ABSTRACT

MOTION COMPENSATION AND VERY LOW BIT RATE VIDEO CODING

by
Shu Lin

Recently, many activities of the International Telecommunication Union (ITU) and the International Standard Organization (ISO) are leading to define new standards for very low bit-rate video coding, such as H.263 and MPEG-4 after successful applications of the international standards H.261 and MPEG-1/2 for video coding above 64kbps. However, at very low bit-rate the classic block matching based DCT video coding scheme suffers seriously from blocking artifacts which degrade the quality of reconstructed video frames considerably. To solve this problem, a new technique in which motion compensation is based on dense motion field is presented in this dissertation.

Four efficient new video coding algorithms based on this new technique for very low bit-rate are proposed. (1) After studying model-based video coding algorithms, we propose an optical flow based video coding algorithm with thresholding techniques. A statistic model is established for distribution of intensity difference between two successive frames, and four thresholds are used to control the bit-rate and the quality of reconstructed frames. It outperforms the typical model-based techniques in terms of complexity and quality of reconstructed frames. (2) An efficient algorithm using DCT coded optical flow. It is found that dense motion fields can be modeled as the first order auto-regressive model, and efficiently compressed with DCT technique, hence achieving very low bit-rate and higher visual quality than the H.263/TMN5. (3) A region-based discrete wavelet transform video coding algorithm. This algorithm implements dense motion field and regions are segmented according to their content significance. The DWT is applied to residual

images region by region, and bits are adaptively allocated to regions. It improves the visual quality and PSNR of significant regions while maintaining low bit-rate. (4) A segmentation-based video coding algorithm for stereo sequence. A correlation-feedback algorithm with Kalman filter is utilized to improve the accuracy of optical flow fields. Three criteria, which are associated with 3-D information, 2-D connectivity and motion vector fields, respectively, are defined for object segmentation. A chain code is utilized to code the shapes of the segmented objects. It can achieve very high compression ratio up to several thousands.

MOTION COMPENSATION AND VERY LOW BIT RATE VIDEO CODING

by
Shu Lin

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Department of Electrical and Computer Engineering

May 1997

APPROVAL PAGE

MOTION COMPENSATION AND VERY LOW BIT RATE VIDEO CODING

Shu Lin

Dr. Yun-Qing Shi, Dissertation Advisor — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Joseph Frank, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Edwin Hou, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Douglas Hung, Committee Member — Date
Associate Professor of Computer and Information Science, NJIT

Dr. Zoran Siveski, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Shu Lin

Degree: Doctor of Philosophy

Date: May 1997

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical and Computer Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1997
- Master of Science in Electrical and Computer Engineering,
  University of Rhode Island, Kingston, RI, 1994
- Master of Science in Electrical Engineering,
  Zhejiang University, Hongzhou, Zhejiang, China, 1991
- Bachelor of Science in Electrical Engineering,
  Zhejiang University, Hongzhou, Zhejiang, China, 1986

Major: Electrical and Computer Engineering

Presentations and Publications:

Shu Lin, and Y. Q. Shi,
" A new Approach to Video Teleconferencing Coding - Another Look at Wireframe Model in Facial Coding", *Proceedings of the Ninth Workshop on Image and Multidimensional Signal Processing*, March 3-6, 1996, Belize City, Belize, IMDSP Belize, 96. pp. 172-173.

Shu Lin, Yun Q. Shi, and Ya-Qin Zhang,
"An Optical Flow Based Motion Compensation Algorithm for Very Low Bit-Rate Video Coding," *IEEE ICASSP97*, Groebenzell, Germany, 1997 (Accepted).

Shu Lin, and Yun Q. Shi,
"Region-Based Adaptive DWT Video Coding Using Dense Motion Field," *The First IEEE Signal Processing Society Workshop on Multimedia Signal Processing*, June 23-25, 1997, Princeton, New Jersey, (Submitted).

Yun Q. Shi, Shu Lin, and Ya-Qin Zhang,

"An Optical Flow Based Motion Compensation Algorithm for Very Low Bit-Rate Video Coding," (Preparing for journal paper)

Jinging Pan, Yun Q. Shi, and Shu Lin,
" A Kalman Filter for Improving Optical Flow Accuracy on Moving Boundaries", (submitted to IEEE trans. on Image Processing).

Shu Lin,
"Pre-fix Coding Generation for Noisy Communication Channels", Master Project, University of Rhode Island, 1994.

Shu Lin,
"Micro-Processor MCS-51 Based Solar Tracking System", Master thesis, Zhejiang University, 1991.

This work is dedicated to
my wife, Yongmei

ACKNOWLEDGMENT

The author wishes to express his sincere gratitude to his advisor, Dr. Yun-Qing Shi, for his guidance, friendship, and moral support throughout this research.

Special thanks to Professors, Joseph Frank, Edwin Hou, Douglas Hung and Zoran Siveski, for serving as members of the committee.

The author is grateful to the NJIT for their supporting his Ph.D education.

Finally, a thank to Jun Li for his help.

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

Since the International Telegraph and Telephone Consultative Committee (CCITT), started its standardization-related activities in video coding in 1984, many efforts have been made to develop audiovisual coding techniques for various applications such as video-telephony, digital video disc, digital compact cassette [24], digital satellite systems (DSS), advanced television (ATV) [2][6], high definition television (HDTV) [25], teleconferencing, CD-ROM storage, digital broadcasting. These efforts have led to national or international standards or recommendations. Some are currently being used widely, such as International Standard Organization (ISO) MPEG-1/2 and ITU-T (International Telecommunication Union, formerly CCITT) H.261. They work very well at the bit rates above 64kbps. Current activities are trying to develop video coding at very low bit rate (below 64kbps), which is expected to be applied to a number of services, like multimedia, mobile personal communications, videophone through existing public switched telephone networks (PSTN). The efforts are leading to new standards such as ITU-T H.263 and ISO MPEG-4. The basic techniques which are used in the previous standards, like MPEG-1/2, H.261, are the block matching based motion compensation and discrete cosine transform (DCT). These existing techniques are becoming mature, but they may not suit well for very low bit rate video coding [18][3]. To solve this problem, the second generation coding schemes [39][40], such as object-oriented [4], region-based, contour-based [54], model-based [3][45], fractal-based, syntax-based [65], and many others have been proposed, but most of them have not been yet mature enough to be used in the very low bit rate video coding. The ITU-T H.263 [32], which is on its way, still uses the classic block matching based DCT technique as its basic coding scheme. The unavoidable blocking artifacts are the intrinsic problem resulting

from the block matching model, and degrade the quality of the reconstructed video frames significantly. In this dissertation four new algorithms are proposed. They are: (1) an optical flow based video coding algorithm with thresholding techniques; (2) an efficient algorithm using DCT coded optical flow; (3) a region-based discrete wavelet transform video coding algorithm; and (4) a segmentation-based video coding algorithm for stereo sequence. All of them are expected to achieve very low bit rate as well as good quality of the reconstructed image frames by drastically reducing blocking artifacts.

1.1 Video Coding Standards

In 1984, the CCITT issued Recommendation H.120 targeted for videoconferencing applications, specifically, for 625/50 and 525/60 TV systems at bit rates of 2.048 Mbps and 1.544 Mbps respectively. In 1989, another standard draft, H.261, was available. It is for the bit rate p x 64 kbps, where p=1,2, ..., 30. And in 1988, the Moving Picture Experts Group (MPEG) was founded under ISO/SC2 to standardize a video coding algorithm targeted for digital storage media at bit rate up to 1.5 Mbps. MPEG-1 was issued in 1992 that is intended to be generic. The quality of MPEG-1 compressed video at 1.2 Mbps are not acceptable for some applications. To extend MPEG-1, MPEG-2 was started in 1990. At the beginning, it is for coding of TV-picture with International Consultative Committee on Broadcasting (CCIR) Rec.601 resolution at bit rate below 10 Mbps. And later it was extended for coding of HDTV up to 20Mbps in 1992, and released in early 1994. Because of the extension of MPEG-2, MPEG-3, which was targeted for HDTV, is no longer needed and died off. MPEG-4 was initiated in 1993 and its target bit rate is below 64 kbps. It is planned to be issued in November, 1998. Meanwhile, the ITU H.263/N (the near term) draft is available, and its bit rate could be as low as 28.8 kbps. H.263/L (the long term) is being accomplished in close collaboration with the ISO MPEG-4 activity.

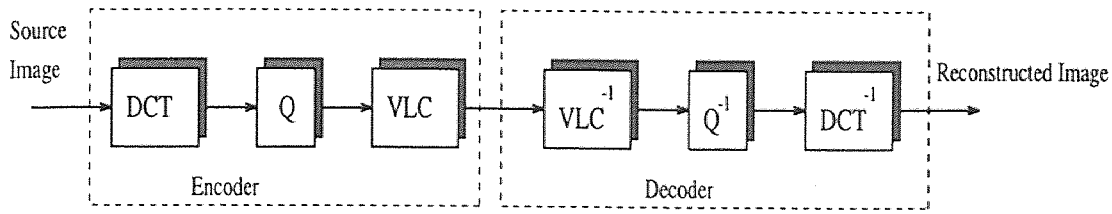


Figure 1.1 The JPEG codec

1.1.1 JPEG

For the sake of completeness, prior to video coding standards, we briefly introduce here the JPEG, an international standard for still image coding. The Joint Photographic Experts Group (JPEG) standard is targeted for full-color still frame applications, achieving 15:1 average compression ratio [63][30]. It defines a family of compression algorithms for continuous-tone, still images [12][63]. It has four modes of operation as follows:

- Sequential DCT-based encoding, in which each component is encoded in single left-to-right, top-to-bottom scan.
- Progressive DCT-based encoding, in which the image is encoded in multiple scans, in order to produce a quick, rough decoded image when the transmission time is long.
- Lossless encoding, in which the image is encoded to guarantee the exact reproduction.
- Hierarchical encoding, in which the image is encoded in multiple resolutions.

The JPEG baseline codec block diagram is given in Figure 1.1. It uses DCT techniques to do intraframe coding, and variable length to do entropy coding. The JPEG standard does not define the meaning or format of the components that comprise the image. Attributes like the color space and pixel aspect ratio must be specified out-of-band with respect to the JPEG bit stream. The JPEG standard

provides a rich set of algorithms for flexible compression. However, JPEG is also applied in some real-time, full-motion video applications by compressing each frame of video as an independent still image and transmitting them in series. Video coded in this fashion is often called Motion-JPEG (MJPEG). For details of the JPEG standard, please consult [30][74][63][41][42].

1.1.2 H.261

H.261 [31][49][8][33] has been developed for videophone and video conferencing, serving over the ISDN at bit rate $p \times 64$ kbps, $p = 1, 2, ..., 30$. It provides very high compression ratio for full-color, real-time motion video transmission with limited motion search and estimation strategies. The standard is intended to cover the entire ISDN channel capacity and for real-time communications allowing minimum delays (maximum coding delay is specified as 150 msec). For $p = 1, 2$, due to limited available bandwidth, only desktop face-to-face visual communications (or video phone) can be implemented using this compression algorithm. However, for $p \geq 6$, more complex pictures are transmitted and the standard is suitable for video-conferencing applications. To permit a single recommendation using 625- (PAL) and 525- (NTSC) line TV standards, the H.261 input picture format is specified as the so-called Common Intermediate Format (CIF). For lower-bit rate applications, a small format, QCIF, has been adopted. The video compression scheme chosen for H.261 standard has two modes: the intra and inter modes. The intra mode, based on block-by-block DCT coding, is coded using information only found in the picture itself. I-frame provides potential random access points into the compressed video data. In the inter mode, first a temporal prediction is employed with or without motion compensation (MC), then the interframe prediction error is DCT encoded. The block diagram of the encoder and decoder is illustrated in Figure 1.2. The algorithm begins by coding an intraframe block using the DCT transform coding and

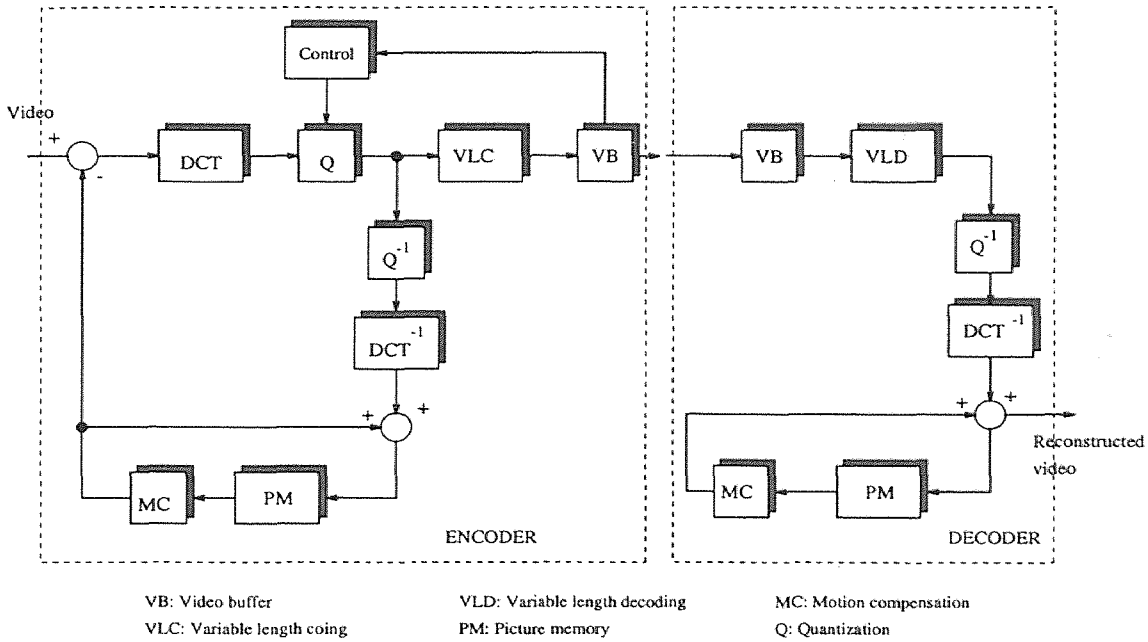


Figure 1.2 H.261 codec

quantization, and then sending it to the video multiplex coder. The same frames are then decompressed using the inverse quantizer and IDCT, and then stored in the picture memory for interframe coding. During the interframe coding, the prediction based on the DPCM algorithm is used to compare every macro block of the actual frame with the available macro blocks of the previous frame. Then, the difference is created, DCT-coded, quantized, and sent to the video multiplex coder. Finally, entropy coding is used to produce more compact code.

The H.261 video codec is very efficient for carrying visual services at $p \times 64$ kbps ISDN networks with constant bit rate transmission. A new H.261 codec proposed in [21] expands the existing H.261 codec to operate in ATM networks. A software based video compression algorithm, called the popular video codec (PVC) proposed in [26], is suitable for real-time systems.

1.1.3 MPEG

The MPEG video compression algorithm is intended for compression of full motion video. The compression method uses interframe compression and can achieve high

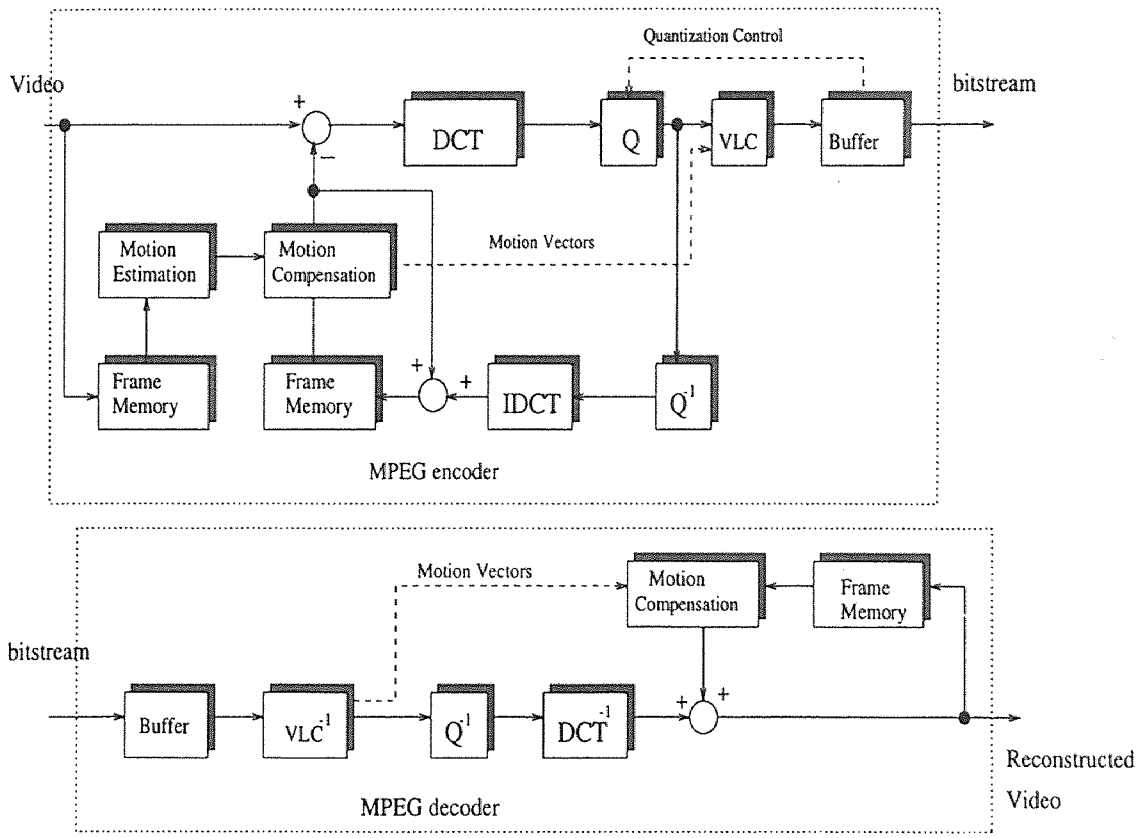


Figure 1.3 The MPEG codec

compression ratios. The MPEG codec block diagram is given in Figure 1.3. It uses motion compensation, DCT and variable length coding techniques. Bit rate control is used to avoid the buffer overflow or underflow and provide guarantee video quality. If the buffer is overflowed, then the quantization step will be increased, so that fewer bits are used to code the next macroblock, consequently, the bit rate is reduced. If the buffer is underflowed, a finer quantization step is used to provide better quality.

MPEG-1 MPEG-1 (ISO/IEC 1117) has been developed for storage of CIF format video and its associated audio at the bit-rate from 1 Mbps to 1.5 Mbps for multimedia systems. MPEG-1 has three parts: (1) Audio (ISO/IEC 1117-1), (2) Video (ISO/IEC 1117-2), (3) System (ISO/IEC 1117-3). MEPG-1 is a generic standard in that it standardizes a syntax for the representation of the encoded bitstream and a method of decoding. The syntax supports operations such as motion estimation, motion

compensation, prediction, discrete cosine transformation, quantization, and variable length coding. MPEG-1 does not define specific algorithms needed to produce a valid data stream, and substantial flexibility is allowed in designing the encoder. A number of parameters, defining the coded bitstream and decoders, are contained in the bitstream itself. This allows the algorithm to be used with pictures of a variety of sizes and aspect ratios and on channels or devices operating at a wide range of bit-rate. MPEG-1 offers:

- Random access to any video storage application at independent access points (I-frame).
- Fast forward/reverse searching, which refers to scanning the compressed bit stream and to display only selected frames to obtain fast forward or reverse search.
- Reasonable coding/decoding delay of about one second, and which provides the impression of interactivity in unidirectional video access.

The maximum decoder buffer size is specified as 376,832 bits. There are two types of interframe encoded pictures, P- and B-pictures. In these pictures the motion-compensated prediction errors are DCT encoded. Only forward prediction is used in the P-pictures, which are always encoded relative to the preceding I- or P-pictures. The prediction of the B-pictures can be forward, backward, or bidirectional relative to other I- or P-pictures. D-pictures contain only the DC component of each block and serve for browing purposes at very low bit-rate.

Generally, the I- and P-pictures are only one third of all frames. The remaining frames can be interpolated from the reconstructed I and P frames. The B-pictures are not used in predicting any future pictures to avoid error propagation. One-half pixel accuracy is allowed for motion estimations. Because of using B-pictures, the frame display order and the transmission order are different (refer to Figure 1.4).

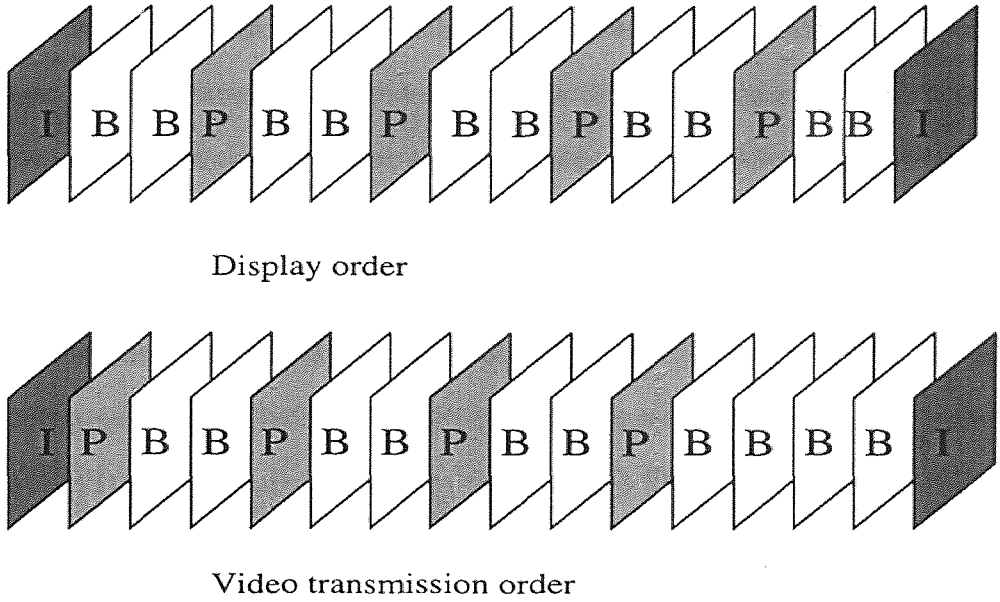


Figure 1.4 The different orders between display and transmission

Motion compensation is a technique for enhancing the compression of P and B frames by eliminating temporal redundancy.

MPEG-2 MPEG-2 standard (ISO/IEC 13818) is an audio/video compression algorithm optimized for broadcasting quality transmissions up to HDTV quality based on the motion compensation and the discrete cosine transform. It defines higher levels (for HDTV) and higher profiles and a multiplexing system, which allows to combine many video, audio, and data streams into one single data stream. It supports both of progressive and interlaced video formats and a number of features for HDTV. The MPEG-2 standard addresses scalable video coding for variety of applications that need different image resolutions, such as video communications over ISDN and ATM networks [14]; allows temporal scalability so that one stream can be displayed at different frame rates; allows for interlaced inputs, higher-definition inputs, and alternative subsampling of the chroma channels; provides improved quantization and coding options, higher bit rates, surround sound, and alternate language channels; allows different scan patterns than the zigzag.

MPEG-2 video syntax provides an efficient way to represent image sequences in the form of more compact coded data. The display picture size and frame rate may differ from the encoded frame size and rate. Hence, downsample or unsample may be applied to the reconstructed sequence according to the source rate, coded rate and display rate. MPEG-2 volume consists of a total of 9 parts[1] under ISO/IEC 13818. The second part is jointly developed with the ITU-T, where it is known as ITU-T recommendation H.262. MPEG-2 can be used in CATV, direct broadcast satellite (DBS), HDTV, digital video tape (DVT), DVD, high density CD, video conferencing and digital camcorder, etc.

MPEG-4 The MPEG committee is currently developing MPEG-4 with wide industry participation. MPEG-4 is to provide an audiovisual coding standard allowing for interactivity, high compression, and/or universal accessibility with high degree of flexibility and extensibility for emergence of the enlarged intersection of telecommunication, TV/film entertainment, and computer industry. This standard is intended for compression of full motion video consisting of small frames and requiring slow refreshments. The bit rate is from 9 kbps to 40 kbps.

