. Kenyon and C. Nightingale, ed., Chapman & Hall, Great Britain, 1992.

[7] R. G. Gallager, Information Theory and Reliable Communication, John Wiley & Sons, USA, 1968, Ch.1, Ch. 9.

[8] C. Chen, "Video compression: Standards and applications," *J. Visual Comm. and Image Rep.*, vol. 4, no. 4, June 1993, pp. 103-111.

[9] T. R. Hsing and K. H. Tzou, "Video compression techniques for visual telephony," *J. Imag. Tech.*, vol. 15(1), Feb. 1989, pp. 15-19.

[10] N. Jayant, "Signal compression: Technology targets and research directions," *IEEE JSAC,* vol. 10, no. 5, June 1992, pp. 796-819.

[11] R. M. Gray, "Vector quantization," *IEEE ASSP Magazine*, April 1984, pp. 4-29.

[12] A. Gersho, "On the structure of vector quantizers," *IEEE Trans. on Info. Theory*, vol. IT-28, no. 2, March 1982, pp. 157-166.

[13] M. Kunt, M. Benard, and R. Leonardi, "Recent results in high-compression image coding," *IEEE Trans. on Circuits and Systems*, vol. CAS-34, no.11, Nov. 1987, pp. 1306-1336.

[14] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second-generation image coding techniques," *Proc. of IEEE*, vol. 73, no. 4, April 1985, pp. 549-574.

[15] J. W. Woods and S. D. O'Neil, "Subband coding of images," *IEEE Trans. Acoust. Speech Sign. Proc.*, vol. 34, Oct. 1986, pp. 1278-1288.

[16] R. Forchheimer and T. Kronander, "Image coding-from waveforms to animation," *IEEE Trans. Acoust. Speech Sign. Proc.*, vol. 37, no. 12, Dec. 1989, pp. 2008-2023.

[17] H. G. Musmann, M. Hotter, and J. Ostermann, "Object-Oriented Analysis-Synthesis Coding of Moving Images," *Signal Processing: Image Communication*, vol. 1, 1989, pp. 117-138.

[18] N. Diehl, "Object-oriented motion estimation and segmentation in image sequences," *Signal Processing: Image Communication*, vol. 3, 1991, pp. 23-56.

[19] N. Diehl, "Model-Based Image Sequence Coding," in "Motion Analysis and Image Sequence Processing," M. I. Sezan and R. L. Lagendijk, ed., Kluwer Academic Publishers, 1993.

[20] K. Aizawa, H. Harashima, and T. Saito, "Model-based analysis-synthesis image coding (MBASIC) system for a person's face," *Signal Processing: Image Communication*, vol. 1, 1989, pp. 139-152.

[21] T. S. Huang, S. C. Reddy, and K. Aizawa, "Human facial motion modeling, analysis, and synthesis for video compression," *SPIE Visual Comm. and Image Proc'91*, vol. 1605, Nov. 1991, pp. 234-241.

[22] W. J. Welsh, "Model-based coding of videophone images," *Electronics and Communication Engineering Journal,* Feb. 1991, pp. 29-36.

[23] W. J. Welsh, S. Searby, and J. B. Waite, "Model-based image coding," in "BT Telecommunication Series, Audiovisual Telecommunication," N. D. Kenyon and C. Nightingale, ed., Chapman & Hall, Great Britain, 1992.

[24] M. Kaneko, A. Koike, and Y. Hatori, "Coding of facial image sequence based on a 3D model of the head and motion detection," *J. Visual Comm. and Image Rep.*, vol. 2, no. 1, March 1991, pp. 39-54.

[25] K. Aizawa, C. S. Choi, H. Harashima, and T. S. Huang, "Human Facial Motion Analysis and Synthesis with Application to Model-Based Coding," in "Motion Analysis and Image Sequence Processing," M. I. Sezan and R. L. Lagendijk, eds., Kluwer Academic Publishers, 1993.

