Techniques for Low Bit-rate Video Coding

Wong Kwok-Wai

A dissertation* submitted in partial fulfillment of the requirements for the degree of

Master of Philosophy

Department of Electronic and Information Engineering
The Hong Kong Polytechnic University

March 2001

*This research work was supported by The Hong Kong Polytechnic University Grand A/C No. G-V709

Techniques for Low Bit-rate Video Coding

Abstract

Digital video compression techniques play an important role in the multimedia information era. Hence, the objectives of this research work are to devise and develop efficient methods for video coding. In this thesis, different approaches for video coding such as block-based motion compensation coding scheme, and foreground/background video coding scheme will be described. Furthermore, several successful video coding standards such as H.261/H.263, and H.263+ will be reviewed in this thesis. Due to the nature of block-based coding, blocky artifacts may appear on the compressed video. We will also review some advanced video coding algorithms that solve this type of problems. As for the foreground/background video coding scheme, the first step is to segment out the targeted object in a video sequence. Since the major component for many applications, such as videophone and video conferencing, is the human face, we will also provide an overview of the techniques.

In our research, we proposed an efficient algorithm for human face detection and facial feature extraction. The genetic algorithm and the eigenface technique are employed for detecting the face and facial feature regions in our approach. The genetic algorithm is applied to search for possible face regions in an image, while the eigenface technique is used to determine the fitness of the regions. As the genetic algorithm is computationally intensive, the searching space is reduced and limited to the eye regions in order to

substantially reduce the computational time. Possible face candidates are then further verified by measuring their symmetries and determining the existence of the different facial features. Furthermore, in order to improve the level of detection reliability in our approach, the lighting effect and orientation of the faces are considered and their related problems solved.

A very low bit-rate video coding algorithm by focusing on moving region has also been proposed. Eight patterns are pre-defined to approximate the moving regions in a macroblock. The patterns are then used for motion estimation and compensation to reduce the prediction errors. Furthermore, in order to improve the compression performance, the residual errors of a macroblock are rearranged into a block with no significant increase of high order DCT coefficients. As a result, both the prediction efficiency and the compression efficiency are improved. Finally, a foreground/background video coding system is developed for application of the videophone. We have also incorporated information about the facial features in our moving region based coding algorithm in order to provide a better picture quality perceptually.

Acknowledgements

I would like to express my sincere gratitude to my Chief Supervisor, Dr. K. M. Lam, and to my co-supervisor, Prof. W. C. Siu. Without their support, this research work would not have been completed. They also offered me many invaluable ideas and suggestions in writing my thesis.

I am also thankful to all the members of the DSP Research Laboratory, past and present, especially Dr. Huaqiu Deng, Mr. K. H. Lin, Mr. W. P. Choi, Mr. Baofeng Gao, Mr. K. C. Lai, Mr. S. K. Yip, Mr. K. C. Hui, Mr. K. P. Cheung, Mr. W. F. Cheung, Mr. K. T. Fung, Mr. Tommy Chan, Mr. K. K. Yiu, Mr. Manson Siu, and Mr. C. B. Chow. The countless discussions I had with them have been proved to be both fruitful and inspiring.

I would also like to thank all members of staff in the Department of Electronic and Information Engineering, as well as the clerical staff in the General Office. They have created a stimulating environment for me to work in.

Finally, it is my pleasure to acknowledge and to thank the Research and Postgraduate Studies Office of The Hong Kong Polytechnic University for its generous support over the past two years.

Without the patience and forbearance of my family, the preparation of this research work would have been impossible. I appreciate their constant and continuous support and understanding.

TABLE OF CONTENTS

LIST OF FIGURES iii

LIST OF TABLES vi

CHAPTER 1 ♦INTRODUCTION P.1
1.1 Motivation of Digital Video Compression P.1
1.2 Introdution to Video Compression Techniques P.2
1.3 Investigated Approaches P.3
1.4 Organization of the Thesis P.4

CHAPTER 2 ♦FACE SEGMENTATION AND FACIAL FEATURE EXTRACTION..P.5
2.1 Introduction P.5
2.2 Problems of Face Segmentation and Facial Feature Extraction P.5
2.3 Face Segmentaion P.6
2.3.1 Shape Analysis Approach P.6
2.3.2 Knowledge-based Approach P.7
2.3.3 Example-based Learning Approach..P.8
2.3.4 Color Analysis Approach P.8
2.4 Faical Feature Extraction P.10
2.5 Applications P.12
2.5.1 Face Recognition P.12
2.5.2 Coding Area of Interest with Better Quality P.13
2.5.3 Model-based Video Coding P.13
2.6 Summary P.15

CHAPTER 3 ♦MOTION COMPENSATED PREDICTIVE-TRANSFORM CODING TECHNIQUES..P.17
3.1 Introduction P.17
3.2 Hybrid Video Coding Techniques P.18
3.2.1 Motion Compensated Prediction P.19
3.2.2 Transform Coding P.19
3.2.3 Hybrid DCT/DPCM Coder P.20
3.3 Motion Estimation and Compensation P.22
3.3.1 Basic Principle of Block Matching Methods P.23
3.3.2 Various Approches for Motion Estimation and Compensation P.24
3.4 Digital Video Coding Standards P.29
3.4.1 ITU-T H.261 and H.263 Video Coding Schemes P.29
3.4.2 ITU-T H.263+ Video Coding Scheme P.33
3.5 Advanced Video Coding Algorithms P.37
3.5.1 Texture Coding Algorithms P.37
3.5.2 Video Coding Using Pre-defined Patterns P.39
3.5.3 Foreground/ Background Video Coding Algorithms P.39
3.5.4 2D/3D Hybrid Video Coding Algorithms P.40

3.6 Rate Control Scheme P.41
3.6.1 Macroblock Type Decision P.42
3.6.2 Quantization Step Size P.43
3.7 Summary P.46

CHAPTER 4 ♦ AN EFFICIENT ALGORITHM FOR HUMAN FACE DETECTION AND FACIAL FEATURE EXTRACTION UNDER DIFFERENT CONDITIONS P.49
4.1 Introduction P.49
4.2 Human Face Detection Using Genetic Algorithm and Eigenface Technique...P.50
4.2.1 Possible Eye Candidates Detection P.51
4.2.2 Structure of a Chromosome P.52
4.2.3 Normalization of the Possible Face Regions P.55
4.2.4 The Fitness Function P.57
4.2.5 Verification of Face Regions and Facial Feature Extraction P.59
4.3 Experiment Results P.63
4.4 Conclusion P.66

CHAPTER 5 ♦ AN EFFICIENT LOW BIT-RATE VIDEO CODING ALGORITHM FOCUSING ON MOVING REGIONS P.69
5.1 Introduction P.69
5.2 Low Bit-rate Video Coding Using Patterns P.70
5.2.1 Moving Region Detection P.71
5.2.2 Architecture of the Encoder and Decoder P.74
5.2.3 Motion Estimation and Compensation P.76
5.2.4 Prediction Errors Encoding P.78
5.3 Simulation Results P.80
5.4 Conclusion P.83

CHAPTER 6 ♦ FACE SEGMENTATION AND FACIAL FEATURE TRACKING FOR VIDEOPHONE APPLICATIONS... P.85
6.1 Introduction P.85
6.2 Efficient Coding Method for the Face Region P.86
6.2.1 Face Detection and Facial Feature Extraction P.86
6.2.2 Face and Facial Fetaure Tracking P.88
6.2.3 Quantization Step Sizes for the Face and Non-face Regions P.92
6.3 Simulation Results P.92
6.4 Conclusion P.95

CHAPTER 7 ♦ CONCLUSION AND FUTURE WORK P.96
7.1 Conclusion P.96
7.2 Future Work P.98

APPENDIX I P.100

AUTHOR'S PUBLICATIONS AND REFERENCES P.101

List of Figures

2.1 An elliptical face location model P.7

2.2 Skin-tone color distribution on Cr-Cb chrominance plane P.9

2.3 Claire image and the result of color segmentation P.9

2.4 An example of defined facial feature geometry P.10

2.5 (a) Deformable eye template P.11

(b) Deformable mouth template P.12

2.6 The "*CANDIDE*" wire-frame model fitted onto the "*Miss America*" image P.14

2.7 Synthesized image by rotating 3D model: (a) Frontal view, and (b) side view P.14

3.1 Block diagram of an hybrid DCT/DPCM video coder P.20

3.2 Block diagram of an hybrid motion compensated DCT video decoder P.21

3.3 Block-based motion estimation P.23

3.4 Initial chromosomes distribute equally in the search space P.26

3.5 (a) Half-pixel precision using bilinear interpolation P.28

(b) Half-pixel accurate motion vector estimation P.28

3.6 Block diagram of an H.263 baseline video encoder P.30

3.7 Motion vector prediction P.31

3.8 Merging types of luminance sub-block pairs in a macroblock P.38

3.9 Block diagram of backward rate control scheme P.42

3.10 The relationships between the buffer occupancy and QP using (a) linear rate control scheme, and (b) non-linear rate control scheme P.44

4.1 Eye candidates detection: (a) original image, (b) possible eye regions, and (c) the good candidates for the detected eye regions P.52

4.2 Structure of a chromosome P.53

4.3 The defined geometry of our head model P.54

4.4 (a) A normal face candidate, and (b) adjusted possible face candidate P.56

4.5 Shirring angle approximation P.56

4.6 (a) Reference face image, (b) original face image with half in shadow,
and (c) the histogram normalized face image. P.57

4.7 The crossover process P.58

4.8 Windows for facial feature extraction P.61

4.9 The y-projection results of (a) eye region, (b) nose region, and (c) the mouth region
.... P.62

4.10 The x-projection results of (a) nose region, and (b) the mouth region P.63

4.11 An example of facial feature extraction by analyzing the projection of normalized
face region P.63

4.12 (a) Experiment results under head-on lighting P.67

(b) Experiment results when the lighting is 45° P.67

(c) Experiment results when the lighting is 90° P.67

(d) Some more experiment results P.68

4.13 Error in facial feature extraction P.68

5.1 Defined patterns: (a) the eight patterns for moving region approximation, and (b) the
training patterns P.71

5.2 Block type selection: (a) an AMB, and (b) a RMB approximated by pattern (1).... P.73

5.3 Block diagram of the encoder P.74

5.4 Block diagram of the decoder P.76

5.5 An example of pattern approximation for the sequence "News": (a) Detected moving regions, and (b) results of pattern approximation P.77

5.6 The principle of interframe coding for RMB P.78

5.7 Rearrangement of the residual errors P.79

5.8 PSNR of the first 100 frames for the video sequence "Akiyo" (target bit-rate=8kbps) P.82

5.9 Simulation results for the video sequence "Akiyo" with different bit-rates P.83

6.1 Detected face region and its respective facial feature points P.88

6.2 Defined "*star*" pattern for facial feature tracking P.89

6.3 Facial feature tracking using the "star" pattern on (a) eye region and (b) mouth region P.91

6.4 Results of facial feature tracking........ P.91

6.5 Segmentation results: (a) eye and mouth regions, (b) face region, and (c) background region. P.93

6.6 The PSNR of "*AKIYO*" sequence with CIF size P.94

6.7 The quality of encoded image: (a) encoded image using our approach, (b) encoded image using H.263, (c) magnified image of (c), and magnified image of (d)........ P.95

List of Tables

3.1 Key differences between H.261 and H.263 .... P.30

3.2 Performance of H.263 and H.261 schemes P.33

3.3 Summary of improvement in PSNR (dB) for H.263 and H.263+ P.36

4.1 Parameter settings P.59

4.2. Experiment results for face detection P.65

4.3 Experiment results for the facial feature extraction P.65

5.1 The rule for classifying the types of macroblocks P.73

5.2 Simulation results P.81

5.3 Encoding time saved per frame compared to H.263 .... P.81

6.1 The proposed quantization step size for the face, facial feature, and background regions P.92

6.2 Simulation results P.94

CHAPTER 1
INTRODUCTION

The main objective of this chapter is to discuss the importance of video compression and its applications, and the different video coding techniques that are available. This Chapter serves as a broad introduction to the rest of the chapters of this thesis.

1.1 Motivation of Digital Video Compression

The demand for digital video communication applications such as video conferencing, video-on-demand, digital video broadcasting, high-definition television (HDTV), and multimedia image/video database services has increased considerably. However, the representation of uncompressed image and video requires a very large number of bits. An uncompressed digitized NTSC video signal with 720 pixels times 480 lines, 8 bits per pixel for each colour component, and 30 frames per second requires a bandwidth of 720×480×8×3×30=249M bits per second for transmission. To store one hour of uncompressed video signal, it requires 112G bytes space. Here it is not efficient to store and transmit the huge amount of video data. Therefore, the goal of image and video compression is to reduce the number of bits as much as possible, and to reconstruct a faithful duplicate of the original picture.

1.2 Introduction to Video Compression Techniques

Digital data compression algorithms can be classified into two categories: lossless compression and lossy compression. A lossless compression algorithm is a method that the identical source data can be reconstructed from the compressed data. Huffman coding and arithmetic coding are examples of lossless compression techniques. However, lossless compression often has a low compression ratio. Sometimes the compression is insignificant in some applications which require a high compression ratio. The lossy compression techniques can achieve a higher compression ratio than the lossless compression. A lossy compression algorithm is a method that the decompressed data is not identical to the original signal but instead is reasonably close to it. Therefore, for purpose of efficient storage and transmission of digital video signals, the lossy techniques are commonly used.

Various lossy approaches for video coding have been proposed, such as block-based motion compensated prediction coding, model-based coding, vector quantization coding, wavelet-based coding, and fractal-based coding. These encoding approaches are designed to discard information that is not perceptible to humans or information that is redundant. In fact, the block-based approaches are being widely used for video coding. There are many important and commonly recognized world-wide standards for image and video coding that make use of the block-based techniques, such as JPEG, Motion JPEG, ITU-T H.261[1], H.263[2], H.263+[3], ISO/IEC MPEG-1, MPEG-2, and MPEG-4[4]. These standards address a wide range of applications and they have different requirements in terms of bit rate, picture quality, complexity, error resilience, and delay. Model-based coding is one of the developing schemes for targets of very low bit-rate. This approach to video coding assumes a parameterized model for each object of an image. The parameters to each object will be coded and transmitted.

1.3 Investigated Approaches

The objective of this research is to investigate and develop an efficient very low bit-rate video coding system. The human face is the most important component for many applications such as video-phone and video-conferencing. The perceptual picture quality can be improved if more bandwidth or more bits are spent on the human face region, and less bits are used to represent the background scene. Hence, in this research work, an efficient very low bit-rate video coding system based on the properties of the face region and its facial features has been developed.

The system consists of three major parts: an automatic human face detection and facial feature extraction, facial feature tracking, and video coding algorithm based on moving regions and the properties of the facial features. In the first part, the location of the human face and its respective facial features are extracted by means of eigenface techniques. After the face region and the facial features have been detected in the first frame, the detected face region and its respective facial features are tracked on the subsequent frames. Finally, the locations of the facial features are passed to the encoding system for compression. The resulting picture quality of the object and the encoding time is superior to that produced by the conventional video coding methods.

1.4 Organization of the Thesis

The rest of this thesis will introduce the face segmentation techniques and their applications, the existing video coding techniques, and our proposed algorithms for face detection and video coding.

Chapter 2 focuses on the specific problems of face segmentation and its applications to videoconferencing. Various approaches to face segmentation and facial feature extraction will be also introduced. Chapter 3 describes the basic principle of the block-based motion compensated prediction coding approaches. It includes many techniques such as motion estimation and compensation, rate control scheme for video coding, and the existing international video coding standards. In Chapter 4, an efficient approach for human face detection and feature extraction using the eigenface technique and the genetic algorithm will be introduced. This approach is used in the first part of the proposed foreground/background coding method described in Chapter 6. In Chapter 5, we proposed a very low bit-rate video coding algorithm by using pre-defined patterns. In comparing the proposed method with the H.263, our experiments show that the coding performance of the proposed is better than that of the H.263 scheme. In Chapter 6, a foreground/background coding method for videophone applications is proposed. It combines our research work on face detection, facial feature tracking, and video coding algorithms. Finally, a summary of the major developments and the conclusion of this research work are provided in Chapter 7.

CHAPTER 2

FACE SEGMENTATION AND FACIAL FEATURE EXTRACTION

2.1 Introduction

Digital images and video are becoming more and more important for multimedia applications. The human face is one of the most important objects in an image or video. Detecting the location of human faces and then extracting the facial features in an image is an important capability with a wide range of applications, such as human face recognition, surveillance systems, human-computer interfacing, video-conferencing, etc. In an automatic face recognition system, the first step is to segment the face in an image or video regardless of whether the background is simple or cluttered. For model-based video coding, the synthesis performance is quite dependent on the accuracy of the facial feature extraction process. In other words, a reliable method for detecting the face regions and locating the facial features is indispensable to such applications. In this chapter, various approaches for face segmentation and facial feature extraction are reviewed.

2.2 Problems of Face Segmentation and Facial Feature Extraction

The task of finding a person's face and its corresponding facial features in an unconstrained image is a challenging process. It is very difficult to locate accurately the position of faces in an image. Several variables affect the detection performance, such as the presence of glasses, skin color, gender, facial hair, and facial expression. Furthermore, being a 3-D object, the human face might be under a distorted perspective and uneven illumination. As a result, the true face may not be detected. Moreover, facial feature extraction is a time-consuming process due to the lack of constraint on the number, location, size, and orientation of faces in an image or a video scene.

2.3 Face Segmentation

Various approaches are available for solving the problems of face segmentation. They are the shape analysis approach, knowledge-based approach, color analysis approach, or more often, a combination of them. Each of the approaches to face segmentation is discussed in this section.

2.3.1 Shape Analysis Approach

One of the common methods for human face detection is ellipse fitting. Since the appearance of a typical human face takes on an oval shape, face detection in an image can be performed by detecting objects with an elliptical shape. An elliptical model for face segmentation is shown in Fig. 2.1, in which an ellipse is defined by its center ($\bar{x}, \bar{y}$), its orientation θ and the length a and b of its minor and major axes. The main purpose of the ellipse fitting approach is to find the $\bar{x}, \bar{y}$, θ, a and b parameters that best represent the shape of the head.

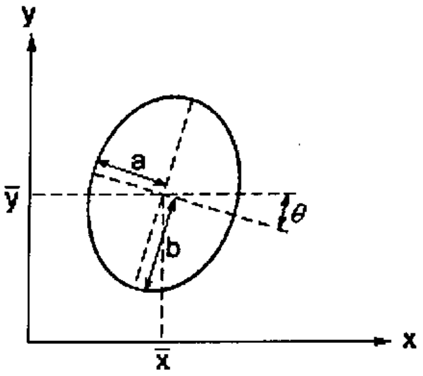


Fig. 2.1 An elliptical face location model

The ellipse fitting process can be applied after the possible outline of a face region has been extracted. In [5][6], the edge of an image is extracted for the processes of ellipse fitting. In [7], the human face is detected by extracting the elliptical shape regions using color information. However, the ellipse fitting process for face detection requires a high degree of computational complexity.

2.3.2 Knowledge-based Approach

A hierarchical knowledge-based algorithm for face detection in a complex background was proposed [8]. The algorithm consists of three levels. The higher two use mosaic images of different resolutions. The third one extracts edges of facial components. Domain knowledge and rules are applied at each level. A true face will be declared if the input region fulfils all the rules at each level. However, this method cannot detect rotated faces and it has difficulty in locating face regions accurately.

A fast approach for detecting human faces using the characteristic of the eye region has been proposed [9]. All the valley regions in an image are detected and tested for the possibility of being the human eyes. Pairs of eye candidates are grouped to form possible face candidates which are then verified as human faces or not by means of the symmetry property of human face. The processing time required for this approach is in the order of a second.

2.3.3 Example-based Learning Approach

Sung and Proggio [10] proposed an example-based learning approach to locate the vertical frontal views of human faces in complex scenes. An initial database of about 1,000 face images was used to construct a distribution-based generic face model with all its permissible pattern variations in a high-dimensional image vector space. A decision procedure was trained based on a sequence of "face" and "non-face" examples. As a result, six face clusters were found according to 4,150 normalized frontal face patterns, and six clusters for non-face were also obtained. Based on the face and non-face clusters, faces can be detected by matching the window patterns at different image locations and scales against the distribution-based face model. The reported detection performance of this approach is good, but it is very time consuming to construct a database contains a huge number of training patterns.

2.3.4 Color Analysis Approach

The use of color information has been studied for face segmentation. Some techniques [7][11]-[14] have been reported that although skin color differs from person to person and race to race, it is distributed over a very small area on the chrominance plane.

Fig. 2.2 shows the skin-tone color distribution on a Cr-Cb chrominance plane. It demonstrates that human faces have a special color distribution that differs significantly from those of the background objects. As a result, the possible human face region can be segmented by using simple thresholding techniques.

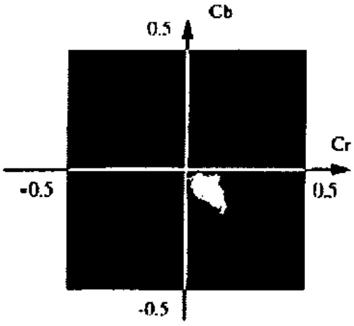


Fig. 2.2 Skin-tone color distribution on Cr-Cb chrominance plane

The use of color for skin segmentation is implemented and the segmented result of the "*Claire*" image is shown in Fig. 2.3. The experimental result, reveals that the skin color segmentation method can provide accurate and reliable results if a good contrast exists between the skin color of the face and the color of the background objects. However, if the color characteristics of the background are similar to that of the skin, the frequency of false alarms will be increased.

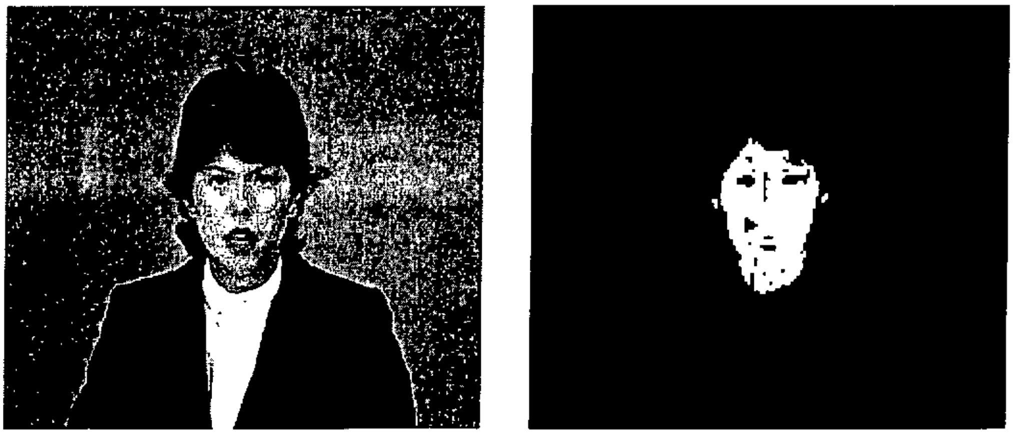


Fig. 2.3 Claire image and the result of color segmentation

2.4 Facial Feature Extraction

The location of facial feature such as the eye, nose, and mouth in a human face is basically the same among different people. Based on this characteristic, some research [15]-[19] defined a head model for extracting the facial feature after the face region has been formed. An example[15] of defined geometry for facial feature is shown in Fig. 2.4. Based on the approximated position of the facial features in a head model, various approaches can be applied to extract the actual location of the facial features. One of the approaches is extraction of the minimum point by computing the horizontal projection and vertical projection [7][15][19]. This is due to the fact that the pixel intensities in the eye, nose, and mouth regions appear darker, a minimum always occurs at the locations of the eye, nose, and mouth regions. Another approach [16][17] finds facial feature points is by determining the edge information. However, these methods work properly only under well-lit conditions. Hence, the pre-processing step of reducing the lighting effect is very important for these methods.

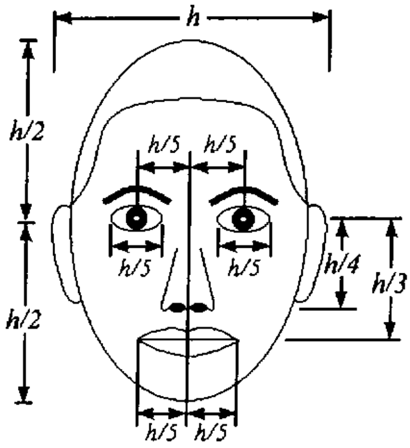


Fig. 2.4 An example of defined facial feature geometry [15]

Facial features can also be extracted by the use of deformable templates [20][21]. Since the general shapes of the human eye and mouth are more or less fixed, it is possible to use specified templates to extract the eye and mouth features. An example of the eye template and the mouth template [20] is shown in Fig. 2.5(a) and 2.5(b), respectively. The deformable eye template consists of a pair of parabolic curves (W_1^j, W_2^j) and a circle with radius r^j. h^j is the pupil point, h_l^j and h_r^j are the eye corner points, and L^j is the width of the eye, and o_o^j and o_u^j are the opening heights of the eyelids. The deformable mouth template consists of four parabolic curves (W_1, W_2, W_3, W_4). h_r^m and h_l^m are the mouth corner points, L^m is the width of the mouth, d_o^m and d_u^m are the lip thickness, and o_o^m and o_u^m are the opening heights of the lips. An energy function is defined which contains terms attracting the template to salient features such as the peaks, valleys, edges, and image intensity itself. The final size, shape and orientation of the eye and mouth templates are obtained by determining the local minimum of the respective energy functions. However, it is time consuming to determine all the parameters from which the minimum energy function is devised.