MPEG-4 combines some of the typical features of other MPEG standards with new ones coming from existing or anticipated manifestations of multimedia:

1. Independence of applications from lower-layer details, as in internet paradigm;

2. Technology awareness of lower layer characteristics (scalability, error robustness etc.)

3. Application software downloadability, as in Java and the network computer paradigm;

[1]MPEG-2: System, video, audio, conformance, software, digital storage medium command and control, non-backward compatible audio, 10-bit video extension, and real-time interface

4. Reusability of encoding tools and data;

5. Possibility to hyperlink and interact with multiple sources information simultaneously as in the Web paradigm;

6. Capability to handle natural/synthetic and real-time/non-real-time information in an integrated fashion;

7. Capability to composite and present information according to user's needs and computer graphics paradigm in general.

At present time, MPEG-4 is structured in terms of four different elements: syntax, tools, algorithms and profiles, using a similar methodology to that of MPEG-2. However, there is a significant difference between MPEG-2 and MPEG-4. In MPEG-2 all profiles are closely related and higher profiles superset lower profiles, while MPEG-4 may have a completely independent and exclusive profile for a specific application since MPEG-4 intends to address so broad applications which may have little in common.

The MPEG-4 applications can be clustered into audiovisual database, audio-visual communications and messaging, and remote monitoring and control. The features of the applications of MPEG-4 are random access, fast forward/reverse searches, reverse playback, audio-visual synchronization, robustness to errors, coding/decoding delay, editability, and format flexibility.

1.1.4 H.263

The H.263/N[32] (H.263 includes H.263 near term, H.263 plus, H.263 long term) is a video coding standard (draft) which specifies the I/O interface, video format, coding algorithms, and bit stream syntax and decoding semantics for narrow telecommunication channels. It will use T.120 as the data interface protocol and V.34[2] as

[2]The V.34 modem recommendation, also known as V.fast, employs multiple modulation methods and multiple impairment compensation techniques. The modem is defined to

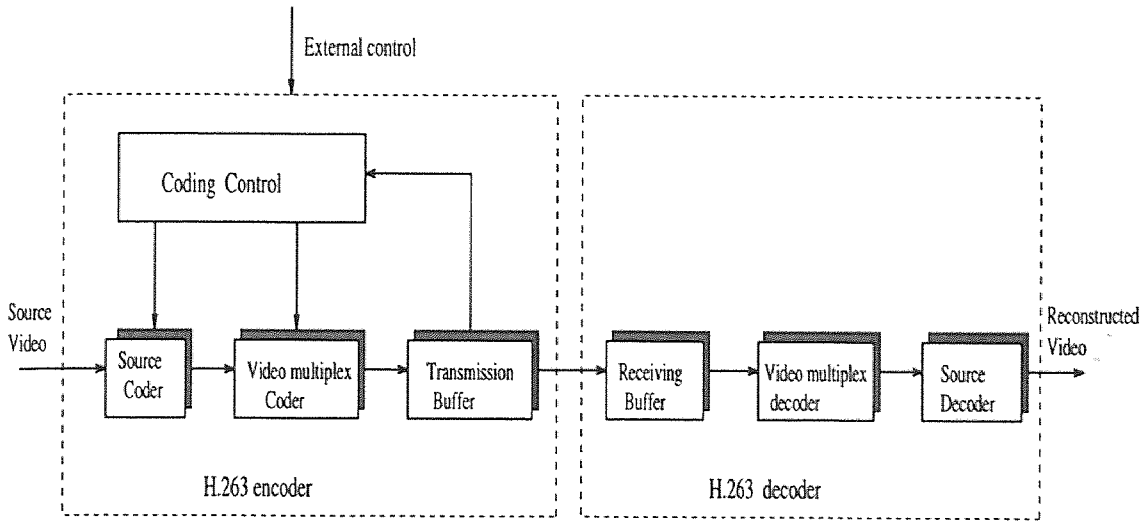


Figure 1.5 The H.263 codec

the modem for transmission. H.263/N uses block-matching motion-compensated prediction, adaptive intra/inter decision at macroblock level, DCT, quantization and entropy coding. The block diagram of H.263 codec is shown in Figure 1.5. The coding control is used to avoid the transmission buffer overflow or underflow.

The main improvements in H.263/N compared with H.261 are as follows:

1. Half pixel motion prediction for higher accuracy, which also eliminates the need for loop filtering.

2. Possible motion search outside of the picture boundary.

3. Possible overlapped motion compensation to obtain a smoother motion field at the expense of computational complexity.

4. Possible incorporation of syntax-based adaptive arithmetic coder as the entropy coder for improved efficiency.

5. Possible use of PB frames to improve the motion prediction efficiency.

be able to automatically and intelligently choose to combine the optimum set of these modulation tools to adapt to any given telephone channel. It defines a 2 wire, full duplex dial and lease line modem supporting both synchronous and asynchronous operations. Bit rate is 28.8 kbps [80].

6. Other optimization for very low bit rate with QCIF and sub-QCIF resolutions and low frame rate (below 10 frames/second).

1.2 Block Matching (BM)

Block matching is an efficient motion estimation method in small displacement between successive frames [20][29]. It is a very simple motion model, which is based on the assumption that all pixels within the block move with a unique motion velocity. The traditional BM treats a block independently. To estimate the displacement by block matching between two successive image frames, one can do the following,

1. Locate the block whose displacement needs to be calculated. Suppose it centers at position (i,j) in frame 2 and its size is m by m pixels. Refer to Figure 1.6.

2. Open a search window with size NxN centered at (i,j). N=m+2d, where d is the largest possible displacement. Refer to Figure 1.6.

3. Define a function as a correlation measure. One such type of functions, named the mean absolute frame difference (MAD) is defined as follows,

$$MAD(x,y) = \frac{1}{m^2} \sum_{p=-\frac{m-1}{2}}^{\frac{m-1}{2}} \sum_{q=-\frac{m-1}{2}}^{\frac{m-1}{2}} \mid I_{k+1}(i+p, j+p) - I_k(x+i+p, y+j+q) \mid \quad (1.1)$$

   where m is odd. Due to its simplicity, it has been used frequently. The function also can be defined with mean square error (MSE) or normalized two-dimensional cross-correlation function (NCCP) instead of MAD.

4. To calculate the most likely displacement, shift the block defined in Procedure 1 to every possible position in the search window to find the best match. Calculate all of the possible correlation measures, and the minimum one indicates a best match. The displacement between the best match position and the original position is the motion vector of this block.

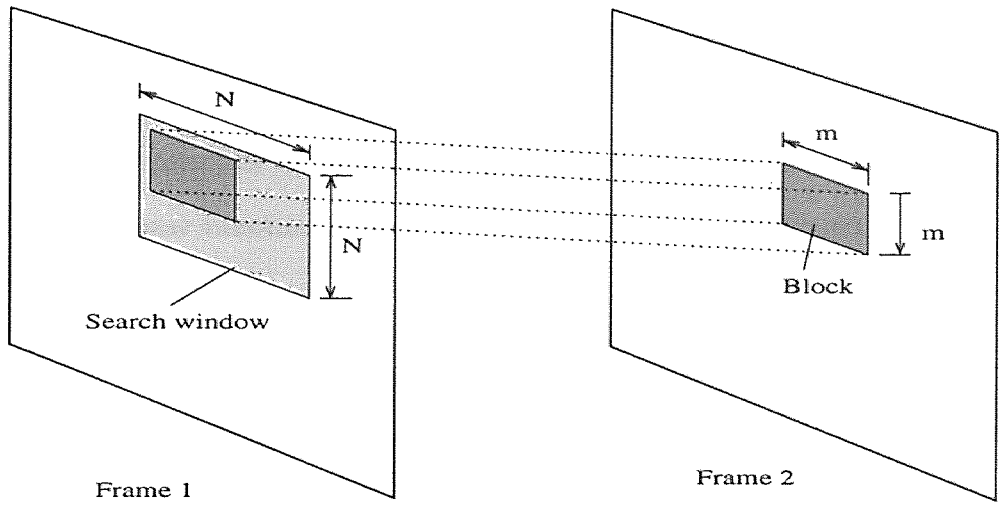


Figure 1.6 Block matching

In the case of small displacements, block matching technique works very efficiently. However when the displacement becomes large, the number of possible shifts increase quadratically. That is, the number of shifts, denoted by NS, is:

$$NS = 4d^2 + 4dm \tag{1.2}$$

To solve this problem, some fast algorithms are investigated, such as 2-D logarithmic search [34] three step search [37], modified conjugate direction search and multiresolution block matching. These algorithms usually save much computation with some assumptions. However, if these assumptions are not satisfied, distortion may take place.

The disadvantage of block matching model comes from its assumption that all pixels within a block have the same translation motion. It results in inaccuracy of motion estimation. Blocking artifacts are known as a major problem.

1.3 The Discrete Cosine Transform (DCT)

The DCT approaches the statistically optimal Karhunen-Loeve transform (KLT) for highly correlated signals, and it is widely used in digital signal processing, especially for video and image coding and speech compressions. Thus, like FFT,

many algorithms and VLSI architectures for the fast computation of DCT have been proposed. The energy compactness of DCT makes it very attractive in video coding. That is, a small number of transform coefficients can represent most of energies. The orthogonal base function of DCT is given as

$$\Phi(r,n) = \frac{1}{c_r}\cos\frac{(2n+1)r\pi}{2N}, \quad 0 \le r \le N-1 \tag{1.3}$$

where $c_r = \sqrt{N}$, when $r = 0$ and $c = \sqrt{N/2}$, when $r \neq 0$.

1.4 Motion Compensation Based Video Coding Scheme

Motion compensation is a major progress which has been made since 1980s for video coding. It drastically reduces temporal redundancies between successive frames and makes it possible that the video can be compressed at high ratio. A block diagram of a general motion compensated video coding scheme is shown in Figure 1.7. The pre-processing is to specify the image size, frame rate, sub or up sampling, and masking, etc. Motion vectors can be calculated with various motion field determination algorithms. The motion field is used to predict the next frame, i.e. motion compensation. The prediction error is obtained by calculating the difference between the original frame and the motion compensation frame. DCT is applied to each block of the predictive error image, and then a variable thresholding is applied to further decrease the number of nonzero DCT coefficients. After that, the DCT coefficients are quantized with reasonable quantization scale. And the quantized coefficients are zigzag scanned, and entropy coded. Then the layer headers are added to that to form the bitstream.

1.5 Very Low Bit-Rate Video Coding

Very low bit rate video coding means that the bit-rate in compression of visual portion is below 64kb/s. A large number of services need to transmit data, audio and

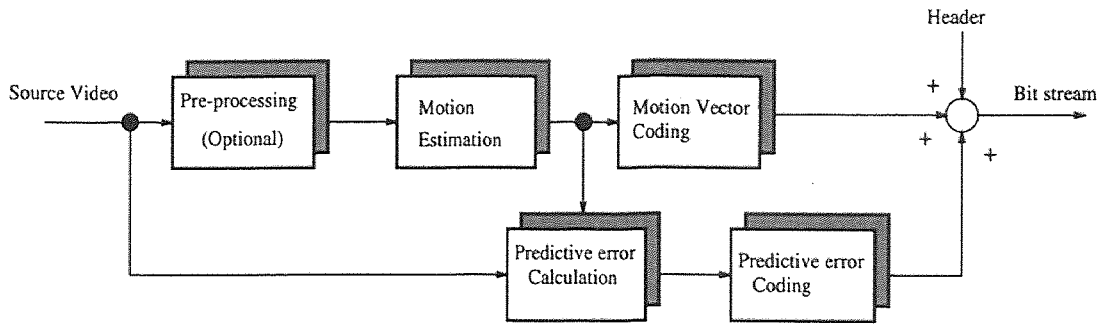


Figure 1.7 Diagram for motion compensated video coding scheme

video in a narrow bandwidth, such as mobile phone, videophone, personal computer which is connected to the traditional telephone line. All of these applications make the compression of data in very low bit rate very important. In this context, it is no doubt that the classic block and DCT based coding schemes have reached a level of saturation in performance. It cannot be improved further very much for very low bit rate applications [18], and the inherent blocking artifacts degrade the quality of the reconstructed images significantly. Many so-called second generation coding schemes have been proposed to solve this problem in the past few years, such as facial model-based, region-based, object-based, and contour, etc. These techniques are expected to be superior to the classic block-based coding in very low bit rate videó coding.

A key problem in high compression video coding is the operational control of the encoder. The nature of the encoder is generally left open to user specification. Ideally, the encoder should balance the quality of the decoded images with channel capacity. Most effective existing video coders utilize several modes of operation which are selected on a block-by-block basis. A given macroblock can be intra-frame coded, or inter-frame coded.

1.6 Organization of the Dissertation

In Chapter 2, three dense motion field determination algorithms are discussed. They are gradient-based, correlation-feedback and correlation-feedback with Kalman

filter. Some simulation and real image sequences have been tested to evaluate these techniques. In Chapter 3, a new algorithm that is optical flow based motion compensated very low bit rate video coding with thresholding techniques is proposed. A statistic model is investigated, based on which, four thresholds are established. The new algorithm is tested and its performance is compared with that of the existing model-based video coding schemes. In Chapter 4, a novel video coding technique is devised. Dense motion field is used for motion compensation, and DCT is applied to the highly correlated motion field. An AR(1) model is used to model optical flow field. Adaptive threshold is used to code the residual images. Two sequences are tested and the performance of the new algorithm is compared with the H.263. In Chapter 5, wavelet transform is discussed. A new region-based discrete wavelet transform technique is presented. Image is decomposed into regions according to their visual significance and bits are adaptively allocated to these regions. The comparison between this algorithm, the algorithm discussed in Chapter 4, and that in H.263 are given. In Chapter 6, a segmentation-based stereo sequence video coding scheme is proposed. The camera setting for stereo images is discussed. Three criteria associated with 3-D information, 2-D connectivity and motion vector fields are used to segment objects. A chain coding scheme is given to code the shapes of the segmented objects. Two stereo sequences are tested. Finally, a summary and directions for future research are given in Chapter 7.

CHAPTER 2
OPTICAL FLOW

Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image [28], and it plays a very important role in motion analysis and video coding. Image sequences consist of a sequential frames taken at different moments by a camera when the objects in the picture are moving. On the other hand, if the objects are not moving while the viewer (like camera) is moving, then at different moments, the objects are moving related to the viewer. The optical flow may reveal useful information not only in analysis of the related motion velocity between the objects and their viewer, but also in recovery of the arrangement of objects in space. There are many algorithms of computing optical flow available now. The techniques used in these algorithms are gradient-based, region-based, energy-based and phase-based. Several algorithms can work in real-time [28]. It is not the author's intention to give a very comprehensive survey about various optical flow techniques here. Only those used in this dissertation work are discussed in this chapter. Specifically, three techniques, gradient-based, correlation-feedback with or without Kalman filter, are discussed.

2.1 Gradient-Based Algorithm

The gradient-based algorithm is based on the assumption that the luminance intensity is invariable in successive frames during a short interval. That is

$$I(x,y,t) = I(x+\delta x, y+\delta y, t+\delta t) \tag{2.1}$$

where $I(x,y,t)$ expresses the intensity of a pixel at position (x,y) at moment t. Taylor series lead to

$$\begin{aligned} I(x+\delta x, y+\delta y, t+\delta t) &= I(x,y,t) + I_x\delta x + I_y\delta y + I_t\delta t... && (2.2) \\ &\approx I(x,y,t) + I_x\delta x + I_y\delta y + I_t\delta t && (2.3) \end{aligned}$$

where

$$u = \frac{\delta x}{\delta t} \tag{2.4}$$

$$v = \frac{\delta y}{\delta t} \tag{2.5}$$

$$I_x = \frac{\partial I}{\partial x} \tag{2.6}$$

$$I_y = \frac{\partial I}{\partial y} \tag{2.7}$$

$$I_t = \frac{\partial I}{\partial t} \tag{2.8}$$

Then we have

$$I_x u + I_y v + I_t = 0 \tag{2.9}$$

according to Equation 2.1 and 2.3. In Equation 2.9 there are two unknown u and v needed to be solved, but we only have one equation, and we need an additional equation. Therefore, another constraint has to be imposed. The smoothness constraint proposed by Horn and Schunck [28] is a constraint frequently used in optical flow determination. Horn and Schunck provided an algorithm derived from these assumptions by minimizing the error in optical flow

$$\int_D \varepsilon_b^2 + \lambda^2 \varepsilon_c^2 dx dy \tag{2.10}$$

where D is the integration domain, $\varepsilon_b = I_x u + I_y v + I_t$, $\varepsilon_c^2 = (\frac{\partial u}{\partial x})^2 + (\frac{\partial u}{\partial y})^2 + (\frac{\partial v}{\partial x})^2 + (\frac{\partial v}{\partial y})^2$, λ decides the strength of the weight of the smoothness term. To solve the above nonlinear equation, Horn and Schunck gave the following iterative solution,

$$u^{k+1} = \bar{u}^k - \frac{I_x(I_x \bar{u}^k + I_y \bar{v}^k + I_t)}{\lambda^2 + I_x^2 + I_y^2} \tag{2.11}$$

$$v^{k+1} = \bar{v}^k - \frac{I_y(I_x \bar{u}^k + I_y \bar{v}^k + I_t)}{\lambda^2 + I_x^2 + I_y^2} \tag{2.12}$$

where k denotes the iteration number, u^0, v^0 are the initial values, and $\bar{u}^k, \bar{v}^k$ are the neighborhood average of u^k and v^k. A presmoothness is used to avoid discontinuities at moving boundaries [10]. For propagation in optical flow determination, the number

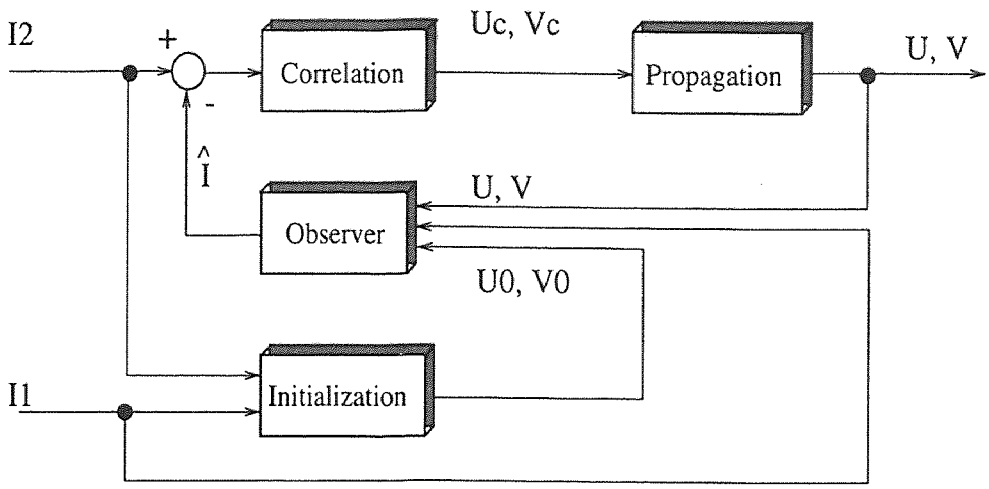


Figure 2.1 The block diagram of the correlation feedback

of iterations should be larger than the number of picture cells across the largest region that must be filled in. Therefore, this technique is suitable for small displacements with respect to great scale of the image intensity variations. But if the scale of the image intensity variation is very small and the displacements are not great enough, then the optical flow will hardly be obtained, refer to [36].

2.2 Correlation-Feedback Algorithm

The correlation-feedback technique is resulted from applying feedback technique to correlation-based algorithm such as Singh's [70]. A diagram of the correlation feedback algorithm is shown in Figure 2.1. It consists of four stages: initialization, observer, correlation and propagation [62].

The initialization stage is to obtain an initial optical flow fields (OFF) by some fast algorithms. The observer stage is to create a bilinear interpolation image with the initial OFF. The correlation stage is to open a search window, to find the best match with the sum-of-square-differences. The propagation stage is to use neighborhood information to improve the image velocities.

Using the sum-of-squared-difference offers several computational advantages over correlation. For each pixel $P(x, y)$ in the first image, a correlation-window w_p

of size $(2n+1) \times (2n+1)$ is formed around the pixel. A search window w_s of size $(2N+1) \times (2N+1)$ is then opened around the pixel at location (x,y) in the second image. The $(2N+1) \times (2N+1)$ sample of error-distribution is computed using sum-of-squared-differences as:

$$\varepsilon_c(u,v) = \sum_{i=-n}^{n} \sum_{j=-n}^{n} (I_1(x+i, y+j) - I_2(x+u+i, y+v+j))^2 \tag{2.13}$$

where $-N \leq u, v \leq N$. And the $(2N+1) \times (2N+1)$ samples of response-distribution is computed as follows,

$$R_c(u,v) = e^{-k\varepsilon_c(u,v)}. \tag{2.14}$$

A point with small response is less likely to be the true match as compared to a point with a high response.

In the correlation stage, the motion velocity can be calculated with the weighted-least-squares,

$$U(x,y) = \frac{\sum_{u=-N}^{N} \sum_{v=-N}^{N} R_c(u,v)u}{\sum_{u=-N}^{N} \sum_{v=-N}^{N} R_c(u,v)} \tag{2.15}$$

$$V(x,y) = \frac{\sum_{u=-N}^{N} \sum_{v=-N}^{N} R_c(u,v)v}{\sum_{u=-N}^{N} \sum_{v=-N}^{N} R_c(u,v)}, \tag{2.16}$$

In the propagation stage we have

$$U^{(n+1)} = \sum_{x=-N}^{N} \sum_{y=-N}^{N} w(u,v) U^{(n)}(i+x, j+y) \tag{2.17}$$

$$V^{(n+1)} = \sum_{x=-N}^{N} \sum_{y=-N}^{N} w(u,v) V^{(n)}(i+x, j+y) \tag{2.18}$$

where w is a Gaussian mask.

2.3 Correlation-Feedback with Kalman Filter

The gradient-based, correlation-feedback, and most other optical flow determining techniques are suffering from blurring in moving boundaries [62].

For the correlation-feedback algorithm, the error decreasing rate, analyzed in [62], is as follows,

$$\frac{\delta^2_{u(k+1)}}{\delta^2_{uk}} = \frac{5c}{(4\bar{I}^2_x\delta^2_{uk}+1)^{\frac{3}{2}}} \tag{2.19}$$

where $\bar{I}^2_x$ is the mean square of I_x in a correlation window, c is a constant. This error decreasing rate is varied for different regions in an image plane. It is larger for the regions where intensity varies more drastically, it is smaller for those where intensity varies more smoothly. This indicates that the iterations needed in optical flow determination should not be uniform for different image regions. That is, for the moving boundaries, where intensity usually changes bigger, fewer iterations are needed than for other regions. Therefore, an optical flow algorithm needs to have fewer iterations along moving boundaries than in other areas so that the better estimations of optical flow along boundaries can be propagated into other areas instead of being blurred by those in other areas. A Kalman filter can realize the task of applying different number of necessary iterations in determining optical flow to deblur boundary and enhance accuracy.

Furthermore, the Kalman filter is based on a linear measurement model, it will not introduce much computing burden. It operates in two phases: the prediction phase and the update phase, as shown in Table 2.1. The block diagram is given in Figure 2.2. For more details, please refer to [61][62].