[26] H. Li, P. Roivainen, and R. Forcheimer, "3-D Motion Estimation in Model-Based Facial Image Coding," *IEEE Trans. Patt. Anal. Mach. Intel.*, Vol. 15, June 1993, pp. 545-555.

[27] J. Ostermann, "Analysis-synthesis Coder based on Moving Flexible 3-D Objects," in Picture Coding Symp. (PCS-93) p. 2.8, March 1993.

[28] J. Ostermann, "Object-based Analysis-Synthesis Coding (OBASC) based on the source model of moving flexible 3-D objects," to appear in *IEEE Trans. Image Proc.*, 1993.

[29] H. Morikawa and H. Harashima, "3D structure extraction coding of image sequences," *J. Visual Comm. and Image Rep.*, vol. 2, no. 4, Dec. 1991, pp. 332-344.

[30] CCITT Recommendation H.261, Video Codec for Audiovisual Services at $p \times 64Kbit/s$, COM XV-R 37-E, 1990.

[31] International Organization for Standardization ISO MPEG, "Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about 1.5Mbits/s," Nov. 1991.

[32] Project COST211 bis, Redundancy Reduction Techniques for Coding of Broadband Video Signals, Final Report.

[33] Description of the BOSCH/UNI-HAN object-oriented analysis-synthesis coder for very-low-bit-rate applications, SIM(92)63.

[34] R. Baker, "Waveform based very low rate video coding," Workshop on very low bit rate video compression, Urbana, IL, May 1993.

[35] M. J. T. Reinders, B. Sankur, and J. C. A. van der Lubbe, "Transformation of a general 3D facial model to an actual scene face," *11th Int. Conf. Pattern Recog.*, 1992, pp.75-79.

[36] G. Bozdağı, A. M. Tekalp, and L. Onural, "An improvement to MBASIC algorithm for 3-D Motion and Depth Estimation," accepted for publication in *IEEE Trans. Image Proc., Special Issue on Image Sequence Processing*, 1993.

[37] A. Verri and T. Poggio, "Motion Field and Optical Flow: Qualitative Properties," *IEEE Trans. Patt. Anal. Mach. Intel.*, vol. PAMI-11, no. 5, May 1989, pp. 490-498.

[38] J. N. Driessen, *Motion Estimation for Digital Video*, PhD Thesis, Delft University of Technology, Department of Electrical Engineering, Information Theory Group, Delft, The Netherlands, Sept. 1992.

[39] F. I. Parke, "Parameterized models for facial animation," *IEEE Computer Grap. and App.*, vol. 2, no. 9, Nov. 1982, pp. 61-68.

[40] N. D. Duffy, J. F. S. Yau, "Facial image reconstruction and manipulation from measurements obtained using structured lighting technique," *Pattern Recog. Letters*, no. 7, 1988, pp. 239-243.

[41] N. Magnenat-Thalman, *et.al.*, "Human prototyping," *Proc. of Comp. Graph. Intern. 88*, pp. 74-84.

[42] M. J. T. Reinders, B. Sankur, and J. C. A. van der Lubbe, "3D scene modeling in videophone," *Proc. of Sixth Int. Sym. Comp. and Info. Sciences, ISCIS VI*, vol.2, 1991, pp. 933-943.

[43] K. Aizawa, H. Harashima, " A model-based analsis/synthesis image coding scheme," *Elec. and Comm. in Japan*, part 1, vol. 73, no. 4, 1990, pp. 1-9.

[44] D. Terzopoulos, K. Waters, "Physically-based facial modeling, analysis and animation," *Journal of Visualization and Computer Animation*, vol. 1, 1990, pp. 73-80.

[45] D. Terzopoulos and K. Waters, "Analysis and synthesis of facial image sequences using physical and anotomical models," *IEEE Trans. Patt. Anal. Mach. Intel.*, vol. PAMI-15, no. 6, June 1993, pp. 569-580.