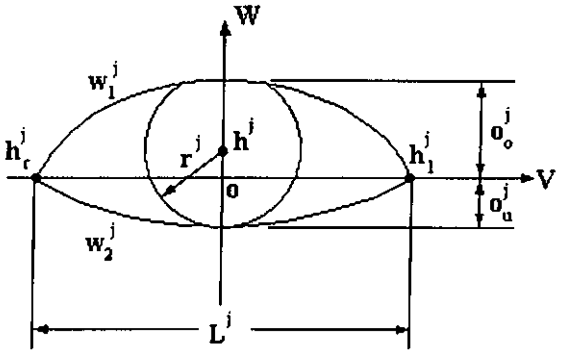


Fig. 2.5 (a) Deformable eye template [20]

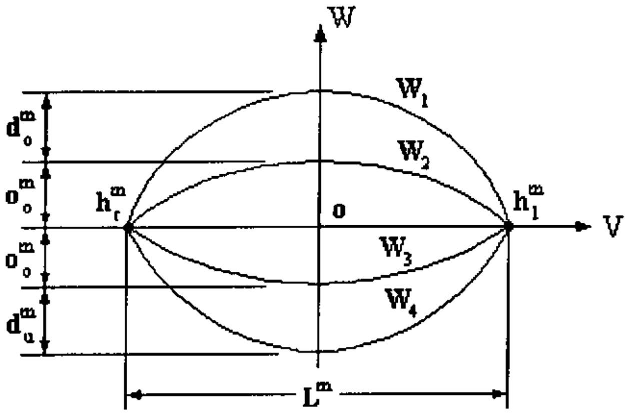


Fig. 2.5 (b) Deformable mouth template [20]

2.5 Applications

Face segmentation and facial feature extraction have important applications to human-to-human and human-to-machine communications. It is an important step in many applications, such as face recognition, object tracking, and image/video coding. Some of the major applications of face segmentation are discussed below.

2.5.1 Face Recognition

Face recognition can be applied to a wide variety of operations, including identification systems, security systems, and human-computer interaction. Various approaches to face recognition have been reported [22]-[24]. However, the first step towards developing a face recognition system is to find the locations of human faces in an unknown image or sequence. This is an important step for any face recognition system due to the fact that the accuracy of the system depends directly on the results of face segmentation and facial feature extraction.

2.5.2 Coding Area of Interest with Better Quality

The human face is a major components for video conferencing and video phone. In order to improve the subjective quality of an encoded videophone sequence, many of the proposed methods [25][26] allow the facial area to be coded at a higher quality than the non-facial regions. Hence, the picture quality of the less important regions such as the background is reduced. Accordingly, face segmentation is a very important process for the foreground/background coding techniques.

2.5.3 Model-based Video Coding

Model-based video coding [27][28] is one of the very low bit-rate coding techniques. Model-based coding uses a 3D model to synthesize the motion of the object by means of a number of parameters. Since the major object in videophone applications is the human face, the first step in these applications is extraction of the face region as well as the facial feature points in order to fit a generic 3D face model to the face images according to several features on the face. A number of 3D face models for video coding have been proposed [27][29][30]. Fig. 2.6 illustrates the implemented result derived from using the "*CANDIDE*" [27] wire-frame model that is fitted onto the "*Miss America*" face image. Different head motions as well as facial expressions can be synthesized by controlling the global motion and local motion parameters. The parameters of the head motion can be obtained by tracking the facial feature points [31]. Hence, the coding performance depends on the accuracy of feature point extraction. The results of a synthesized image using the "*CANDIDE*"model are shown in Fig. 2.7.

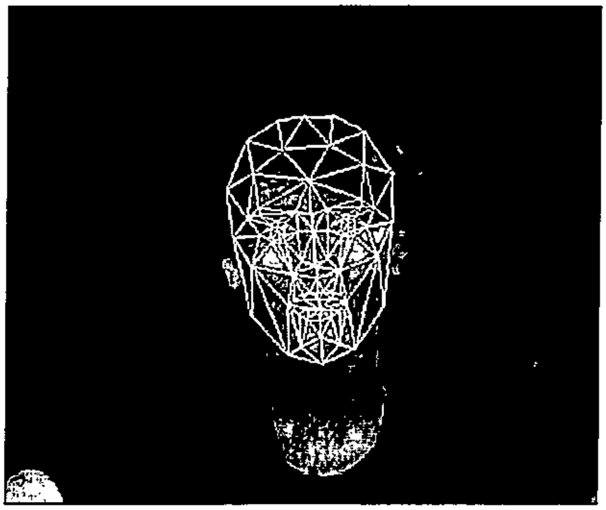
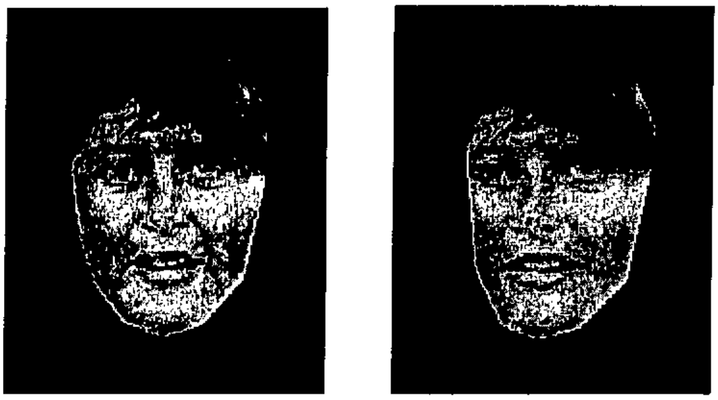


Fig. 2.6 The "*CANDIDE*" wire-frame model fitted onto the "*Miss America*" image



(a) (b)

Fig 2.7 Synthesized image by rotating 3D model: (a) Frontal view, and (b) side view

2.6 Summary

Face segmentation and facial feature extraction are important steps in many applications, such as face recognition, object tracking, image coding, and video coding. However, locating the positions of faces and their corresponding facial features in an unknown image is a difficult task. The detection performance is affected by several variables, such as the presence of glasses, skin color, gender, facial hair, and facial expression. In order to solve these problems, various approaches to face segmentation and facial feature extraction have been introduced. These include the ellipse fitting method, knowledge-based method, example-based learning method, and skin color segmentation method. In the ellipse fitting method, the shape of a head is assumed to be of elliptical and hence an ellipse with a specific size and orientation angle is used to obtain the elliptical shape in an image. In the knowledge-based method, some prior knowledge and rules regarding human faces are defined for face segmentation. A human face region is declared if the input image fulfils all the rules. As for the example-based learning method, a huge number of face and non-face images are used for obtaining the distribution in a face image. The input image is classified into the face clusters or non-face clusters according to the training results. Another popular face segmentation method is color segmentation. Based on the chrominance plane distribution of the human skin, a human face is segmented by using a simple thresholding technique to detect the skin like color region.

For facial feature extraction, an approach based on the properties of pixel intensity of facial features has been introduced. Since facial features appear darker in pixel intensity than the non-facial features, the location of the feature points on the x-axis and y-axis can be determined by finding the minimum projected locations on the x-axis and the y-axis. The approach using the eye and mouth templates for feature extraction has also been discussed. Based on a defined energy function, the respective final template for the different facial features can be obtained by finding the value of the parameters that achieve the local minimum of the energy function. However, finding an optimal solution under the templates approach is highly complex computational-wise.

CHAPTER 3

MOTION COMPENSATED PREDICTIVE-TRANSFORM CODING TECHNIQUES

3.1 Introduction

Nowadays, digital video compression techniques play an important role in the multimedia information era. Several video coding techniques, such as block-based motion compensation coding, model-based coding, vector quantization (VQ) coding, fractal-based coding, and wavelet-based coding, have been introduced. These video coding techniques aim to achieve a high compression ratio while maintaining the desired visual quality.

In this chapter, the block-based motion compensation techniques and the international standards for very low-bit rate video coding will be reviewed. We will describe the basic principle of the hybrid DCT/DPCM video coder in Section 3.2. This coder is the most popular coding technique for video sequences. It can operate in either an intraframe or interframe coding mode. With intraframe coding, spatial redundancy reduction techniques are applied for image coding. With interframe coding, both the temporal redundancy reduction and the spatial redundancy reduction techniques are employed. Temporal redundancy reduction can be done by motion estimation and compensation. However, the computational complexity of traditional motion estimation is too high for real-time applications. Therefore, fast motion estimation algorithms are necessary for such applications. In Section 3.3, the matching criteria for motion estimation and the fast algorithms for motion estimation are reviewed.

The ITU-T H.261[1]/H.263 [2] and H.263+[3] are the successful worldwide video coding standards over ISDN. The working principle and the performance of these coding schemes will be discussed in Section 3.4. In order to improve both of the image quality and the compression efficiency, various algorithms for video coding based on these standards have been proposed. We will present these algorithms in Section 3.5. In addition, rate control is also important for video coding: it controls the bit rate generation as well as the time delay. The key to bit rate generation lies in controlling the quantization step size and the macroblock type. In Section 3.6, we will describe the rate control methods for video coding. Finally, a summary of the various video coding techniques is provided in Section 3.7.

3.2 Hybrid Video Coding Techniques

Redundancy reduction is an efficient technique for reducing the amount of data that needs to be encoded. For image and video compression, motion compensated prediction and transform coding are popular techniques for redundancy reduction. The temporal redundancy between pictures can be reduced by the motion compensated prediction techniques, and the spatial redundancy of an image can be reduced by the transform coding techniques. When combined, the two coding techniques can improve the compression performance, and is commonly referred to as hybrid motion compensated predictive transform coding.

3.2.1 Motion Compensated Prediction

Motion compensated prediction is a powerful prediction technique that reduces the temporal redundancies between frames. The concept of motion compensation is based on the estimation of motion between frames, from which a prediction frame that is similar or equal to the input frame is generated. If the scene remains unchanged, the motion between frames can be described by motion vectors due to the small differences between frames. The motion compensated frame can be generated according to the description of the motion vectors. However, it is unnecessary to encode the motion vectors of each pixel in an image. Usually, one motion vector is used to represent the motion of a marcoblock (16×16 pixels). This means that the input images are usually separated into disjoined macroblocks for the processes of motion estimation and compensation. A more detailed description of motion estimation and compensation is provided in section 2.3. The prediction errors between the motion compensated frame and the actual frame with the motion vectors are usually entropy coded and transmitted to the receiver.

3.2.2 Transform Coding

Transform coding is another very popular compression technique for reducing the spatial redundancy of an image. The JPEG standard is an example of image compression that uses the transform coding techniques. The purpose of transform coding is to de-correlate the image content and then encode the transform coefficients. It is more efficient to encode the coefficients than the original pixels of the image. Usually, the input image is split into disjoined blocks (NxN pixels), and transformation will then be applied in each block. As a result, the NxN coefficients will be obtained for each block.

Discrete cosine transform (DCT) is one of the successful methods for still image and video coding. DCT based implementations are used in most image and video coding standards due to their high de-correlation performance and the availability of fast DCT algorithms that are suitable for real-time applications.

3.2.3 Hybrid DCT/DPCM Coder

The hybrid DCT/DPCM coder [32][33] is an efficient and popular video coding method that uses the block matching techniques. The coder structure of video coding standards such as H.261/H.263, and MPEG is based on the hybrid DCT/DPCM coder. Since each of the motion compensated prediction technique and the transform coding method has its own advantages and disadvantages, the general approach is to combine the techniques of transform coding and predictive coding to achieve a better compression performance.

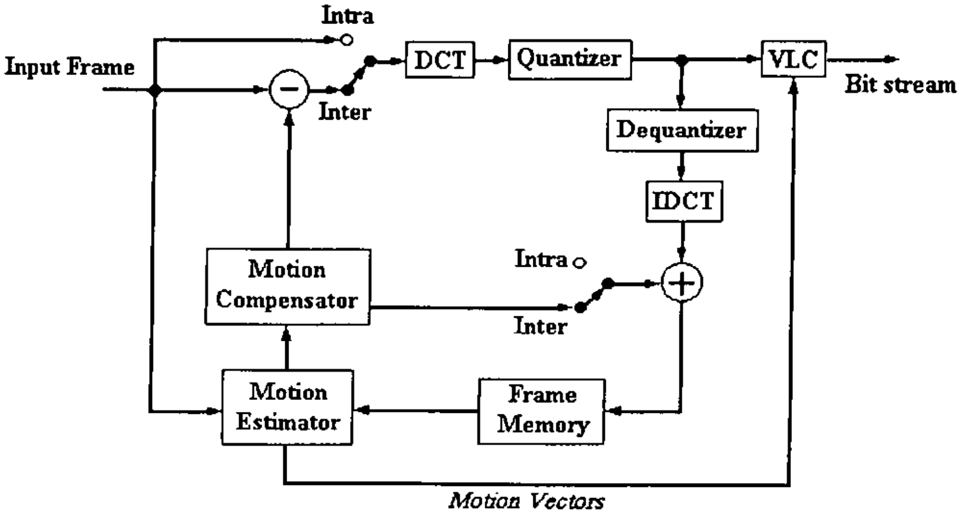


Fig. 3.1 Block diagram of an hybrid DCT/DPCM video coder

The block diagram of an hybrid DCT/DPCM video coder is shown in Fig. 3.1. This coder can operate on either intraframe or interframe mode. For intraframe (I-frame) coding, the transform coding methods is applied. Each input marcoblock is segmented into four 8×8 blocks, and then transformed into the frequency domain by using 2D-DCT. The insignificant coefficients are discarded by the "quantizator" unit. The quantized DCT coefficients will be entropy encoded by variable length coding (VLC). For interframe (P-frame) coding, both the predictive coding and transform coding methods are applied. The prediction errors between the previous reconstructed frame and the motion compensated frame will be transformed into the DCT domain. The quantized DCT coefficients and the motion vectors will be entropy encoded by the VLC. The reconstructed frame is stored in frame memory to be used as a reference frame for motion estimation and compensation. The reconstructed frame is reconstructed by adding the decoded predicted errors to the compensated frame. Decoding of the prediction errors can be achieved by inverse quantization and inverse DCT. The compensated frame is generated according to the estimated motion vectors from the "motion estimator" unit.

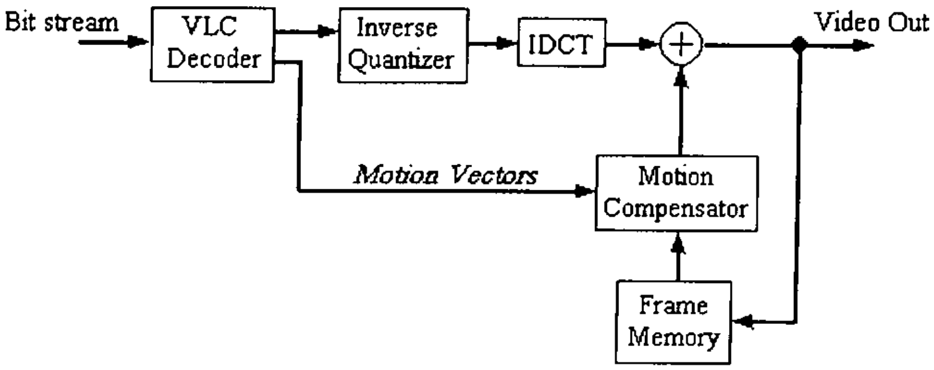


Fig. 3.2 Block diagram of an hybrid motion compensated DCT video decoder

The decoder of an hybrid coder is shown in Fig.3.2. Firstly, the motion vectors and the coefficients are decoded by the VLC decoder. The predicition errors are recontructed by inputting the decoded coefficients into the "inverse quantizer" and "IDCT" units for inverse quantization and inverse transform, respectively. According to the decoded motion vectors and the previously reconstructed frame, a motion compensated frame is generated by the "motion compensator" unit. As a result, the current frame is reconstructed by adding the prediction errors to the motion compensated frame. Furthermore, the decoded current frame is stored in the frame memory to form the new reference frame.

3.3 Motion Estimation and Compensation

Motion estimation and compensation is an important process for video coding. Motion compensated prediction provides a significant gain in compression. In video coding, block-based motion estimation and compensation techniques are likely to be used to reduce the temporal redundancy.

The traditional full search method for motion estimation is very time consuming, hence fast motion estimation algorithms are very important for real-time implementation. In this section, the criteria of block matching techniques and the various fast algorithms for motion estimation will be reviewed.

3.3.1 Basic Principle of Block Matching Methods

The block matching methods are the most popular motion estimation methods, which are adopted by various video coding standards. The methods assume that a block of pixels has the same translational motion from frame to frame. Fig. 3.3 illustrates the basic principle of block-based motion estimation and compensation. A current frame is divided into small rectangular blocks. For each block in frame N, a motion vector (MV) is obtained by finding the displaced coordinate of a match block within the search window of the reference frame. Supposing that the block size is N×N pixels and the maximum displacement of a motion vector is ±*d* in both the horizontal and vertical directions, a motion vector (*u*,*v*) is obtained by finding a match block within a search window of size (2*d*+1)×(2*d*+1) in the reference frame.

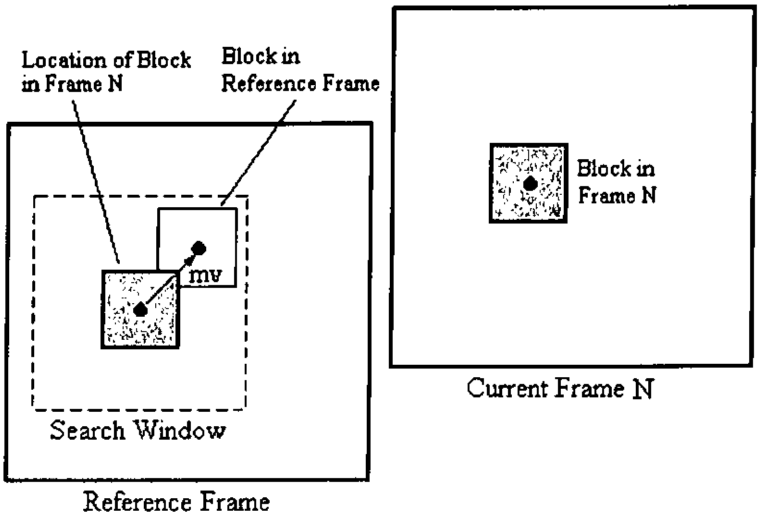


Fig. 3.3 Block-based motion estimation

The motion vector for the block matching techniques can be obtained by minimizing a cost function. The cost function is used to measure the differences or similarities between two given blocks. The smaller the values returned by the cost function, the more similar the two blocks are to each other. There are many choices [32] for the matching criterion, such as mean absolute difference (MAD), mean squared difference (MSD), and cross correlation function (CCF). Among the criteria, the MAD is the most popular one because it does not require any multiplication and performs in a manner similar to the MSD. The matching criterion is defined as:

$$MAD(dx,dy) = \frac{1}{mn}\sum_{i=-n/2}^{n/2}\sum_{j=-m/2}^{m/2}\left|F(i,j) - G(i+dx, j+dy)\right| \qquad (3.3-1)$$

where $F(i,j)$ represents a $(m \times n)$ macroblock from the current frame, $G(i,j)$ represents the same macroblock from a reference frame, and (dx,dy) is a vector representing the search location.

3.3.2 Various Approaches for Motion Estimation and Compensation

The most straightforward block matching algorithm is the full search method. The full search method finds a global optimal motion vector from all the motion vector candidates within the search window. If the maximum displacement of a motion vector is $\pm d$, it will require $(2d+1)^2$ block distortion measure calculations. However, it is not suitable for real-time implementations due to its high computational complexity. Many fast block matching algorithms for motion estimation have been developed and evaluated in the literature [34][35]. They include exhaustive search, three-step search, 2-D logarithmic search, and conjugate direction search. These conventional block matching algorithms, such as the three-step search, use a uniformly allocated search pattern in their first step. However, experimental results [36][37] show that the block motion field of a real world image sequence is usually gentle, smooth, and varies slowly. This indicates that the global optimum motion vector distribution is highly biased at the central area. As a result, these conventional block matching algorithms are not efficient for capturing small motions that appear in stationary blocks. Several algorithms based on the characteristic of center-biased motion vector distribution have been proposed [36]-[39]. A new three-step search algorithm is proposed in [36]. In this approach, the original three-step search method is employed, but eight extra checking points are added to the first step. The eight points are the neighbors of the search window center. If the minimum block distortion measure point in the first step is one of the eight neighbors of the window center, the search in the second step will be performed only for the eight neighboring points of the checked minimum point, after which the search will end. Otherwise, the search window size for the next step is reduced by half and the process is the same as that of the three-step search. In [37], a smaller search window is employed in the first step. The search size of the next step depends on the location of the minimum cost function point. If the minimum cost function point is close to the center, the search window or the search step size will be reduced. Otherwise, the search size remains unchanged. In conclusion, the common target of these methods is to reduce the computational complexity of motion estimation by using a smaller search window in the first step to capture the center-biased motion. As a result, the computational requirement of the center-biased methods is reduced when compared to the conventional search algorithms that are used for capturing small motions, but the computational requirement may be increased for activity motion blocks.

An hybrid adaptive search algorithm for solving the above problems has been reported [40]. Since some algorithms are better at searching large motions while others are better at searching small motions, the idea of the approach is to combine two kinds of search algorithms to capture more efficiently both small and large motions. Based on the threshold techniques, the input block is classified as stationary motion, small motions, or large motions block. Depending on the motion type of the block, different search algorithms are employed for motion estimation. As a result, the computational requirement for capturing the small motions and large motions is reduced when compared to that of the center-biased approaches.

In fact, the local minima sticking problem is always present when the conventional motion estimation methods are used. In order to solve this problem, genetic algorithms(GA) [41] are applied to perform block-based motion estimation [42][43]. The main advantage of this GA-based block-matching algorithm is its small initial population which leads to a reduction of the computational time. In order to increase the chance of capturing different kinds of motion, the initial chromosomes are distributed equally in the search space as shown in Fig. 3.4.

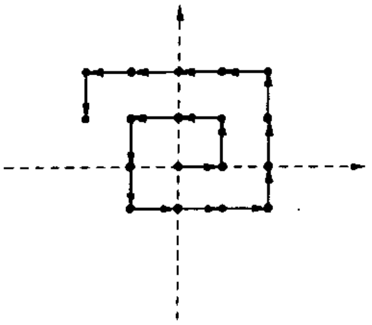


Fig. 3.4 Initial chromosomes distribute equally in the search space

The traditional GAs involve a crossover process, which is not employed in this approach because the efficiency of crossover for small population is not great. Furthermore, the probability of a mutation process is usually low, but the setting in this approach is very high. As a result, more search points can be reached. The experimental result shows that the image quality is much higher than that of the three-step search, and the prediction performance is close to that of the full search method.

Most of the motion estimation approaches are based on the assumption that all pixels within each block are moved by the same amount, but this may not be the true motion vector when the cost function reaches its minimum. For example, if $\mathrm{MAD}_{\min}= \mathrm{MAD}(x_0, y_0) \neq 0$, there is a chance that $\mathrm{MAD}(x_1, y_1)>\mathrm{MAD}(x_0, y_0)$, but (x_1, y_1) might still be the true motion vector. The larger the $\mathrm{MAD}_{\min}$, the higher the chance that this will happen. In other words, if less pixels are used to find the motion vector, the value of the cost function will get smaller, thereby reducing the chance of finding a wrong motion vector. An adaptive pixel selection method for motion estimation according to the image activity in the spatial domain has been proposed [44]. The number of selected pixels can be varied according to the image details. Fewer pixels will be used if the block has uniform intensity. In high activity blocks, more pixels will be employed for the matching criterion. The experimental result shows that the computation is reduced by a factor of four when compared with the exhaustive search. Moreover, the measured MSD of the adaptive pixel decimation is also smaller than that of the exhaustive search.

In all of the above discussions on motion estimation, we restricted the motion vector estimation to integer pixel grids. Thus, the motion vector would be pixel or pel-accurate. With fractional or sub-pixel precision motion estimation, the quality of the reconstructed images will be improved. The international video coding standard H.263 is one of the standards that permit motion vectors to be specified to a half-pixel accuracy.

Motion vector estimation with half-pixel accuracy can be easily found by interpolating the current and reference pictures by a factor of two using any of the motion estimation methods. However, the computational complexity and the storage requirements of this method are intensive. To reduce the computational complexity, the following steps are usually preferred.

Step 1. The motion vector with integer-pixel accuracy is found by using any of the motion estimation methods.