2.4 Unified Optical Flow Field (UOFF)

An image brightness function can be described by

$$g(x, y, t, \vec{s}) \tag{2.20}$$

where x and y are coordinates on image plane, t is time, $\vec{s}$ indicates the sensors, camera position in 3-D world space

$$\vec{s} = (x, y, z, \beta, \gamma) \tag{2.21}$$

Table 2.1 Kalman filter

Models	System model	$u_k = \Phi_{k-1} u_{k-1} + \eta_k.\ \eta_k \sim N(0, Q_k)$
	Measurement model	$D_k = H u_k + \xi_k.\ \xi_k \sim N(0, R_k)$
	Prior model	$E[u_0] = \hat{u}_0.\ cov[u_0] = P_o$
	(other assumption)	$E[\eta_k \xi_k^T] = 0$
Prediction	State estimate extrapolation	$\hat{u}_k^- = \Phi_{k-1}\ \hat{u}_{k-1}^+$
	State covariance extrapolation	$P_k^- = \Phi_{k-1} P_{k-1}^+ \Phi_{k-1}^T + Q_{k-1}$
Update	State estimate update	$\hat{u}_k^+ = \hat{u}_k^- + K_k(D_k - H_k \hat{u}_k^-)$
	State covariance update	$P_k^+ = (I - K_k H_k) P_k^-$
	Kalman gain matrix	$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$

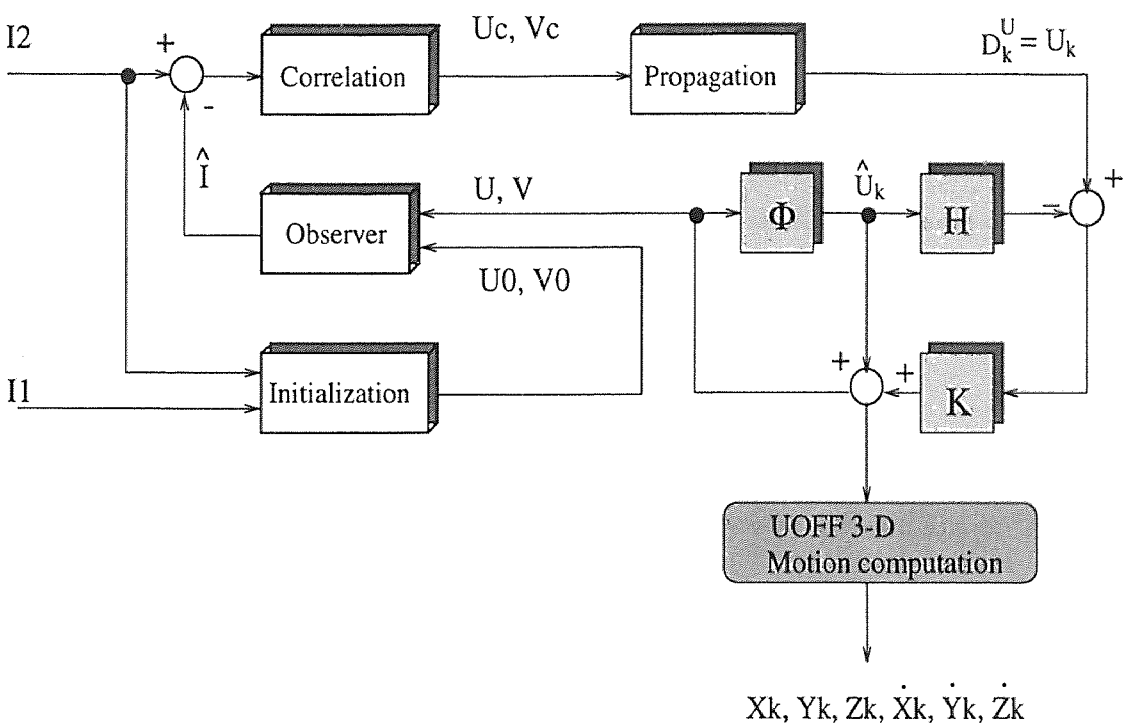


Figure 2.2 The correlation feedback with Kalman filter

(x, y, z) is the position of the optical center of the sensor in 3-D world space, β and γ represent the orientation of the optical axis of the sensor in 3-D world space. A world point P in 3-D space is projected onto the image plane as a pixel with the coordinates x_p and y_p. Then x_p, y_p are also dependent on t and $\vec{s}$, i.e. $x_p = x_p(t, \vec{s})$ and $y_p = y_p(t, \vec{s})$. Then

$$g = g(x_p(t, \vec{s}), y_p(t, \vec{s}), t, \vec{s}) \tag{2.22}$$

Due to the assumption of the time and space invariance of brightness, one can get

$$g(x(t, \vec{s}), y(t, \vec{s}), t, \vec{s}) = g(x(t + \delta t, \vec{s} + \delta\vec{s}), y(t + \delta t, \vec{s} + \delta\vec{s}), t + \delta t, \vec{s} + \delta\vec{s}) \tag{2.23}$$

To expand the right hand side of the above equation in the Taylor series leads to:

$$(\frac{\partial g}{\partial x}u + \frac{\partial g}{\partial y}v + \frac{\partial g}{\partial t})\delta t + (\frac{\partial g}{\partial x}\vec{u^s} + \frac{\partial g}{\partial y}\vec{v^s} + \frac{\partial g}{\partial \vec{s}})\delta\vec{s} + \varepsilon = 0 \tag{2.24}$$

1. If $\delta\vec{s} = 0$, the sensor is static in a fixed spatial position.

2. If $\delta t = 0$, i.e. at a specific time moment, the images generated with sensors at different spatial positions can be viewed as a spatial sequence of images. Hence, one has

$$\frac{\partial g}{\partial x}\vec{u^s} + \frac{\partial g}{\partial y}\vec{v^s} + \frac{\partial g}{\partial \vec{s}} = 0 \tag{2.25}$$

   This equation can be used to deal with stereo imagery and can be solved with smoothness constraints [62].

The UOFF approach [71] can calculate the u^L, v^L, u^R, v^R, u^S, and v^S, where superscripts L and R indicate the left and right cameras, respectively. Using these six UOFF parameters and a set of equations, one can solve for all 3-D motion and position parameters, which completely describe the object in space status [71].

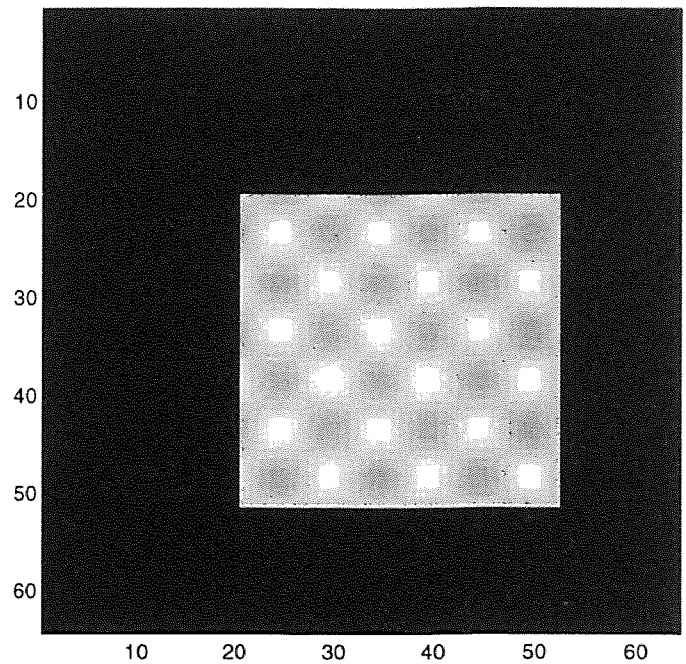


Figure 2.3 A sinusoidal square

2.5 Experiment and Conclusion

In this dissertation research, some experiments are conducted to evaluate the performances of the gradient-based and correlation-feedback algorithm with and without Kalman filters. Three experiments are presented below.

2.5.1 Experiment I: A Moving Sinusoidal Square

A sinusoidal square moves east at a speed of a pixel/frame, refer to Figure 2.3. The expected truth-ground optical flow is shown in Figure A.1. The optical flow fields obtained by Horn and Schunck's gradient-based algorithm with or without Kalman filter are given in Figure A.2 and Figure A.3. The Figure A.4 and Figure A.5 demonstrate the results of the correlation feedback with or without Kalman filter. Obviously, the correlation feedback algorithm outperforms the gradient-based algorithm and the Kalman filter preserves moving boundaries better.

2.5.2 Experiment II: Real Sequence of Moving Boxes

Refer to Figure A.6, there are three boxes in this image. Two of them are moving together, and the other is not moving. Figure A.7 is the optical flow u calculated with correlation-feedback algorithm, and Figure A.8 is the optical flow calculated with correlation-feedback with Kalman filter. The optical flow field obtained with Horn-Schunck's algorithm is shown in Figure A.9, and that obtained with an additional Kalman filter is shown in Figure A.10. We can observe that the Kalman filter does improve the optical flow along moving boundaries.

2.5.3 Experiment III: Hamburg Taxi

The Hamburg Taxi sequence can be obtained from *ftp.csd.uwo.ca/pub/vision* via anonymous *ftp*. Refer to Figure A.11. There are four moving objects in this sequence. The first is a taxi turning the corner; the second is a car in the lower left, driving from left to right; the third is a van in the lower right driving right to left; and the fourth is a pedestrian in the upper left. A needle diagram of the optical flow field is shown in Figure A.12. It is noted that the moving pedestrian cannot be shown because of the scale used in the needle diagram. The optical flow fields obtained by Horn and Schunck's, Singh's (correlation-based), and the correlation-feedback algorithms are shown in Figure A.13, A.14, and A.15, respectively. Only the turning taxi portion is given in these figures in order to give a detail needle diagram. Obviously, the correlation-feedback algorithm achieves the best quality.

CHAPTER 3

OPTICAL FLOW BASED VIDEO CODING WITH THRESHOLDING TECHNIQUES

By analyzing the video sequence properties and the existing model-based facial video coding algorithms, we developed a new very low bit rate video coding algorithm [46], which is based on thresholding and optical flow techniques. This algorithm achieves a very good performance in terms of better quality of reconstructed video sequence, higher data compression ratio and much simpler computational complexity than some typical model-based facial video coding algorithms.

3.1 Introduction

It is well-known that wireframe model leads to very low bit rate in facial video coding [3][45][44], but it encounters two major problems: quality of reconstructed video sequences, and complexity of computation. In order to obtain high quality of reconstructed video sequences, some 3-D wireframe model algorithms [76][3] clip and paste some portions of speaker's face such as lips and eyes, which usually contain rich information of facial complexion, and transmit them to the receiving end. In doing so, however, identification and segmentation of these portions will be necessary. For different speakers, these sensitive portions will be quite different from one another. Consequently, computational complexity increases drastically. On the other hand, the quality of reconstructed video sequences heavily depends on the accuracy of clipping and pasting of these portions. This method relies on, in turn, the above-mentioned difficult jobs: identification and segmentation of these portions, and it is therefore questionable. Another method commonly adopted [3][45][44] utilizes so-called action unit (AU). Similar to the clip-and-paste method, it involves identification of various facial complexion for various people, which not only increases computational complexity but also causes distortion of reconstructed video sequences

seriously. Next, take a look at the case of 2-D wireframe models [3]. It is found that 2-D model algorithms are more robust than 3-D models. But, it is quite obvious that equal-spaced 2-D model is too dense in areas like background and too sparse in areas which are rich in a speaker's complexion. It then still suffers from the two problems, quality and computational complexity.

In our experimental works, it has been noticed that when handling improperly a reconstructed video sequence can differ quite from its original counterpart. That is, severe distortion can take place even though the reconstructed video sequence does look like the very speaker but with speaker's facial complexion quite different from that in original video sequence. The key here is the speaker's facial complexion. This observation agrees with a recent survey paper on model-based image coding [3].

In order to avoid above two problems, a new algorithm for facial video coding is proposed. This algorithm uses optical flow and thresholding techniques instead of the wireframe model. In order to reduce transmitted data, and increase the quality of reconstructed image sequence, four thresholds are established and utilized in the process of coding.

3.2 Statistic Model and Thresholds

After analyzing the existing video sequences, such as Miss America, Claire and Salesman, we found that the distribution of the difference between two successive frames is very close to the zero-mean Gaussian distribution, except for the portions of large differences. For example, the histogram of the difference between the third frame and the second frame of Claire sequence is shown in Figure 3.1, while the corresponding Gaussian distribution is shown in Figure 3.2. Comparing these two curves, it is observed that there are more percentages of large difference in Figure 3.1 than that in Figure 3.2. It is conjunctured that these differences mainly come from the object motions. From statistics, consider two pictures taken at two

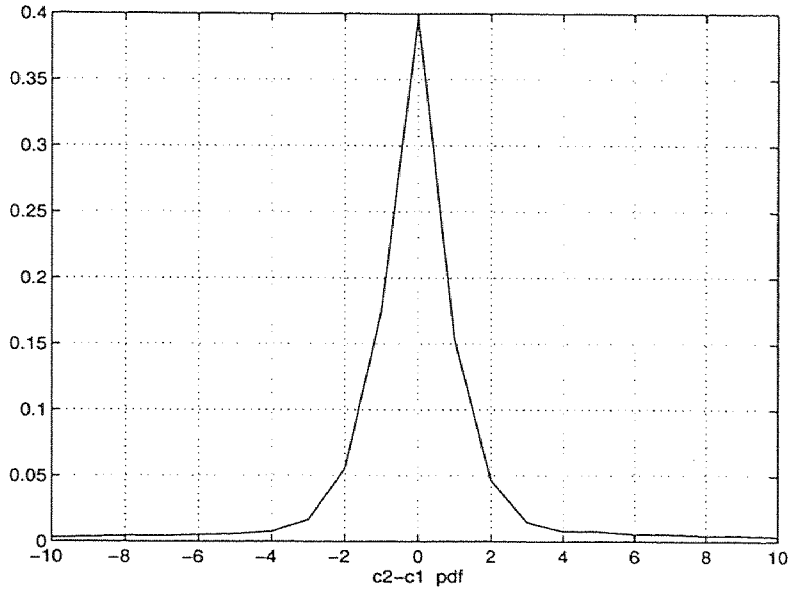


Figure 3.1 The distribution of the SFD

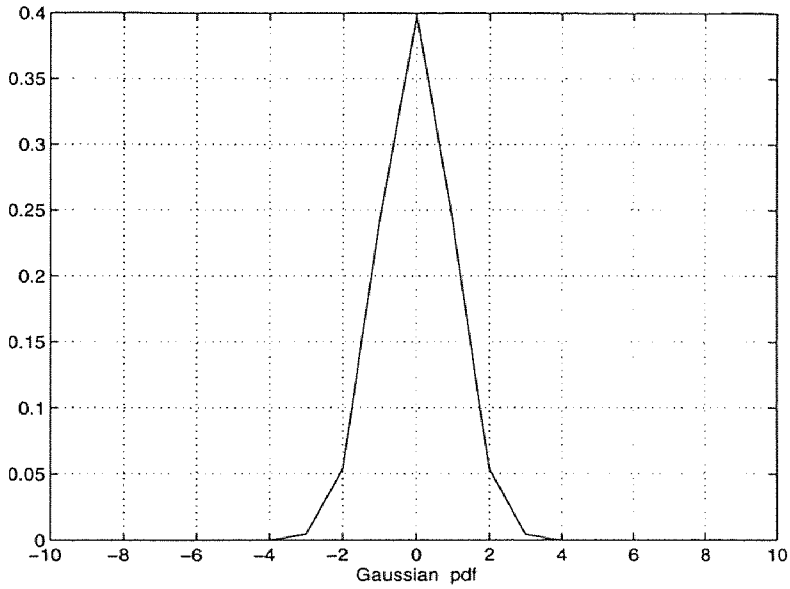


Figure 3.2 The standard Gaussian distribution

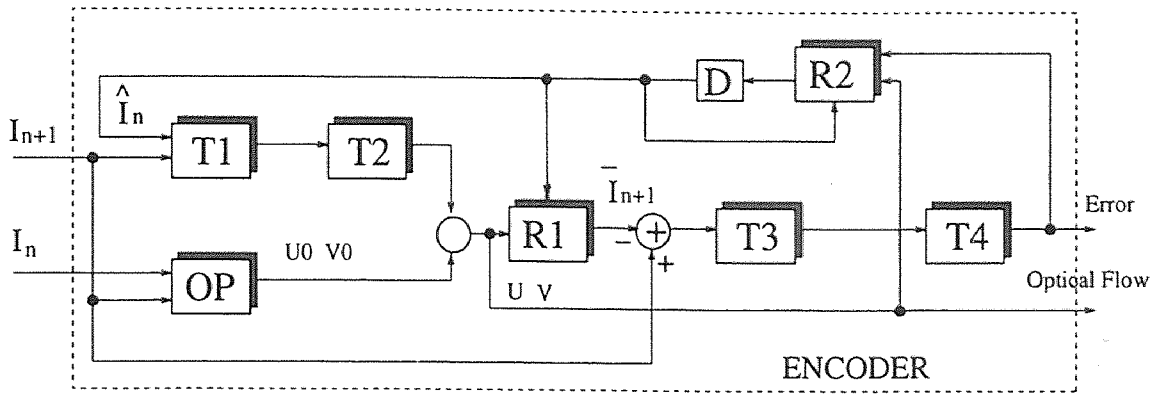
different moments, even though there is no motion existing, some white Gaussian noises may be added during this time interval. The differences of the intensities between these two images obey white Gaussian distribution, so that we can use the zero-mean Gaussian distribution as statistic model to decide the thresholds which will be discussed below.

1. Consulting Figures 3.1 and 3.2. As we discussed the differences between two images, the magnitudes of difference which is greater than three are more likely resulted from object motions. The first threshold is set and used so that the positions of the optical flow vectors associated with those pixels may be transmitted if the difference between the intensity of a pixel and that of the displaced pixel by the corresponding optical flow vector exceeds the defined threshold T_1, that is,

$$|g_n(i+u, j+v) - g_{n+1}(i+u, j+v)| > T_1 \tag{3.1}$$

   where i,j denote pixel position, u,v are the components of the optical flow vector, and g_n and g_{n+1} represent image intensity function at n and n+1 moments, respectively. Otherwise the $g_{n+1}(i+u, j+v)$ can be replaced by $g_n(i+u, j+v)$ directly, and the associated optical flow vector does not need to be transmitted.

2. Some pixels associated with large intensity differences may not experience meaningful motion. These pixels are likely isolated and corrupted with random noises. The second threshold T_2 is set that if the number of the moving pixels in a neighborhood is less than T_2, then this pixel is considered static and its motion vector does not need to be transmitted. The purpose of this threshold is to eliminate noises, most in the background caused by inaccuracy in optical flow calculation.



R1, R2, R3: Reconstrucors
OP: Optical flow calculator
T1, T2, T3, T4: Thresholds
D: Delay

In: Original nth frame
In+1: Original (n+1)th frame
Î n: Reconstructed nth frame
Ī n+1: Motion compensation (n+1)th frame

Figure 3.3 The block diagram of the new algorithm

3. In order to increase the quality of the reconstructed image, in addition to position and motion vectors of certain pixels being transmitted, the predictive error information will be transmitted as well. By the predictive error information we mean the difference between reconstructed video picture and original one. The large errors need to be transmitted, and the third threshold T3 is set that the error which is greater than T3 maybe transmitted.

4. The fourth threshold is set to avoid transmission of unnecessary errors that are isolated. Only the error of those pixels satisfying the threshold T_4, which is similar as T_2, will be transmitted.

Only if the magnitude of the error of a pixel exceeds T_3 and this error is not isolated from that of its neighboring pixels determined by T_4, then this error should be transmitted. The threshold T_3 can be controlled by considering the required PSNR (say 35 dB) and percentages of pixels whose error needs to be transmitted (4-5% is usually considered).

A block diagram of the algorithm is given in Figure 3.3.

3.3 Threshold Selection

The first threshold can be chosen as, say, between 2 to 4 in 256 gray levels. Obviously, compared with 256 different gray levels, the range between 2 to 4 means a very small portion. Therefore, not sending positions and optical flow vectors for those pixels, whose intensity value change after motion compensation is less than this threshold, will not seriously affect quality of reconstructed video sequence at the receiving end. From the video sequences of Miss America, Salesman and Claire, it's noticed that even for the static pixels intensity changing between 3 and 4 takes place very often. The second threshold can be between 6 and 8, with neighborhood of 5×5. This step further eliminates transmission amount drastically.

The third threshold is chosen as $T_3 = 4$ to 16. The T_4 is determined similarly to T_2.

These four thresholds work well in our experiments.

3.4 Expected Advantages

With the above thresholdings, the new algorithm avoids difficult tasks of identification and segmentation of the sensitive portions in a speaker's face. The optical flow computation becomes a major computational load of the new algorithm. It is known that optical flow computation can be implemented in real-time [44] and many algorithms are available. Furthermore, the transmission of the error information eases accuracy requirement of optical flow computation. For the relatively uniform regions, optical flow might even not need to be calculated following the reasoning of the first two thresholdings. Therefore, iterations needed in optical flow algorithm can be quite less. It thus simplifies computational complexity. Also, the transmission of the error information can fairly alleviate accumulative error. The transmission of the error information exceeding certain criterion raises quality without increasing computation very much.

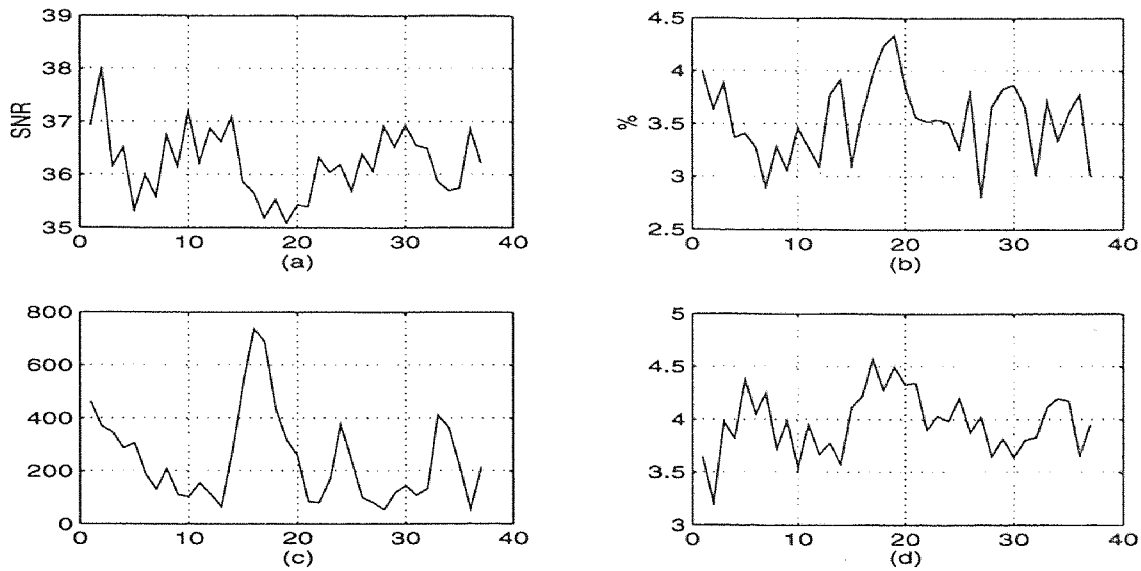


Figure 3.4 Miss America sequence: The horizontal axis is the frame number of the sequence. The vertical axis is (a) the PSNR which our algorithm achieved, (b) the percentages of the pixels whose error information needs to be transmitted, (c) the number of velocities need to be transmitted, (d) the STD

3.5 Experiments

In order to compare data compression ratio between our new algorithm and the algorithm using 3-D wireframe model, we applied our new algorithm to the video sequences of Miss America, Salesman and Claire. The results are shown in Figures 3.4, 3.5, and 3.6.

With these three sequences for the central portion (256×256) of the CIF format, our algorithm transmits the position and optical flow vectors for about 300 pixels, as well as error information of about 5% pixels. Only the first frame of each sequence is transmitted, and all the subsequent frames are reconstructed from the transmitted velocities and errors. The results show that no obvious accumulative error has occurred. In this sense, it's more robust than many other algorithms.

The compression ratio is higher than that achieved by the algorithm using 3-D wireframe model [3]. This can be justified as follows. In [3] a similar sequence (a Japanese woman) is tested. Also only the central portion, 256×256 pixels, is treated.