[46] Y. Nakaya, Y. C. Chuah, and H. Harashima,"Model-based/waveform hybrid coding for videophone images," *Int. Conf. on ASSP'91*, M9.8, pp. 2741-2744.

[47] C. S. Choi, K. Aizawa, H. Harashima, and T. Takebe, "Analysis and synthesis of facial expressions in model based image coding," *PCS'90*, 9.15-1.

[48] T. Fukuhara, K. Asai, T. Murakami, "Hierarchical division of 3D wireframe model and VQ in a model-based coding of facial image," *PCS'90*, 7.2-1.

[49] F. Lavagetto, S. Zappatore, "Customized wireframe modeling for facial image coding," *Third Int. Workshop on 64kbits/s Coding of Moving Video*, 1990, 4.2.

[50] K. Aizawa, *et.al.*, "Model-based synthesis image coding system," *Proc. GLOBECOM'87*, Nov. 1987, pp. 45-49.

[51] H. Harashima, K. Aizawa, and T. Saito, "Model-based analysis-synthesis coding of videotelephone images-Conception and basic study of intelligent image coding," *Trans. of the IEICE*, vol. E72, no. 5, May 1989, pp. 452-459.

[52] K. Aizawa, H. Harashima, and T. Saito, "Model-based synthetic image coding system-Construction of a 3D model of a person's face," *PCS'87*, 3.11.

[53] M. Rydfalk, "CANDIDE: A parametrised face," Dept. Elec. Eng. Rep. LiTH-ISY-I-0866, Linköping Univ., Oct. 1987.

[54] H. Huang, M. Ouhyoung, and J. Wu, "Automatic feature point extraction on a human face in model based image coding," Opt. Eng., vol. 32, no. 7, July 1993, pp. 1571-1580.

[55] J. B. Welsh and W. J. Welsh, "Head boundary location using snakes," in "BT Telecommunication Series, Audiovisual Telecommunication," N. D. Kenyon and C. Nightingale, ed., Chapman & Hall, Great Britain, 1992.

[56] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images - A review," *Proc. IEEE*, vol. 76, no. 8, Aug. 1988, pp. 917-935.

[57] P. Ekman, and W. V. Friesen, Facial Action Coding System, Consulting Psyhologists Press, 1977.

[58] R. Forcheimer, "The motion estimation problem in semantic image coding," *PCS'87*, pp. 171-172.

[59] A. Koike, M. Kaneko, and Y. Hatori, "Model-based image coding with 3D motion estimation and shape change detection," *PCS'90*, 9.5-1.

[60] C. S. Choi, H. Harashima, and T. Takebe, "Analysis and synthesis of facial expressions in knowledge-based coding of facial image sequences," *Proc. ICASSP'91*, M9.7, pp. 2737-2740.

[61] D. Terzopoulos, K. Waters, "Analysis of facial images using physical and anatomical models," *IEEE 3rd ICCV*, Dec. 1990, pp. 727-732.

[62] M. Kass, A. Witkin, D. Terzopoulos, "Snakes: Active contour models," *Int. Journal of Comp. Vision*, 1988, pp. 321-331.

[63] K. Waters, "A muscle model for animating three-dimensional facial expressions," *Computer Graphics*, vol. 21, no. 4, July 1987, pp. 17-24.

[64] V. Seferidis, "Facial feature estimation for model-based coding," *Electronic Letters*, vol. 27, no. 24, Nov. 1991, pp. 2226-2228.

[65] A. L. Yuille, D. S. Cohen, and P. W. Hallinan, "Feature extraction from surfaces using deformable templates," *Proc. IEEE Conf. on CVPR*, June 1989, pp. 104-109.

[66] M. A. Shackleton, W. J. Welsh, "Classification of facial features for recognition," *Proc. IEEE Conf. on CVPR*, 1991.