Step 2. The 8 surrounding half-pixels of the detected motion vector in step 1 are determined by bilinear interpolation as shown in Fig. 3.5(a). The one with the minimum block distortion measure is the half-pixel accuracy motion vector as shown in Fig. 3.5(b).

The performance of this half-pixel accuracy in motion estimation and compensation is better than that of integer-pixel accuracy, hence the experimental result also shows that both the compression ratio and the quality of the reconstructed images are improved.

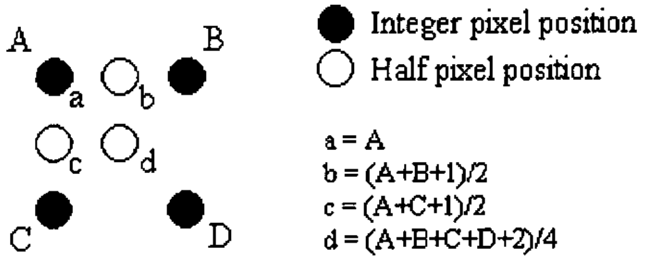


Fig. 3.5(a) Half-pixel precision using bilinear interpolation

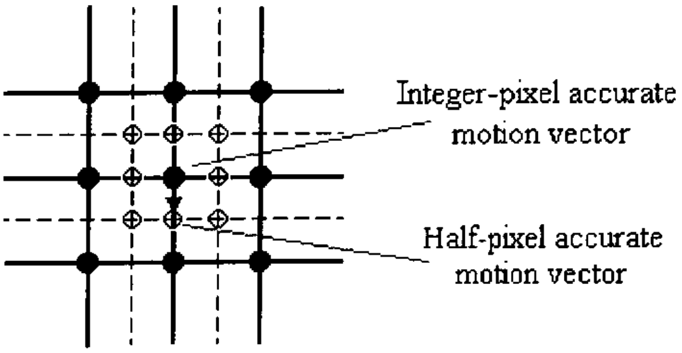

Integer-pixel accurate motion vector
Half-pixel accurate motion vector

Fig. 3.5(b) Half-pixel accurate motion vector estimation

3.4 Digital Video Coding Standards

Block-based motion estimation and compensation are the most popular approaches in digital video coding standards. ITU Recommendation H.261 is one of the international standards for video compression using block-based motion compensation techniques. However, the general model used in this standardized coding algorithm provides only a basic and incomplete description of the video scenes. In general, a very good picture quality is obtained at several megabits per second. Furthermore, H.261 generates annoying blocking artifacts below 64kbps, resulting in low a temporal resolution and end-to-end delay. Hence, additional coding standards have emerged, such as H.263 and H.263+, which aims at reaching an acceptable picture quality at a low bit rate.

3.4.1 ITU-T H.261and H.263 Video Coding Schemes

ITU-T H.263 is one of the successful video compression standards employed in video conferencing and video phone services over the ISDN at rates below 64 kbps. A block diagram of an H.263 baseline encoder is shown in Fig.3.6. The architecture of the baseline encoder is the same as that of the hybrid DCT/DPCM coder. Motion compensation prediction first reduces temporal redundancies. Discrete cosine transform (DCT)-based algorithms are then used for encoding the motion compensated prediction errors. The quantized DCT coefficients, motion vectors, and side information are entropy coded using variable length codes (VLC).

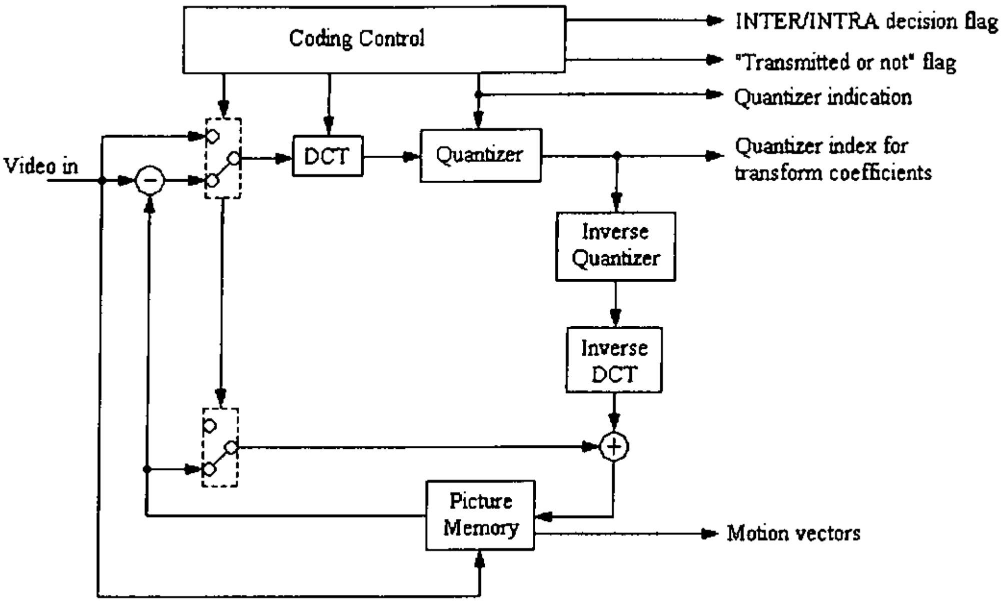


Fig.3.6 Block diagram of an H.263 baseline video encoder

The coding structure of H.263 is based on H.261, but it provides a better picture quality at low bit rates with little added complexity. The key differences between these two video coding schemes are the target bit rate, picture format, precision of motion compensation, loop filter, VLC table and motion vector coding. Table 3.1 summarizes the key differences between them.

Codec	**H.261**	**H.263**
Target bit rate	p×64 kbits/s	below 64 kbits/s
Picture format	CIF, QCIF	sub-QCIF,QCIF, CIF, 4CIF, 16 CIF
Precision of motion estimation and compensation	integer pixel accuracy	half-pixel accuracy
Loop filter	present	absent
Motion vector coding	the difference between the motion vectors of the previous macroblock and the current macroblock is encoded	the difference between the median of the motion vectors of three previously coded blocks and the motion vector of the current macroblock is encoded
VLC table	2-dimensional variable length codes	3-dimensional variable length codes

Table 3.1 Key differences between H.261 and H.263

H.261 employs a spatial-domain loop filter in the coding loop to reduce the blocky effects due to the use of block-based motion estimation. However, H.263 does not employ such a filter since the bilinear interpolation used in H.263 for half-pixel motion compensation introduces some low-pass filtering as a side-effect. Furthermore, the H.263 VLC table relies on the probabilistic nature of a run/length combination being at the end of a block which results in a saving of approximately 2 bits per block. Bits are also saved for coding the motion vectors in H.263. The difference between the median of the motion vectors of the three neighboring macroblocks and the motion vectors of the current macroblock is encoded. Fig. 3.7 indicates the motion vector prediction for H.263. The result is a smaller average vector difference that needs to be coded. Therefore, the overall performance of the compression ratio of H.263 is much better than that of H.261.

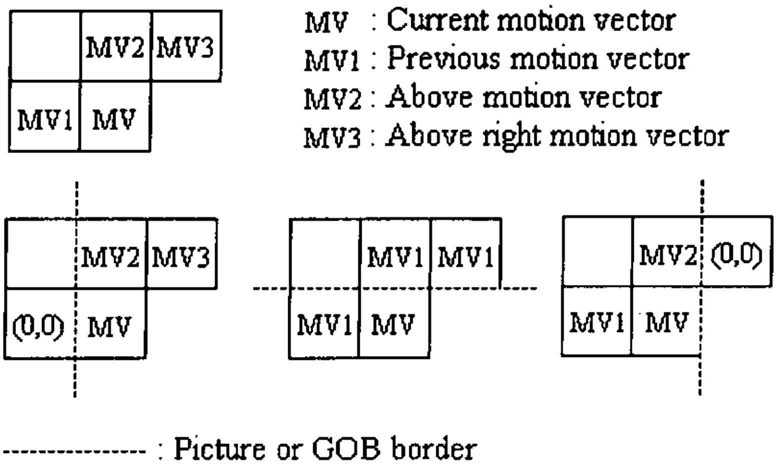

MV2
MV3
MV1
MV
MV : Current motion vector
MV1 : Previous motion vector
MV2 : Above motion vector
MV3 : Above right motion vector
MV2
MV3
(0,0)
MV
MV1
MV1
MV1
MV
MV2
(0,0)
MV1
MV
--------------- : Picture or GOB border

Fig. 3.7 Motion vector prediction

In order to improve its compression performance, H.263 has 4 optional modes: unrestricted motion vectors, advanced prediction, P-B frames, and syntax based arithmetic coding. The first two modes are used to improve inter picture prediction. The

P-B frames mode improves temporal resolution with a small amount of bit rate increase. When the syntax-based arithmetic coding mode is enabled, arithmetic coding replaces the default VLC coding. These optional modes allow the developers a trade-off between compression performance and complexity. The 4 optional modes are briefly described below:

1) Unrestricted motion vector mode (UMV): In baseline H.263, motion vectors can only reference pixels that are within the picture area. As a result, macroblocks at the border of a picture may not be well predicted. When the unrestricted motion vector mode is used, motion vectors can take on values in the range of [-31.5, 31.5] instead of [-16, 15.5], and are allowed to point outside the picture boundaries.

2) Syntax-based arithmetic coding mode (SAC): Baseline H.263 employs variable-length coding as a means of entropy coding. In this mode, syntax-based arithmetic coding is used instead of variable-length coding, and the bit rate can be reduced by approximately 5%.

3) Advanced prediction mode (AP): This mode allows for the use of four motion vectors per macroblock, one for each of the four 8×8 luminance blocks. Furthermore, overlapped block motion compensation is used for the luminance macroblocks, and motion vectors are allowed to point outside the picture as in the unrestricted motion vector mode. This mode improves inter picture prediction, and a significant improvement in subjective picture quality is achieved for the same bit rate by reducing the blocking artifacts.

4) PB-frames mode (PB): In this mode, the frame structure consists of a P picture and a B picture. The quantized DCT coefficients of the B and P pictures are interleaved at the macroblock layer such that a P picture macroblock is immediately followed by a B picture macroblock. Therefore, the maximum number of blocks transmitted at the macroblock layer is 12 rather than 6. The P picture is forward predicted from the previously decoded P picture, and the B picture is bidirectionally predicted from the previously decoded P picture and the P picture currently being decoded.

A more detailed description of the H.261 and H.263 video coding schemes can be found in [33][45]-[50]. Table 3.2 shows the compression ratio and the PSNR for the "Salesman" sequence based on the H.261 and H.263 schemes [51]. The experimental result shows that the P-B frames mode achieves the maximum compression ratio, and the advanced prediction mode achieves the maximum PSNR.

	Q Step = 8		Q Step = 16		Q Step = 24	
Codec/mode	Compression ratio	PSNR (db)	Compression ratio	PSNR (db)	Compression ratio	PSNR (db)
H.261	45.15	34.02	86.55	29.67	110.41	27.59
H.263 (base level)	87.37	32.99	217.17	29.17	343.13	27.37
H.263 (AP+UMV)	95.17	33.09	218.86	29.31	312.67	27.54
H.263 (PB)	121.41	32.81	273.51	29.11	374.36	27.42
H.263 (SAC)	91.00	32.99	223.81	29.17	349.09	27.37
H.263 (all 4 modes)	138.43	32.87	286.82	29.27	349.47	27.56

Table 3.2 Performance of H.263 and H.261 schemes [51]

3.4.2. ITU-T H.263+ Video Coding Scheme

H.263+ is an extension of H.263, providing 12 new negotiable modes and other additional features. The objective of H.263+ is to broaden the range of applications and to improve the compression efficiency. It allows the use of scalable bit streams, enhanced performance over packet switched networks, support custom picture size and clock frequency, and external usage capabilities. The 12 new optional coding modes including the modification of H.263's unrestricted motion vector mode are summarized below:

1) Unrestricted motion vector mode (UMV): When this mode is enabled, new reversible VLCs (RVLCs) are used for encoding the difference motion vectors. The idea behind RVLCs is that decoding can be performed by processing the received motion vector part of the bit stream in the forward and reverse directions. This improves the error resilience of the bit stream. Furthermore, the motion vector range is extended to up to ±256.

2) Advanced intra coding mode (AIC): This mode improves the compression performance of coding intra macroblocks. In this mode, intra-block prediction from the neighboring intra blocks, a modified inverse quantization of intra DCT coefficients, and a separate VLC table for intra coded coefficients are employed.

3) Deblocking filter mode (DF): This mode introduces a block edge filter within the coding loop. The main purpose of the block edge filter is to reduce the blocking artifacts. The filtering is performed on 8×8 block edges.

4) Slice structured mode (SS): A slice structure instead of a GOB structure is employed in this mode. This allows the subdivision of a picture into segments containing variable numbers of macroblocks.

5) Supplemental enhancement information mode (SEI): In this mode, supplemental information is included in the bit stream in order to offer display capabilities within the coding framework. This information includes support for picture freeze, picture snapshot, video segmentation, progressive refinement, and chroma keying.

6) Improved PB-frames mode (IPB): This mode is an enhanced version of the H.263 PB frames mode. The main difference is that the H.263 PB-frames mode allows only bidirectional prediction to predict B-pictures in a PB frame, but the IPB-frames mode permits forward, backward and bi-directional predictions.

7) Reference picture selection mode (RPS): In order to suppress temporal error propagation due to inter picture coding, it is possible to select the reference picture for prediction in this mode. Multiple pictures must be stored at the decoder, and the encoder should signal the necessary amount of additional picture memory by external means.

8) Temporal, SNR, and spatial scalability mode: This mode specifies a syntax to support temporal, SNR, and spatial scalability capabilities. Temporal scalability provides a mechanism for enhancing perceptual quality by increasing the picture display rate. SNR scalability is achieved by using a finer quantizer to encode the difference picture in an enhancement layer. Spatial scalability allows for the creation of multi-resolution bit streams to meet various display requirements for a wide range of applications.

9) Reference picture resampling mode (RPR): This mode describes an algorithm to warp the reference picture prior to its use for prediction. It can be useful for resampling a reference picture having a different source format from that of the picture being predicted.

10) Reduced resolution update mode (RRU): This mode allows the encoder to send updated information for a picture encoded at a lower resolution, while maintaining a higher resolution for the reference picture, to create a final image at the higher resolution.

11) Independently segmented decoding mode (ISD): In this mode, picture segment boundaries are treated as picture boundaries in the sense that no data dependencies across the segment boundaries are allowed.

12) Alternative inter VLC mode (AIV): In this mode, the intra VLC table designed for encoding the quantized intra DCT coefficients in the AIC mode can be used for inter block coding.

13) Modified quantization mode (MQ): The modified quantization mode allows modification of the quantizer to any value thus providing the rate control methods with more flexibility.

Sequence / Mode	FOREMAN		AKIYO	
	32kbps	128kbps	8kbps	32kbps
Baseline	30.44 dB	35.83 dB	33.90 dB	39.26 dB
UMV	+0.61	+0.64	-0.01	-0.04
SAC	+0.08	+0.06	-0.12	+0.08
AP	+0.28	+0.58	+0.19	+0.33
PB B-picture	-0.57	-1.64	+0.51	+0.61
P-picture	+0.11	+0.37	+0.60	+1.05
AIC	+0.04	0.00	+0.14	-0.03
DF	+0.18	+0.48	-0.24	-0.23
IPB B-picture	-0.17	-1.21	+0.54	+0.74
P-picture	+0.36	+0.62	+0.61	+1.12
AIV	+0.02	+0.06	-0.02	+0.01
MQ	+0.40	+0.20	+0.40	-0.05

Table 3.3 Summary of improvement in PSNR (dB) for H.263 and H.263+

A summary of the compression improvements resulting from the use of individual modes is given in Table 3.3 [52][53]. Results are presented for low and high bit rates using two QCIF video sequences at 10 fps: an active video sequence, Foreman, and a typical head-and-shoulder videophone sequence, Akiyo. It can be observed that a given mode is not always suitable for all bit rates or all sequences. For example, the alternate inter VLC mode achieves compression gains only at high bit rates. Moreover, the deblocking filter mode may yield a decrease in PSNR, but the resulting picture subjective quality is usually better. Another observation is that the modified quantization mode does not lead to compression gains at high bit rates for low motion sequences. The unrestricted motion vector mode shows PSNR improvements for sequences with motion across picture boundaries, or at CIF and larger resolutions.

3.5 Advanced Video Coding Algorithms

Although the compression performance of H.263 or H.263+ seems very good, these block-based motion estimation methods do not take into account the arbitrary shape or structure of objects in a picture. Its prediction efficiency will not be as large as expected when these objects have different motions. Furthermore, the texture coding performance for arbitrary shaped objects using conventional DCT methods is not efficient. Since the image quality and the compression performance both depend on the texture coding techniques employed, we will review the texture coding algorithms and some efficient video coding algorithms in this section.

3.5.1 Texture Coding Algorithms

Padding techniques is an efficient method for arbitrary shaped image coding. Moon *et al.* [54] proposed a texture coding method which enhances the coding efficiency of conventional DCT with padding techniques for arbitrary shaped objects in video coding. Firstly, the macroblock within a separated object is divided into four sub-blocks. The two sub-blocks, sub-block-1 and sub-block-2, will be merged if there are no overlapping pixels between the shape of sub-block-1 and the 180° rotated sub-block-2. Merging processes can be done in the horizontal, vertical and diagonal positions. Fig. 3.8 shows the merging types of luminance sub-block pairs in a macroblock. Finally, the background pixels of the merged sub-blocks are filled with the average of the collocated padding values of the two sub-blocks. This method may reduce the number of blocks to be encoded. However, the first step of this approach is extraction of the boundary of an object in an image and then sending the shape information of the extracted object to the receiver; hence a good compression ratio may not be obtained if the number of shape information to be sent is large.

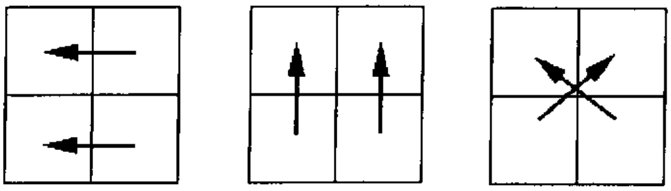


Fig. 3.8 Merging types of luminance sub-block pairs in a macroblock

Filling zero values or mean values to the region that is outside the object boundary may increase the values of the high order transform coefficients. In order to reduce the higher frequency components, an extension-interpolation method for arbitrarily shaped texture coding has also been proposed [55]. The idea is that with the block length N, interpolate the N pixels with the object segment with length M and replace the whole block with the interpolated N pixels. The experiment results show that the high order transform coefficients are reduced because the frequency in the spatial domain of the interpolated block is reduced.

3.5.2 Video Coding Using Pre-defined Patterns

Fukuhara *et al.* [56] proposed a very low bit-rate video coding method based on H.263. They used four kinds of rectangular-shaped partitions to roughly represent the object's shape. According to the four pre-defined patterns, each macroblock is divided into two partitions, and motion estimation is performed on each partition separately. Thus, a macroblock may have one or two motion vectors to be coded. In addition, two forward reference frames are used for motion estimation and compensation: a short-term frame memory (STFM) and a long-term frame memory (LTFM). These processes reduce the prediction errors, hence both the compression ratio and the image quality are improved. However, the disadvantage of this method is its high computational complexity. We therefore propose a simple but efficient method based on H.263 to improve both the compression ratio and the image quality (see Chapter 5).

3.5.3 Foreground/ Background Video Coding Algorithms

In a video phone or video conferencing system, the main object of the image is the human face. Users pay more attention to the face region and the facial expression. A number of researches on efficient coding of the face region have been reported [25][57][58]. In [25], the face is segmented by using the skin color segmentation approach (see Chapter 2). In this approach, the quantization step size for the face region is much smaller than that of the non-face region. Hence, the subjective quality of the face region is improved. In [57], a method for reducing the prediction errors with respect to the face region is proposed. Affine transformation is applied to estimate the motion of the face region, eye region, and mouth region. As a result, the prediction errors of the face region are reduced, and the PSNR of the face region is increased. Furthermore, encoding speed is also important for video phone and video conferencing applications. Ding *et al.* [58] proposed two methods to reduce encoding time. The first method assumes that the motion vectors of regions such as the forehead and hair areas are basically the same as the motion vectors of the mouth and eye regions, and the motion vectors of the non-face region are assumed to be stationary. Thus, the motion vectors of the forehead and hair regions can be predicted according to the motion vectors of the eye and mouth regions, and the motion vectors of the background region are assumed to be zero. Therefore, the time spent on motion estimation is greatly reduced. The second method reduces the computation of the DCT by reducing the transformation size. Since the head movement does not change quickly from frame to frame, a lot of zero coefficients in the higher frequency components will occur. Therefore, using 6×6 points DCT instead of the conventional 8×8 points DCT is significant for the facial region. The remaining coefficients of the block are filled by zero value. For less detailed regions such as the background, shoulders and hair, using either 2×2 or 4×4 points DCT is satisfactory. As a result, the reported frame rate is increased and there is less degradation of the image quality. However, the time spent on facial feature extraction and tracking has not been reported. The encoding speed may be affected by these detection processes.

3.5.4 2D/ 3D Hybrid Video Coding Algorithms

In standard block-based video coders, motion compensation is performed on the assumption that the scene contents can be modelled using 2D rectangular blocks.

Blocking artifacts always occur at the block boundaries due to the block-based motion estimation and the omission of high order DCT components. Model-based coding in which the scene is modelled using irregular shapes and sizes is introduced to solve this problem. However, the performance is poor for large rotational motions due to a lack of 3D structure and failures of object tracking. Hence, the switching coder [59]-[62] is proposed to solve the problems of model-based and block-based video coders. Both the model-based and H.261/H.263 coders are employed for encoding the video in the switching coder. The working principle is that switching between the two coders generates a compensated image for encoding. As a result, these switching coders might outperform the H.261/H.263 coder that works by itself.

3.6 Rate Control Scheme

In a video coding system, each of the video frames generates different amounts of data according to its activity. The rate control schemes must therefore to handle the bit rate generated by the source coder in order to preserve a constant image quality. In general, rate control schemes can be classified into forward and backward rate control methods. In video coding systems with a forward rate control, the quantization parameter(QP) and all of the rate control variables are determined by examining the input image's activity such as its variance, number of object, etc. On the other hand, in backward rate control scheme, the bandwidth of the transmission line, buffer size, and amount of data generated for the previous frames are critical factors for deciding of the coding parameters. The backward rate control method is the most popular rate control scheme due to its implementation simplicity. The block diagram of a backward rate control scheme is shown in Fig. 3.9. In this section, we will devote our attention to such rate control schemes.

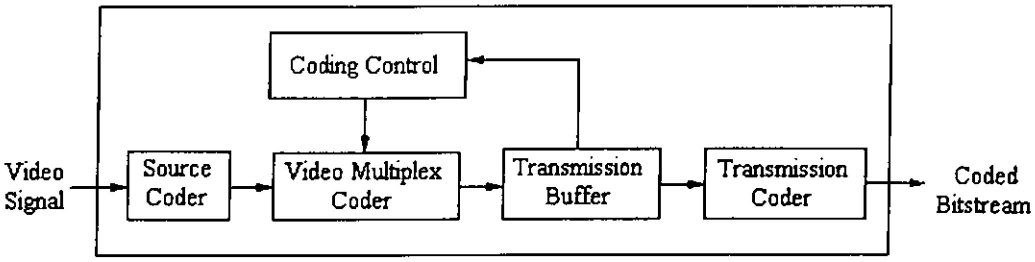


Fig. 3.9 Block diagram of backward rate control scheme

3.6.1 Macroblock Type Decision

In block-based motion compensation coding, blocks can be coded in INTRA, INTER or SKIPPED modes depending on the contents of the block. Since the output image quality and the bit rate of these modes are different, an efficient mode selection method can improve both the image quality and the output bit rate.

Generally, if the buffer is empty, the INTRA/INTER mode will be selected to fill up the buffer. If the buffer is full, the SKIPPED mode will be selected. In [63], the INTRA or INTER mode is selected according to the mode decision curve. The curve is obtained according to the target bit rate, the relationship between the SAD of the motion compensated frame and the SAD without motion compensation. For the block without sufficient change, the SKIPPED mode will be selected. This means that there will be no need for motion compensation and encoding.

Video Codec Test Model, Near-Term, Version 8 (TMN8) [64] is the rate control scheme for H.263. The criterion for the INTRA/INTER mode decision is defined as:

$$A = \sum_{i=1,j=1}^{16,16} \left| original - MB\_mean \right| \qquad (3.6-1)$$

where *MB_mean* is the average pixel value of the macroblock. The INTRA mode will be chosen if A<(MAD(x,y)-500). Otherwise, the INTER mode will be chosen. Eqn. (3.6-1) implies that if the difference between the original and the previous decoded macroblock is large, INTRA mode coding will be more efficient.