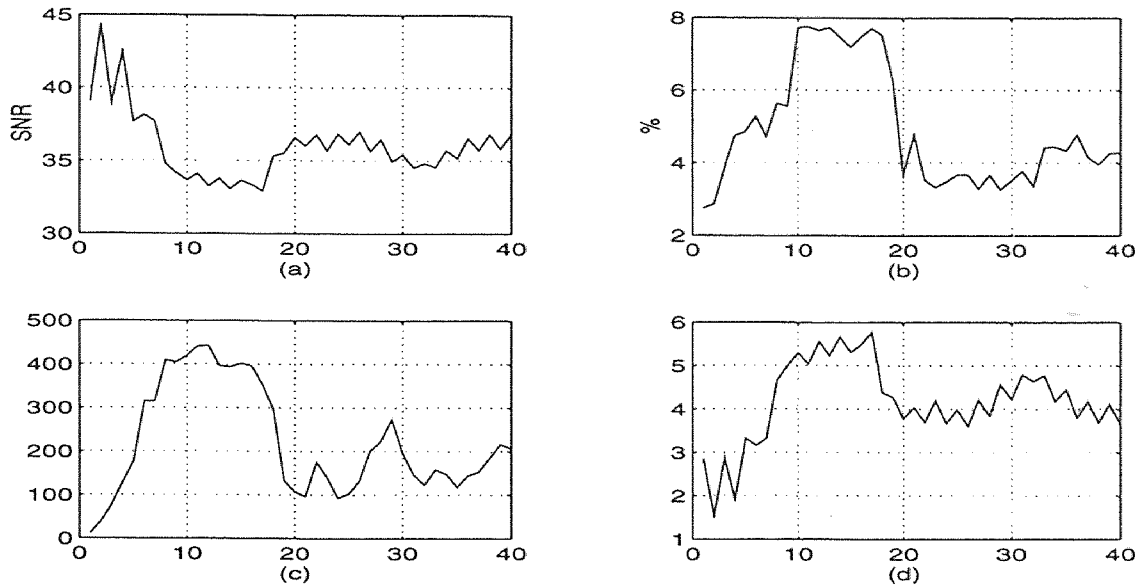


Figure 3.5 Salesman sequence: The horizontal axis is the frame number of the sequence. The vertical axis is (a) the PSNR which our algorithm achieved, (b) the percentages of the pixels whose error information needs to be transmitted, (c) the number of velocities need to be transmitted, (d) the STD

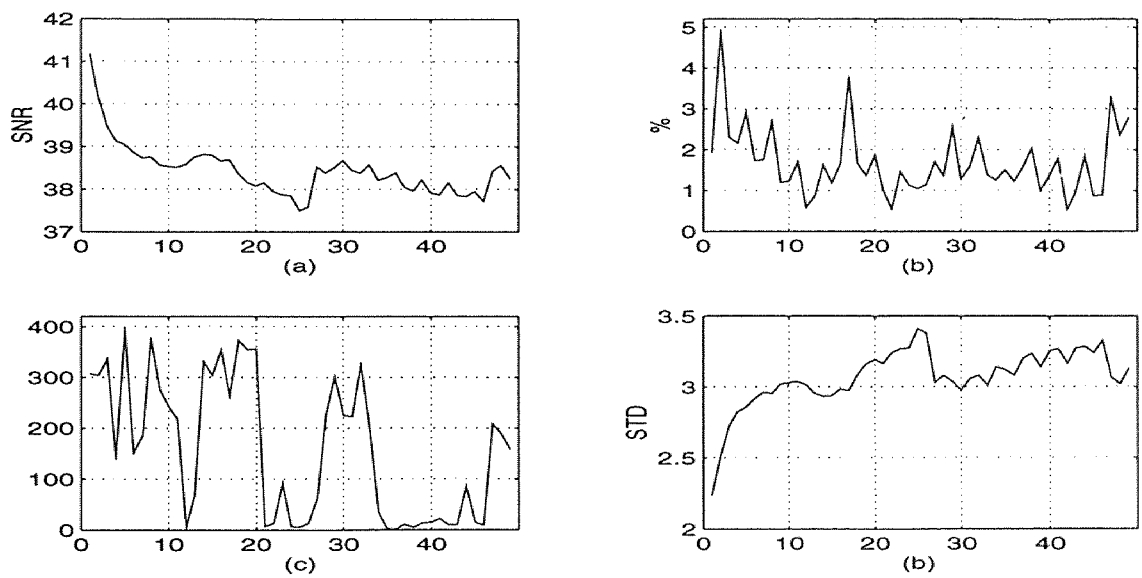


Figure 3.6 Claire sequence: The horizontal axis is the frame number of the sequence. The vertical axis is (a) the PSNR which our algorithm achieved, (b) the percentages of the pixels whose error information needs to be transmitted, (c) the number of velocities need to be transmitted, (d) the STD

The motion vectors (3 parameters each) of vertices about 400 triangles need to be transmitted. For our algorithm only about velocities (two parameters each) of about 300 pixels are needed to be transmitted. Using 3-D wireframe model, the number of pixels clipped and pasted, whose intensity values are transmitted in order to raise quality of reconstruction, is reported as 5% of the total pixels, while the number of pixels whose error information needs to be transmitted in our algorithm working on Miss America sequence is found to be 3.5%.

In terms of quality, our algorithm achieves the PSNR of 36.22 dB for Miss America sequence, 36.0 dB for Salesman sequence and 38.4 dB for Claire sequence, while the algorithm using 2-D wireframe model in achieves 35.5 dB for Miss America sequence. Compared with the algorithm reported in [45][44], which uses AU and achieves the standard deviation of 7.5 (for Claire sequence), our algorithm achieves the standard deviations of 3.1 for the same sequence.
Refer to Table 3.7 for detail. The performance of our algorithm is also better than 2-D model algorithms.

3.6 Conclusion

Following an analysis of existing wireframe model-based facial coding algorithms, a new algorithm is developed. It utilizes optical flow and thresholding techniques. As a result, it avoids completely identification and segmentation of those highly complicated regions in human face containing rich facial complexion information, thus raising quality of reconstructed video sequences as well as data compression ratio, and simplifying computation drastically.

Performances of Different Algorithms

		3-D Wireframe Model [1,4]	2-D Wireframe Model [4]	3-D Wireframe Model (AU) [2,3]	Our Algorithm		
Testing Sequence		A Japanese Woman	Miss America	Claire	Miss America	Salesman	Claire
Quality	PSNR (dB)	Not Available	35.5	Not Available	36.22	36.0	38.45
	Standard Deviation	Not Available	Not Available	7.5	4.0	4.17	3.07
Data Compression		5% Pixels are Clipped and Pasted. Their Intensities are Transmitted.	Not Available	Not Available	Error information of 3.5% pixels are Transmitted	4.9% (See Left for Detail)	1.7% (See Left for Detail)
		Velocities (Three Parameters each) of Vetecies of about 400 Triangles are Transmitted			Velocities (Two Parameters each) of 300 Pixels Transmitted		
Identification and Segmentation		Clip-and-Paste is involved	Not Needed	AU is involved	Not Needed		

Figure 3.7 The comparison of different algorithms

CHAPTER 4

A DCT CODED OPTICAL FLOW ALGORITHM FOR VERY LOW BIT RATE VIDEO CODING

In this chapter, we propose an efficient compression algorithm for very low bit-rate video applications. The algorithm [47] is based on (1) optical-flow motion estimation to achieve more accurate motion prediction fields; (2) DCT-coding of the motion vectors from the optical-flow estimation to further reduce the motion overheads; and (3) region adaptive threshold technique to match optical flow motion prediction and minimize the residual errors. Unlike the classic block-matching based discrete cosine transformation (DCT) video coding schemes in MPEG-1/2 [43][14] and H.261/3 [58][32], the proposed algorithm uses optical flow for motion compensation and the DCT is applied to the optical flow field instead of predictive errors. Thresholding techniques are used to treat different regions to complement optical flow technique and to efficiently code residual data. While maintaining comparable peak signal to noise ratio (PSNR) and computational complexity with that of ITU-T H.263/TMN5, the reconstructed video frames of the proposed coder are free of annoying blocking artifacts, and hence visually much more pleasant. The computer simulation are conducted to show the feasibility and effectiveness of the algorithm. It achieves a bit-rate of 11 kbps for interframe coding and can be used for transmission of both audio and video signals through the existing public switched telephone network.

4.1 Introduction

The successful applications of ISO MPEG 1/2 and ITU-T H.261 [14][58][72][18][32] for video communications at relatively high bit-rates demonstrate that the block matching based motion compensation and DCT coding algorithm (BMDCT) works quite well at a bit-rate above 64 kbps for digital video coding. However, it is well known that the block matching techniques have several serious drawbacks: unreliable

motion fields in the sense of the true motion in the scene, block artifacts, and poor motion compensated prediction along moving edges [17] . At very low bit-rates (below 64 kbps), the block artifacts become severe and the quality of the reconstructed images are degraded considerably. It is especially true in facial video coding because most of the facial expressions involve nonrigid motion. Considerable research efforts have been made on very low bit rate coding, which result in the current H.263/TMN5 [32]. It achieves much improved performance over H.261 with advanced motion prediction options, optimized entropy coding, and fine-tuning of the parameters to fit very low bit rate nature of the applications. Despite its good performance, H.263 still suffers from the blocking effects, an intrinsic problem of the block-matching motion prediction and block DCT implementation.

Many techniques have been proposed to overcome the drawbacks of block matching techniques. One approach is to use overlapped windows [77][57][9], which was also adopted as an optional feature for H.263. Another approach is to apply variable block size (VBS) motion prediction to adapt the motion prediction to object-level. One example of VBS is the locally adaptive multigrid block matching motion estimation [17]. There a multigrid structure and a modified three-step search are used. The entropy criterion is established to optimally balance the amount of information corresponding to the prediction error and the representation of the motion. Another example uses a quadtree structure and different criterion to remove the constraint of fixed block size and translational motion to allow more flexibility in the motion field [53].

Different from these techniques, in this paper we present a new algorithm as an improvement of H.263/TMN5. The proposed algorithm utilizes optical flow (dense motion vector field) instead of block matching (block motion vector field) for motion compensation. Due to the usage of optical flow technique, it is expected to overcome the drawbacks of the traditional block matching. That is, it is expected to provide

more accurate predictions and to eliminate the annoying block artifacts. However, the dense motion field implies more overhead information. How to handle this issue becomes a key in our new algorithm. We preprocess images to lower the computational burden as well as side information related to optical flow field. Based on an analysis, which shows that dense motion vectors are highly correlated, we use DCT to code the optical flow vectors instead of predictive errors. A thresholding technique is developed to treat different regions to complement with the optical flow technique to further decrease side information. Due to the usage of dense motion vector field, it is expected that the predictive errors will be drastically reduced and less correlated. To save computation, we directly code the residual data without involving the DCT. Another thresholding technique is devised to code residual data. The new algorithm is free of blocking effect and hence achieves better visual quality than the H.263/TMN5 while maintains comparable PSNR and computational complexity. It achieves a bit-rate of 11 kbps for interframe coding and can thus be used for transmission of both video and audio signals in PSTN and cellular networks.

4.2 Description of the New Algorithm

An overall block diagram of the proposed algorithm is shown in Figure 4.1. After preprocessing, optical flow is estimated from frames of a video sequence. These optical-flow vectors are divided block by block with each block of 8×8. Those flow vectors which do not satisfy certain thresholds in the successive frame difference (SFD) are eliminated from coding. Those which pass the thresholds are then transformed into discrete cosine domain. The position information of these DCT coefficients are quantized, zig-zag scanned, and run-length coded followed by Huffman coding. The magnitudes of these DCT coefficients are Huffman coded. The predictive errors are thresholded first. They are then divided block by block in the same fashion. The

magnitude and position information of these non-zero residual data are treated in a similar way to that for the DCT coefficients of motion vectors.

The new algorithm is described in more details below.

4.2.1 Optical Flow Estimation

Modified Horn and Schunck algorithm The motion compensation plays a key role to exploit high correlation existing in video sequences. In this new algorithm, the dense motion vectors are used together with the bilinear interpolation for motion compensation. Using the dense motion vector field is expected to achieve much less prediction errors than using block-based motion model. The reconstructed images are expected to look much more natural, and the facial expression is much closer to the original.

In video coding, the ultimate goal of motion estimation is not to assess the motion present in a scene, but to transmit the video frames with satisfactory quality and less bit-rate. Therefore, it is the changes in the spatiotemporal intensity, i.e. the optical flow, instead of 2D motion field, that need to be estimated. Optical flow field is defined as the distribution of apparent velocities of movement of brightness patterns in an image. From this perspective, the optical flow can arise from not only relative motion of objects and viewer but also intensity variation.

There are two models of optical flow field. One is a deterministic model in which the motion can be considered as an unknown deterministic signal and can be estimated with maximum likelihood by maximizing the probability of the observed sequence with respected to the unknown motion. The other is a random model in which the motion is assumed to be a random variable. The motion field is modeled by a Markov random field. It can be estimated by maximizing a posteriori or minimizing expected cost. The deterministic model is usually taken.

A comprehensive survey of existing optical flow techniques is recently conducted by Barron, Fleet and Beauchemin [10]. There nine algorithms, classified into the following four different techniques: differential [28][56], region-based matching [5] energy-based [1][11], and phase-based techniques [19], are studied thoroughly and their performance are compared with each other. Some of the techniques can work in real time. What we used in the experiments is the so-called modified Horn and Schunck algorithm since it runs fast and gives quite good accuracy [10]. The Horn and Schunck algorithm [28] is among differential techniques. It is also frequently referred to as the gradient-based approach.

The gradient-based approach is based on the assumption that intensity of light reflected by a point on a surface of an object and recorded in the image remains constant during a short time interval, although the location of the image of that point may change due to motion. This can be mathematically stated as,

$$I(x, y, t) = I(x + u\Delta t, y + v\Delta t, t + \Delta t) \tag{4.1}$$

where $\vec{U} = (u, v)$ is an optical flow vector at the point (x, y) and it is assumed to be constant during the interval $(t, t + \Delta t)$. This equation is called intensity constant equation. $I(x, y, t)$ is the image intensity at point (x, y) in the image at time t. In the limit, when the time interval Δt tends to zero, the intensity-constancy assumption leads to the following equation:

$$I_x u + I_y v + I_t = 0 \tag{4.2}$$

where I_x, I_y, and I_t are partial derivatives of I with respect to x, y, and t, respectively. This is because the Taylor series expansion of the right hand side of Equation(4.1) is as follows,

$$\begin{aligned} I(x + u\Delta t, y + v\Delta t, t + \Delta t) &= I(x, y, t) + I_x u\Delta t + I_y v\Delta t + I_t \Delta t \\ &+ \textit{higher order terms.} \end{aligned} \tag{4.3}$$

Ignoring the higher order terms in Equation (4.3), using Equation (4.1) in Equation (4.3) and taking the limit as $\Delta t \rightarrow 0$, Equation (4.2) can be obtained.

The collection of image velocity vectors $\vec{U}$ for the entire image constitutes the optical flow field for the image. Equation (4.2) embodies two unknowns u and v, and is not sufficient by itself to specify the optical flow uniquely. This problem is known as the aperture problem. But Equation (4.2) does constrain the solution. It is possible to regularize the ill-posed problem and to compute optical flow for images by introducing an additional constraint. A frequently utilized assumption is the smoothness constraint, i.e., motion field varies smoothly in most parts of the image, which was introduced by Horn and Schunck [28]. That is the minimization of $\| \Delta u \|_2^2$ and $\| \Delta v \|_2^2$, i.e., the squares of the magnitude of the gradient of the optical flow velocity components u and v respectively:

$$(\frac{\partial u}{\partial x})^2 + (\frac{\partial u}{\partial y})^2$$

and

$$(\frac{\partial v}{\partial x})^2 + (\frac{\partial v}{\partial y})^2.$$

As a result, the optical flow is calculated by minimizing the error in optical flow.

$$\int_D (\Delta I \cdot \vec{U} + I_t)^2 + \alpha^2(\| \Delta u \|_2^2 + \| \Delta v \|_2^2) dX \tag{4.4}$$

where D is integration domain, the magnitude of α reflects the influence of the smoothness term, $\Delta I = (I_x \ I_y)$, and $\| \Delta u \|_2^2 + \| \Delta v \|_2^2$ are the measure of the departure from smoothness in the optical flow.

Horn and Schunck solved this minimization problem by using the variation calculus. An iterative procedure was derived:

$$u^{k+1} = \bar{u}^k - \frac{I_x[I_x\bar{u}^k + I_y\bar{v}^k + I_t]}{\alpha^2 + I_x^2 + I_y^2} \tag{4.5}$$

$$v^{k+1} = \bar{v}^k - \frac{I_y[I_x\bar{u}^k + I_y\bar{v}^k + I_t]}{\alpha^2 + I_x^2 + I_y^2} \tag{4.6}$$

where k denotes the iteration number, u^0 and v^0 denote initial velocity estimates which are set to zero, and $\bar{u}^k$ and $\bar{v}^k$ denote neighborhood averages of u^k and v^k.

On the boundary, however, the smoothness constraint of optical flow may not hold. Much efforts have been made to improve optical flow determination along the boundaries. A detailed discussion about the quantitative error and reliability analysis of the gradient-based approach can be found in [36].

One of the problems with this algorithm is that the intensity derivatives are estimated by using a first-order difference, which is a relatively crude form of numerical differentiation and can be the source of considerable error. Barron et al modified it with spatiotemporal presmoothing and 4-point central differences for differentiation (with mask coefficients $\frac{1}{12}(-1, 8, 0, -8, 1)$), resulting the modified Horn and Schunck algorithm. It uses a spatiotemporal Gaussian prefilter with a standard deviation of 1.5 pixels in space and 1.5 frames in time (1.5 pixel-frames), sampled out to three standard deviations. It performs better than the original Horn and Schunck algorithm [10].

Number of iterations As mentioned above, the ultimate goal of optical flow used for video coding is somehow different from that for robot vision. Hence, optical flow computation for video coding has different features. Specifically, optical flow in smooth areas does not need to be propagated from other areas for many times since optical flow accuracy is not vital in these areas. Consequently, we only use five iterations of the fast modified Horn and Schunck algorithm, rather than more than 100 iterations suggested in [10] . This saves lots of computation. Moreover, the problem caused by the smoothness constraint will not be serious with only five iterations. The optical flow estimation in fixed background and foreground is even not required, since these fixed background and foreground may be copied from one frame to its next frame.

Preprocessing The analysis in above section suggests a preprocessing to reduce optical flow vectors. This preprocessing is to decide in which areas the optical flow vectors need to be calculated. To coup with H.263, the whole image is divided into blocks with the size of a block being 16 by 16. A measure of SFD of a block is defined as follows:

$$SFD_l = ||V_l(t) - V_l(t-1)||, \tag{4.7}$$

where l is the index of the block, $V_l(t)$ represents a vector formed in a certain manner with all the pixels within the block in the image $I(x, y, t)$, and $||.||$ means vector norm. The optical flow of block l will not be calculated except that its SFD_l is greater than a preset threshold T_{SFD}. Therefore T_{SFD} is an adjustable parameter of the algorithm. In our experimental work, the T_{SFD} is usually chosen such that only less than 25% blocks need optical flow determination. Concretely, we calculate SFD_l for each possible l, then arrange them, say, in a descending order. The T_{SFD} is chosen as such a value that there are only less than 25% of SFD values in the sequence are larger that it. The reason to do so is that there is usually only small motion experienced during the time interval between two consecutive frames, and the change of brightness patterns mainly occurs around moving boundaries. For uniform or smooth regions in the image plane, there is no need to calculate flow vectors for video coding. In our experiments, this choice of T_{SFD} can produce both satisfactory reconstructed image quality and required bit-rate. This preprocessing saves not only huge computation, but also huge side information.

4.2.2 DCT Coding of the Motion Vectors with Thresholding

AR model It is considered in general that the motion vectors are transmitted as side information in motion compensated video coding schemes. However, when the very low bit-rate video coding is dealt with, the amount of the motion vector data

becomes comparable with and even more than that of the error data. Therefore the motion vector coding for very low bit-rate becomes more important than that in the case of high bit-rate. Obviously, to transmit all the flow vectors needs many more bits since even after the preprocessing the number of motion vectors is much more than that in the BMDCT technique. The bits used to encode the optical flow substantially affect the transmission bit-rate.

How to code optical flow vectors? Let us consider human facial expressions. It is noted that the motion of any point in a face is not free or independent of its neighboring points, it is constrained by some muscles and skin. That is, the motion of a point correlates with that of its neighborhood very closely. It is well known that the DCT works very efficient for highly correlated data. Hence, in the new algorithm we use the DCT to code optical flow vectors.

Using DCT to code optical flow vectors can also be justified from the theoretical sense. Figure 4.2 (a) shows a diagram of probability density function (pdf) of a first order AR model:

$$f_n = \rho f_{n-1} + \nu_n$$

where $\rho = 0.8$ and ν is a random variable, obeying Gaussian distribution with mean being 0 and variance 0.17. Figure 4.2 (b) and (c) show the pdf of u and v components of optical flow field associated with Claire sequence (the 20th and 21st frames). The similarity between Figures 4.2 (a), and (b) and (c) supports this AR modeling of optical flow field. It is well-known that DCT is most effective for coding AR process.

Thresholding To further reduce motion vector data, we use thresholding technique in our new algorithm. Most background of head-shoulder type of video frames such as in the Miss America and Claire sequences is fixed. These regions need not to be transmitted in interframe coding. They can be copied from the previous frame to the current frame directly. To avoid complex segmentation and merging of these

patches, the whole image is divided into fixed-size blocks (generally, 8×8 to cope with H.263). The mean and variance of the difference between estimated frame $\hat{I}_n$ and given frame I_{n+1}, named SFD in Figure 4.1, for those blocks whose optical flow has been estimated, are calculated. It is noted that the SFD used here is slightly different from that used in above section. Estimated frame $\hat{I}_n$ is used here, while actual frame I_n is used there. To decide which block's optical flow vectors need to be coded and transmitted, two thresholds, T_1 and T_2, are set. If the mean of a block is less than T_1 and the variance is less than T_2, then this block is assumed to be a non-motion block, all its contents can be replaced by the corresponding block in the previous frame and nothing is to be coded and transmitted. On the other hand, if the mean of a block is greater than T_1 or its variance is greater than T_2, this block's motion vectors then need to be coded and transmitted. The DCT is applied to this 8x8 block. The coefficients of DCT are then quantized. The positions of the nonzero coefficients are zig-zag scanned. The amplitudes and positions of the non-zero DCT coefficients are run-length coded, followed by Huffman coding. Refer to [72] for details. The thresholds T_1 and T_2 can be viewed as adjustable parameters. Their selection will affect the quality of reconstructed frames and the bit-rate. In summary, the preprocessing discussed in above section aims at reducing the number of optical flow vectors, while the thresholding discussed here is to reduce the number of flow vectors that need to be coded. Both contribute to data saving.

DCT, quantization and zigzag scanning The 8×8 motion vectors are transformed into the frequency domain using the 2-D forward discrete cosine transform (FDCT) using equation

$$F(u,v) = \frac{C(u)}{2}\frac{c(v)}{2}\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16} \tag{4.8}$$

where $C(u) = \frac{1}{\sqrt{2}}$ for $u = 0$, $C(u) = 1$ for $u > 0$, $C(v) = \frac{1}{\sqrt{2}}$ for $v = 0$, $C(v) = 1$ for $v > 0$,

The transformed 64-point discrete signal is called DCT coefficients. The $F(0,0)$ coefficient is called the dc coefficient and the remaining 63 coefficients are called the AC coefficients. The DC coefficients represent the average values of the 64 points motion. If all the 63 AC coefficients are discarded, then it is reduced to the block matching model with a single motion vector to describe the whole block motion. Next, the coefficients whose amplitudes have trivial contribution to the quality of the image are dropped out, and thus increasing the number of zero-value coefficients. Quantization further discards DCT coefficients that are not visually significant. The positions and the nonzero coefficients are zigzag scanned.

Runlength coding and Huffman coding After quantization, most of the DCT coefficients are zeros. The runlength coding is used to further save the bits. It works as follows. In the intermediate symbol sequence, each nonzero AC coefficient is represented by a pair of symbols, where

Symbol - 1	*Symbol - 2*
(RUNLENGTH, SIZE)	*(AMPLITUDE)*

RUNLENGTH is the number of consecutive zero AC coefficients preceding the nonzero AC coefficient. *SIZE* is the number of bits used to encode *AMPLITUDE*. *AMPLITUDE* is the amplitude of the nonzero AC coefficient. Entropy coding (Huffman coding) is then used to follow the run-length coding.