[67] J. F. S. Yau, N. D. Duffy, "A texture mapping approach to 3D facial image synthesis," *Computer Graphics Forum*, no. 7, 1988, pp. 129-134.

[68] D. Pearson, "Texture Mapping in Model-Based Image Coding," *Signal Processing: Image Comm.* Vol. 2, pp. 377-395, 1990.

[69] Y. Nakaya, K. Aizawa, and H. Harashima, "Texture updating methods in model-based coding of facial images," *PCS'90*, 7.3-1.

[70] B. Klaus and P. Horn, Robot Vision, MIT Press, 1986.

[71] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 239, 1981, pp. 133-135.

[72] B. Girod, "Motion estimation and very low bitrate video compression," Workshop on very low bit rate video compression, Urbana, IL, May 1993.

[73] R. Y. Tsai and T. S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Trans. Patt. Anal. Mach. Intel.*, vol. PAMI-6, 1984, pp.13-27.

[74] K. Kanatani, "Structure and motion from optical flow under orthographic projection," *Comp. Vision Graph. Image Proc.*, vol. 35, 1986, pp. 181-199.

[75] B. K. P. Horn and B. G. Schunk, "Determining optical flow," *Artif. Intell.*, vol. 17, 1981, pp. 185-203.

[76] K. Zeger and A. Gersho, "Stochastic relaxation algorithm for improved vector quantizer design," *Electronic Letters*, vol. 25, no. 14, July 1989, pp. 96-98.

[77] A. Pentland, "Photometric Motion," *IEEE Trans. Patt. Anal. Mach. Intel.*, vol. PAMI-13, no. 9, Sept. 1991, pp. 879-890.

[78] A. Pentland, "Finding the illuminant direction," *J. Opt. Soc. Am.*, vol. 72, no. 4, April 1982, pp. 448-455.

[79] H. Lee and A. Rosenfeld, "Improved methods of estimating shape from shading using light source coordinate system," in *Shape from Shading,* (B.K.P. Horn and M.J. Brooks, eds.), Cambridge, MA, MIT Press, 1989, pp. 323-569.

[80] R. Szeliski, "Fast shape from shading," *CVGIP: Image Understanding*, vol. 53, no.2, March 1991, pp. 129-153.

[81] Q. Zheng and R. Chellappa, "Estimation of illuminant direction, albedo, and shape from shading," *IEEE Trans. Patt. Anal. Mach. Intel.*, vol. PAMI-13, no. 7, July 1991, pp. 680-702.

[82] A. A. Zhigljavsky, Theory of Global Random Search, ch. 3, Kluwer Academic Publishers, 1991.

[83] G. Bozdağı, A. M. Tekalp, and L. Onural, "3-D Motion and Structure Estimation including Photometric Effects with Application to Model-Based Coding of Facial Image Sequences," IEEE MDSP'93 Workshop, MO-01, pp. 114-115, 1993, France.

[84] G. Bozdağı, A. M. Tekalp, and L. Onural, "Simultaneous estimation of 3-D motion and structure parameters for model-based image coding," COST211ter European Workshop on New Techniques for Coding of Video Signals at Very Low Bitrates, pp. 4.4.1-4.4.3, 1993, Hannover.

[85] Description of the BOSCH/UNI-HAN Object-oriented analysis-synthesis coder for very low bit rate applications, internal report of COST211ter, 1992.

[86] Simulation model for object-based coding, internal report of COST211ter, 1993.

[87] I. Nikolaidis and I. Akyildiz, "Source characterization and statistical multiplexing in ATM networks," submitted for publication.

Vita

Gözde Bozdağı was born in Ankara, Turkey, in 1967. She received her B.Sc. degree from the Middle East Technical University, Ankara, Turkey, and M.Sc. degree from the Bilkent University, Ankara, Turkey, in 1988 and 1990, respectively, both in electrical and electronics engineering. Her current research interests include very-low-bit-rate image coding, 3-D motion estimation and holography.