3.6.2 Quantization Step Size

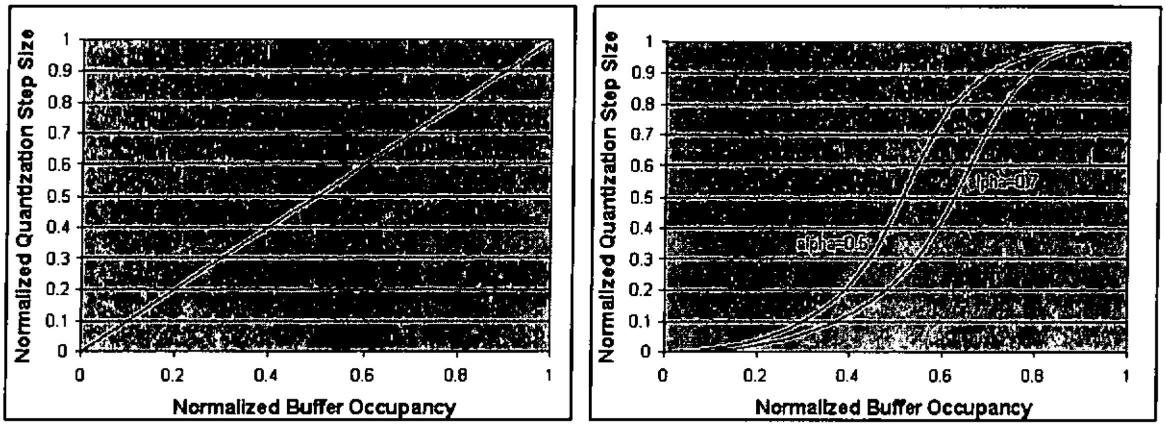
One of the popular methods of controlling the output bit rate at a constant image quality is by controlling the quantization step size. Oehler *et al.* [65] proposed a method that chooses the quantization step according to the target MSD. However, this method dose not consider the effect of buffer fullness. Since the encoded bit stream is stored in the buffer before transmission, the time delay will be long if the buffer is empty. If the buffer is full, bits will be lost and a degraded image quality will result. A linear rate control method is proposed to solve the above problems. The quantization step of the linear rate control strategy is linearly related to the buffer occupancy as shown in Fig. 3.10(a). The relationship is defined as:

$$\Delta(n) = b(n-1) \qquad (3.6-2)$$

where $\Delta(n)$ is the normalized quantization step size of the nth macroblock, and $b(n-1)$ is the normalized rate buffer occupancy at the (n-1)th macroblock. However, linear rate control has a high risk of underflow or overflow. To overcome this, a non-linear rate control scheme using a non-linear relationship between the buffer occupancy and quantization parameter QP is proposed [66]. The relationship is defined as:

$$\Delta(n) = \begin{cases} \alpha(\alpha^{-1}b(n-1))^k, & if\ 0 \le b(n-1) \le \alpha \\ 1-(1-\alpha)((1-\alpha)^{-1}(1-b(n-1)))^k, & if\ \alpha \le b(n-1) \le 1 \end{cases} \qquad (3.6-3)$$

where k is the steepness factor, and α is a real number between 0 and 1. Fig. 3.10(b) shows the relationship between the buffer occupancy and QP. When the buffer occupancy is close to 1, the predicted QP will be increased rapidly. This means fewer bits will be generated when the buffer is close to full. If the buffer occupancy is close to 0, the predicted QP will be small. This will cause a high output bit rate, which in turn will fill up the buffer faster. As a result, the time delay which is a function of buffer occupancy will be reduced.



(a) (b)

Fig. 3.10 The relationships between the buffer occupancy and QP using (a) linear rate control scheme, and (b) non-linear rate control scheme

To determine the QP value, both the linear and non-linear methods are dependent on buffer occupancy without considering the input video characteristics. In [67][68], quantization algorithms for adjustment of step size according to the activity of the block are proposed. In [67], the quantization step size of the macroblock is determined by using TM5 [69], and then refined according to a factor "ratio". The "ratio" factor is computed by comparing the average SAD of the macroblock to the average SAD of the image. This means that if the computed SAD of the current macroblock is below the average, the ratio will be smaller than 1. Oh *et al.* [68] proposed an adaptive rate control method which controls the QP according to the bit-rate generated from the previous frame. The most popular QP is obtained from the history of the relationship between the QP and the amount of bit generated at that quantization value. The QP for the n^{th} macroblock QP(n) is determined by:

$$QP(n) = \min\{QP \mid C(n-1) + f(QP) - mb < BS \times (1-\mu)\} \qquad (3.6-4)$$

$$C(n) = C(n-1) + B(n) - mb \qquad (3.6-5)$$

where *QP(n)* is the quantization parameter for the n^{th} macroblock, *C(n-1)* is the amount of buffer content up to the $(n\text{-}1)^{th}$ macroblock, *mb* is the mean of the outgoing bit rate for a macroblock, *BS* is the physical buffer size, μ represents the buffer utilization factors, *f(QP)* is the amount of bit to be generated, and *B(n)* is the number of bits generated by the source coder for the n^{th} macroblock. From eqn. (3.6-4), the minimum value of the QP can be selected without buffer overflow. The amount of bits in the QP-bitrate table is updated using the most recently used QP and the related amount of bit generated during the video coding. This process of updating the QP-Bitrate table reduces the prediction errors of the bit generation at a specified QP. Moreover, two QP-Bitrate tables are used: the first one for intra mode prediction, and the second one for inter mode prediction. As a result, the amount of bits to be generated more accurately, and performs much better than the linear and non-linear methods.

3.7 Summary

Different hybrid DCT/DPCM video coding techniques have been described in this chapter. These approach combines the motion compensated prediction and transform coding techniques. The encoder architecture of most video coding standards such as ITU-T H.263 and MPEG is based on the hybrid DCT/DPCM coder. We have also described the coding structure of H.261, H.263 and H.263+. The key differences between the coding schemes of H.261 and H.263 lie in the target bit rate, picture format, precision of motion compensation, loop filtering, VLC table and motion vectors encoding. Furthermore, H.263 has four additional optional modes for improving the image quality and the compression performance. These four optional modes are the unrestricted motion vector, syntax-based arithmetic coding, advanced prediction, and PB-frames. The H.263+ is an extension of H.263, and provides 12 new negotiable modes and additional features. Experiment results show that the overall compression performance of H.263+ is better than that of both H.261 and H.263. However, the computational complexity requirement for the optional modes in H.263+ is higher than that of H.263 and H.261. Hence the trade-off is between computational complexity and compression performance.

The goal of motion estimation and compensation is to find the motion vectors that best describe the motion between frames with the least complexity. We have discussed the center-based search algorithms, the pixel decimation search algorithm, the GA-based block matching algorithm, and the half-pixel precision techniques for motion estimation. The basic principle of the center-based search algorithms is to use a smaller search window in the first step to capture small motions of a block. For the pixel decimation search algorithm, the essential idea is to select some pixels within the macroblock instead of using the whole macroblock information for motion estimation. The GA-based block-matching algorithm is designed to solve the local minima sticking problem using conventional fast search algorithms. We have been described the half-pixel precision technique for motion estimation and compensation.

The efficiency of texture coding is also very important for video compression. Some efficient arbitrarily shaped texture coding techniques such as padding and interpolation methods have been viewed. The main idea of the padding method is to reduce the high order DCT coefficients by filling some values to the background region. The interpolation method interpolates the object segment to the size of the block to reduce the frequency in the spatial domain. Moreover, some of the advanced video coding algorithms such as the block partitioning coding method, facial region video coding method, and switching model-based coder have been introduced. The block partitioning coding method is designed for improving the whole image quality, while the facial region coding methods are designed for improving the image quality of face region only. For the switching coder, both the block-based coder and the model-based coder are employed for motion compensation. Hence, the image quality can be improved with additional complexity.

Finally, we have reviewed some rate control schemes. In general, we have described the mode decision and the quantization step size for rate control. A block or a frame can be coded in intra, inter or skipped modes. It is very important for controlling the image quality and the bit rate generated. An efficient mode selection method using TMN8 has been introduced. In addition, some algorithms for efficient control of the quantization step size have also been described. The refinement of the quantization step size is generally dependent on the buffer fullness, the previous decoded image quality, and the activity of the sequence. The experimental results shown that video coding systems with rate control have a reduced output bit rate; hence also an improved image quality.

CHAPTER 4
AN EFFICIENT ALGORITHM FOR HUMAN FACE DETECTION AND FACAIL FEATURE EXTRACTION UNDER DIFFERENT CONDITIONS

4.1 Introduction

In Chapter 2, we have addressed the problem of face segmentation and its applications. Various approaches to human face segmentation and facial feature extraction, such as shape analysis approaches, knowledge-based approaches, and color analysis, have been considered with a view to solving the problem. In this chapter, a reliable algorithm for face detection and facial feature extraction using the genetic algorithm and the eigenface techniques will be introduced.

In our approach, the possible eye candidates in a gray-level image with a complex background are identified by means of valley features on the human eyes. A genetic algorithm is applied in order to pair the possible eye candidates to form possible face region. The fitness value of the possible face region is determined by means of eigenface techniques. The facial features are then extracted from the detected face regions. In order to improve the level of detection reliability, the lighting effect is also considered and alleviated for the possible face regions. This method is tested with the MIT face database and some other complex images. Experiment results show that faces can be detected more reliably and efficiently now compared with our previous work [9]. The details of our approach for face and facial feature detection will be described in the following sections.

4.2 Human Face Detection using the Genetic Algorithm and Eigenface Technique

Our method for detecting and extracting the facial features in a gray-level image is divided into two stages. Firstly, the possible human eye regions are detected by testing all the valley regions in an image. A pair of eye candidates are selected by means of the genetic algorithm [41] to form a possible face candidate. The fitness value of each candidate is measured based on its projection on the eigenfaces [22]. In order to improve the level of detection reliability, each possible face region is normalized for illumination and the shirring effect, when the head is tilted, is also considered. After a number of iterations, all the face candidates with a high fitness value are selected for further verification. At this stage, the face symmetry is measured and the existence of the different facial features is verified for each face candidate. The facial features are determined by evaluating the topographic relief of the normalized face regions. The facial features extracted include the eyebrow, the iris, the nostril, and the mouth corner.

Genetic algorithm is an optimization technique that operates on a population of individual solutions. It has been successfully applied for many purposes, such as object recognition [70], human face detection [5][6], facial feature extraction [71], and motion estimation for video coding [43]. In our approach, genetic algorithm is also applied to search for possible facial regions in an image. The first step in locating the face regions in our approach is to select a pair of eye candidates using genetic algorithms. The fitness value for each face candidate is calculated by projecting it onto the eigenfaces space. Since eigenfaces is a successful approach for face recognition, we therefore adopt it as a fitness function.

4.2.1 Possible Eye Candidates Detection

In our approach, the possible eye regions are located by detecting the valley points in an image. Since the human iris in a gray-level image is of low intensity, a valley exists at an eye region. The valley field, Φv, can be extracted using morphological operators [72]. The equation for valley field extraction is

$$\Phi_v = f(x,y) \bullet B - f(x,y) \qquad (4.2-1)$$

where *f*(*x*,*y*) is the image and *B* is the structuring element. The valley image is obtained by performing a closing operation, which is then subtracted by the original image. A pixel at (*x*,*y*) is considered as a possible eye candidate if the following criteria are satisfied.

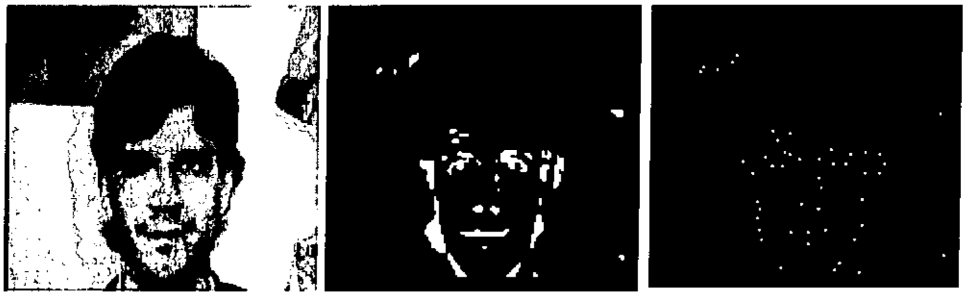
$$f(x,y) < t_l \quad and \quad \Phi_v(x,y) > t_v \qquad (4.2-2)$$

where t_l and t_v are thresholds. Figure 4.1(a) and 4.1(b) show the original image and its corresponding possible eye regions, respectively. The segmented possible eye regions are then reduced to a point or a number of points by choosing the good candidates in each region. The good eye candidates are those having large values in the functions, *F*1(*x*,*y*) and *F*2(*x*,*y*). The two functions are defined as follows:

$$F1(x,y) = W_{1,1}\left(\frac{f(x-2,y)+f(x+2,y)}{2} - S_{1,1}(x,y) \right) + W_{1,2}\Phi_{1,1}(x,y)$$

$$F2(x,y) = W_{2,1}\left(\frac{f(x-3,y)+f(x+3,y)}{2} - S_{2,1}(x,y) \right) + W_{2,2}\Phi_{2,1}(x,y) \qquad (4.2-3)$$

where W's are the weighting factors, $S_{1,1}(x,y)$ and $S_{2,1}(x,y)$ are the average gray-level intensities of the region under 3x3 and 5x5 windows, respectively. $\Phi_{1,1}(x,y)$ and $\Phi_{2,1}(x,y)$ are the average value of the valley field under the 3x3 and 5x5 windows, respectively. This arrangement allows us to detect the eyes according to different scales. Fig. 4.1(c) illustrates those good eye candidates for the segmented regions in Fig. 4.1(b). The locations of the possible eye candidates are stored in a buffer. In the genetic algorithm, two entries are selected from the buffer to form a possible face candidate. Therefore, the search space is limited to the possible eye candidates, which can then greatly reduce the required runtime.



(a) (b) (c)

Fig 4.1 Eye candidates detection: (a) original image, (b) possible eye regions, and (c) the good candidates for the detected eye regions.

4.2.2 Structure of a Chromosome

Each generated solution for a problem using the genetic algorithm is called a chromosome or string, which is represented in binary format. In our approach, two components are used to specify a face region in a chromosome. The two components which represent the position of the left eye (L_{eye}) and the right eye (R_{eye}) are the index numbers to the buffer. The structure of the chromosome is illustrated in Fig. 4.2. The number of bits required to represent the L_{eye} or R_{eye} is $B=\lceil \log_2 N \rceil$, where N is the total number of detected eye candidates. Thus, the total number of bits in each chromosome is $2B$.

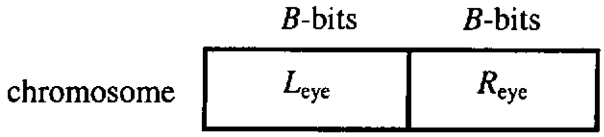


Fig. 4.2 Structure of a chromosome

Since the size of a human face is proportional to the distance between the two eyes (d_{eye}), a possible face region which contains the eyebrows, eyes, nose, and mouth can be formed based on this relationship. In our method, a square block is used to represent the detected face region. Fig. 4.3 shows an example of a selected face region based on the location of an eye pair. The line passing through the centers of the eye pair is called the base line. The extracted possible face regions are subsampled and interpolated to a resolution of 28×31. A low resolution has been proved to be sufficient for face identification. Moreover, the required computation is also reduced due to the fact that fewer pixels need to be manipulated. The relationships between the eye pair, the face size, and the orientation angle θ between the base line and the x-axis are defined as follows:

$$h_{face} = 1.8 d_{eye} \tag{4.2-4a}$$

$$h_{eye} = \frac{1}{5} h_{face} \tag{4.2-4b}$$

$$w_{eye} = 0.225 h_{face} \tag{4.2-4c}$$

$$
\begin{aligned}
&a = y_2 - y_1, \\
&b = x_2 - x_1, \\
&c = x_2 y_1 - x_1 y_2, \\
&\theta = \tan^{-1}\left(-\frac{a}{b}\right), \quad -\frac{\pi}{2} \le \theta \le \frac{\pi}{2}
\end{aligned}
\qquad (4.2-4d)
$$

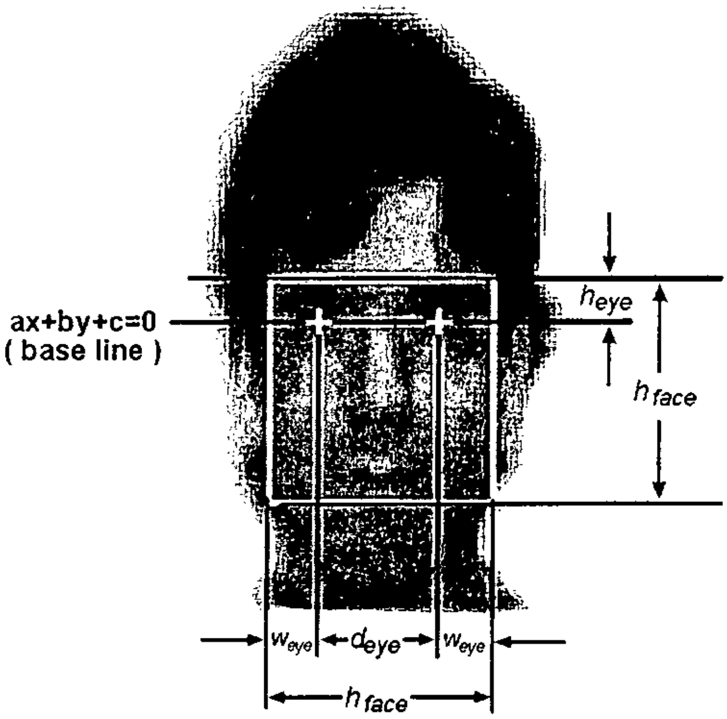


Fig. 4.3 The defined geometry of our head model

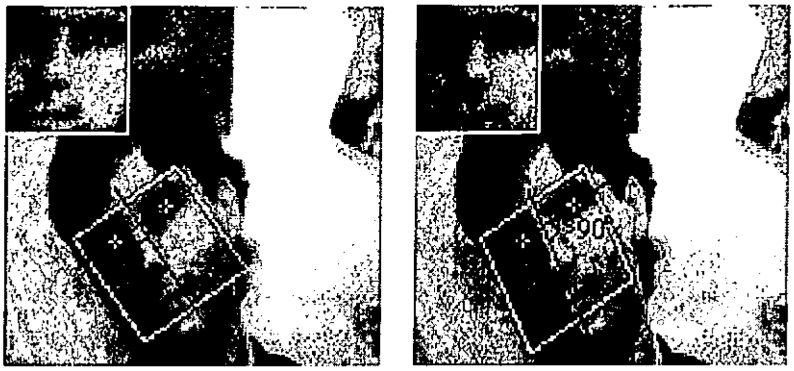
Based on the locations of the eye pairs, a population of possible face regions of different locations, sizes, and orientations can be generated. An initial population of the chromosomes is generated by pairing the possible eye candidates, depicted as white dots in Fig. 4.1(c). If the total number of detected eye candidates is N, the total number of pairing combinations for the initial chromosomes is $N(N-1)/2$. Therefore, members of the initial population are produced by selecting randomly from the $N(N-1)/2$ chromosomes.

4.2.3 Normalization of the Possible Face Regions

The orientation angle of a face candidate can be determined based on the gradient of the eye pair. However, the human face is not a rigid object; it will suffer from a shirring effect if the head is rotated too much, as illustrated in Fig. 4.4. In Fig. 4.4(a), if the face region is considered to be rectangular, the extracted face will be distorted. However, if the face region is a parallelogram, as shown in Fig. 4.4(b), the shirring effect is alleviated and a more upright face can be extracted. In our approach, the shirring effect will be compensated when the rotation angle θ >10°. In this case, the shirring is estimated to be $\theta/3$, which is based on the measurement of over 50 rotated human faces. If the rotation angle is less than 10°, the shirring effect may be neglected. If the rotation angle is larger than 10°, two possible face candidates will be generated for a chromosome in calculating the fitness values. One candidate uses a rectangular face region, while the other one is adjusted based on the shirring angle of the face. The shirring angle, ϕ, is defined as shown in Fig. 4.5. This normalization process for the shirring effect is performed using the following transformation.

$$\begin{bmatrix} xr \\ yr \end{bmatrix} = \begin{bmatrix} \tan\phi \cdot \sin\theta + \cos\theta & -\tan\phi \cdot \cos\theta + \sin\theta \\ -\sec\phi \cdot \sin\theta & \sec\phi \cdot \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad (4.2-5)$$

where xr and yr denote the coordinates of x and y after compensating for the shirring effect. The derivation of this equation is shown in Appendix I.



(a) (b)

Fig. 4.4 (a) A normal face candidate, and (b) adjusted possible face candidate.

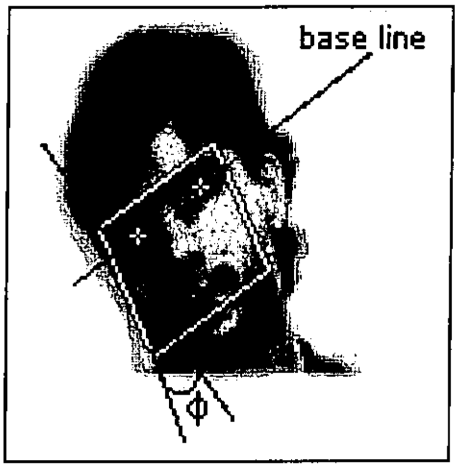


Fig. 4.5 Shirring angle approximation

The detection performance is also affected by the external environment, such as the direction of the lighting source. The uneven lighting conditions make a face become asymmetrical; a true face may not be detected. In order to reduce the lighting effect, the possible face candidates would be normalized by transforming their histograms to the histogram of a reference face image [73] before calculating the fitness value. This can be achieved due to the fact that all human faces have basically the same shape and general illumination properties. The advantage of the histogram normalization is that the size of the reference face image and the input region can be unequal. Thus, it is unnecessary to resample the face candidate to the size of the reference face. Fig. 4.6 shows an example of the histogram normalization of a face region. After the shirring effect and the histogram normalization processes, the fitness value of the face region will then be computed.

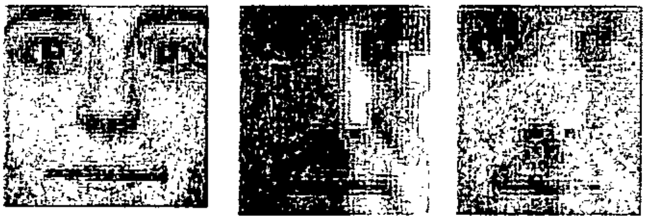


Figure 4.6 (a) Reference face image, (b) original face image with half in shadow, and (c) the histogram normalized face image.

4.2.4 The Fitness Function

To determine whether the normalized face candidate is a face or not, the fitness values of the possible face regions are computed by means of the eigenfaces. The eigenfaces are obtained by extracting the principle components from a training-set of pre-processed face images. The training images are also pre-processed by a histogram normalized to reduce the lighting effect. The normalized possible face region is then projected onto the eigenface space in order to calculate the fitness. The fitness function is a measure of the distance between its projection and that of the training-set face images. The distance between the mean adjusted faces Φ of the training images and the projection of the mean adjusted input region $\Phi_f(n)$ for the n^{th} chromosome on the face space is calculated by:

$$\varepsilon(n) = \left\| \Phi - \Phi_f(n) \right\| \qquad (4.2-6)$$

The value of $\varepsilon(n)$ is a measure of the distance between the input candidate and the training images. Thus, the fitness function of the possible face region for the nth chromosome is defined as:

$$f(n) = \frac{1}{\varepsilon(n)} \qquad (4.2-7)$$

From eqn.(4.2-7), it follows that a chromosome with a smaller distance will have a larger fitness value. A new population is then generated by means of the genetic operations: selection, crossover, and mutation. A chromosome with a higher fitness value will have a better chance of being chosen for the next generation. In the crossover process, two chromosomes are selected from the mating pool. In our method, the two-point crossover method is employed. Two cutting points are selected randomly within the chromosome for exchanging the contents. The crossover process is illustrated in Fig. 4.7. Since the probabilities of a chromosome being selected for the crossover and mutation processes are proportional to its computed fitness value, so those good offspring will probably be passed to the next generation. In order to increase the successful rate, the best candidate in one generation can pass directly to the next generation. After a number of iterations, those good candidates will be further verified as to whether they are human faces. The

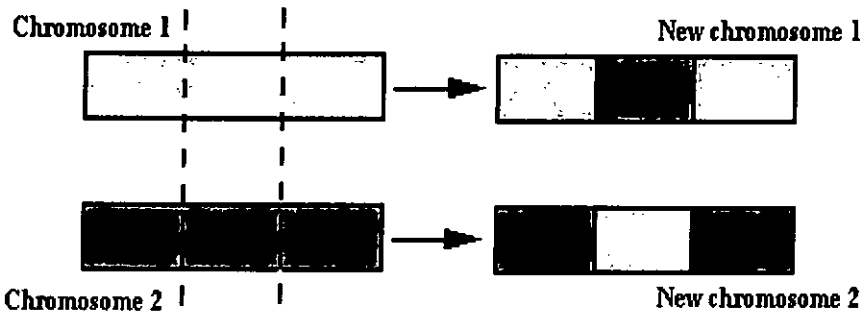


Fig. 4.7 The crossover process

parameter settings used in our approach are shown in Table 4.1. The extracted good face candidates are then input to the next stage for further verification and facial feature extraction.