4.3 Adaptive Coding of the Residual Pictures

Like all other motion compensated coding schemes, this algorithm also transmits error information. In order to find out the predictive errors, we reconstruct an image from the motion vectors and the previous frame as follows. After we obtain the nonzero quantized DCT coefficients and their positions, we reorder these coefficients according to their positions, and the nontransmitted coefficients are replaced with

zeros. Then the inverse DCT is applied to recover the motion vectors. Using these motion vectors together with bilinear interpolation if necessary, we estimate the current frame from the previous one. The difference between the estimated and actual current frames thus gives the predictive error. Because the pixel-based motion vectors rather than block-based motion vectors are used, the predictive errors are much less than that obtained by the BMDCT. The less the predictive errors, the lower the correlation they have, and the less effective the DCT, if applied, will be. For this reason, we do not apply the DCT to the error information. Instead, we transmit the errors directly. In order to use the bit-rate more effectively, another threshold T_3 is set so that only the predictive errors which are greater than T_3 will be quantized and transmitted. From experiments, it is observed that the predictive errors, needed to be transmitted in order to achieve good quality of the reconstructed image, are not scattered sparsely. Most of them are concentrated in the regions near the eyes and mouth. Using the 8×8 blocks defined above for optical flow vectors, the positions of these errors are zig-zag scanned and run-length coded in the same way as used for the nonzero coefficients of the DCT of the optical flow vectors. It is noted that T_3 is the only parameter left after T_{SFD}, T_1 and T_2 have been determined. Adjustment of this threshold T_3 may play a role to reach a desired compromise between the reconstructed frame and quality and bit-rate.

The values of both the quantized nonzero coefficients of the DCT of motion vectors and the quantized predictive errors, and the positions of these values are coded by variable length coding (Huffman codes) to further reduce the bit-rate.

4.4 Experimental Results

In order to evaluate the performance of our new algorithm, we applied it to Miss America and Claire sequences in the format of QCIF. The optical flow field is calculated with the modified Horn and Schunck algorithm. The coefficients of DCT

of motion vectors are quantized by 16 levels and the predictive errors are quantized by 32 levels. In these experiments, the threshold T_1 is chosen as 3.0, T_2 is 10.0, and T_3 is adaptive, ranging from 15 to 30.

In the case of Miss America, for 10 frames/s (30 frames/s, 2 frames are skipped for every 3 frames) interframe coding, the proposed algorithm achieves a bit-rate of 11 kbps and the PSNR of about 37.1 dB. Both are averaged for the first 36 frames of the sequence. These reconstructed frames are free of blocking artifacts. The 21st reconstructed frame is shown in Figure B.1 (b), the corresponding frame reconstructed with H.263/TMN5 is shown in Figure B.1 (c), and the original 21st frame in Figure B.1 (a). It is obvious that prominent blocking effects can be observed in the reconstructed frame with H.263/TMN5, particularly in these areas: the hair at right side, the lower portion of lips and chin, and the neck below chin, the background and so on. These blocking artifacts do not exist in the reconstructed frame with our algorithm. Our algorithm also improves the reconstructed image in the regions with rich information, like eyes, mouth and facial expression considerably, and the reconstructed frame is visually much more pleasant. Figure B.2 (a) and (b) show the PSNR curves frame indexes for our algorithm and H.263/TMN5, respectively.

The similar comparison and observations are made for Claire sequence. The 39th reconstructed frames with the new algorithm and with H.263/TMN5, and the original frame are shown in Figure B.3 . The PSNR curves are illustrated in Figure B.4.

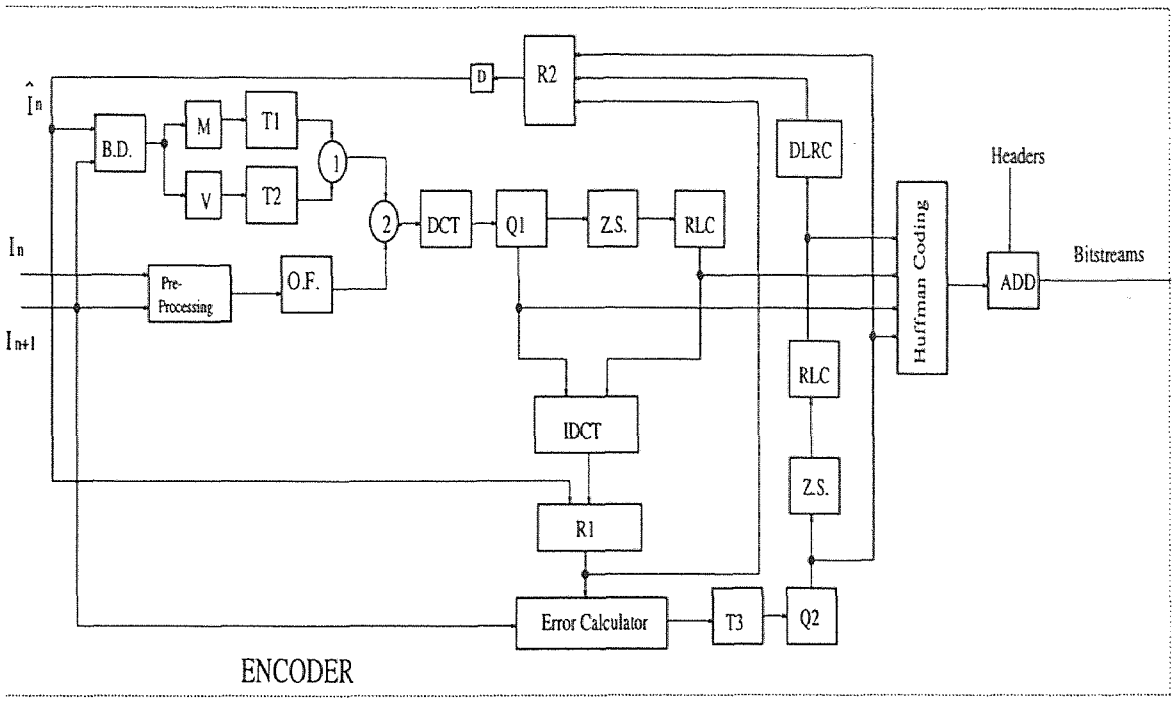
4.5 Conclusion and Discussion

An efficient optical flow based motion compensation video coding scheme for very low bit-rate application is presented and implemented in this paper. The employment of the DCT to highly correlated optical flow motion vectors reduces data needed

for motion vectors considerably. Adaptive thresholding techniques contribute to the coding efficiency of the algorithm as well.

It is noted that the optical flow based motion estimation for video compression has been applied for many years. However, the high bit overhead prevents it from practical usage in video coding. One of the new elements in this paper is the DCT coding of the dense motion vectors and the adaptive coding of the residual pictures, which enable the proposed optical flow based algorithm to efficiently work for very low bit rate video coding.

This algorithm enhances the reconstructed image quality significantly by eliminating the blocking artifacts resulted from block matching model. Experiments demonstrate that the proposed algorithm achieves superior performance than H.263 at very low bit-rate in terms of subjective evaluation. It can be widely used in various very low bit-rate applications such as video phone, teleconferencing, and wireless communication.



O.F.: Optical flow calculator

T1,T2,T3: Thresholds

R1, R2: Reconstructors

M: Mean of blocks in SFD

V: Variance of blocks in SFD

Z.S.: Zip-Zig Scanners

B.D. : Block division in SFD

DCT: Discrete Cosine Transformation

IDCT: Inverse DCT

Q1, Q2: Quantizers

D: Delay

RLC: Run-length Coder

DRLC: Decoder of Run-length Coding

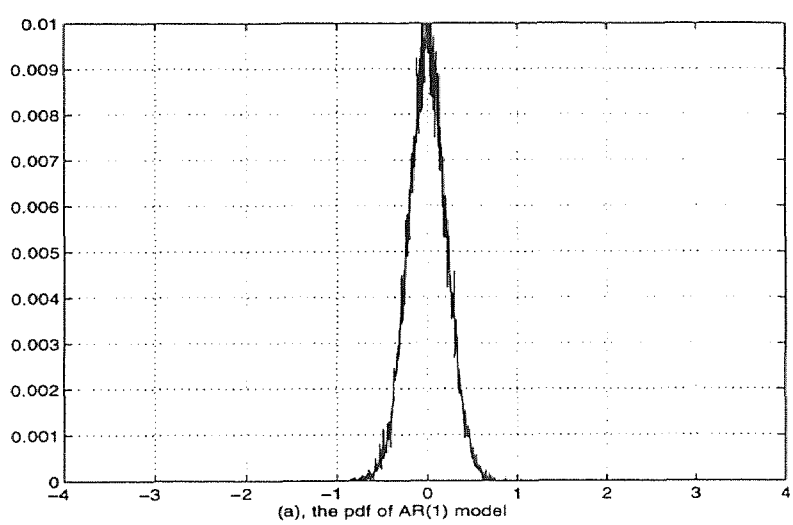
In: the nth original image frame

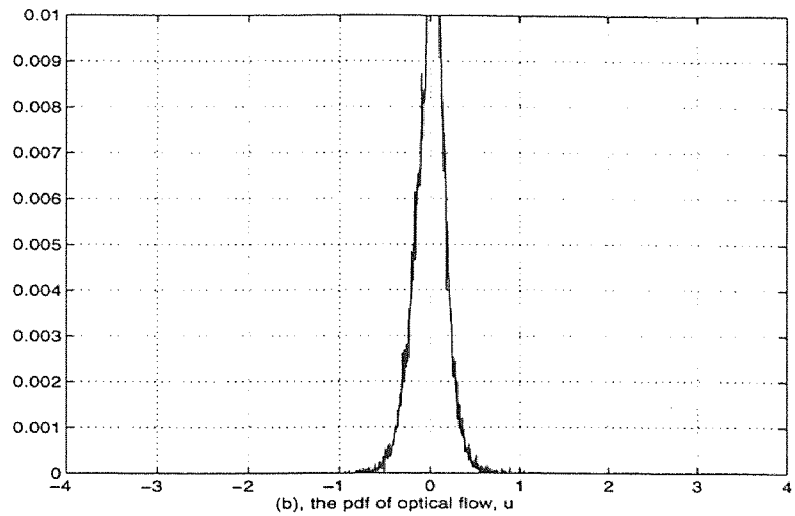
$\hat{I}_n$: the nth estimated image frame

(1) : Decide which blocks in optical flow need to be transmitted.

(2) : Pick up those blocks in optical flow field.

Figure 4.1 The encoder of the proposed algorithm





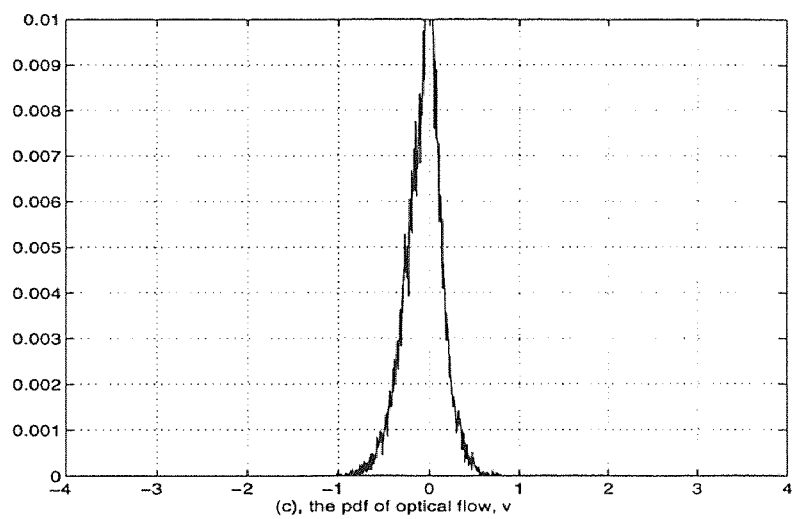


Figure 4.2 The pdfs of AR(1) model and optical flow

CHAPTER 5

REGION-BASED ADAPTIVE DWT VIDEO CODING USING DENSE MOTION FIELD

The emergence of the wavelet has led to a convergence of linear expansion methods used in signal processing and applied mathematics. Recently, the wavelets is one of the hostest research topics, it can be used in analysis acoustic signals [23], digital communications [78], computer graphics [68], biomedical engineering[73], computer vision [51] and subband coding [64]. Wavelet transform has a good time-frequency location, and multiresolution representation [50]. Some best bases have been exploited [64]. An embedded zero-tree wavelet algorithm (EZW) was proposed by Shapiro [69] for image coding. With the EZW, the code generated is fully embedded, which means that the encoder can terminate the encoding at any point and the image can be reconstructed with quality that corresponds to the number of bytes of the code. The EZW may outperform the classic DCT based still image coding, and might have competitive performance at high bit rate video coding. At very low bit rate video coding, however, like DCT based motion compensation algorithm, it suffers from significant distortion too, due to the very limited bits available, and the usage of two passes to code the amplitude and position information [69] or the usage of the priority-position coding relying on arithmetic coding to encode the position index [27].

In this chapter, firstly, the mathematical background of wavelet transform is discussed. Then the applications of the wavelet transform in image coding is discussed. Finally, a new algorithm of video coding for very low bit rate applications is presented. This algorithm [48] is based on (1) dense motion field, which can achieve better motion compensation than sparse (say, block-wise) motion field; (2) DCT applied to the dense motion field to drastically save overhead information; (3) region-based segmentation with morphological techniques which can segment video frames

into different regions according to their content significance; (4) discrete wavelet transform (DWT) applied to residual data with adaptive bit allocation. Consequently, this algorithm avoids annoying block artifacts, thus making reconstructed video frames much more visually pleasant, which is similar to the previously proposed DCT coded dense motion field coding method. Moreover, it provides more flexibility, which means that it can adaptively allocate bits to the regions according to their content significance, while maintains similar bit-rate to H.263.

5.1 Short Time Fourier Transform (STFT)

Wavelet coding is a kind of transform coding schemes. It has good localization in both time domain and frequency domain. It has potential power to be widely used in digital image coding and video coding besides its applications in other areas. Fourier transform has been widely used in signal processing, such as communications, speech, control, system analysis, etc. It has perfect frequency resolution, but there is no time resolution. So that windowing is used to provide time localization, that is the short-time Fourier transform.

There is a fundamental difference between the STFT and wavelet transform. In the STFT, at any analyzing frequency of ω_0, changing the window width will increase or decrease the number of cycles of ω_0 inside the window. In the wavelet transform, at a carrier frequency of ω_0, variation of window width causes dilation or compression, namely, the carrier frequency becomes $\frac{\omega_0}{a}$ for a window width change from T to aT. And the number of cycles inside the window remains constant. The frequency resolution is directly proportional to the window width in both the STFT and the wavelet transform.

Generally, a transform for video signals can be described as

$$\Psi = W^T \Phi W \tag{5.1}$$

where Φ is $N \times N$ matrix whose elements are samples of an image. W could produce a transformed image matrix Ψ that is sparse and with most of its large magnitude elements concentrated in a small region of Ψ. This is the idea of decorrelation and energy compaction by transformation, typically, only 15% of the elements of Ψ need be retained without an adverse effect on the image quality.

Most transforms which are widely used in image, video and audio signals are orthogonal transforms. Because the components of the transform bases are orthogonal to each other and dropping any component of an orthonormal transformation results in a truncated representation that could still be best in the least square sense. STFT maps a single dimensional signal into the time-frequency domain,

$$STFT(\tau, \omega) = \frac{1}{2\pi} \int e^{-j\omega t} s(t) h(t - \tau) dt. \tag{5.2}$$

where $h(t)$ is a Gaussian window, or other windows (Hamming, Blackman, etc.) A time localization can be obtained by suitably pre-windowing the signal. Good time resolution of STFT requires a short window, and good frequency resolution of STFT requires a narrow band filter. Therefore, the joint time-frequency resolution of STFT is inherently limited, improving the time resolution results in a loss of frequency resolution and vice versa.

5.2 Wavelet Transform

A wavelet is an orthogonal function which can be applied to a set of finite data. Wavelet theory is based on the multiresolution[1] analysis concept. A multiresolution wavelet package is shown in Figure 5.1. When the wavelet package is used to code the video signals, we can get different bitstream with different resolutions. The video signals pass through the wavelet package, we get different subimages. Figure C.1 is the original image; Figure C.2 gives four subimages and their size is approximately

[1]Multiresolution: we approach the original signal by successively adding details to it, successively refine it.

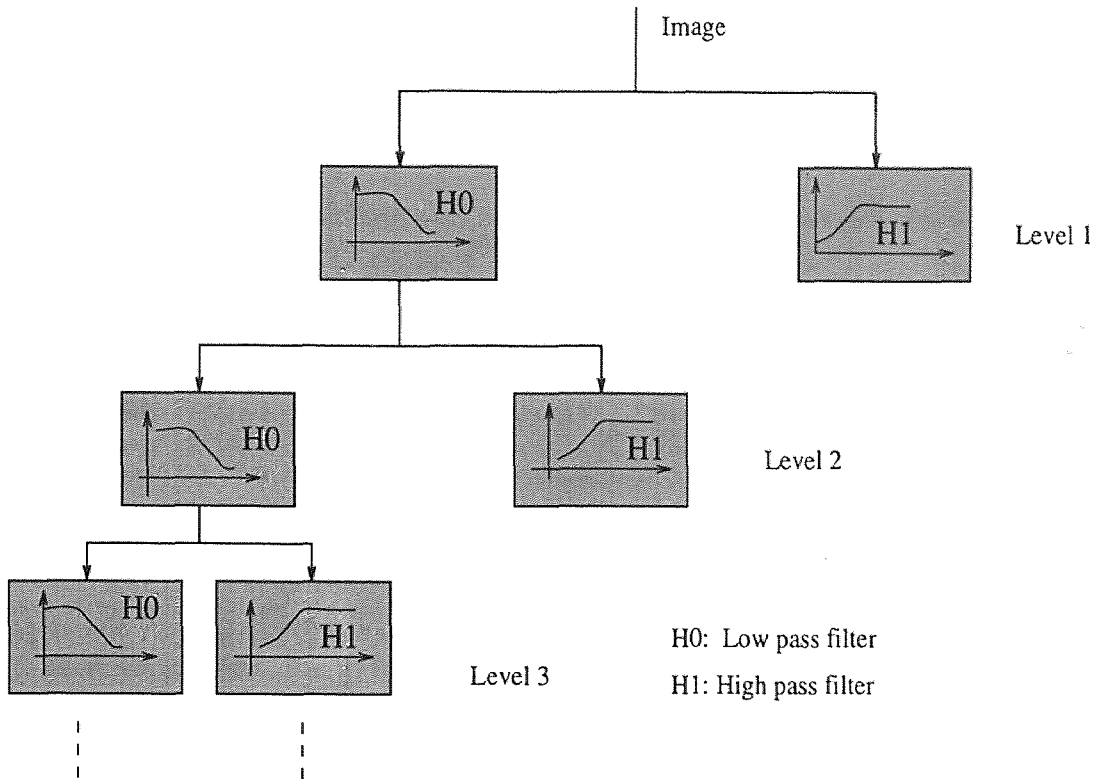


Figure 5.1 The multiresolution wavelet package

a quarter of the original image size. The top-left subimage in Figure C.2 is further decomposed into another four subimages, shown in Figure C.3.

Wavelet transform is generally given as

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a}), a > 0, b \in R \tag{5.3}$$

ψ is a fixed function, called "mother wavelet," that is well localized both in time and frequency domains. The baby wavelets are generated from the dilation and/or translation of the mother wavelet. In Equation 5.3 a is a scale, it scales a function by compressing or stretching it; b is a translation of the wavelet function along the time axis.

A continuous wavelet transform of a function $f \in L_2(R)$ is

$$T[f(a,b)] = < f, \psi_{a,b} > = \frac{1}{\sqrt{a}} \int f(t)\psi(\frac{t-b}{a})dt. \tag{5.4}$$

The inverse transform is

$$f(t) = C_\psi^{-1} \int_0^{-\infty} \frac{da}{a^2} \int_{-\infty}^{\infty} T[f(a,b)]\psi_{a,b}(t)db \tag{5.5}$$

where

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} < +\infty \tag{5.6}$$

In practice, the continuous wavelet transform can only be computed on a discrete grid of points.

5.2.1 Examples of Wavelet Transform

There are lots of wavelet functions devised, and the following are some typical ones,

1. Modulated Gaussian (Morelet), shown in Figure 5.2.

$$\psi(t) = e^{j\omega_0 t} e^{-\frac{t^2}{2}} \tag{5.7}$$

$$\psi(\omega) = \sqrt{2\pi} e^{-\frac{(\omega-\omega_0)^2}{2}} \tag{5.8}$$

2. Second derivative of a Gaussian, shown in Figure 5.2.

$$\psi(t) = (1-t^2)e^{-\frac{t^2}{2}} \tag{5.9}$$

$$\psi(\omega) = \sqrt{2\pi}\omega^2 e^{-\frac{\omega^2}{2}} \tag{5.10}$$

3. Shannon, shown in Figure 5.3.

$$\psi(t) = \frac{\sin(\frac{\pi t}{2})}{\frac{\pi t}{2}} \cos(\frac{3\pi t}{2}) \tag{5.11}$$

$$\psi(\omega) = \begin{cases} 1 & \pi < |\omega| < 2\pi \\ 0 & otherwise \end{cases} \tag{5.12}$$

4. Haar, shown in Figure 5.3.

$$\psi(t) = \begin{cases} 1 & 0 \le t \le \frac{1}{2} \\ -1 & \frac{1}{2} \le t < 1 \\ 0 & otherwise \end{cases} \tag{5.13}$$

$$\psi(\omega) = je^{-j\frac{\omega}{2}} \frac{\sin^2(\frac{\omega}{4})}{\frac{\omega}{4}} \tag{5.14}$$