Population Size	100
Selection Probability	0.9
Crossover Probability	0.8
Mutation Probability	0.08
Chromosome length	2*B* bits

Table 4.1 Parameter settings

4.2.5 Verification of Face Regions and Facial Feature Extraction

Those possible face candidates with a high fitness value are passed on to the second stage. The functions of the second stage are to verify whether the candidates are human faces or not, and to extract the respective facial features in the face region. The verification process is based on the characteristics of the projected face images.

At this stage, the symmetry of a face candidate is measured. As every face region is normalized for the shirring effect and the illumination effect, the difference between the left half and the right half of a face region should be small due to its symmetry. In our method, the size of a face region is normalized to 28×31, and the symmetrical measure is calculated as follows:

$$T_S = \frac{1}{434}\sum_{y=0}^{30}\sum_{x=0}^{13}\left|f(x,y) - f(27-x,y)\right| \qquad (4.2-8)$$

where *f*(*x*,*y*) represents a possible face candidate. If the value of T_S is smaller than a threshold, the face candidate will be selected for further verification. With any overlapping regions, the one with the lowest value of T_S is chosen.

After measuring the symmetry of a face candidate, the existence of the different facial features is also verified. The position of the facial features is determined by analyzing the projection of the normalized face candidate region. The facial feature regions will exhibit a low value on the projection. A normalized face region is divided into three parts; each of which contains the respective facial features. In our method, the y-projection is performed in each part to determine the vertical position of the facial features. The y-projection is the average of the gray-level intensities along each row of pixels in a window. In order to reduce the effect of the background in a face region, only the white windows as shown in Fig. 4.8 are considered in computing the projections. The two top windows contain the eyebrows and the eyes; the middle window contains the nose; and the bottom window contains the mouth. In each of the windows, the position where the projection value is a minimum is identified. For each of the two top windows, two significant minima will be detected due to the eyebrow and eye, respectively. These minima indicate the vertical position of the eyebrow and the eye. Similarly, the minima in the middle and the bottom windows represent the vertical position of the nostril and the mouth, respectively. The results of the y-projection for the windows in Fig. 4.8 are shown in Fig. 4.9. A valid minimum is identified by measuring the difference between the minimum and its neighboring maximum. If the vertical position of any of the facial features cannot be found, the face candidate is then declared as a non-facial image, and is rejected from the x-projection process.

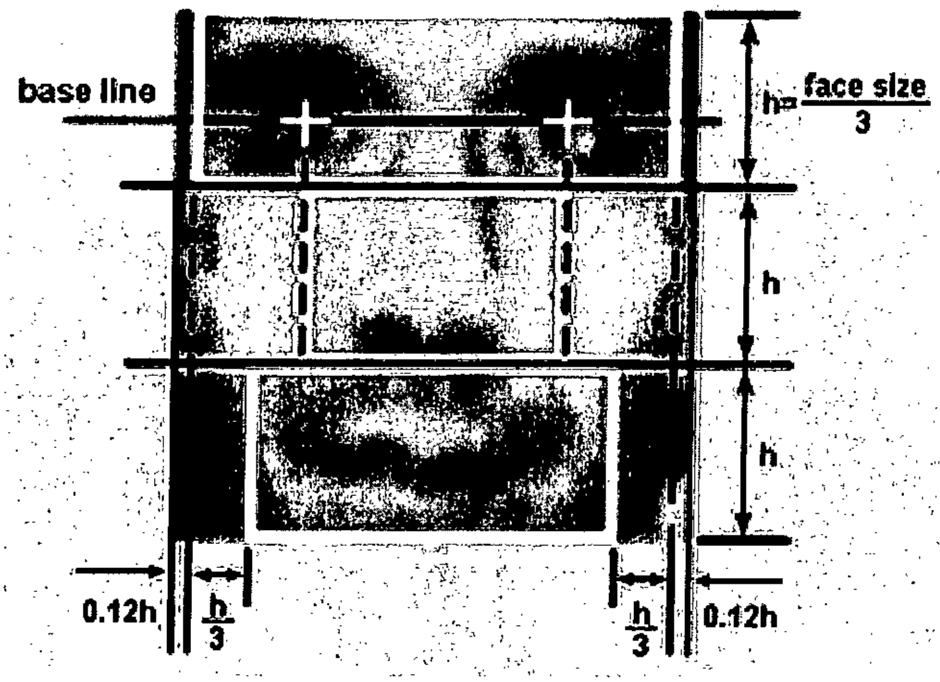
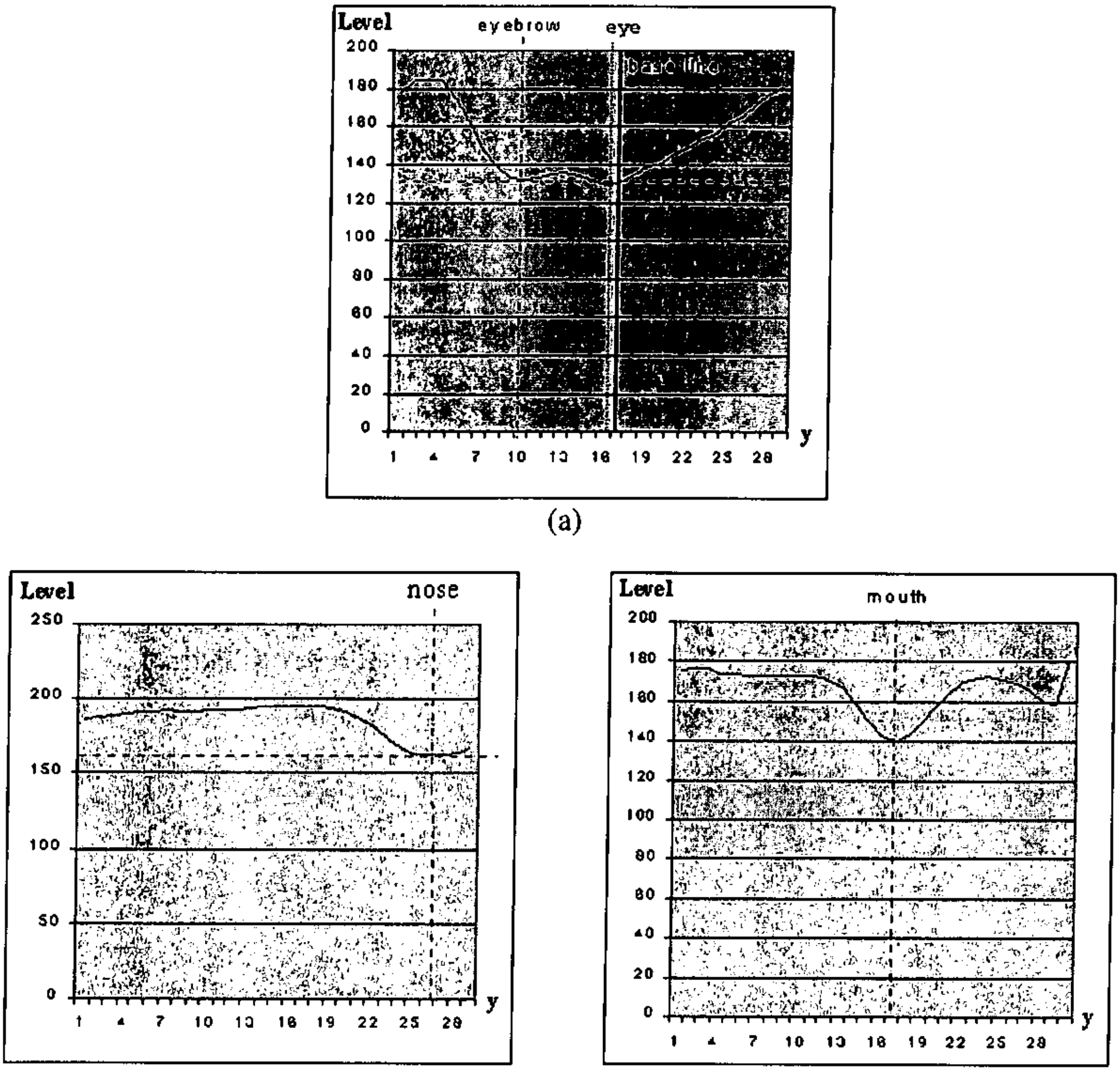


Fig. 4.8 Windows for facial feature extraction

Having obtained the vertical position of the respective facial features, the horizontal position of the facial features is then determined by the x-projection. The x-projection is computed by averaging the gray-level intensities on each column in a window. The position of the eyes can be estimated by performing an x-projection around their vertical position and identifying the location of the two minimal points of the projection. For the eyebrows, sudden changes in the x-projection values signify the end points of the eyebrows. To detect the horizontal position of a nostril in the middle window, two significant minima and a maximum between the two minima will be obtained. The first minimum represents the horizontal position of the left nostril, while the second minimum represents the right nostril. Fig. 4.10(a) shows the x-projection for determining the nostrils. For the bottom window, the mouth corner can be detected based on two assumptions; the mouth corners are close to the horizontal position of the corresponding iris and the gray-level intensity changes significantly at the mouth corner. Fig. 4.10(b) illustrates the x-projection and the determination of the detected mouth corners. The detection result for the respective facial features is shown in Fig. 4.11. Similarly, if any horizontal position of the facial features cannot be located, the candidate is assumed to be a non-facial image. Otherwise, a true face region is declared, as are the different facial features being located.

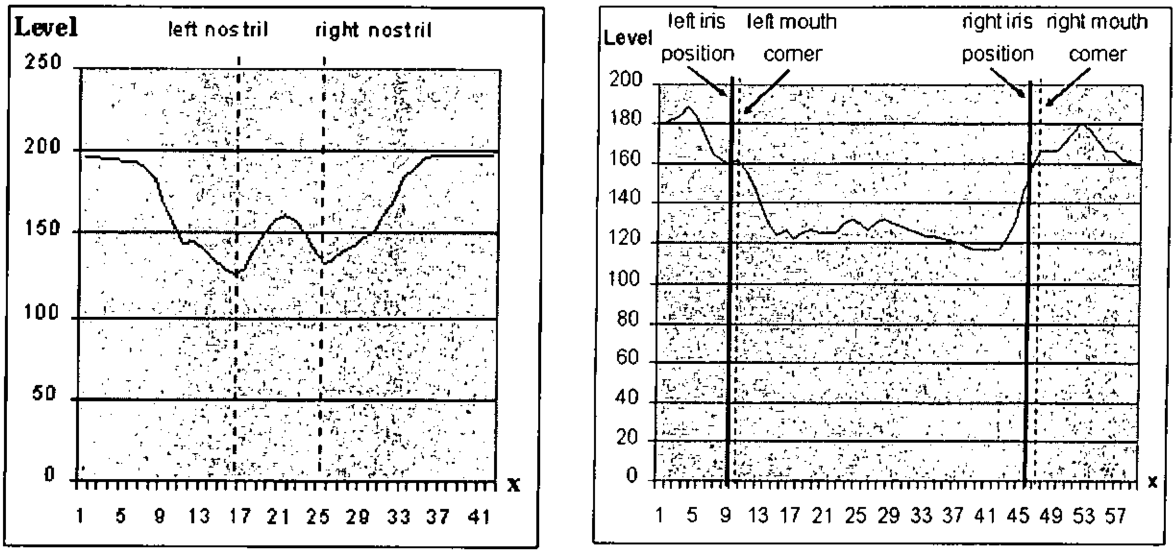


(a)

(b)

(c)

Fig. 4.9 The y-projection results of (a) eye region, (b) nose region, and (c) the mouth region



(a) (b)

Fig. 4.10 The x-projection results of (a) nose region, and (b) the mouth region

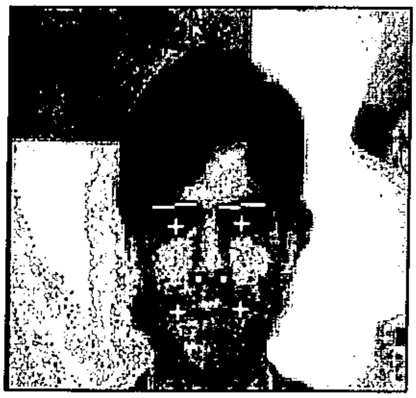


Fig. 4.11 An example of facial feature extraction by analyzing the projection of normalized face region

4.3 Experiment Results

In our approach, if the fitness value of the chromosome is greater than a threshold, it is assumed to be a possible face candidate. These possible face candidates will pass into the second stage for further verification. In the second stage, the symmetry of a face candidate will be calculated. If the difference between the left-half and right-half regions of the candidate is greater than a threshold, it is declared a non-facial image. Otherwise, the projection processes will be applied to detect the respective facial features. If the projection results of the face candidate do not fulfill the defined rules for facial features, the face candidate will also be declared a non-facial image.

The detection performance of our method is tested using the face database from MIT and some images with a number of faces. In the experiment, the training set images are different from the test images. Table 4.2 shows both the hit rate and the miss rate of our method of face detection under different conditions. This approach can achieve an overall hit rate of 100% without head tilt and under head-on lighting. When the heads tilt to the left or right, the hit rate is 95.3%. When the light source to the faces is 45°, the hit rates for the upright and tilted faces are 87.5% and 82.8%, respectively. When the lighting is 90°, the hit rate for the upright face is 93.75% and the hit rate for the tilted face is 81.25%. The experiment shows that the hit rates for a titled face after performing shirring normalization have a great improvement over our previous work [5] on face detection.

The hit rates for facial feature detection are tabulated in Table 4.3. In this part, only those faces detected successfully are considered. The reasons for the failure in detecting the facial features can be summarized as follows: facial images with glasses may affect the determination of the eyebrows; nostril detection is highly affected by the lighting conditions; and a moustache in a facial image covers the mouth corners. Fig. 4.12 shows the detection results under different lighting conditions and different angles of rotation, while Fig. 4.13 illustrates some errors in locating the facial features.

Our method is extended to the detection of multiple faces in an image. A user may choose to make either a single-face or a multiple-faces detection. The respective processes for detecting a single face and multiple faces are very similar. The major difference is in the threshold setting in stage one: the threshold value for single-face detection is greater than that for multiple-faces detection. This means that more face candidates may pass into the next stage in multiple-faces detection. Thus, false alarms will happen in this case. We have tested 20 images with multiple-faces (2 to 3 faces in each of the image). The total number of false alarms is 6, while the hit rate is 92%. The experiments were performed on a Pentium II 400MHz computer. The average processing time for locating faces and the facial features in a picture of size 128×120 is about 2.18s. In conclusion, this method outperforms those used in our previous work.

	Lighting	Head on		45°		90°	
	Head tilt	no	tilt	no	tilt	no	tilt
Full scale	hit	16	30	15	26	16	26
	miss	0	2	1	6	0	6
Medium scale	hit	16	31	13	27	14	26
	miss	0	1	3	5	2	6

Table 4.2. Experiment results for face detection

	Lighting	Head on		45°		90°	
	Head tilt	no	tilt	no	tilt	no	tilt
Full scale	hit rate of first part (eyebrow and iris)	16/16	30/30	15/15	25/26	16/16	25/26
	hit rate of middle part (nostril)	16/16	29/30	15/15	24/26	15/16	23/26
	hit rate of bottom part (mouth corner)	15/16	28/30	14/15	24/26	14/16	22/26
Medium scale	hit rate of first region (eyebrow and iris)	16/16	31/31	13/13	26/27	13/14	24/26
	hit rate of middle part (nostril)	16/16	31/31	12/13	26/27	13/14	23/26
	hit rate of bottom part (mouth corner)	15/16	29/31	11/13	24/27	12/14	22/26

Table 4.3 Experiment results for the facial feature extraction

4.4 Conclusion

In this chapter, we have proposed a more reliable face detection approach based on the genetic algorithm and the eigenface technique. Firstly, possible eye candidates are obtained by detecting the valley points. Based on a pair of eye candidates, possible face regions are generated by means of the genetic algorithm. Each of the possible face candidates is normalized by approximating the shirring angle due to head movement. Furthermore, the lighting effect is reduced by transforming their histograms into the histogram of a reference face image. The fitness value of a face candidate is calculated by projecting it onto the eigenfaces. Selected face candidates are then further verified by measuring their symmetries and determining the existence of the different facial features. The advantages of our approach are that a tilted human face can still be detected robustly even if the face is shirred, under shadow, of a different scale, under bad lighting conditions, and is wearing glasses. In conclusion, this method can achieve a high performance level in detecting human faces and extracting facial features in complex and simple backgrounds.

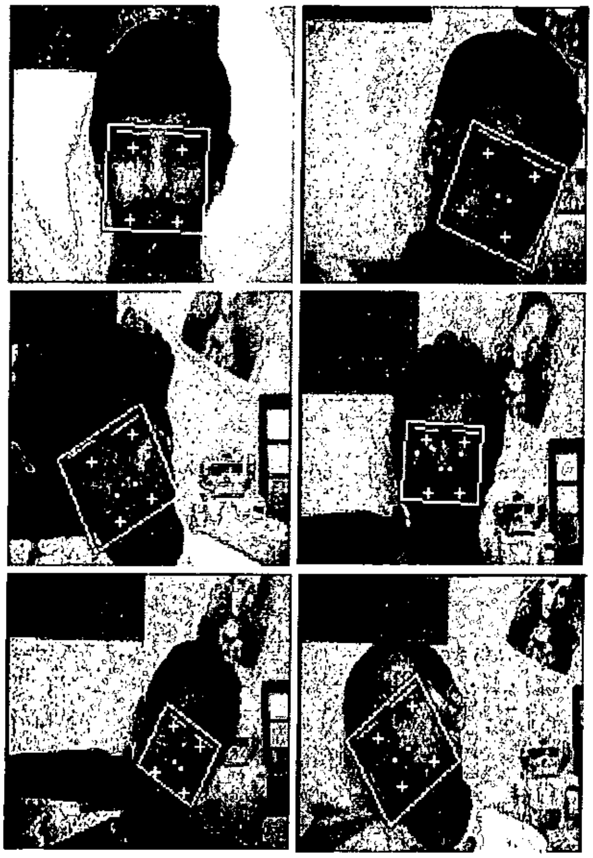


Fig. 4.12(a) Experiment results under head-on lighting

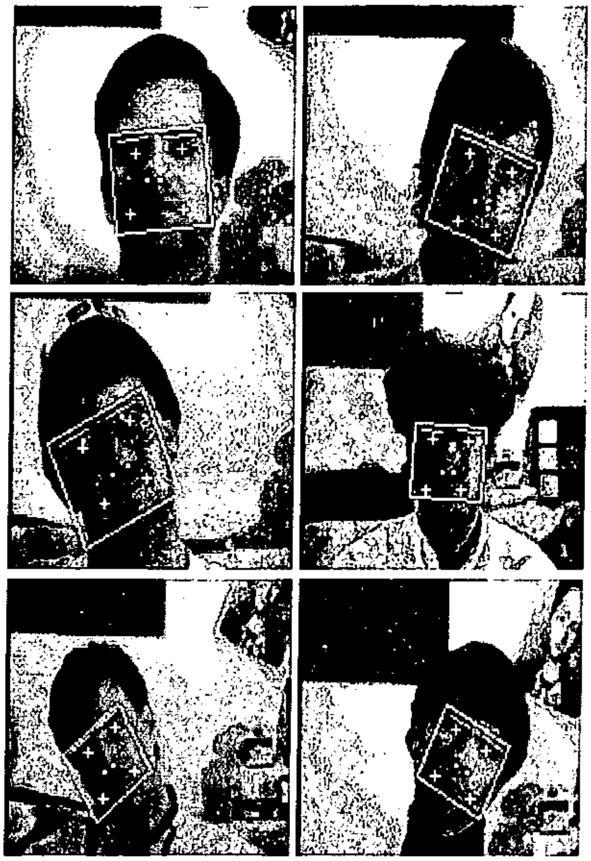


Fig. 4.12(b) Experiment results when the lighting is 45°

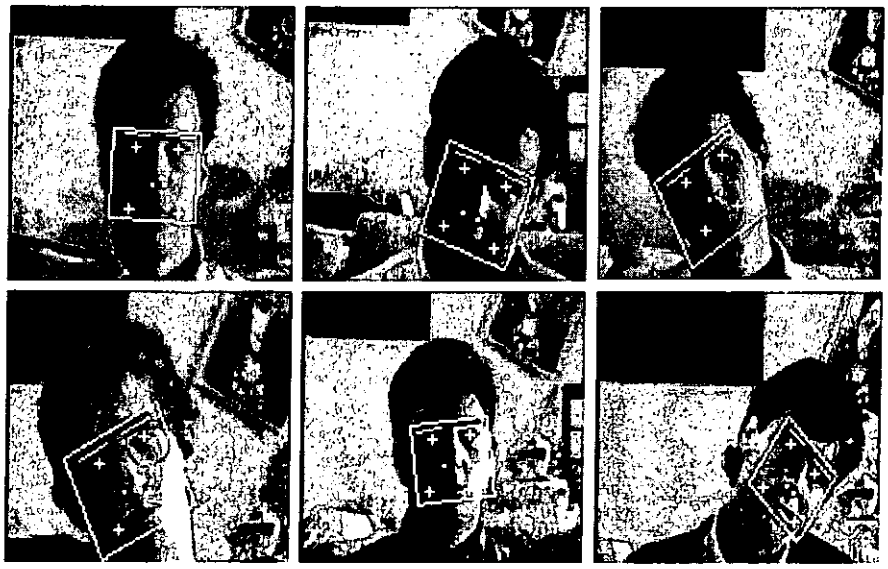


Fig. 4.12(c) Experiment results when the lighting is 90°

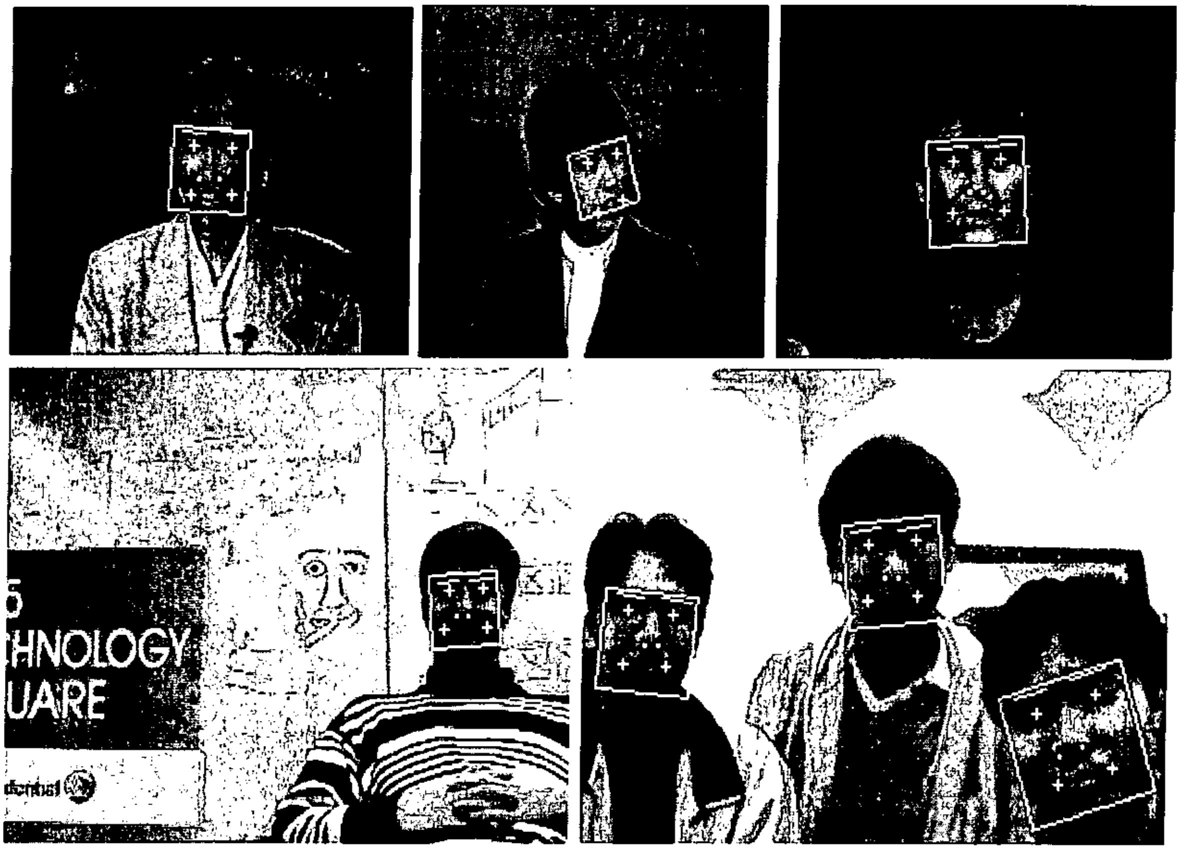


Fig. 4.12(d) Some more experiment results

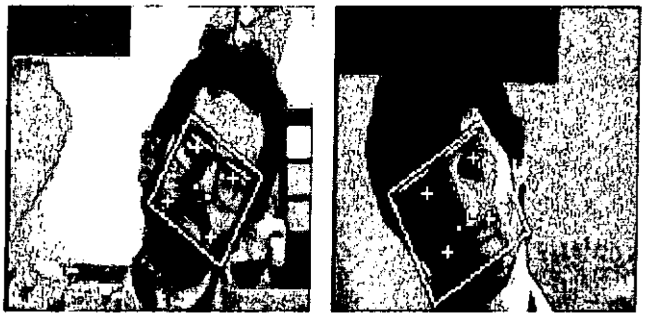


Fig. 4.13 Error in facial feature extraction

CHAPTER 5
AN EFFICIENT LOW BIT-RATE VIDEO CODING ALGORITHM FOCUSING ON MOVING REGION

5.1 Introduction

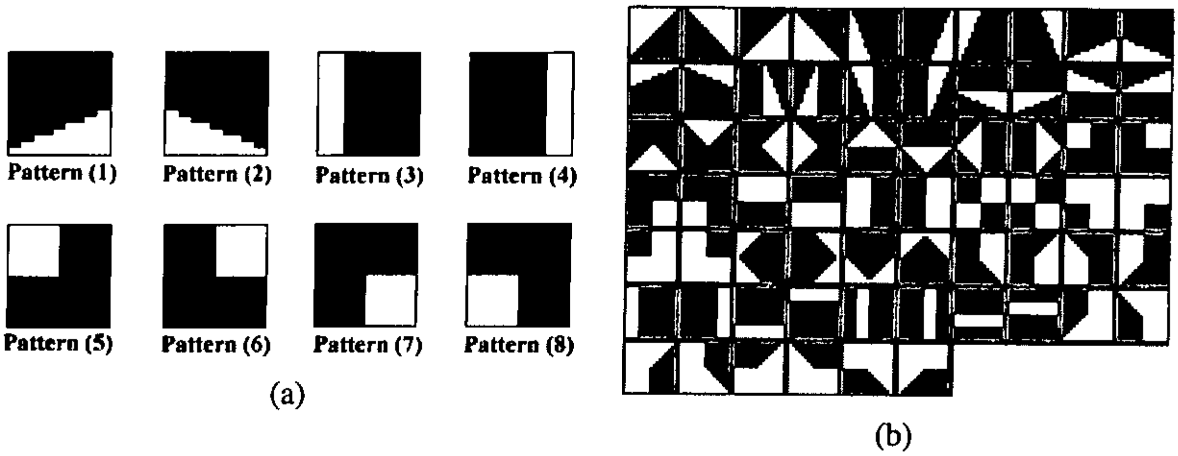
As the existing block-based motion compensated prediction coding schemes, such as H.263, do not take into account the arbitrary shape of moving objects, their level of prediction efficiency will not be as high as expected. In Chapter 3, various approaches to solving the above mentioned problem have been introduced. However, the computational complexity of the algorithms may be too high for real-time applications. Hence, an efficient encoding method is proposed which has less complexity by using eight pre-defined patterns.

In fact, motion estimation is the most important process for video coding. A precise motion predictor can achieve a good performance with image quality as well as a high compression ratio. However, motion estimation is the most time-consuming process, so a fast and precise motion predictor is necessary. A number of fast motion estimation algorithms, which have been reviewed in Chapter 3, can achieve better results than that of the conventional fast search algorithms such as the three-step search, the 2-D logarithmic search, the conjugate direction search, etc. In our approach, an efficient method for motion estimation that focuses on moving regions is devised.

The basic encoding and decoding structures of our approach are based on H.263. It includes motion estimation and compensation, DCT transformation and quantization. In our approach, the moving regions in a frame are detected and then partitioned into macroblocks. One of the eight pre-defined patterns will be used to represent the moving regions for the purpose of motion estimation and compensation. The residual errors of the macroblock will be rearranged into a block without a significant increase in high order DCT coefficients. However, if the patterns are insufficient to represent the moving regions in a macroblock, the conventional DCT-based coding method will be employed. Experimental results show that both the picture quality and the run-time are improved.

5.2 Low Bit-rate Video Coding Using Patterns

A frame in a sequence may consist of moving regions and static regions. It is unnecessary to encode the static regions in a frame, as they can be obtained from the reference frames directly. The moving regions should be encoded precisely; this is important for visual quality. In our approach, the moving regions in a frame are detected, and one of the pre-defined patterns is used to encode the moving region in a macroblock. The details of our algorithm are described as follows.



(a)

(b)

Fig. 5.1 Defined patterns: (a) the eight patterns for moving region approximation, and (b) the training patterns

5.2.1 Moving Region Detection

The pre-defined eight patterns which approximate the moving region in a macroblock are shown in Fig. 5.1(a). The white areas represent the moving region, while the black areas are the static region. The eight pre-defined patterns were selected from 66 patterns (see Fig. 5.1(b)) based on four head-and-shoulders video sequences. In our approach, the eight most popular pre-defined patterns appeared in the training video sequences were selected to represent moving regions. The moving region in a frame is detected by comparing the current frame *C*(*x*,*y*) to its previous frame *P*(*x*,*y*). The moving region *M*(*x*,*y*) in a frame is obtained as follows:

$$M(x,y) = (T|\,C(x,y) \bullet B - P(x,y) \bullet B\,|) \qquad (5.2-1)$$

where *T*(.) is a thresholding function and *B* is the structuring element of morphological operations [73][74]. After applying the morphological operations to the current and previous frames, small holes and narrow breaks are eliminated. This is a useful and efficient pre-processing step in our approach for extracting the moving regions. From eqn.(5.2-1), the computed value of the static regions in a frame is zero, while the value of the moving regions is non-zero. The processed frame is then divided into macroblocks (MB) for further classification. Three types of MB are defined in our approach: static MB (SMB), active MB (AMB), and active-region MB (RMB). If the contents of the MB are all zero or the size of moving region is smaller than a threshold, it is assumed to be a SMB. Otherwise, the MB will be divided into four sub-blocks for further classification. If the contents of all the sub-blocks have a non-zero value, the MB is defined as an AMB. The remaining MBs are assumed to be candidates of RMB. In this case, one of the eight pre-defined patterns will be used to approximate the moving regions. The best match pattern is obtained by finding a pattern that has the minimum value of $D_{K,N}$ as shown in the function below:

$$D_{K,N} = \frac{1}{256}\sum_{i=0}^{15}\sum_{j=0}^{15}\left|M_K(i,j) - P_N(i,j)\right| \qquad (5.2-2)$$

where M_K represents the K^{th} macroblock in the being processed frame $M(x,y)$, and $P_N(i,j)$ represents the pre-defined pattern number N. The value of the patterns for moving regions is defined as one, while that of the static regions is zero. If the computed $D_{K,N}$ is greater than a threshold, this implies that the patterns are not good enough to represent the moving region. In this case, the MB is assumed to be an AMB, and the conventional motion estimation and DCT-based methods will be employed. Otherwise, the MB is declared to be a RMB, and one of the pre-defined patterns will be used for encoding. The rule for classifying the types of macroblocks is shown in Table 5.1.

Macroblock Type	SMB	AMB	RMB
Condition(s) for Macroblock Type Decision	$\sum_{i,j=0}^{15} M_K(i,j) < T_{SMB}$	$\left(\sum_{i1,j1=0}^{7} M_{K1}(i1,j1) \neq 0\text{, and}\right.$ $\sum_{i2,j2=0}^{7} M_{K2}(i2,j2) \neq 0\text{, and}$ $\sum_{i3,j3=0}^{7} M_{K3}(i3,j3) \neq 0\text{, and}$ $\left.\sum_{i4,j4=0}^{7} M_{K4}(i4,j4) \neq 0 \right)$ or $\min(D_{K,N}) > T_{RMB}$	$\min(D_{K,N}) < T_{RMB}$

Table 5.1 The rule for classifying the types of macroblocks

The M_{K1}, M_{K2}, M_{K3}, and M_{K4} in Table 5.1 represent the four sub-blocks of M_K, T_{SMB} and T_{RMB} are the thresholds for macroblock type classification. The function of T_{SMB} is to verify if there is no significant movement inside a MB, while T_{RMB} is used to determine a suitable pre-defined pattern for approximating the moving region. Examples of an AMB and a RMB are illustrated in Fig. 5.2(a) and (b), respectively.

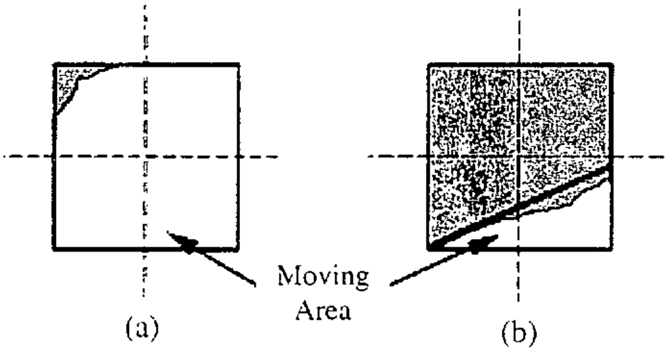


Fig. 5.2 Block type selection: (a) an AMB, and (b) a RMB approximated by pattern (1)

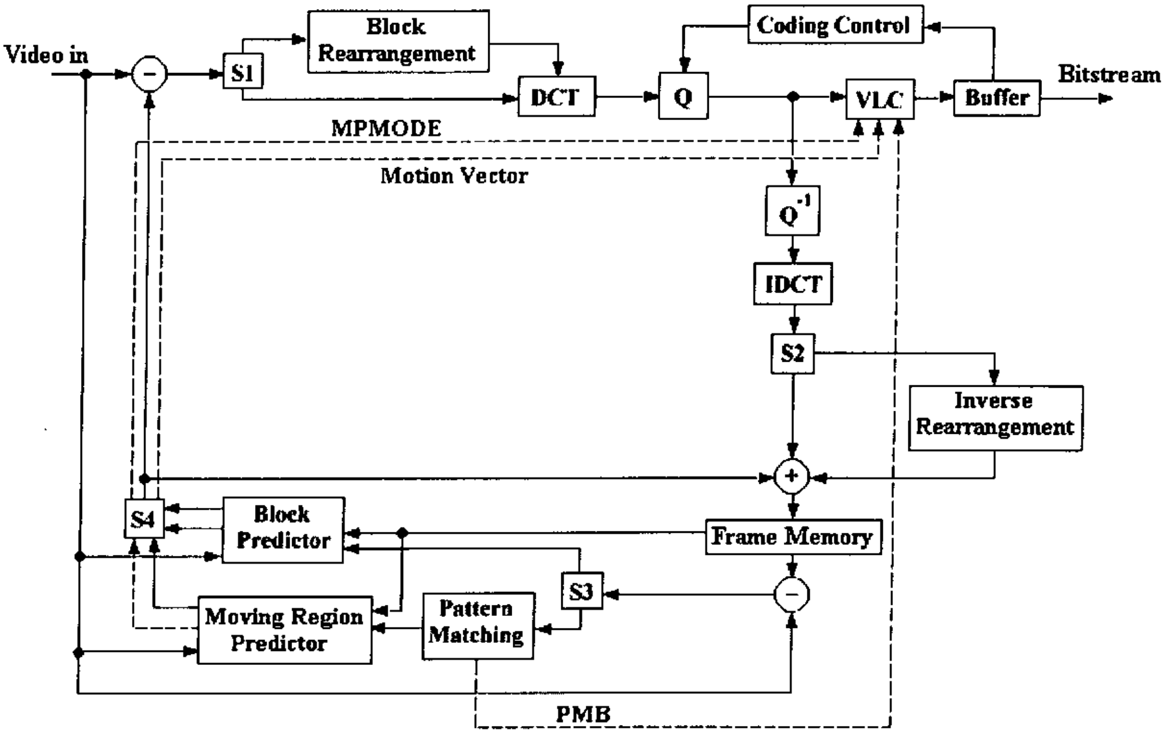


Fig. 5.3 Block diagram of the encoder

5.2.2 Architecture of the Encoder and Decoder

The block diagram of the proposed video encoder is shown in Fig. 5.3. The structure is similar to a conventional video encoder, but includes additional features for encoding the moving regions:

1. Type of Macroblock (MB): A MB will be classified according to the results of moving region detection. Different encoding methods will be applied for different types of macroblock. If a SMB is detected, it is encoded as a skipped MB, which is supported by H.263. For the AMB, the conventional block-based encoding method is used. This includes block motion estimation, motion vector encoding, and residual error encoding. If RMB is detected, the best match pre-defined pattern will be extracted by the "Pattern Matching " unit for motion estimation and compensation.

2. Pattern Matching: The function of the "Pattern Matching" unit is to find a pre-defined pattern which is the best representation of the moving region in a RMB. The PMB (pre-defined pattern information) will be generated from this unit, and then encoded by VLC.
3. Block Rearrangement: If the input is a RMB, the switch S1 will be connected to the "Block Rearrangement" unit to rearrange the residual errors of the moving regions into a block of size 8×8. Hence, fast DCT algorithms can be used to encode the block. Similarly, the output from the IDCT will be rearranged in an inverse manner by connecting the switch S2 to the "Inverse Rearrangement" unit.
4. Motion Estimation: The respective prediction procedures for AMB and RMB are different. The switch S3 is connected to the upper path for AMB, while it is connected to the lower path for RMB, to which moving regions prediction is applied. The block predictor employs the conventional block-based motion estimation for encoding the AMBs. For RMB, the proposed moving region predictor as described in Section 5.2.3 is used for finding the motion vector which best represents the motion of the moving region. The static region is obtained directly from the reference frame. According to the MPMODE (motion predictor mode), the switch S4 will select the compensated image from either the block predictor or the moving region predictor. The MPMODE together with the motion vector are encoded by the VLC.

The architecture of the decoder is shown in Fig. 5.4. When the MPMODE indicates the current input MB to be a RMB, the switch S1 will connect to the "Region Moving Predictor" to generate the compensated image. The switch S2 will also select the upper path, the "Inverse Rearrangement" unit, to rearrange the residual error. For the AMB, the conventional block predictor will be employed, and the residual error will be decoded by inverse quantization, and then inverse DCT.

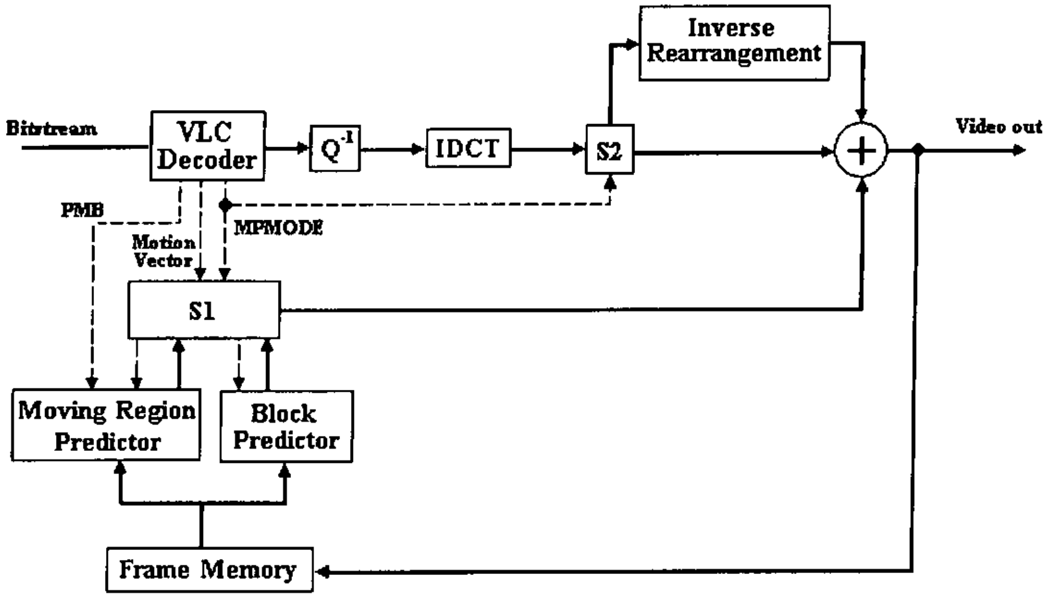


Fig. 5.4 Block diagram of the decoder

5.2.3 Motion Estimation and Compensation

The coding performance, which includes the picture quality, the run-time, and the compression ratio, is affected by the motion estimation and compensation process. In our approach, the contents of SMB are copied from the reference frame directly. For the AMB, the conventional motion estimation and DCT-based methods are used. For RMB, both moving and static regions exist, but it is not necessary to use the static region in motion estimation. In [75], an efficient method was proposed for reducing the missing edge effect by using the edge information. In our approach, the motion vectors for the RMB are obtained by considering the moving regions only, which can reduce the complexity of the motion estimation process as well as prevent the missing edge effect.

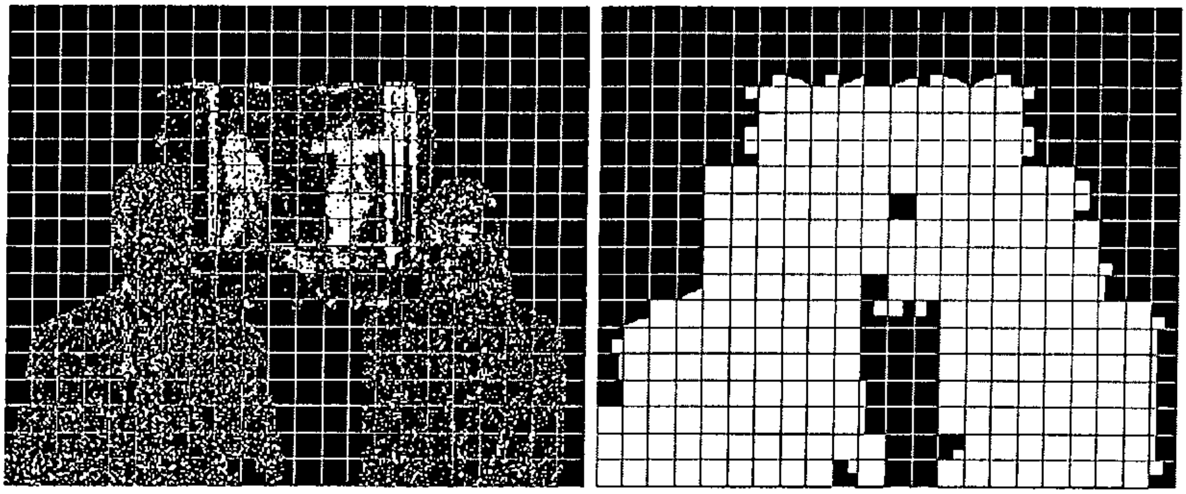


Fig. 5.5 An example of pattern approximation for the sequence "News": (a) Detected moving regions, and (b) results of pattern approximation

The matching process is more precise as the static region is neglected in motion estimation.

The mean-absolute difference (MAD) criterion function is employed for motion estimation. The motion vector for the moving region in a RMB can be obtained by finding the minimum value of the function defined as follows:

$$MAD(dx,dy)=\frac{1}{64}\sum_{\substack{\forall P_v(i,j)\\=1}}\left|C(i,j)-G(i+dx,j+dy)\right| \qquad (5.2-3)$$

where G(*i,j*) represents the MB from a reference picture, and (*dx,dy*) is a vector representing the search location. From eqn. (3), the total number of points used for calculating the MAD is reduced. For example, in each search location using conventional methods, 256 "minus" operations, 255 "sum" operations, and 256 "absolute" operations are required. With our approach, it requires only 64 "minus" operations, 63 "sum" operations, and 64 "absolute" operations. For the motion estimation and compensation of a RMB, the contents of the static region are directly copied from the previous frame, while the motion vector of the moving region is determined by using eqn.(5.2-3). The determination of the best matched pattern in a video sequence and the principle of interframe coding for RMB are illustrated in Fig. 5.5 and Fig. 5.6, respectively. As a result, both the prediction errors and the complexity required for motion estimation are reduced.

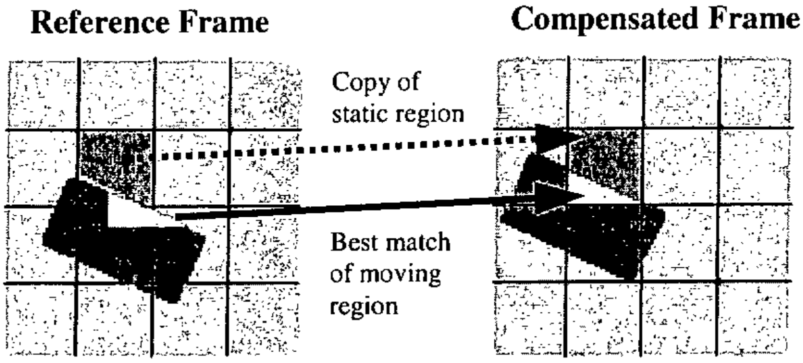


Fig. 5.6 The principle of interframe coding for RMB

5.2.4 Prediction Error Encoding

The encoding of residual errors for an AMB is based on the conventional DCT-based coding methods. However, if this conventional method is applied directly for encoding a RMB, two blocks will have to be processed. In our approach, the prediction errors of the moving region in a RMB are rearranged to a block of size 8×8, and DCT transform is then employed. Based on the distributions of the residual errors, the rearrangement method is devised without a significant increase in the high order transform coefficients. As a result, only a block is needed to encode for a RMB, and the compression ratio is increased. The rearrangements of the residual errors in the spatial domain for four of the patterns are illustrated in Fig. 5.7. The arrows in the diagrams indicate the rearrangement order in a MB. The rearrangements for patterns 5-8 are not illustrated, as they are already in the form of a block.

In our coding method, two extra codewords, "MPMODE" and "PMB", are needed. The "MPMODE" uses 1 bit to represent the prediction mode: the conventional block-based prediction or the moving region prediction. The "PMB" uses 3 bits to indicate the pattern being selected for moving region prediction.

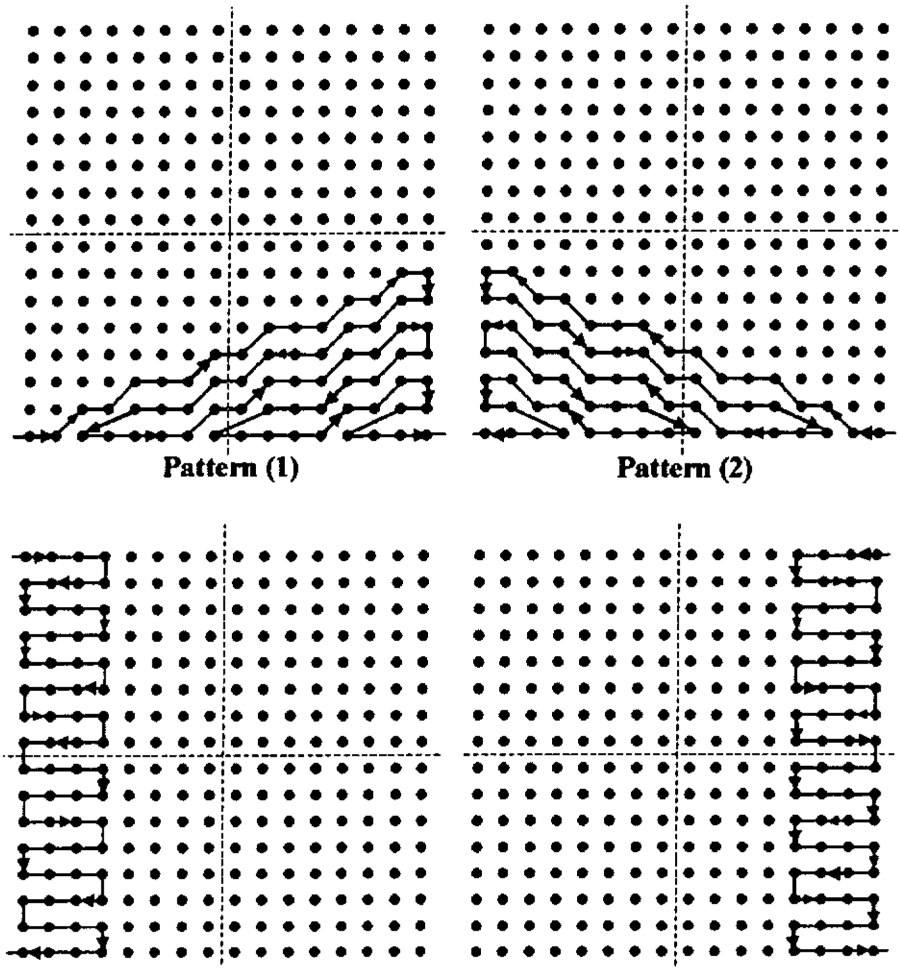


Fig.5.7 Rearrangement of the residual errors

5.3 Simulation Results

The performance of the proposed algorithm is evaluated based on a variety of image sequences, which include typical head-and-shoulders video sequences, smooth motion sequences, and active motion sequences. The results of our proposed method are compared to the H.263. Full search motion estimation and the TMN-8 [64] rate control method are employed for obtaining the encoding results. Table 5.2 and Fig. 5.8 illustrate the encoding results of the first 100 frames using our approach, as well as the H.263 scheme. It can be observed that our proposed method generally gives better image quality than that of the H.263 for low and high bit rates. However, its performance regarding motion-intensive video (e.g. Foreman) is reduced. There is no prediction gain for such sequences using our approach. This is due to the fact that these kinds of video sequences result in a small number of RMBs, while our approach requires additional bits to represent the MB type. Consequently, there is no advantage of encoding high motion-intensive sequences using our approach. Table 5.3 shows the improvement in the encoding time and the total number of detected RMBs in the first 100 frames. Experiment results show that the encoding time of our approach is much less than that of the H.263. The amount of time saved for encoding a frame for head-and-shoulders or smooth motion sequences varies from 8.69% to 53.53%. For encoding the active sequence "Foreman", the approximating simulation time is 2.69% longer than that using the H.263. As a small number of RMBs are detected in this sequence, so the time saved on motion estimation is insufficient to compensate for the time spent on the pre-processing step in our approach. In general, when the number of RMBs detected increases, the required encoding time will also decrease, while the PSNR will increase.