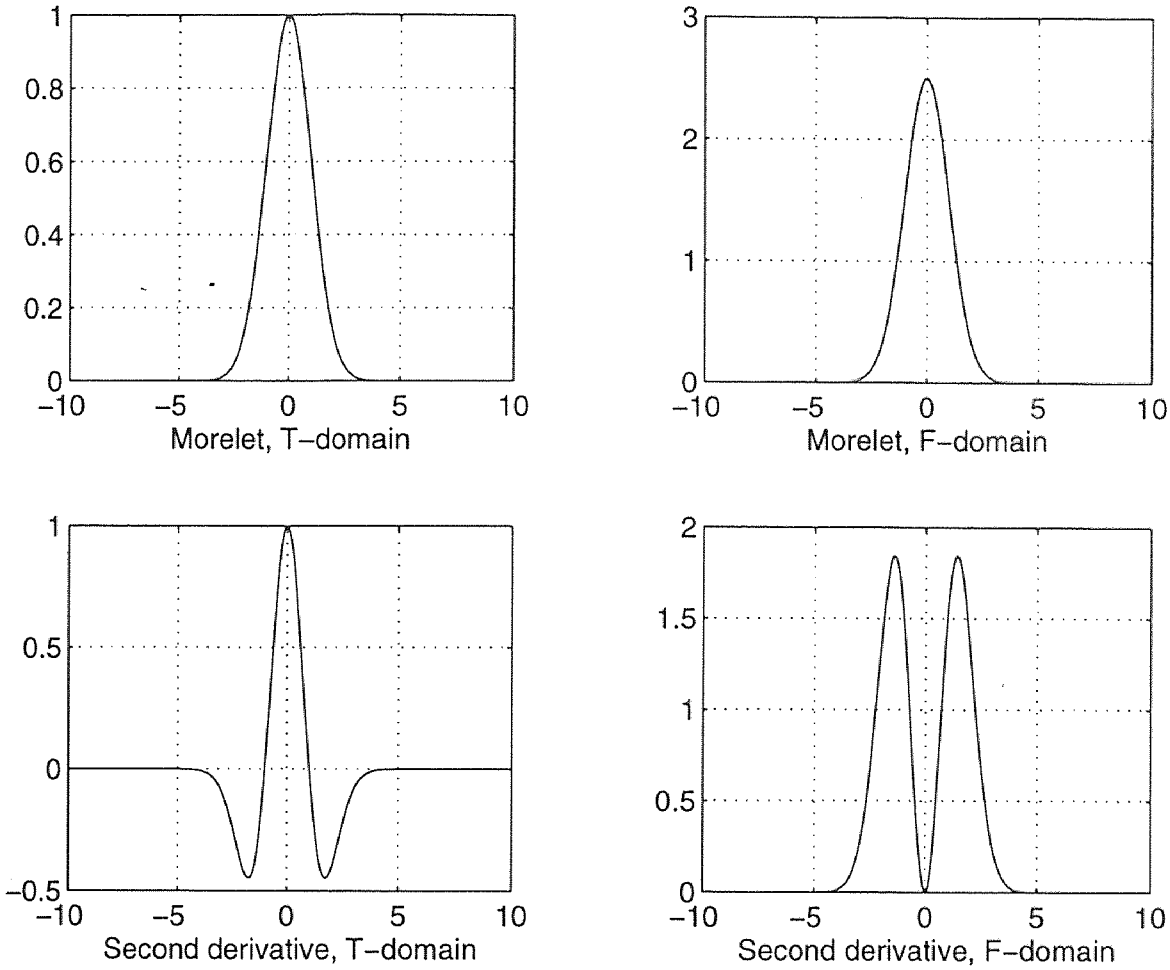


Figure 5.2 The Gaussian and the 2nd derivative Gaussian wavelets

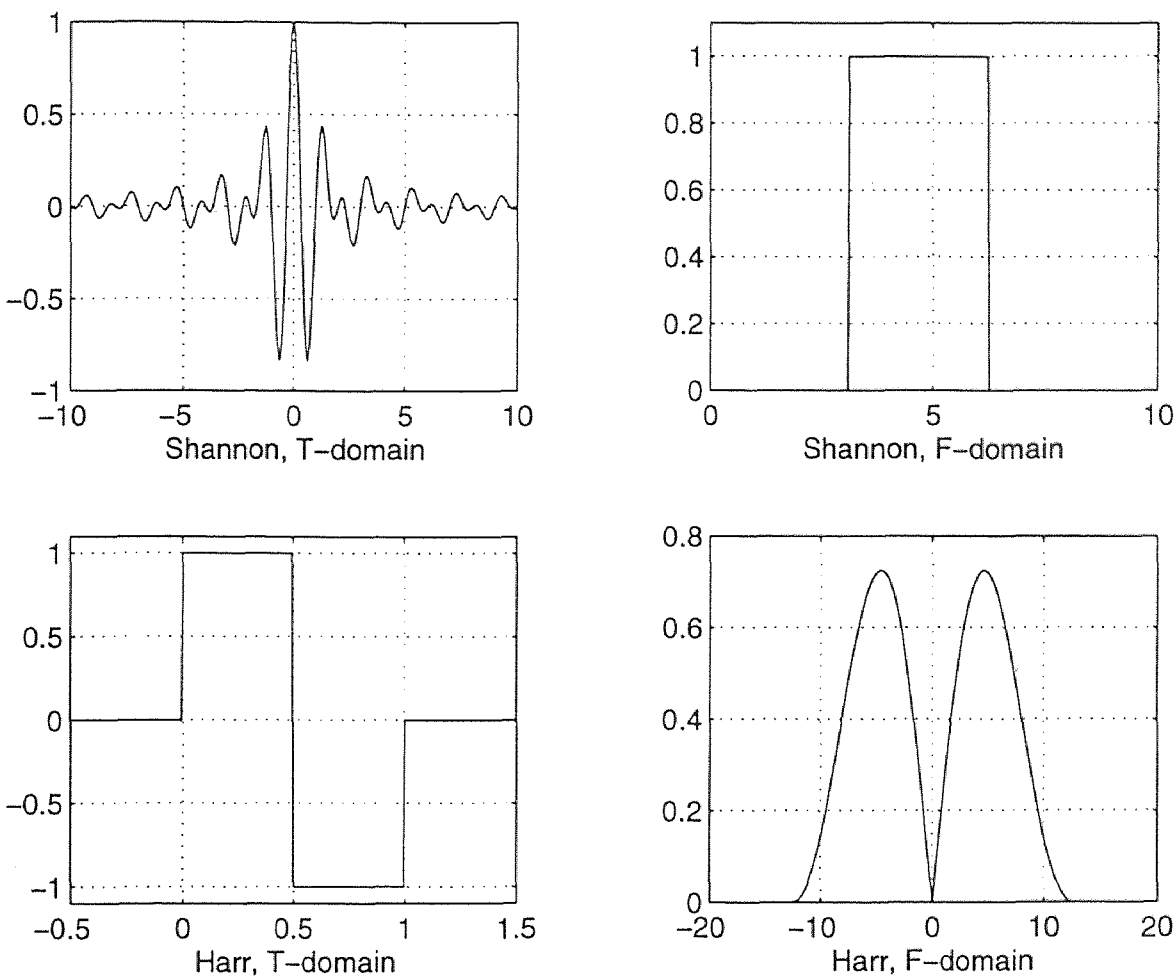


Figure 5.3 The Shannon and the Haar wavelets

5.2.2 Wavelet Transform Analysis in Video Coding

The wavelet function $\psi(t)$ in practice should have compact support in order to have good time localization. Compact support is that the wavelet transform is able to operate on a finite set of data. The value of the transform coefficients c_k is determined by constraints of orthogonality and normalization. Generally, the area under the wavelet curve over all space should be unity, which requires that

$$\sum_k c_k^2 = 2 \tag{5.15}$$

Since most of the information exists in the output of the low-pass filter, one can take this filter output and transform it again. This process can be repeated if necessary. this is the wavelet transform dilations shown in Figure 5.4. There are three parameters which describe a wavelet transform, the filter length, which affects the number of butterfly stages required in the lattice filter, and the number of delays involved; the block size of the input data to be transformed; the number of nested levels. Unfortunately the wavelet reconstruction is carried out such that the deepest nested dilation calculated in the decomposition must be first reconstructed. This means that all transformed data must be saved in memory. So the size of the input blocks and the resolution of the wavelet decomposition are limited constrained by the memory size.

The delay necessarily existing in convolution results in an output stream which is half the length of the input plus the number of delay stages. It is not desirable to output all these numbers of delay stages, because that would require an increasingly greater rate of processing for the output of each decomposition level. There are two ways to solve this problem, one is to truncate the output and discard the extra endpoint values. This gives a reconstruction which is very close to the original if it is done correctly. The second is circular, the input stream is wrapped around and fed back through the filter, and the new outputs replace the first outputs. This method

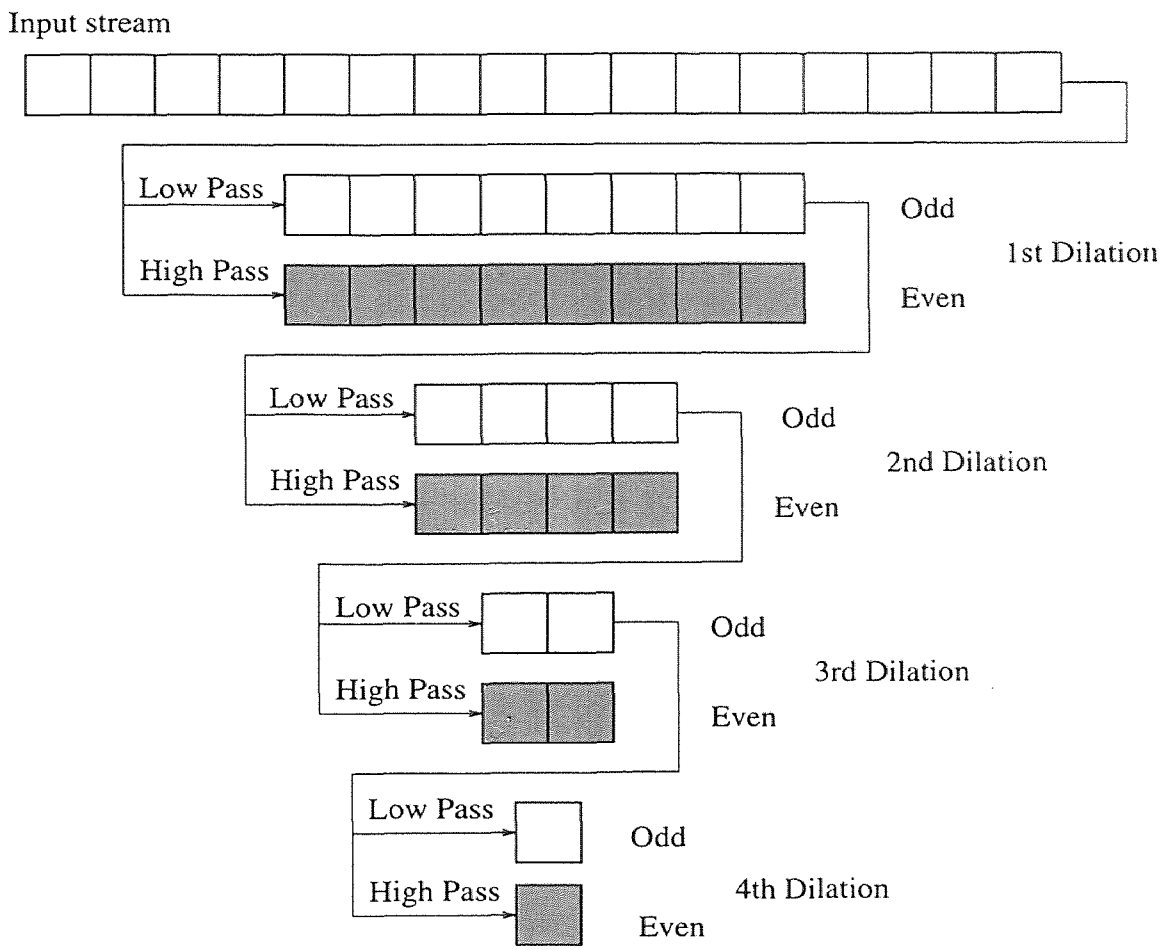


Figure 5.4 Wavelet dilation

yields a perfect reconstruction, but makes scheduling and data routing even more difficult.

The wavelet transform can be divided into

1. continuous wavelet transform;
2. discrete parameter wavelet transform;
3. discrete time wavelet transform;
4. and discrete wavelet transform (DWT). The DWT is given as

$$DWT(m,n) = 2^{\frac{m}{2}} \sum_k s(k)\psi(2^{-m}k^{-n}). \tag{5.16}$$

5.2.3 Perfect Reconstruction

There are two kinds of multiresolution. One is the full balanced tree, which is shown in Figure 5.5, and every subband can be further decomposed. The other is unbalanced tree, in which only the lowest subband will be further decomposed, this is the so-called wavelet package, shown in Figure 5.1.

Perfect reconstruction (PR) is a basic requirement in filter bank designing, even though after compression, the reconstructed image will only be an approximation of the original. From Figure 5.6, we can obtain the following solution for perfect reconstruction.

$$\hat{X}(z) = T(z)X(z) + E(z)X(-z) \tag{5.17}$$

$$T(z) = \frac{1}{2}[G_0(z)H_0(z) + G_1(z)H_1(z)] \tag{5.18}$$

$$E(z) = \frac{1}{2}[G_0(z)H_0(-z) + G_1(z)H_1(-z)] \tag{5.19}$$

The conditions for PR are:

$$T(z) = z^{-d} \tag{5.20}$$

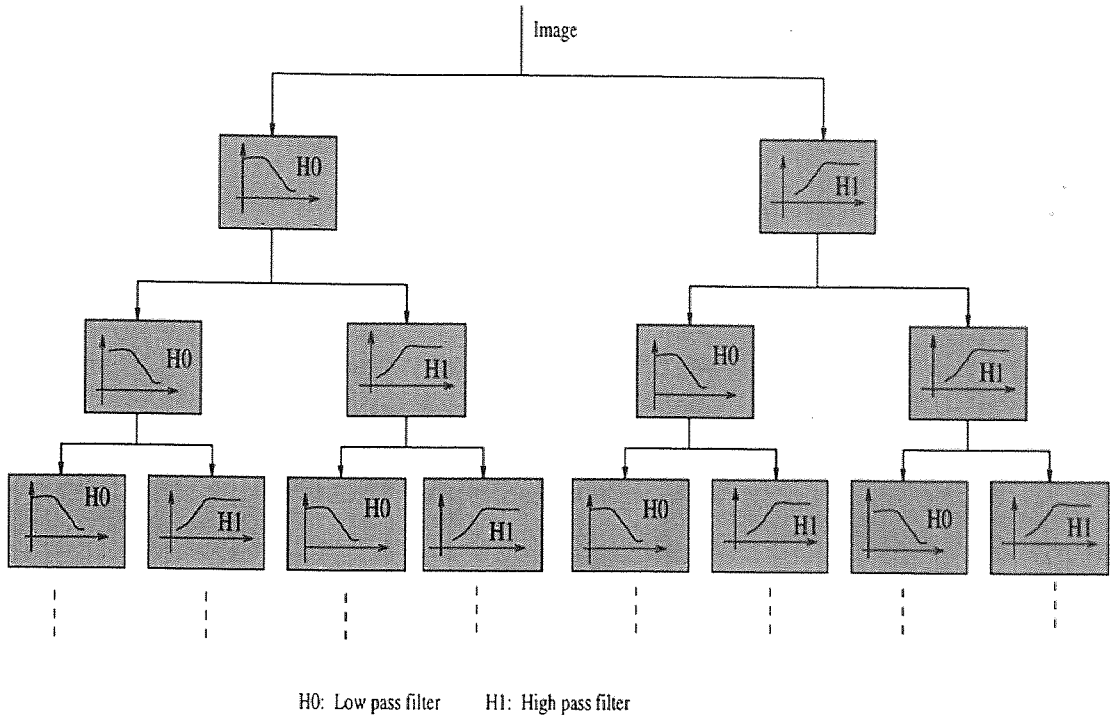


Figure 5.5 Balanced binary tree subbands

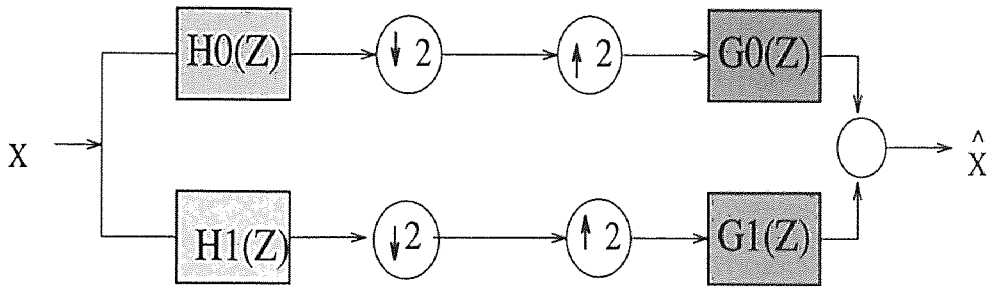


Figure 5.6 Biorthogonal wavelet filters

$$E(z) = 0 \tag{5.21}$$

$$\tag{5.22}$$

For paraunitary (lossless) FIR filters of order p, which is an even number, we have

$$H_0(z) = -z^{-(p-1)}G_0(-z^{-1}) \tag{5.23}$$

$$T(z) = z^{-(p-1)}[R(z) + R(-z)] \tag{5.24}$$

with

$$R(z) = G_0(z)G_0(z^{-1}) \tag{5.25}$$

$$G_0(z) = \sum_{l=0}^{p-1} g_0(l)z^{-l} \tag{5.26}$$

$$h_0(l) = (-1)^l g_0(p-1-l) \tag{5.27}$$

$$h_1(l) = (-1)^{l+1} g_0(l) \tag{5.28}$$

$$g_1(l) = g_0(p-1-l) = h_0(p-1-l). \tag{5.29}$$

Another very useful filter bank is QMF (quadrature mirror filter) which satisfies

$$| H_0(\omega) |=| G_0(\pi - \omega) | = | G_0(\omega + \pi) | \tag{5.30}$$

$$| G_0(\omega) |^2 + | H_0(\omega) |^2 = 1 \tag{5.31}$$

The requirement for lowpass PR-QMF is

$$| G_0(\omega) |^2 + | G_0(\omega + \pi) |^2 = 1 \tag{5.32}$$

while for highpass PR-QMF

$$| H_0(\omega) |^2 + | H_0(\omega + \pi) |^2 = 1 \tag{5.33}$$

$$G_0(\omega)H_0^*(\omega) + G_0(\omega + \pi)H_0^*(\omega + \pi) = 0 \tag{5.34}$$

or

$$G_0(z)H_0(z^{-1}) + G_0(-z)H_0(-z^{-1}) = 0 \tag{5.35}$$

5.3 DWT and Video Coding

The application of digital video coding has enormously increased in the past decade, such as teleconferencing, video phone, DVD, DSS, DBS, HDTV, WWW, etc. To meet requirements of these applications, several video coding standards, such as ISO/IEC MPEG 1/2 and ITU-T H.261, have been successfully developed. For very low bit rate video coding, H.263/N is almost finished, and MPEG-4 is expected to be completed in 1998. Many coding schemes have been devised, such as block matching based DCT (BMDCT), DWT, model based, object-based, fractal, etc. The BMDCT is widely used in the standards. It works very well at high bit rate applications, but at very low bit rate, it suffers from the well-known block artifacts, an intrinsic problem of the block-wise model. Since only very limited bits are available at very low bit rate, the block artifacts then become very severe and the quality of the reconstructed images are degraded considerably. It is especially true in facial video coding, because most of the facial expressions involve nonrigid motion.

An efficient way to code video sequence is to divide the coding process into two stages, motion compensation and residual image coding. DWT has already been shown to outperform DCT technique when used to code still images, generally, 2 to 3 dB will be enhanced over DCT. While DWT is used to code video sequence in order to reach better performance, it faces two major problems. First, if the block matching is used in motion compensation stage, then block artifacts will manifest themselves in the residual images, and many sharp edges can be observed. After DWT, the number of DWT coefficients with large magnitude in the high frequency subbands will increase, coding these coefficients requires more bits, hence deteriorating the DWT performance. Second, the DWT is applied to the whole image, and bits are allocated to subbands according to their variances only, ignoring the visual perception. Therefore the visual quality of the reconstructed image is not optimum to human visual system.

To overcome blocking artifacts, instead of the block based motion vectors, the dense motion field is used in our proposed algorithm for motion compensation, because dense motion vectors can reflect true motion and intensity variation more accurately than the block based sparse motion vectors do, especially near the motion boundaries, hence, the prediction error is decreased, and the number of sharp edges is reduced. To compress the overhead information resulted from using dense motion fields, DCT is applied to the highly correlated dense motion vectors as discussed in the previous chapter. To optimize the bit allocation to different regions according to their visual perceptual significance, morphological segmentation techniques are used to segment video frame. And the DWT is then applied to residual data region-wise, and the bits are allocated depending on not only the subband variances but also visual perception. Experimental results demonstrate that the block artifacts have been eliminated. Furthermore for regions with significant contents, both of the PSNR and the subjective visual effect have been improved. Due to DCT applied to dense motion field and other thresholding techniques utilized, the bit rate is compatible with that achieved by H.263.

Wavelet transform has not only good localization in both frequency and time domains but also flexible scalibility in video coding. Figure 5.7 explains how to deal with the bitstream resolution scaling. The input video signals are filtered and downscaling, then we get a base layer bitstream and an enhancement layer stream. For the low resolution application, only a base layer bitstream is needed. For high resolution applications, both of them are required. This is very useful for different applications, especially in video server and multimedia distributions.

5.4 New Algorithm

A new algorithm is proposed in this paper to reduce the block artifacts resulted from the block-wise motion estimation and motion compensation. Dense motion fields are

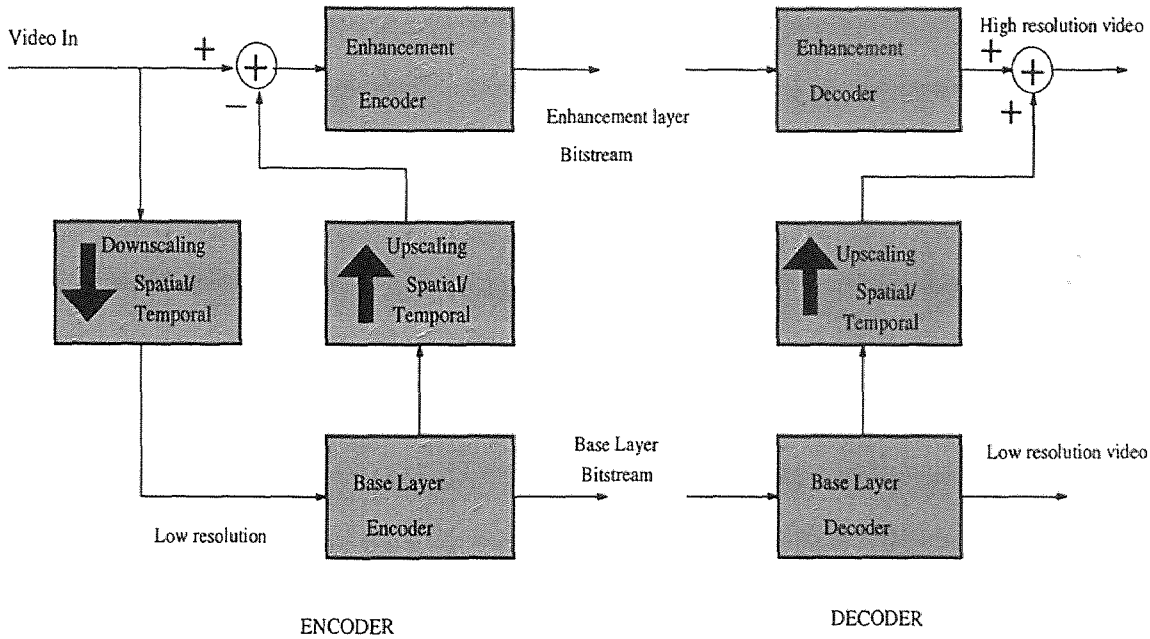


Figure 5.7 Wavelet transform scaling

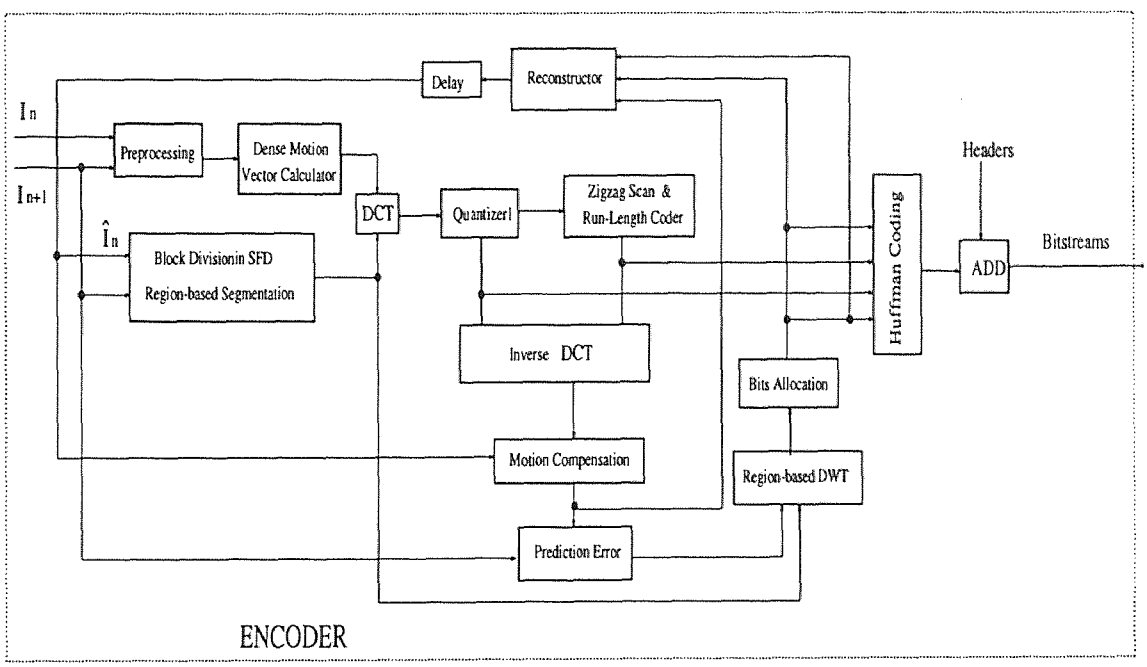


Figure 5.8 The encoder of the proposed algorithm

exploited to predict motion in this algorithm, since pixel based motion compensation turns out to be much more accurate than that of the block based, this is especially true for motion estimation near motion boundaries as well as for non-rigid motion. To optimize bit allocation to different areas, morphological segmentation is used to segment different regions according to their significance, based on the light intensity, neighborhood information, and motion estimation. DWT is then applied to the prediction error of these regions and bits are adaptively allocated to regions according to their content significance. An overall block diagram of the proposed algorithm is shown in Figure 5.8. For videoconferencing the encoding algorithm assumes that the video sequence is a set of head-and-shoulder type images and that the subjectively most important features are the eyes, mouth, and facial expression. It showed that people, when observing a moving head-and-shoulder image, concentrate mainly on the speaker's face.