However, the run-time will become longer and the PSNR will be reduced slightly when fewer RMBs are detected.

Sequence	Picture format	Target bit rates	Average bit per frame		Average PSNR (dB)	
			H.263	Proposed method	H.263	Proposed method
Akiyo	QCIF	8kbps	735.67	734.73	35.15	35.39
Salesman	QCIF	8kbps	844.81	842.05	31.27	31.35
Akiyo	QCIF	10kbps	914.02	913.11	36.80	36.99
Claire	QCIF	10kbps	940.65	935.75	35.83	36.07
Salesman	QCIF	10kbps	1020.90	1020.86	32.42	32.46
Akiyo	CIF	16kbps	1508.11	1510.84	33.02	33.19
Miss America	QCIF	16kbps	1474.58	1463.46	38.42	38.53
Mother-daughter	QCIF	16kbps	1518.28	1515.08	32.60	32.68
Akiyo	CIF	24kbps	2216.69	2211.39	35.75	35.82
Mother-daughter	QCIF	24kbps	2313.95	2310.36	35.45	35.49
Foreman	QCIF	24kbps	2372.27	2409.95	30.59	30.57
News	CIF	32kbps	3183.74	3198.74	30.25	30.44
Foreman	QCIF	48kbps	4700.82	4730.65	34.50	34.45
News	CIF	48kbps	4805.11	4801.83	33.55	33.58

Table 5.2 Simulation results

Sequence	**Picture format**	**Target bit rates**	**Total number of RMB in the first 100 frames**	**Encoding time saved per frame**
Akiyo	QCIF	8kbps	1803	18.01%
Salesman	QCIF	8kbps	4359	42.31%
Claire	QCIF	10kbps	3226	53.53%
Miss America	QCIF	16kbps	3545	38.71%
Mother-daughter	QCIF	16kbps	1976	8.69%
Akiyo	CIF	24kbps	6009	32.13%
News	CIF	32kbps	9669	34.07%
Foreman	QCIF	48kbps	737	-2.69%

Table 5.3 Encoding time saved per frame compared to H.263

Fig. 5.8 PSNR of the first 100 frames for the video sequence "Akiyo" (target bit-rate=8kbps)

We have also selected four pre-defined patterns only, patterns 1 to 4, to encode the sequence "Akiyo". The experiment results are shown in Fig. 5.9. It can be observed that the PSNR using 8 patterns is higher than that of using 4 patterns and the H.263 scheme at the same bit-rates; the maximum prediction gains using 8 patterns compared to the H.263 scheme and the 4-pattern approach are about 0.6dB and 0.5dB, respectively. However, the H.263 scheme outperforms the 4-pattern approach at high bit-rates. This is due to the fact that the four patterns are insufficient to approximate the moving regions, so the prediction gain is not as good as when using eight patterns or the H.263. In conclusion, the coding efficient using 8 patterns is better than that of using 4 patterns as well as the H.263 scheme.

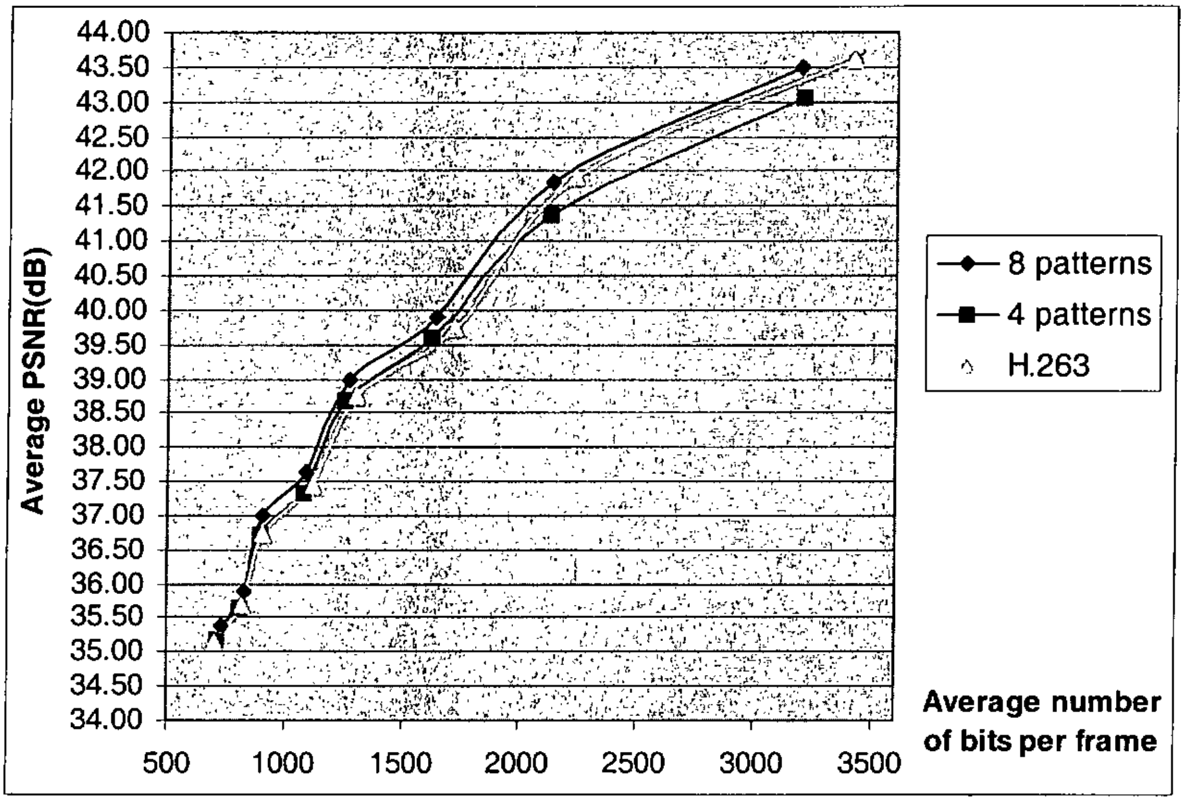


Fig. 5.9 Simulation results for the video sequence "Akiyo" with different bit-rates

5.4 Conclusion

We proposed an efficient very low bit-rate video coding scheme for the applications of video-conferencing and video-phone. Eight pre-defined patterns were chosen by experiments to represent moving regions. Based on the pre-defined patterns, the computation required for motion estimation is reduced. The encoding time of our approach is much less than that for H.263 for smooth motion sequences. Furthermore, in order to reduce the size of a MB to be encoded, we devised a rearrangement method to compact the residual errors of a MB into a block of size 8×8. The simulation results show that our approach can produce the same image quality with faster encoding speed compared to the H.263 scheme for the head-and-shoulders and smooth motion sequences. However, the total number of detected RMB will decrease in motion-intensive video sequences, in which its performance will be degraded and close to that of the H.263. In conclusion, this approach outperforms the H.263 in terms of the run-time for sequences of smooth motion.

CHAPTER 6

FACE SEGMENTATION AND FACIAL FEATURE TRACKING FOR VIDEOPHONE APPLICATIONS

6.1 Introduction

In the previous chapters, we introduced the applications of face segmentation for foreground/background video coding, pattern recognition, indexing, and object tracking. In this chapter, the applications of face segmentation and facial feature tracking for video-conferencing systems will be presented.

In traditional block-based video coding, blocky artifacts always occur in pictures at low bit rates. Hence, our objective is to devise a video coding algorithm which can produce a better perceptual quality of the encoded picture. In our approach, the important objects such as the human face and its respective facial features will be extracted. More bits are then allocated to these important objects, while fewer bits will be reserved for the non-face region.

In this video coding approach, the encoding structure is based on the proposed method as described in Chapter 5. The major difference between the two lies in the use of different qunatization step sizes in the face region, the facial feature region, and the background scene. The first step in our approach is to segment out the face region and its respective facial features. In Chapter 4, we proposed a method for extracting the face region and the facial features by using the eigenface and the genetic algorithm. Based on this face segmentation method, a modified method will be proposed which can extract the face region and its respective facial features more efficiently. As for facial feature tracking, a fast and robust method has also been developed and implemented. Finally, the location of the facial features is passed into the encoder for compression. Our experiment shows that both the subjective and the objective image quality of the face region are improved.

6.2 Efficient Coding Method for the Face Region

The computational complexity and the accuracy of the facial feature tracking process are important issues to be considered. They affect the encoding time and the coding performance with respect to the face region. A modified approach for face segmentation based on the method proposed in Chapter 4 will be introduced in the next section. Furthermore, a fast and robust method for tracking the facial feature using a "*star*" pattern will be described. The details of our approach are described below.

6.2.1 Face Detection and Facial Feature Extraction

In Chapter 4, we proposed a method for face detection. Firstly, the possible human eye regions are detected, and then pairs of eye candidates are selected by means of genetic algorithms to form possible face candidates. A fitness function based on the eignface technique is defined to determine whether the input candidate is a face image or not. This approach will perform well if these are numerous possible eye candidates. However, the use of genetic algorithm to search for face regions will not be efficient if the number of possible eye candidates is not large. In order to improve the runtime and the detection performance, the number of possible eye candidates is used to determine the search method to be used. If the total number of possible eye candidates is smaller than a threshold, the genetic algorithm approach will not be used. All the possible eye candidates will be investigated to determine possible face candidates. On the other hand, if the total number of possible eye candidates is larger than the threshold, the genetic algorithm approach will be employed, and the initial population size will be defined as below:

$$initial\ population\ size = \frac{total\ number\ of\ possible\ eye\ candidates}{n} \tag{6.2-1}$$

where n is a number whose value depends on the maximum number of iterations to be allowed. With this formulation, the initial population size will be set at a large value if the possible population is large, and it will be set at a smaller value if the number of iterations allowed for searching is large. This arrangement adapts the population size to be used to the problem situation, hence making more efficient the searching process. If a large number of possible eye candidates are detected, more possible face candidates will be generated which will in turn increase the chance of finding the best solution. If a small number of possible eye candidates are detected, a smaller number of chromosomes will be generated and the searching time can be reduced.

The approach for facial feature detection is the same as that proposed in Chapter 4. All the face candidates with a high fitness value are selected for further verification. The facial features are determined by evaluating the topographic relief of the normalized face regions. Fig. 6.1 shows a detected face region and its respective facial feature points in the first frame of the "*AKIYO*" sequence. In this coding approach, the extracted facial features include the two eyes, and the two mouth corners.

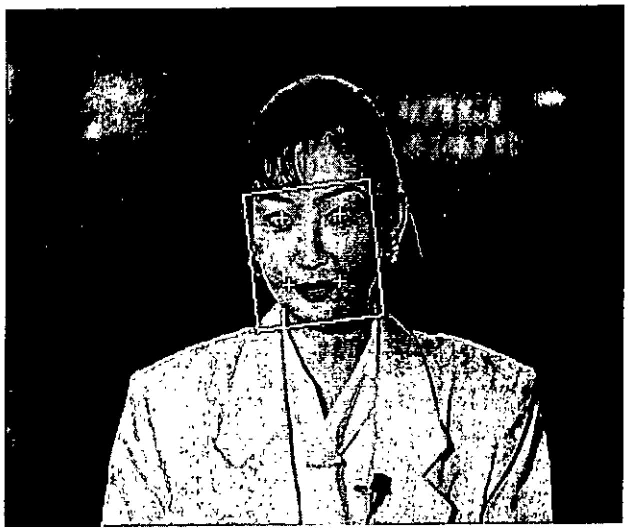


Fig. 6.1 Detected face region and its respective facial feature points

6.2.2 Face and Facial Feature Tracking

Tracking the facial feature points is an important process in our approach. The accuracy and the detection time for the facial feature points would affect the perceptual quality of the encoded picture and the encoding time. In [76][77], the facial feature points are tracked by using the principle components analysis. However, the computational complexity may be too high for real-time applications. In this approach, a less complex method for feature point tracking is introduced.

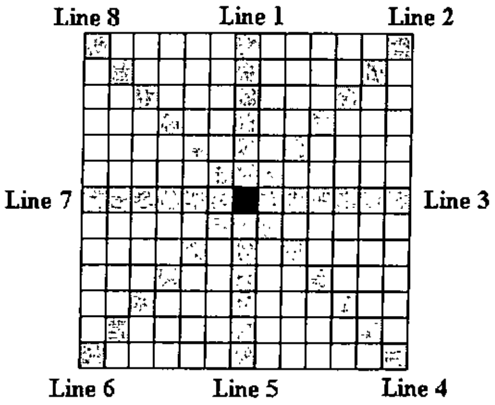


Fig. 6.2 Defined "*star*" pattern for facial feature tracking

In [7], a block of $N \times N$ pixels is used for tracking the feature points within a search window. However, the computational complexity of this method may be too high, and the errors generated during the tracking process may be cumulative. In our approach, we choose some of the important points within a block for the tracking process by using a "*star*" pattern, as shown in Fig.6.2. This means that a "*star*" pattern instead of the whole block is used for searching the new position of the feature point within a search window. An advantage of using the defined pattern is that the computational complexity required for matching can be reduced. Lines 1 to 8 containing l pixels each, as shown in Fig. 6.2, represent the structure of the "*star*" pattern, and the darkest pixel represents the centre. In this pattern, the pixel density near the point under consideration is denser than that far away from the point. The facial features are extracted in the first frame by means of the proposed method as described in Section 6.2.1. A new feature point is then obtained by minimizing a cost function. This cost function consists of two parts: the first part, L_n, measures the similarity between the extracted facial feature in the first frame and the possible facial feature in the current frame with displacement (dx,dy), the second part, S, determines the symmetrical properties of the possible facial feature with displacement

(dx, dy). The purpose of the first part is to prevent the accumulation of errors during the tracking process, while the purpose of the second part is to track the eye and mouth regions robustly regardless of whether the eye and mouth are open or close. The cost functions for tracking the iris $F_{eye}(dx, dy)$, and the left mouth corner $F_{mouth}(dx, dy)$ are defined as follows:

$$F_{eye}(dx, dy) = \frac{1}{8}\sum_{n=1}^{8} L_n(dx, dy) + S_{eye}(dx, dy) \qquad (6.2-2a)$$

$$F_{mouth}(dx, dy) = \frac{1}{8}\sum_{n=1}^{8} L_n(dx, dy) + S_{mouth}(dx, dy) \qquad (6.2-2b)$$

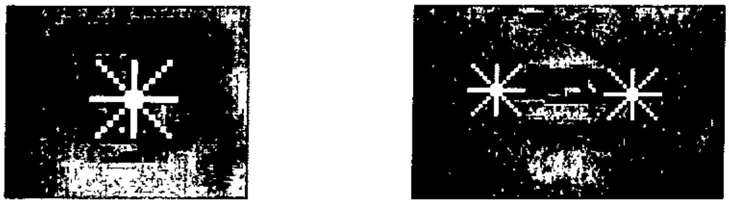
$$L_n = \frac{1}{l}\sum_{k=1}^{l} \left| N_n(k) - P_n(k) \right| \qquad (6.2-2c)$$

where $N_n(k)$ and $P_n(k)$ represent the pixel intensity in the current frame and the first frame at position k of line n, respectively. l is the length of the line, and $L_n(dx, dy)$ represents the absolute difference between line n of the first frame and line n of the current frame with displacement (dx, dy). The pixel intensities of the lines should be related to each others according to the characteristics of iris and mouth corners. For example, the differences between the lines 1 and 5, the lines 6 and 8, and the lines 2 and 4 of the mouth corners should be small. Hence, the second term in the cost functions for the eyes and the two mouth corners are defined as follows:

$$\begin{aligned} S_{eye} = \frac{1}{4l}\sum_{m=0}^{l} \Big(&\left| P_1(m) - P_5(m) \right| + \left| P_2(m) - P_8(m) \right| \\ &+ \left| P_3(m) - P_7(m) \right| + \left| P_4(m) - P_6(m) \right| \Big) \end{aligned} \qquad (6.2-3a)$$

$$\begin{aligned} S_{mouth} = \frac{1}{3l}\sum_{m=0}^{l} \Big(&\left| P_1(m) - P_5(m) \right| + \left| P_6(m) - P_8(m) \right| \\ &+ \left| P_2(m) - P_4(m) \right| \Big) \end{aligned} \qquad (6.2-3b)$$

where $P_n(m)$ is the pixel intensity at line n and position m. Fig. 6.3 shows the tracking process using the "*star*" pattern. In our approach, a 5x5 search window is used. Furthermore, the length of the line in the pattern can be changed according to the distance between the left and the right iris. This means that the length of the line will be changed according to the face size. A larger "star" pattern will be used for a larger face. Some results generated by the proposed tracking method are illustrated in Fig. 6.4. In our approach, an open eye, close eye, open mouth and close mouth can be tracked robustly. All the tracked facial feature points will be passed into the encoder for purposes of controlling the quantization step sizes for these regions.



(a) (b)

Fig. 6.3 Facial feature tracking using the "star" pattern on (a) eye region and (b) mouth region

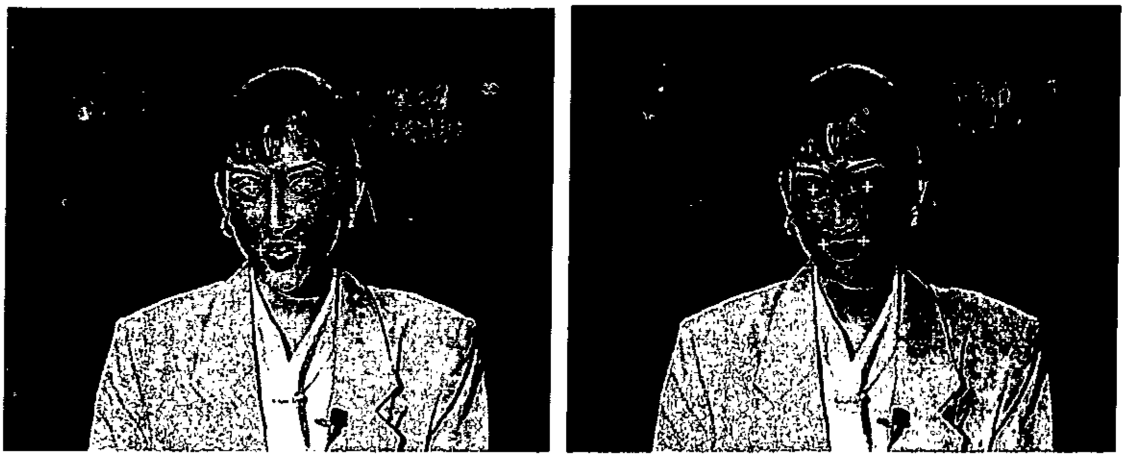


Fig 6.4 Results of facial feature tracking

6.2.3 Quantization Step Sizes for the Face and Non-face Regions

The coding approach is based on the proposed method as described in Chapter 5. Firstly, moving regions are extracted and represented by using a number of pre-defined patterns. Motion estimation is performed based on the patterns, and the prediction errors of an MB are rearranged to a block size. Under this coding approach, more bits are spent on the detected face region, and fewer bits are used for encoding the non-face regions such as the background scene. This can be done by using smaller quantization step sizes for the facial feature region, and larger quantization step size for the background region. The proposed quantization step sizes for the facial region, face region, and non-face regions are shown in Table 6.1.

Active region	facial feature region	use the finest quantization Q_1=Q_P
	face region	use second finer quantization Q_2=Q_P+4
	other active region	use the coarsest quantization Q_3=Q_P+14
Static region	background	skip

Table 6.1 The proposed quantization step size for the face, facial feature, and background regions

6.3 Simulation Results

The performance of the proposed algorithm was evaluated using the "*AKIYO*" video sequence with a CIF size. Our proposed face detection and facial feature extraction algorithm are used to separate each input frame into facial feature regions, face region, and background region at block level. The results of the segmented regions are shown in Fig. 6.5.

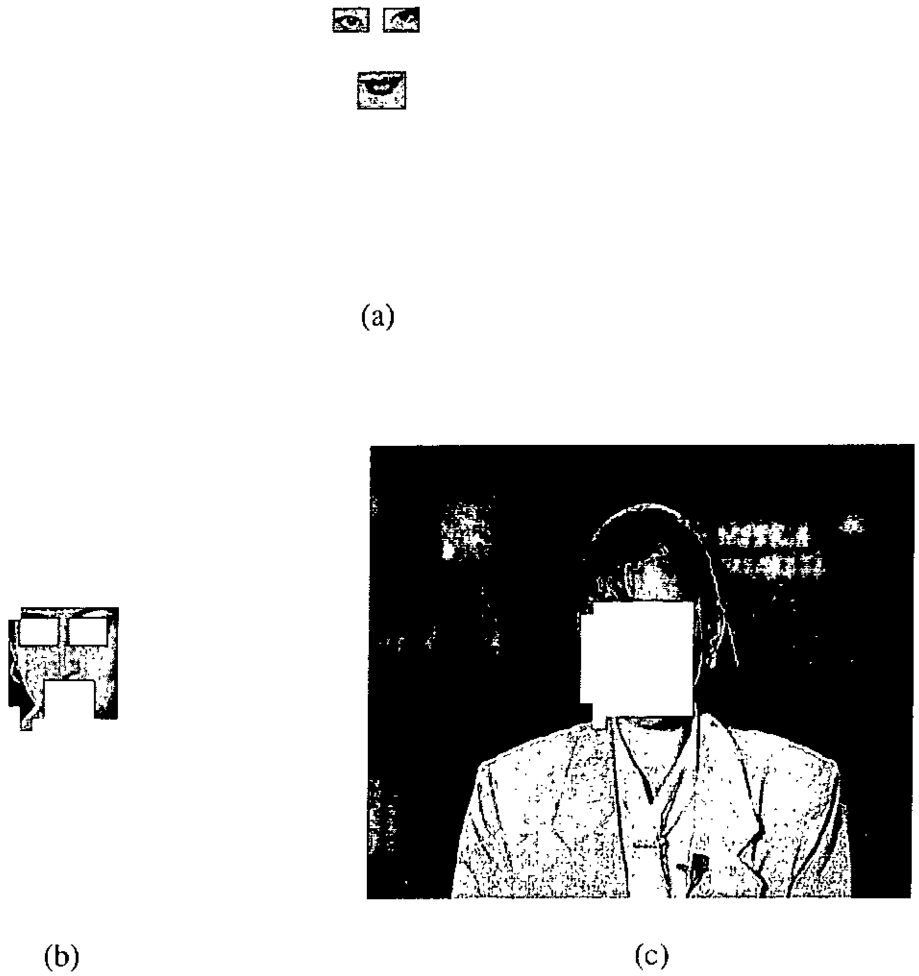


Fig. 6.5 Segmentation results: (a) eye and mouth regions, (b) face region, and (c) background region.

The smallest quantization step size Q_1 is applied to encoding the extracted eye and mouth regions. As for the background region, the largest quantization step size Q_3 is used for this less important region. The quantization step size for the face region is defined between Q_1 and Q_3. The "*AKIYO*" sequence with CIF size is used to evaluate the performance of our approach. Table 6.2 and Fig. 6.6 illustrate the coding results based on our approach and the H.263 scheme for the first 100 frames. The encoding image quality is improved for the facial feature regions and face region, while the quality for the background region is reduced. With the same targeted bit-rate, the average PSNR of the picture quality using our approach is smaller than that of H.263. However, the subjective quality of the picture is better than that of H.263. The quality of the encoded image using our approach and H.263 is shown in Fig. 6.7(a) and Fig. 6.7(b), respectively. Fig. 6.7 (c) and Fig. 6.7(d) illustrate the magnified face region in Fig. 6.7(a) and Fig. 6.7(b), respectively.