5.4.1 Dense Motion Field Estimation

In typical applications such as teleconferencing, videophone, the frame-to-frame change is relatively slow. Therefore, most areas in the frame will not involve motions. To save computation of the dense motion field, a preprocess is used to detect motion areas. This preprocess is the same as that discussed in Chapter 7.

5.4.2 DCT Coding of the Motion Vectors

In general, the motion vectors are transmitted as side information in motion compensated video coding schemes. To transmit all the dense motion vectors needs many bits, and it excesses the bit budget. The motion in a face is constrained by some muscles and skin, and it is highly correlated with that of its neighborhood. Hence, in the new algorithm we use the DCT to code the dense motion vectors. Please refer to Chapter 7 for an analysis.

5.4.3 Region-Based Segmentation

In order to adaptively assign bits to regions with different visual perception, these regions need to be segmented. The human visual perception is not linear with the light intensity of the image. We have noticed that the intensity variant in dark areas is less perceptual than that in bright areas; noise in the areas with an abundance of texture is generally ignored by human eyes; human attention is usually focused on human face rather than shoulders or background. For the head-and-shoulder type sequence, in order to use visual perceptual weighting, the image is decomposed into three regions, significant region (human face), less significant region (shoulders), and non-significant region (background) with the following formula,

$$(1-\lambda)||I(i,j)|| + \lambda Var > T \tag{5.36}$$

where Var is the pixel variance obtained within its neighborhood, generally a Gaussian mask is used; λ is a weighting coefficient, from 0.2 to 0.8; $I(i,j)$ is the intensity of a pixel at position (i,j); T is a threshold which is decided by the contents of the image and is region dependent. With different T, different regions will be segmented. Some small regions will be segmented in this stage. To code these small regions separately is neither economic nor necessary. The morphological erosion and dilation are used to eliminate and merge these small regions. Unlike the object-based, the region-based techniques rely on the regions rather than objects. For example, the objects of eyes and nose are segmented, they will be possibly coded individually with object-based techniques, while using region-based techniques, they will be merged an coded together because they belong to the same class of significant region and they are adjacent. The three-class regions of Miss America sequence are segmented in Figure 5.9.

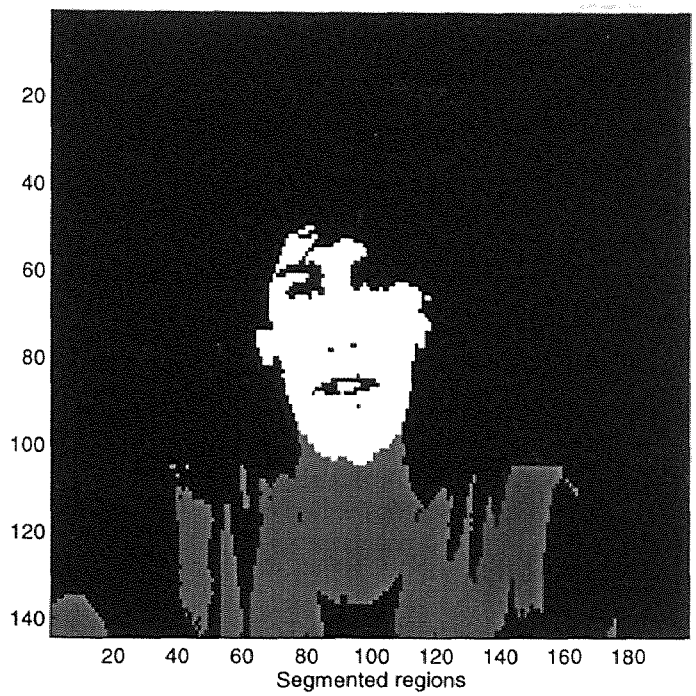


Figure 5.9 Three-class regions

5.4.4 Adaptive DWT Coding of Residual Pictures

Like all other motion compensated coding schemes, this algorithm also transmits error information. In order to find out the predictive errors, we reconstruct an image from the motion vectors and the previous frame as follows. After we obtain the nonzero quantized DCT coefficients and their positions, we reorder these coefficients according to their positions, and the nontransmitted coefficients are replaced with zeros. Then the inverse DCT is applied to recover the motion vectors. Using these motion vectors together with bilinear interpolation if necessary, we estimate the current frame from the previous one. The difference between the estimated and actual current frames thus gives the predictive error. Because the pixel-based motion vectors rather than block-based motion vectors are used, the predictive errors are much less than that obtained by the block matching based DCT. The less the predictive errors, the lower the correlation they have, and the less effective the DCT, if applied, will be. For this reason, we do not apply the DCT to the error information. Instead, we use DWT to code the prediction error.

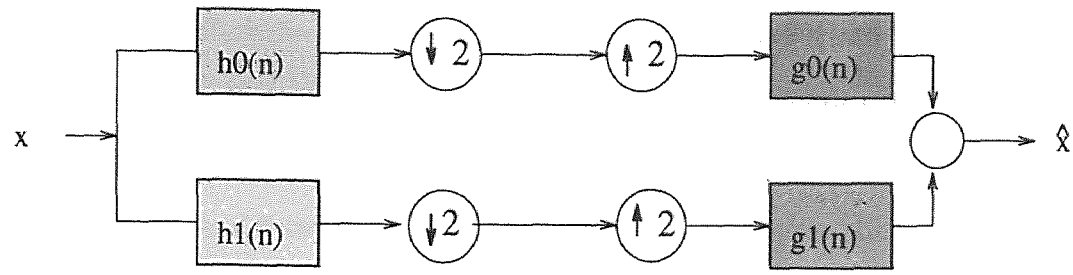


Figure 5.10 Discrete wavelet transform

Wavelet transform is well known for its good localization in both time and frequency domains. Its perfect reconstructed filter banks (best bases) are widely used in image and video coding. Figure 5.10 gives us a diagram of a simplest wavelet transform. A signal sequence x input to a pair of bi-orthonormal filters $h_0(n)$ and $h_1(n)$, two subsequences are produced and down sampled by a factor of two, these two sub-signals locate in different frequency bands so that they can be coded separately. At the decoder end, they will be upsampled by a factor of two, and then pass through a pair of synthesis filters, $g_0(n)$ and $g_1(n)$, and added up to get the reconstructed signal. DWT is traditionally applied to rectangular shaped regions rather than an arbitrary shaped region. The regions segmented in the previous section are generally not rectangular, so that they will be regularized into rectangular (in our experiments) before they are input to the DWT. The significant region is coded first, it can use up to 100% bits. The less significant region will not be coded every frame. Generally, coding of every other frame is sufficient. If it is coded, it will use about 20% bits assigned to the frame. The non-significant region will be coded every fifteen frames and will use about 10% bits of the frame.

5.5 Experimental Results

To evaluate the performance of our new algorithm, we applied it to Miss America sequence and Claire sequence in the format of QCIF. The dense motion field is calculated with the modified Horn and Schunck algorithm (refer to the previous

chapter). The coefficients of DCT of motion vectors are quantized by 16 levels and the DWT coefficients are quantized by 32 levels. In the experiment, the threshold T_1 is chosen as 3.0, T_2 is 10.0, and T_3 is adaptive, ranging from 15 to 30. The Daubechies 4-tap and 6-tap wavelets [16] are used to decompose the residual images. At frame rate of 10 frames/s (30 frames/s, 2 frames are skipped for every 3 frames), the proposed algorithm achieves a bit-rate of 12 kbps for interframe coding and the PSNR of about 34.8 dB for the significant region. Both are averaged for the first 36 frames of the sequence. These reconstructed frames are free of blocking artifacts. The 21st reconstructed frame is shown in Figure 5.11 (b), the corresponding frame reconstructed with H.263/TMN5 is shown in Figure 5.11 (c), and the original 21st frame in Figure 5.11 (a). It is obvious that prominent blocking effects can be observed in the reconstructed frame with H.263/TMN5.

5.6 Conclusion and Discussion

A new region based DWT video coding algorithm which is based on dense motion field is proposed and discussed in this chapter. This algorithm utilizes DCT technique to code dense motion field, morphological segmentation to decompose image into different regions, and DWT to code the residual images. The bits are adaptively assigned to the regions according to their visual perceptual significance, and different refresh rates are applied to different regions as well. It improves the reconstructed image visual quality subjectively by reducing the block artifacts and assigning more bits to the significant regions. It achieves 12 kbps of interframe coding while working on Miss America sequence. This algorithm is expected to be used in various very low bit rate applications, such as video phone, teleconferencing, etc. Compared with the algorithm discussed in Chapter 4, and the H.263, for Miss America sequence, this algorithm achieves a bit rate of 12 kbps for interframe coding, and a PSNR of 37.04 dB for the whole image and 34.8 dB for the significant regions, while the algorithm

(a) The 21st original frame of Miss America sequence

in Chapter 4 achieves about 11 kbps, PSNR of 37.1 dB for the whole image and 34.3 dB for the significant regions, and the H.263 achieves 10.5kbps, PSNR of 37.4 dB for the whole image and 34.2 dB for the significant regions. This algorithm and the algorithm presented in the previous chapter have no annoying blocking artifacts, hence, their visual quality of the reconstructed frames is better than that of the H.263/TMN5.

(b) The recovered frame 21 with DWT, Miss America

(c) The reconstructed 21st frame with H.263/N

Figure 5.11 Miss America

CHAPTER 6

SEGMENTATION-BASED STEREO SEQUENCE VIDEO CODING

Segmentation-based video compression has been a very active research area over the past few years[4]. It has been viewed as a potential alternative to traditional schemes suffering from the "blockiness" of image intensities at very low bit rate video coding. In conventional motion compensation methods, a translational motion model of square blocks is employed. The independently moving square blocks lead to visible block artifacts in the very low bit rate video coding. A relative large predictive error along moving object boundaries is expected. To solve this problem, a new segmentation-based motion compensation algorithm is proposed in this chapter. This algorithm utilizes stereo images to estimate the 3-D information (3-D coordinates and 3-D velocity). The 3-D information along with the connectivity and velocity fields are used to segment objects. A chain code is used to efficiently code the contour of the segmented objects. Because the 3-D information is used, the occlusion can be handled. The number of velocities is equal to the number of segmented objects, and the segmentation-based method essentially has a superior prediction ability for moving objects. Therefore, this algorithm can achieve very high compression ratio which can be up to several thousands at acceptable reconstructed image quality in our experimental works. It is noted that the 3-D information (say, depth) is utilized in the segmentation. In doing so, the disparity vector fields have to be determined. This disparity vector field can late be used to predict one image of the stereo image sequence from the other in stereo video coding.

6.1 Introduction

High compression video coding is an essential technique for digital video applications, such as videophone, digital video disc, etc. For these purposes, there are several

international standards developed, e.g. ITU-T H.261, ISO MPEG-1/2. However, these coding schemes do not achieve a high enough level of compression performance for mobile video communication or full motion video transmission through integrated services digital network (ISDN), or public switched telephone network (PSTN). In recent years, ISO MPEG-4 and ITU-T LBC groups have started active researches on the standardization of audiovisual services at very low bit rate through PSTN, LAN, ATM and mobile networks. Many advanced techniques have been studied to avoid blocking effects resulting from classic block matching motion compensated DCT algorithm. Among these techniques, region-based [7], object-oriented [60][67][55], and segmentation-based [35][52][75][59] techniques are based on the models of moving regions, moving objects, and segmentation, respectively. All of these techniques inherently require an effective video segmentation algorithm. Video segmentation is a process of partitioning each frame of a video sequence into disjoint homogeneous regions. Recursive shortest spanning tree (RSST) or centroid linkage region growing (CLRG) have been often used as a segmentation algorithm for region-based coding. However, RSST and CLRG are believed to be impractical in real-time processing because of irregular structure in nature. In the following, we will give a new algorithm which is used for stereo video sequence and based on object-segmentation.

6.2 Camera Setting for Stereo Sequence and 3-D Information Calculation

At least two cameras are needed for taking stereo images. The general camera setting for stereo images is shown in Figure 6.1. The distance between two camera optic centers O^R and O^L is the base line l. Z^L, Z^R are the depths of the left and right cameras, and f^L, f^R are the left and the right camera focal length, respectively. In the following text, superscript L denotes the left camera and R denotes the right camera. The parameters (X, Y, Z) without superscript are the 3-D world coordinates. From

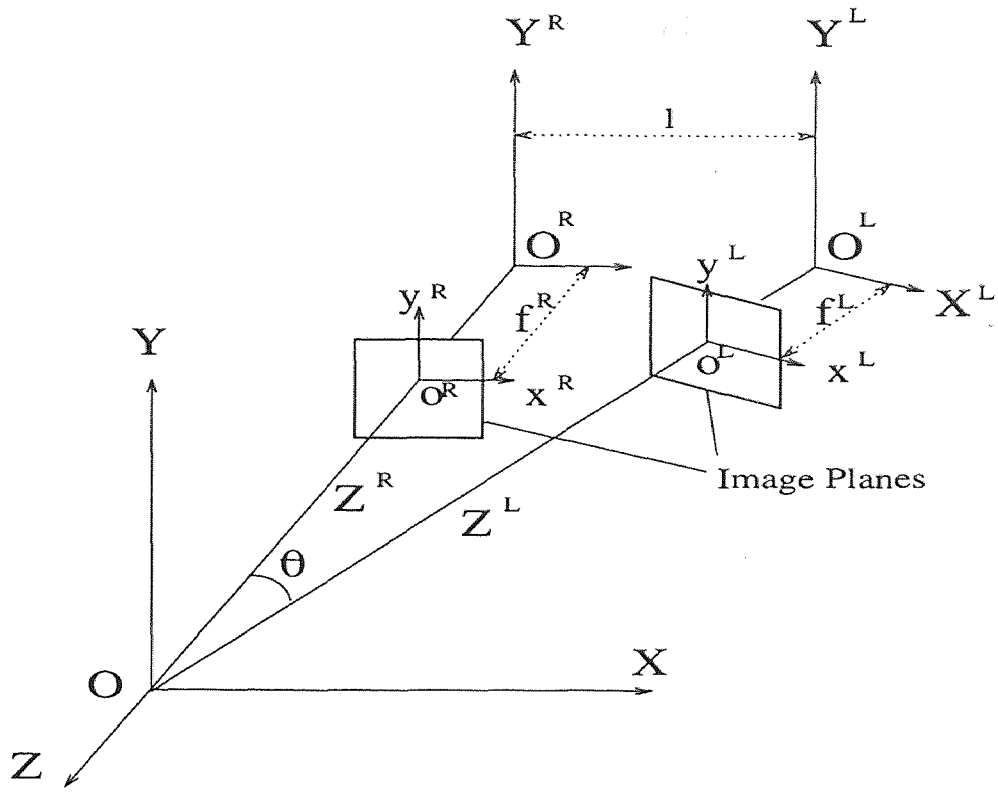


Figure 6.1 The camera setting for stereo images

the perspective projection theory, for the left camera we have

$$x^L = -\frac{f^L}{Z^L} X^L \tag{6.1}$$

$$y^L = -\frac{f^L}{Z^L} Y^L \tag{6.2}$$

where (x^L, y^L) is the coordinate in the image plane. X^L, Y^L, Z^L are the left camera coordinates in 3-D space.

Let $s^L = Z^L$, and rewrite the above two equations for the left camera as

$$\begin{bmatrix} s^L x^L \\ s^L y^L \\ s^L \end{bmatrix} = \begin{bmatrix} -f^L & 0 & 0 \\ 0 & -f^L & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X^L \\ Y^L \\ Z^L \end{bmatrix} \tag{6.3}$$

The relationship between the left camera 3-D coordinate system and the world coordinate system can be expressed with a rotation matrix R^L and a translation matrix T^L. Thus, we have

$$\begin{bmatrix} X^L \\ Y^L \\ Z^L \end{bmatrix} = R^L \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T^L \tag{6.4}$$

Inserting it into Equation 6.3, we have the relationship between the left image coordinates and the world coordinates as following:

$$\begin{bmatrix} s^L x^L \\ s^L y^L \\ s^L \end{bmatrix} = F^L R^L \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + F^L T^L \tag{6.5}$$

where

$$F^L = \begin{bmatrix} -f^L & 0 & 0 \\ 0 & -f^L & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{6.6}$$

Similarly, we have the following equation for the right camera,

$$\begin{bmatrix} s^R x^R \\ s^R y^R \\ s^R \end{bmatrix} = F^R R^R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + F^R T^R \tag{6.7}$$

The elements in the matrices of $R^L, T^L, F^L, R^R, T^R, F^R$ are determined by the parameters of the camera settings. From Equation 6.5, we have

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R^{L^{-1}} F^{L^{-1}} \left(\begin{bmatrix} s^L x^L \\ s^L y^L \\ s^L \end{bmatrix} - F^L T^L \right) \tag{6.8}$$

and plug it into Equation 6.7,

$$\begin{aligned} \begin{bmatrix} s^R x^R \\ s^R y^R \\ s^R \end{bmatrix} &= F^R R^R R^{L^{-1}} F^{L^{-1}} \left(\begin{bmatrix} s^L x^L \\ s^L y^L \\ s^L \end{bmatrix} - F^L T^L \right) + F^R T^R & (6.9) \\ &= F^R R^R R^{L^{-1}} F^{L^{-1}} \begin{bmatrix} x^L \\ y^L \\ 1 \end{bmatrix} s^L - F^R R^R R^{L^{-1}} T^L + F^R T^R. & (6.10) \end{aligned}$$

We reorder them,

$$\begin{bmatrix} x^R \\ y^R \\ 1 \end{bmatrix} s^R - F^R R^R R^{L^{-1}} F^{L^{-1}} \begin{bmatrix} x^L \\ y^L \\ 1 \end{bmatrix} s^L = F^R T^R - F^R R^R R^{L^{-1}} T^L \tag{6.11}$$

it is rewritten as,

$$\begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} s^R \\ s^L \end{bmatrix} = c \tag{6.12}$$

where

$$\begin{aligned} a &= \begin{bmatrix} x^R & y^R & 1 \end{bmatrix}^T & (6.13) \\ b &= -F^R R^R R^{L^{-1}} F^{L^{-1}} \begin{bmatrix} x^L \\ y^L \\ 1 \end{bmatrix} & (6.14) \\ c &= F^R T^R - F^R R^R R^{L^{-1}} T^L. & (6.15) \end{aligned}$$

We obtain s^R, s^L,

$$\begin{bmatrix} s^R \\ s^L \end{bmatrix} = \begin{bmatrix} a^T a & a^T b \\ b^T a & b^T b \end{bmatrix}^{-1} \begin{bmatrix} a^T \\ b^T \end{bmatrix} c, \tag{6.16}$$

and the 3-D coordinates are

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R^{L^{-1}} F^{L^{-1}} \left(s^L \begin{bmatrix} x^L \\ y^L \\ 1 \end{bmatrix} - F^R T^R \right) \tag{6.17}$$

From Equations 6.13 to 6.15, all parameters are known except one pair of (x^R, y^R) or (x^L, y^L). In this algorithm, we use unified optical flow method to find (x^R, y^R) or (x^L, y^L) depending on which one known and which one unknown [71]. One way in the lab to take stereo images as follows. A camera takes a picture at the left camera position, and then moves to the right camera position, takes another picture while all other conditions are preserved. This is equivalent to the case that the left camera and the identical right camera take pictures at the same time. Using the unified optical flow technique, we can find the displacement between these two images. For a simple case, the left camera 3-D coordinate system is coincident with the world coordinate system, and both cameras locate at the same altitude. There is an angle θ between the two camera optical axes. Then the depth can be simplified as:

$$Z^R \approx \frac{-fl(x^L \sin\theta + f\cos\theta)}{f\cos\theta(x^R - x^L) + f^2 \sin\theta} \tag{6.18}$$

$$\approx \frac{-fl\cos\theta}{u^S \cos\theta + f\sin\theta} \tag{6.19}$$

where u^S is the horizontal displacement between left and right images. Please consult to [71] for detail.

6.3 Segmentation

In the previous section, we have discussed the stereo image camera setting, and the depth calculation from the optical flow u^S in the space domain. In this section, we will use them to segment objects and code the video sequence.

In segmentation-based coding, the image is first segmented into a set of homogeneous objects. Then their contours are coded and transmitted to the receiver. Finally, the prediction error is coded and transmitted. Some of existing techniques use morphological technique [66][22][18]. We propose a different method below.

6.3.1 Optical Flow Calculation

There are many optical flow determination techniques [10][28][34][36][62]. In order to obtain an accurate motion field, the discussed correlation-feedback with Kalman filter technique is used. Four image frames, i.e. two frames from the right sequence and two frames from the left sequence are used to calculate the six parameters: $u^L, v^L, u^R, v^R, u^S, v^S$. u^L is the horizontal component of the pixel velocity associated with the left sequence, and v^L is the vertical component. u^R and v^R are the counterparts for the right sequence. u^S is the horizontal component of the pixel velocity associated with the spatial image sequence (here formed by the left and right images), and v^S is the vertical component. The initial optical flow field generally are obtained with some fast algorithms (e.g. Horn's gradient-based algorithm in our experiment). The initial matrices, Φ, H and P_0^+ in Table 2.1 are set to identical matrices. And the choice of the parameter c depends on the accuracy and the speed of convergency desired.

6.3.2 Three-Dimensional Coordinate Calculation

For every pixel, its 3-D coordinates, X, Y, Z, can be calculated with Equations 6.12 $\sim$ 6.17. In this algorithm, only the depth is needed for segmentation. The depth could be calculated with Equation 6.19.

6.3.3 Three-Criterion Segmentation

The points belonging to the same object must have some common properties, such as depth, connectivity and velocity. For instances, they are probably in the same plane, $aX + bY + cZ = m$, or on the same surface, say, $aX^2 + bY^2 + cZ^2 = d$. After the motion vector field and the 3-D coordinates (X, Y, Z) are obtained, the image will be segmented with the following three criteria.

1. **3-D Coordinates**

   The image is segmented first according to an object's 3-D information. We define function $F(X, Y, Z)$ such that a point (X_k, Y_k, Z_k) in 3-D space may belong to the object Γ, if

   $$| F(X_k, Y_k, Z_k) | < \delta_\Gamma \tag{6.20}$$

   where δ_Γ is a threshold. The function F can be linear or non-linear. For the simplest case, only the depth is used as,

   $$F(X, Y, Z) = Z - Z_0 \tag{6.21}$$

   where Z_0 is the object depth. All points with depth between $Z_0 - \delta_\Gamma$ and $Z_0 + \delta_\Gamma$ may belong to the object Γ.