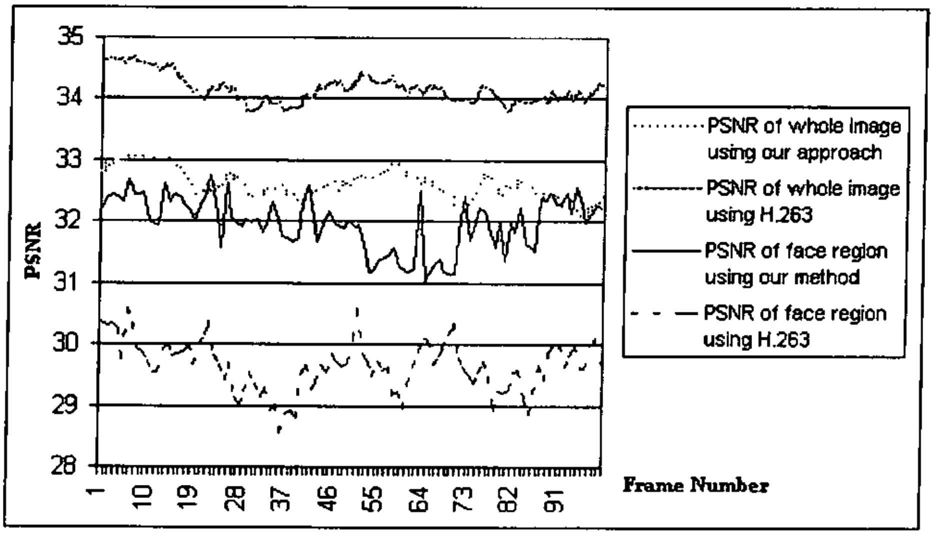
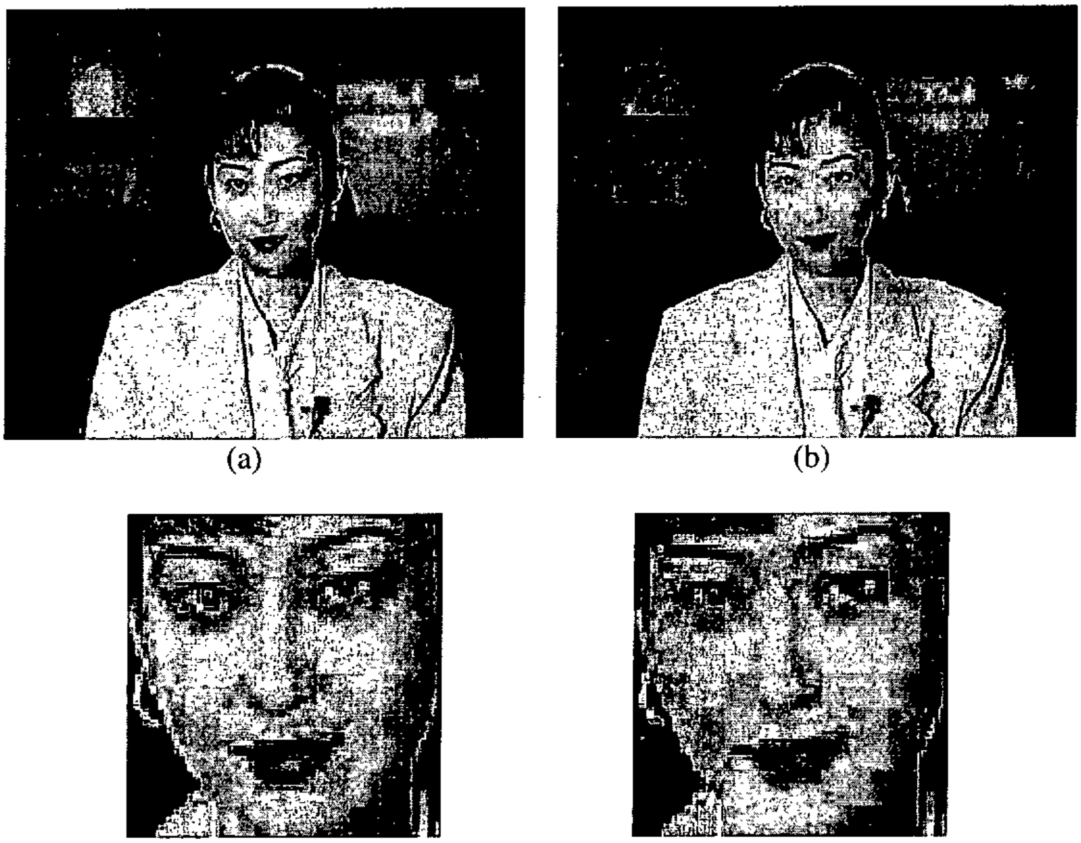


Fig. 6.6 The PSNR of "*AKIYO*" sequence with CIF size

	Average bits/frame	Average PSNR of the whole image	Average PSNR of face region
Our approach	721.57	32.62	31.97
H.263	739.69	34.16	29.65

Table 6.2 Simulation results


(a) (b)

(c) (d)

Fig. 6.7 The quality of encoded image: (a) encoded image using our approach, (b) encoded image using H.263, (c) magnified image of (a), and (d) magnified image of (b)

6.4 Conclusion

In this Chapter, we have proposed an algorithm which can improve the quality of important regions in a video for videophone application. Firstly, the face region and its respective facial feature regions are extracted. The segmented regions are then passed into an encoder for encoding. More bits are allocated to the facial feature regions and the face region, while fewer bits are used for encoding the background scene. This is accomplished by using a smaller quantization step size to encode the facial feature regions and the face regions, while a larger quantization step size is employed for the background region. Our experiment shows an improved perceptual picture quality over that of H.263.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

In this thesis, we have introduced various techniques for face segmentation, facial feature extraction, and video compression. For face segmentation, the shape analysis approach, the knowledge-based approach, the example-based learning approach, and the color analysis approach have all been reviewed. From the experiment results, we may conclude that the computational complexity of the shape analysis approach and the example-based approach are too high for real-time applications. The required runtimes for both these approaches are in the order of minutes. The detection speed of the color analysis approach and the knowledge-based approach is faster. However, the color analysis approach can work properly only under well controlled lighting conditions or with a simple background.

For facial feature extraction, we have reviewed some algorithms such as the projection method and the deformable templates. The experiment results have shown that the projection method is simple and easy for implementation, but the performance is quite dependent on the accuracy of the extracted face region as well as the lighting conditions. The computational complexity required by the deformable templates techniques is too high. The edge information is used as a component of the energy functions for the templates. Hence, the pre-processing step for reducing the lighting effect is important for these approaches.

We have also described various approaches for video coding. The block-based motion compensated prediction coding is one of the most popular techniques for video compression. Many successful coding standards, such as H.261/H.263, and MPEG, are based on the block-based motion compensated prediction coding techniques. However, in block-based video coding method, blocking artifacts always occur in the areas of the picture that are annoying to the viewers. Hence, many advanced video coding algorithms such as the foreground/background coding scheme and 2D/3D hybrid video coding methods have been examined for improving the perceptual quality of the encoded picture.

In this research, we have devised a reliable algorithm for face detection and facial feature extraction. Possible face regions are formed by means of genetic algorithm. Eigenface techniques is employed for determining whether the input region is a face image or not. Finally, the projection method is applied to detect the facial feature points. The advantages of our approach are that a tilted human face can still be detected robustly even if the face is shirred, under shadow, of a different scale, under bad lighting conditions, and is wearing glasses. Furthermore, we have implemented the H.263 scheme and devised a coding method based on it. Since the H.263 does not take into account the arbitrary shape of an object in coding, we pre-defined eight patterns to represent the arbitrary shape region, and used those patterns for motion estimation and compensation. As a result, both the computation for motion estimation and the prediction errors are reduced. However, additional information is required to represent the selected pattern. In order to further improve the performance, we have proposed a rearrangement method for the residual errors that will reduce the required block size for coding.

Finally, a foreground/background video coding system has been developed that combines our research work on human face detection, facial feature extraction, facial feature tracking, and the video coding algorithm based on patterns. Firstly, the face and the respective facial features in an image are segmented. The facial features are tracked frame by frame, and then the detected location of the feature points is passed to the encoding system. In the encoder, more bits will be allocated to the face region and the facial feature regions, while fewer bits are used for the background scene. As a result, the perceptual picture quality can be improved.

7.2 Future Work

In the previous chapters, we introduced various approaches to video coding. The demand for digital video communication applications such as video conferencing, videophone, and high-definition television has increased considerably. We found that the 2D/3D hybrid video coding scheme can achieve a better performance for both objective and subjective picture qualities in these applications. Hence, in future work, we will focus on the 2D/3D hybrid video coding scheme.

For a 2D/3D hybrid video coder in videophone applications, a 3D head-and-shoulders model for the synthesis of the face image is necessary. The synthesis process relies on the accuracy of the face and the feature points extracted. Therefore, both a fast and robust face detection and facial feature extraction method, and a less complex face and facial feature tracking method are necessary. Since we have developed a face detection algorithm, a facial feature tracking method, and a block-based video coding system, the next step will be to develop a 2D/3D hybrid video coder and to devise a 3D head-and-shoulders model for synthesising the motion of a face as well as the facial expressions. Finally, we will combine all the developed modules to form an efficient 2D/3D hybrid video coder.

Appendix I

A face region under shirring effect is illustrated in Fig. I(a), where θ and ϕ are the angle of rotation and the shirring angle, respectively. Rotating the region about the point O by an angle θ, the rotation transformation is as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{1}$$

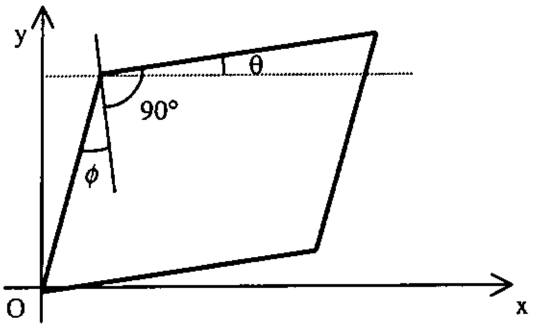


Fig. I(a)

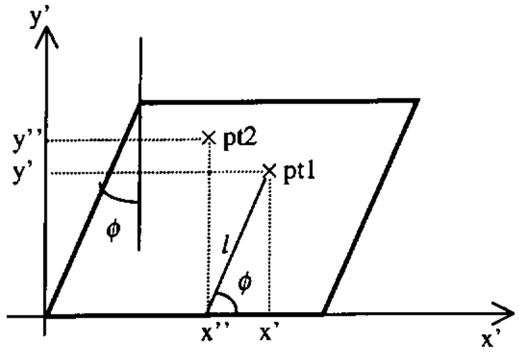


Fig. I(b)

The rotated region is illustrated in Fig. I(b). Let pt1 be the point before normalization and pt2 be the point after shirring normalization, then

$$\frac{x'-x''}{y'} = \tan\phi \quad \Rightarrow x'' = x' - y'\tan\phi$$

$$(y'')^2 = l^2 = y'^2 + (x'-x'')^2$$

$$(y'')^2 = (y')^2 + (x'-(x'-y'\tan\phi))^2$$

$$\Rightarrow y'' = y'\sec\phi$$

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} 1 & -\tan\phi \\ 0 & \sec\phi \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

From (1), we have

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} 1 & -\tan\phi \\ 0 & \sec\phi \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\vartheta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} \cos\theta + \tan\phi\sin\theta & \sin\theta - \tan\phi\cos\theta \\ -\sin\theta\sec\phi & \cos\theta\sec\phi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Author's Publications

1. Kwok-Wai Wong, Kin-Man Lam, and Wan-Chi Siu, "An Efficient Algorithm for Human Face Detection and Facial Feature Extraction under Different Conditions" *Accepted by Pattern Recognition.*

2. Kwok-Wai Wong, Kin-Man Lam, and Wan-Chi Siu, "An efficient low bit-rate video coding algorithm focusing on moving regions," *Submitted to IEEE Transactions on Circuits and Systems for Video Technology.*

3. Kwok-Wai Wong; Kin-Man Lam, "A reliable approach for human face detection using genetic algorithm," *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 499 –502, 1999.

4. Kwok-Wai Wong, Kin-Man Lam, and Wan-Chi Siu, "A very low bit-rate video coding algorithm by focusing on moving regions," *International Conference on Multimedia and Exposition*, 2000.

5. Kwok-Wai Wong, Kin-Man Lam, and Wan-Chi Siu, "A reliable approach for detecting human face under different conditions," *Accepted by the First IEEE Pacific-Rim Conference on Multimedia*, 2000.

References

1. ITU Telecom. Standardization Sector of ITU, "Video codec for audiovisual services at p×64 kbit/s," *ITU-T Recommendation H.261*, 1993.

2. ITU Telecom. Standardization Sector of ITU, "Video coding for low bitrate communication," *Draft ITU-T Recommendation H.263*, 1996.

3. ITU Telecom. Standardization Sector of ITU, "Video coding for low bitrate communication," *Draft ITU-T Recommendation H.263 Version 2*, 1998.

4. Generic Coding of Audio-Visual Objects: (MPEG-4 video), *Final Draft International Standard*, 1999.

5. Y. Yokoo and M. Hagiwara, "Human faces detection method using genetic algorithm," *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp.113-118, May 1996.

6. Y. Suzuki, H. Saito, and S. Ozawa, "Extraction of the human face from the natural background using GAs," *Proceedings of IEEE TENCON. Digital Signal Processing Applications*, vol. 1, pp.221-226, 1996.

7. Karin Sobottka and Ioannis Pitas, "A novel method for automatic face segmentation, facial feature extraction and tracking," *Signal Processing: Image Communication*, vol. 12, Issue 3, pp. 263-281, June 1998.

8. G. Yang and T.S. Huang, "Human face detection in a complex background," *Pattern Recognition*, vol. 27, no. 1, pp. 53-63, 1994.

9. K.M Lam, A fast approach for detecting human faces in a complex background, *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol.4, pp. 85 -88, 1998.

10. Kah-kay Sung and Tomaso Poggio, "Example-based learning for view-based human face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no.1, Jan. 1998.

11. K. Sobottka, and I. Pitas, "Segmentation and tracking of faces in color images," *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pp. 236 –241, 1996.

12. Hualu Wang and Shih-Fu Chang, "A highly efficient system for automatic face region detection in MPEG video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 4, pp. 615 –628, Aug. 1997.

13. Q.B. Sun, W.M. Huang, and J.K. Wu, "Face detection based on color and local symmetry information," *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition*, pp.130 –135, 1998.

14. D. Chai, and K.N. Ngan, "Locating facial region of a head-and-shoulders color image," *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 124 –129, 1998.

15. A.M. Alattar, S.A. Rajala, "Facial features localization in front view head and shoulders images," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, pp. 3557 –3560, 1999.

16. S.H. Jeng, H. Y. M. Liao, C.C. Han, M.Y. Chern, and Y.T. Liu, "Facial feature detection using geometrical face model: an efficient approach," *Pattern Recognition*, Vol. 31, no. 3, pp. 273-282, 1998.

17. A. Al-Qayedi, A.F. Clark, "An algorithm for face and facial-feature location based on grey-scale information and facial gemomerty," *Seventh International Conference on Image Processing and Its Applications*, vol. 2, pp. 625 –629, 1999.

18. Wu Haiyuan, Chen Qian, and M.Yachida, "Facial feature extraction and face verification," *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 3, pp. 484–488, 1996.

19. Fu-Che Wu, Tzong-Jer Yang, and Ming Ouhyoung, "Automatic feature extraction and face synthesis in facial image coding," *Sixth Pacific Conference on Computer Graphics and Applications*, pp. 218 –219, 1998.

20. Liang Zhang, "Automatic adaptation of a face model using action units for semantic coding of videophone sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 6 , pp. 781 –795, 1998.

21. A.L. Yuille, P.W. Hallinan, and D.S. Cohen, "Feature extraction from faces using deformable templates," *International Journal of Computer Vision*, pp.99-111, 1992.

22. M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neurosci.*, vol. 3, no.1, pp.71-86, 1991.

23. Kin-Man Lam, and Hong Yan, "An analytic to holistic approach for face recognition based on a single frontal view," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, July 1998.

24. Maxim A. Grudin, "On internal representations in face recognition systems," *Pattern Recognition*, vol. 33, pp. 1161-1177, July 2000.

25. D. Chai, K.N. Ngan, "Face segmentation using Skin-color Map in Videophone Applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 551 –564, June 1999.

26. D. Chai, K.N. Ngan, "Foreground/background video coding scheme ," *Proceedings of IEEE International Symposium on Circuits and Systems*, vol.2, pp. 1448 –1451, 1997.

27. R. Forchheimer and T. Kronander, "Image coding – from waveforms to animation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 12, pp. 2008-2023, Dec. 1989.

28. W.J. Welsh, S. Searby, and J.B. Waite, "Model based image coding," *J. Br. Telecom. Tech.*, vol. 8, no. 3, pp. 94-106, Jul. 1990.

29. K. AIZAWA and H.HARASHIMA, "Model-based analysis synthesis image coding (MBASIC) system for a person's face," *Signal Processing: Image Communication*, vol. 1, pp. 139-152, 1989.

30. Li-An Tang and T.S.Huang, "Automatic construction of 3D human face models based on 2D images," *Proceedings of International Conference on Image Processing*, vol.3, pp.467-470, 1996.

31. Soo-Chang Pei, Ching-Wen Ko, and Ming-Shing Su, "Global motion estimation in model based image coding by tracking three-dimensional contour feature points," IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, no. 2, pp. 181 –190, April 1998.

32. A.Murat Tekalp, "Digital video processing," *Prentice Hall PTR*, 1995.

33. K.R. Rao, and J.J. Hwang, "Techniques and standards for image, video, and audio coding," *Prentice Hall PTR*, 1996.

34. Borko Furht, Joshua Greenberg and Raymond Westwater, "Motion estimation algorithms for video compression," *Kluwer Academic Publishers*, 1997.

35. Borko Furht, Stephen W. Smoliar, and HongJiang Zhang, "Video and image processing in multimedia systems," *Kluwer Academic Publishers*, 1995.

36. Li Reoxiang, Zeng Bing; M.L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 438 -442 Aug. 1994.

37. Lai-Man Po, and Wing-Chung Ma, "A novel four step search algorithm for fast block motion estimation" *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 313 –317, June 1996.

38. Lai-Man Po, and Chok-Kwan Cheung, "A new center-biased orthogonal search algorithm for fast block motion estimation," *Proceedings of IEEE TENCON: Digital Signal Processing Applications*, vol. 2, pp. 874 –877, 1996.

39. Jae-Yong Kiam, and Sung-Bong Yang, "An efficient hybrid search algorithm for fast block matching in video coding," *Proceedings of the IEEE TENCON*, vol. 1, pp. 112–115, 1999.

40. Chok-Kwon Cheung, and Lai-Man Po, "Hybrid search algorithm for block motion estimation," Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, vol.4, pp. 297 –300, 1998.

41. David E. Goldberg, "Genetic algorithms in search, optimization, and machine learning," *Addision-Wesley*, 1989.

42. Chun-Hung Lin and Ja-Ling Wu, "Genetic block matching algorithm for video coding," *Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems*, pp. 544-547, 1996.

43. Chun-Hung Lin and Ja-Ling Wu, "A lightweight genetic block-matching algorithm for video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 4, Feb. 1998.

44. Yui-Lam Chan and Wan-Chi Siu, "New adaptive pixel decimation for block motion vector estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 1, Feb. 1996.

45. K. Rijkse, "H.263: video coding for low-bit-rate communication," *IEEE Communications Magazine*, vol. 34, no. 12, pp. 42 -45, Dec. 1996.

46. A. Murat Tekalp, "Digital video processing," *Prentice Hall PTR*, 1995.

47. Arun N. Netravali and Barry G. Haskell, "Digital pictures: representation, compression, and standards," *New York: Plenum Press*, 1994.

48. Keith Jack, "Video demystified: a handbook for the digital engineer," *San Diego, Calif.: HighText Publications*, 1996.

49. weidong kou, "Digital image compression : algorithms and standards," *Boston: Kluwer Academic Publishers*, 1995.

50. Barrt G. Haskell, Paul G. Howard, Yann A. LeCun, Atul Puri, Jöern Ostermann, M. Reha Vivanlar, Lawrence Rabiner, Leon Bottou, and Patrick Haffner, "Image and video coding — emerging standards and beyond," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 7, Nov. 1998.

51. G. Ashraf and M.N. Chong, "Performance analysis of H.261 and H.263 video coding algorithms," *ISCE '97, Proceedings of 1997 IEEE International Symposium on Consumer Electronics*, pp.153 -156, 1997.

52. G.Cote, B.Erol, M.Gallant and F. Kossentini, "H.263+: Video coding at low bit rates," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 849 -866, Nov. 1998.

53. B.Erol, M.Gallant, G.Cote, and F.Kossentini, "The H.263+ video coding standard: complexity and performance," *Proceedings of Data Compression Conference*, pp. 259–268, 1998.

54. Joo-Hee Moon, Ji-Heon Kweon, and Hae-Kwang Kim, "Boundary block-merging (BBM) technique for efficient texture coding of arbitrarily shaped object," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, Feb. 1999.

55. Jong-Won Yi, Soon-Jae Cho, Wook-Joong Kim, Seong-Dae Kim, and Sang-Jee Lee, "A new coding algorithm for arbitrarily shaped image segments," *Signal Processing: Image Communication* 12, pp. 231-242, 1998.

56. Takahiro Fukuhara, Kohtaro Asai, and Tokumichi Murakami, "Very low bit rate video coding with block partitioning and adaptive selection of two time-differential frame memories," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, Feb. 1997.

57. K. Ishikawa, and O. Nakamura, "Very low bit-rate video coding based on a method of facial area specification," *IEEE Canadian Conference on Electrical and Computer Engineering*, 1998.

58. Li Ding, and K. Takaya, "H.263 based facial image compression for low bitrate communications," *IEEE Conference Proceedings WESCANEX : Communications, Power and Computing*, pp. 30-34, 1997.

59. Peter Eisert, Thomas Wiegand, and Bernd Girod, "Model-aided coding: a new approach to incorporate facial animation into motion-compensated video coding," *IEEE Transactions on Circuit and Systems for Video technology*, vol. 10, no. 3, April 2000.

60. M.F. chowdhury, A.F. Clark, A.C. Downton, E. Morimatsu, and D.E. Pearson, "A switched model-based coder for video signals," *IEEE Transactions on Circuit and Systems for Video technology*, vol. 4, no. 3, June 1994.

61. J.C. Woods, S. Ramanan, and D.E. Peatson, "Selective macroblock level switching for model-based coded video," *Electronics Letters*, vol. 35, no. 5, March 1999.

62. J.C. Woods, S. Ramanan, and D.E. Peatson, "Low level switched model-based coder for low bit-rate applications," *Seventh International Conference on Image Processing and Its Applications*, vol.2, pp. 605 –609, 1999.

63. Chul Ryu, and Seung P. Kim, "Rate control in video coding by adaptive mode selection," *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 186 –189, 1998.

64. ITU Telecom. Standardization Sector of ITU, "Video Codec Test Model, Near-Term, Version 8 (TMN8)," *Video Coding Experts Group*, June 1997.

65. K. Oehler, J.L.H Webb, " Macroblock quantizer selection for H.263 video coding," *Proceedings of International Conference on Image Processing*, vol.1, pp.365 –368, 1997.

66. Yoonho Kim and Hansoo Kim, "A new buffer control strategy for image data compression," *IEEE Transactions on Consumer Electronics*, vol. 40, no. 4, Nov. 1994.

67. Nam Ik Cho, Heesub Lee, and Sang Uk Lee, "An adaptive quantization algorithm for video coding," IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 4, pp. 527 –535, June 1999.

68. Hwang-Seok Oh and Heung-Kyu Lee, "Adaptive rate control scheme for very low bit rate video coding," *IEEE Transactions on Consumer Electronics*, vol. 42, no. 4, Nov. 1996.

69. "Coded representation of picture and audio information -- MPEG2 Test Model 5," ISO/JTC1/SC29/WG11/N0400, Apr. 1993.

70. Daniel L. Swets, Bill Punch, and John Weng, "Genetic algorithms for object recognition in a complex scene," *Proceedings of International Conference on Image Processing*, vol.2, pp.595 –598, 1995.

71. C.H. Lin and J.L. Wu, "Automatic facial feature extraction by genetic algorithms," *IEEE Transactions on Image Processing*, vol. 8, no. 6, 1999.

72. P. Maragos, "Tutorial on advances in morphological image processing and analysis," *Optical Engineering*, vol. 26, no. 7, pp. 623-632, 1987.

73. P. Jonathon Phillips and Yehuda Vardi, "Efficient illumination normalization of facial images," *Pattern Recognition Letters 17*, 921-927, 1996.

74. Rafael C. Gonzalez, Richard E. Woods, "Digital image processing," *Mass. Reading: Addison-Wesley*, 1992.

75. Yui-Lam Chan and Wan-Chi Siu, "Block Motion Vector Estimation using Edge Matching : An Approach with Better Frame Quality as Compared to Full Search Algorithm," *IEEE International Symposium on Circuits and Systems*, vol.2, pp.1145–1148, 1997.

76. Paul M. Antoszczyszyn, John M. Hannah, and Peter M. Grant, "Primciple components analysis for tracking of facial features," *First International Workshop on Wireless Image/Video Communications*, pp. 32 –37, 1996.

77. P.M. Antoszczyszyn, J.M. Hannah, and Peter M. Grant, "Reliable tracking of facial features in semantic-based video coding," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 145, no. 4, pp. 257 –263, Aug. 1998.