2. **Connectivity**

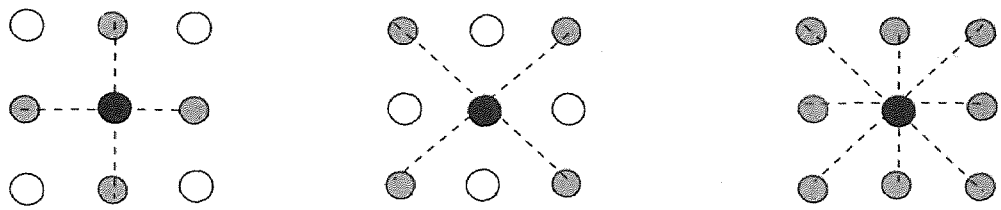
   A pixel, p, at position (x,y) has four neighbors, their coordinates are given by

   $$(x+1, y), (x-1, y), (x, y+1), (x, y-1).$$

   denoted by $N_4(p)$. In Figure 6.2 (a), the black pixel is p, and the 4 shaded pixels are its 4-neighbors. Four diagonal neighbors of p have coordinates:

   $$(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1),$$

   and are denoted by $N_D(p)$, shown in Figure 6.2 (b). The four diagonal neighbors together with the 4-neighbors are called the 8-neighbors of p, denoted by $N_8(p)$, which is shown in Figure 6.2 (c).



(a) 4-neighbors (b) 4-diagonal neighbors (c) 8-neighbors

Figure 6.2 The 4-neighbors and 8-neighbors

We have three kinds of connectivities:

- 4-connectivity: Two pixels p and q with values from a set of V are 4-connected if q is in the set $N_4(p)$.
- 8-connectivity: Two pixels p and q with values from a set of V are 8-connected if q is in the set $N_8(p)$.
- m-connectivity: Two pixels p and q with values from a set of V are m-connected if (i) q is in $N_4(p)$ or
  (ii) q is in $N_D(p)$ and the set $N_4(p) \bigcap N_4(q)$ is empty.

The 8- and m- neighbors of a pixel are shown in Figure 6.3. The shaded dots have the values from the set V. If a point under consideration connects to one of the points of the objects, this point may be considered to belong to the object.

3. **Motion Velocity**

The motion velocity is the third criterion. If a point satisfies the first two criteria, but has significantly different motion vector from that of the other points which belong to the object Γ, this point is not considered on the object, since we assumes an object has a unique motion vector. This criterion can be implemented as

$$W(X, Y, Z) < \epsilon_\Gamma \tag{6.22}$$

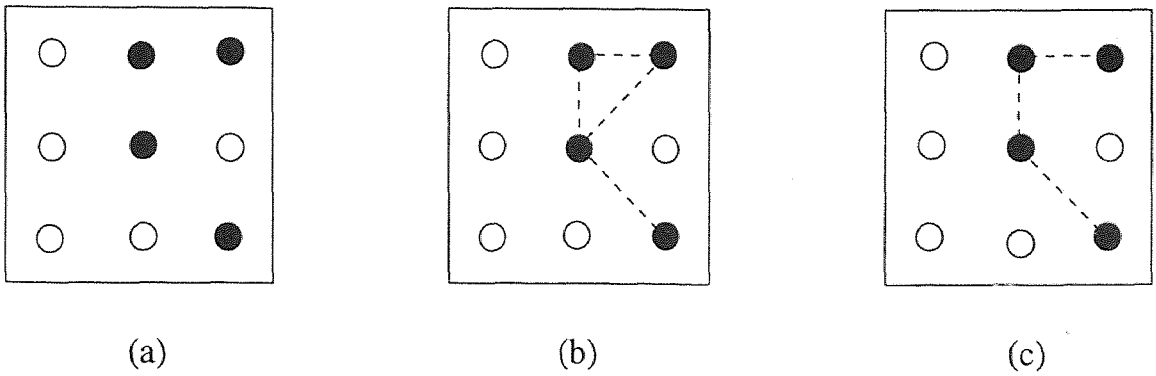


Figure 6.3 The neighbors, (a) arrangement of pixels, (b) 8-connectivity, (c) m-connectivity

where W is the motion function and ϵ_Γ is a threshold. If 3-D motion is considered, then W has two components, rotation ω and translation T.

$$W(X_k, Y_k, Z_k) = (\lambda \parallel T_k - T_0 \parallel^2 + (1-\lambda) \parallel \omega_k - \omega \parallel^2)^{\frac{1}{2}}, \tag{6.23}$$

where λ is a parameter. If only 2-D motion field is considered, it can be simplified as

$$W(x, y) < \epsilon_\Gamma, \tag{6.24}$$

where W is a 2-D motion vector in image plane.

Any point which satisfies the above three criteria belongs to the object Γ. There may exist very small segments after the segmentation. Those small objects need to be eliminated or merged to its nearest large object.

6.4 Motion Vector Calculation

After segmentation, the vector associated with an object is calculated as follows,

$$W_\Gamma = \sum_{(i,j)\in\Gamma} w(i,j) W_{(i,j)} \tag{6.25}$$

where $W_{(i,j)}$ is the motion vector of the pixel at position (i,j), and w is a weight function, it could be an average for simplicity. For rigid motion, a motion vector is good enough to describe the object motion. For non-rigid motion, if the adjacent

frames are taken in a very short time interval, then the motion can be considered approximately as rigid motion. This assumption is reasonable if the deformation of the object shape is not very fast. Because an object only has a motion vector, the number of motion vectors equals to the number of segmented objects. The motion information is thus drastically decreased, and there is no blocking artifacts resulted from the block matching technique.

6.5 Contour Coding

A major problem in object oriented video coding is the efficient encoding of object boundaries. There are two common approaches to encode the segmentation information. One approach is based on a spline approximation of the boundary, and the other is based on chain codes. The original boundary represented with pixel accuracy can be losslessly encoded by an 8-connect chain code. Using 4-connect chain code can encode with fewer bits.

Let $B = b_0, b_1, ..., b_{N_B-1}$ denote the connected boundary which is an ordered set, where b_j is the j-th point of B and N_B is the total number of points in B. Let $P = p_0, p_1, ..., p_{N_p-1}$ denote the contour used to approximate B which is an ordered set, where p_k is the k-th point of P and N_p is the total number of points in P. The approximation used in our proposed algorithm is described shortly in this subsection.

The bits to encode the entire contour is

$$R(p_0, p_1, ..., p_{N_{p-1}}) = \sum_{k=0}^{N_p-1} r(p_{k-1}, p_k) \tag{6.26}$$

where R is the total bits, and r is the bits used to code two consecutive points.

This encoding scheme which is used in the proposed algorithm is a combination of an 8-connect or 4-connect chain code, and a run-length encoding scheme. There are two passes used in this coding scheme. The first pass records the direction values while the second pass records the run-length of a direction.

The algorithm is,

1. Choose an object.

2. Choose an arbitrary contour point on the object.

3. Find the next contour point and compute its direction value.

4. If the direction value is equal to the previous one, then the run-length increased by 1. Otherwise, a new direction value is added to the first pass. Update the contour point.

5. Go to step 3 until no more contour point can be found on the object. Then leave this object.

6. Go to step 1 until all objects are coded.

7. The two passes are entropy coded.

The direction values are given in Figure 6.4. Figure 6.4(a) has 4 directions, which will be coded as binary sequence (00), (01), (11), and (10). The binary codes for 8 directions are shown in Figure 6.4(b).

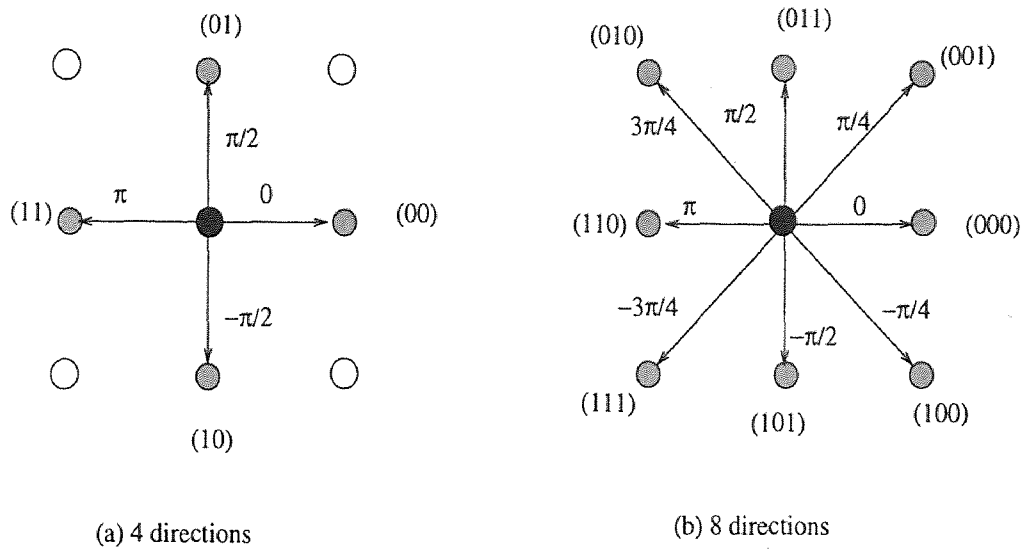
After the objects are segmented at the encoder, this contour information will be transmitted to the decoder, and the decoder uses it to reconstruct the image. Because this information is preserved as long as the object appears in the scene, and it does not need to be resent.

6.6 Predictive Error Coding

The transmission of the predictive error information is still necessary because

1. The segmentation is not perfect.

2. Some covered scene exposes due to disocclusion take place.



(a) 4 directions

(b) 8 directions

Figure 6.4 The contour directions

3. There is error between the estimated motion and the true motion.

It is noted that the disocculusion may affect the bit rate severely. These areas can be predicted from the previous frame. The predictive error is divided into 8×8 blocks, and DCT is applied block by block.

6.7 Experiments

In order to evaluate the new algorithm, two video sequences, a simulation sequence with three moving objects and a real image sequence with two moving boxes and a fixed box, are used in this experiments.

6.7.1 A Simulation Sequence

The camera setting for the simulation sequence is in Figure D.1. $\theta = 2.5^o$. The box_1 and box_3 have sinusoidal textures on their surfaces, and box_2 has uniform surfaces. The first four frames of the sequence are shown in Figure D.2. The small box with sinusoidal (box_1) moves with 2 pixels/frame horizontally and 2 pixels/frame vertically. The largest box (box_3) moves with -1 pixel/frame horizontally and 1 pixel/frame vertically. The other box (box_2) moves with -1 pixel/frame horizontally

and -2 pixels/frame vertically. The minus sign means the opposite direction. White Gaussian noises are added to the background. The optical flow field is obtained with the correlation-feedback with Kalman filter algorithm, and the depth obtained with Equation 6.19. The objects are segmented according to the three criteria. Only depth is used in the first criterion, Z_0 is chosen to be 1050, 980, 940 for different objects, and δ_Γ=20. 8-connectivity is tested in the second criterion, and $\epsilon_\Gamma = 20\%$ in criterion 3. The contours of the three objects are coded with 4-directions and the prediction error is DCT coded. The bit rate and the PSNR of the reconstructed frames are shown in Figure D.3. It achieves PSNR 38.46dB at bit rate about 0.09 bpp. The contours are coded at 0.04 bpp for the first frame, and the motion vectors are coded at 0.008 bpp.

6.7.2 A Real Image Sequence

The camera setting for this real sequence is in Figure D.4. $\theta = 2.5^o$. The box_2 and box_3 move together, and box_1 is fixed. The first frame of the sequence is shown in Figure D.5. The optical flow field is obtained with the correlation-feedback with Kalman filter algorithm, too. Only depth is used in segmentation. Z_0 is chosen to be 1080, and 875; δ_Γ=40. 8 connectivity is tested in second criterion too, and $\epsilon_\Gamma = 20\%$ in criterion 3.

The predictive error and the PSNR of the reconstructed frames are shown in Figure D.6. At bit rate 0.12 bpp, it achieves PSNR 37.536 dB. The results for different quality of the reconstructed image is given in Table 6.1,

6.8 Conclusion

A new video coding algorithm which is based on segmentation is given in this chapter. Three criteria are defined for segmentations and a chain code combined with run length is implemented to code the shapes of the segmented objects. The advantages

Table 6.1 The results of the real sequence

PSNR	27.2	31.4	36.5	37.5	39.0
bpp	0.004	0.011	0.08	0.12	0.34

of this algorithm are that only a few bits are required to code the motion vectors, and it can handle the occlusions since the depth information is used so that very low bit rate is obtained with good quality of reconstructed video frames. On the other hand, it can easily reconstruct the right image sequence from the left sequence with the disparity vector fields, which can be calculated from the given stereo video sequence, or vice versa. This algorithm works well for regular shape objects and rigid motions.

CHAPTER 7

SUMMARY

This chapter contains a summary of our major research contributions, a review of some unsolved problems and a discussion of some possible directions for future research.

7.1 Major Contributions

It is known that in intraframe coding spatial redundancy is reduced. The I-pictures, defined in MPEG, are coded independently of each other. Therefore, every I-picture can be accessed randomly. Any frame in a sequence consisting of the I-pictures exclusively, such as coded with MJPEG, can be displayed easily and with less buffer memory than that coded with the interframe coding. However, it needs high bit-rate. The motion compensated video coding, used in interframe coding, reduces temporal redundancy drastically. Compared with MJPEG, it increases compression ratio by a factor of more than ten at the same quality of reconstructed frames. The motion compensation is widely used in the video coding standards. At high bit-rate, motion compensation with block matching techniques works very well for video coding. But, at very low bit-rate, because very limited bits are used to code prediction error, annoying blocking artifacts become more significant than that at high bit-rate. From our experiments, the bits used to code motion vectors is comparable to the bits used to code prediction error in the case of very low bit-rate video coding. More specifically, at high bit-rate, the bits to code motion vectors are only less than 5%, while at very low bit-rate, it will need 30% to 50%. Those limited bits for coding prediction error are not sufficiently enough to eliminate blocking artifacts which arises from block matching model, and many efforts have been made to reduce artifacts. To eliminate blocking artifacts, one of the potential approaches is to use dense motion

fields. The dense motion field is pixel-based rather than block-based, so that it can eliminate blocking artifacts. To transmit dense motion vectors needs, however, more bits than to transmit block-based motion vectors. This difficulty has made optical flow field little used for motion compensation in video coding after some trials in the early 80s. In this dissertation research, it is found that dense motion vectors can be efficiently compressed with DCT techniques. For example, points in a human face are controlled by skin and muscles, and a point motion is highly correlated with its neighboring points. Further theoretical analyses show that dense motion fields can be modeled by the first order auto-regressive model with correlation coefficient 0.8. Although the bits used to code dense motion fields may be a few more than that required by the block matching technique, the prediction error is much less due to the usage of dense motion field. To reduce the computational burden of pixel-based motion field determining, a pre-processing is applied to detect motion areas. Only motion vectors in those areas that involve obvious motion will be calculated. The pre-processing reduces computation to one third in general. These analyses result in an efficient coding algorithm for very low bit-rate video coding which uses DCT coded dense motion field as motion compensation. Compared with H.263, the new algorithm eliminates annoying blocking artifacts, hence making reconstructed video frames much more visually pleasant while maintaining almost the same bit-rate. However, for very complicated video sequence, its performance will decrease because the computation in optical flow determination and the amount of prediction error will increase.

The region-based DWT algorithm also uses DCT coded dense motion field. It decomposes the picture into three regions, significant, less significant, and non-significant regions. The motivation of doing that is to assign bits to regions according to their content significance. For example, when people observe a head-and-shoulder type sequence, their attention, in general, is focused on the speaker's face. Assigning

more bits to the facial portion will improve the quality of reconstructed frame subjectively. It achieves similar performance to that by the previous algorithm. However, it has more flexibility in terms of adaptively assigning bits to regions according to their content significance and thus improve the quality of significant regions. Furthermore, the performance of this algorithm can be potentially enhanced once the DWT technique is improved. Similarly to the previous algorithm, its performance will decrease when complexity of video sequences increases.

After studying the model-based video coding techniques, we find several problems preventing this powerfully potential technique from being practically used for very low bit-rate video coding. The first problem involves identification and segmentation. In order to obtain high quality of reconstructed frames, some 3-D wireframe model algorithms and AU methods require identification and segmentation of some sensitive portions, such as lips and eyes. Consequently, the computational complexity increases drastically. The second problem is model fitting. To fit a generic model to a special human face is not straightforward. It is either non-automated or not fitting well. Not only the shape of face need to be fitted but also eyes and mouth need to be fitted. The third problem is motion estimation. For the 3-D wireframe model, the 3-D motion vectors need to be calculated with relatively high accuracy. If the motion vectors are not accurate enough, it will result in high prediction error, and it will need many bits to code prediction error, as a result, the compression ratio will decrease.

After realizing these problems, we proposed an algorithm which is based on dense motion field and thresholding techniques. This algorithm is much simpler than the model-based techniques, and it outperforms some typical model-based techniques in terms of the bit-rate and the quality of reconstructed frames.

Stereo sequence coding is a research topic on which relatively less effort has been made. From stereo sequence, we can obtain the 3-D information which can be

used for object segmentation. In this dissertation, the proposed algorithm for stereo video sequence coding, which utilizes three criteria to segment objects, can achieve very high compression ratio for regular shape of objects. A combination of chain codes and run-length coding is used to code the contour of the segmented objects. This algorithm is good for sequences with rigid motion and regular shapes of objects. With many non-regular shape objects, its performance will deteriorate.

In a summary, four efficient video coding techniques at very low bit-rate are proposed in this dissertation. These four algorithms are all based on dense motion fields used for motion compensation. Besides, following an optical flow determining technique devised by Pan et al [71][62], some simulation and real image sequence experiments have been carried out to evaluate the correlation-feedback with a Kalman filter [62]. It is concluded that the correlation-feedback technique is robust against noise. The Kalman filter does improve the motion field accuracy near moving boundaries.

7.2 Major Unsolved Issues

Our efforts in this dissertation are aimed at finding out some very efficient techniques to reduce or eliminate the blocking artifacts. The computational efficiency is another important issue. However, the exact number of operations in these proposed algorithms has not been quantitatively analyzed, this is because the operations in computation is quite complicated.

The segmentation-based stereo video coding algorithm works very well for relatively simple video sequences. For highly complicated sequences, it needs to be tested and further improved.

7.3 Directions for Further Research

Firstly, how to overcome the problems associated with the model-based video coding technique, which are mentioned in the first section of this chapter. These problems need to be solved before the model-based technique can be practically used for video coding. Secondly, how to reduce or eliminate the blocking artifacts at very low bit-rate is still a research topic. Thirdly, the most existing dense motion field determining techniques were not proposed specifically for video coding. Therefore, these techniques should be modified before they are used for video coding. New techniques need to be devised to match this application. Finally, for some video coding schemes, such as content-based, region-based, and object-based, in order to compress video sequence more efficiently, prior knowledge of the image content is very helpful and necessary. Hence how to extract the content information from the first frames is a very important research topic.

It is expected that the future research for video coding will be in both very low bit-rate and high bit-rate. For very low bit-rate video coding some very efficient coding schemes need to be developed, and for high bit-rate video coding some problems in applications need to be solved.

APPENDIX A

EXPERIMENTAL FIGURES IN CHAPTER 2

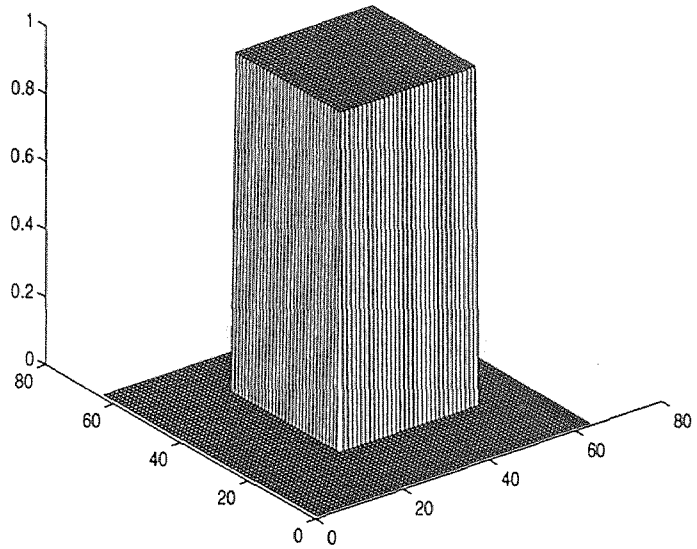


Figure A.1 True optical flow

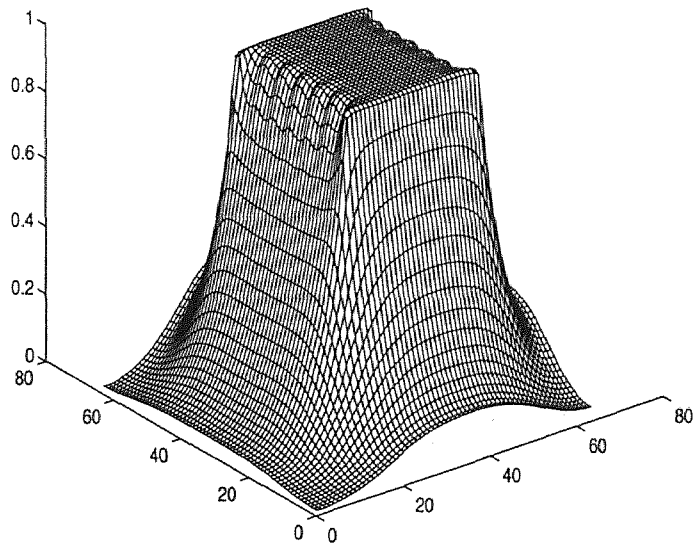


Figure A.2 Optical flow obtained with gradient-based

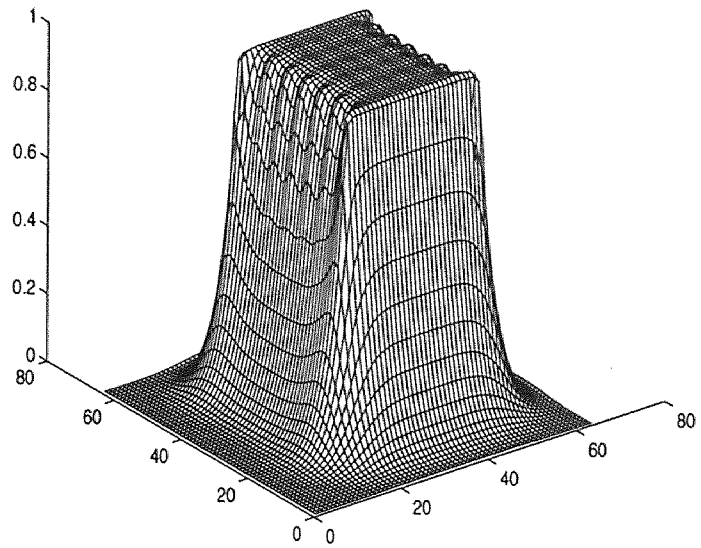


Figure A.3 Optical flow obtained by gradient-based with Kalman filter

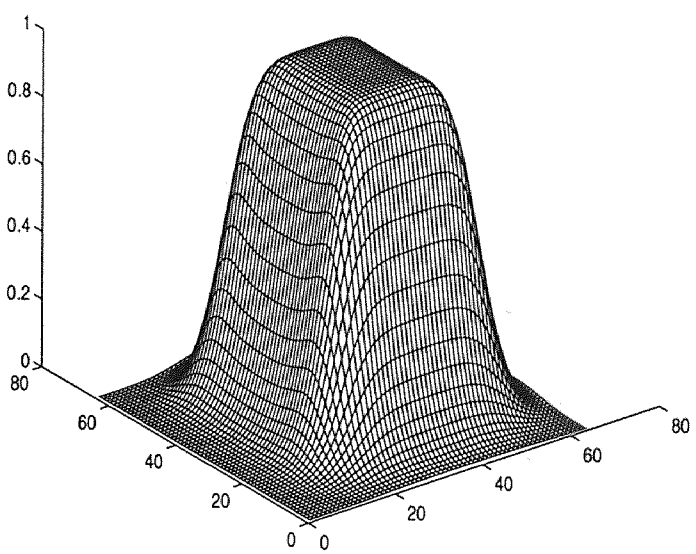


Figure A.4 Optical flow obtained with correlation feedback algorithm

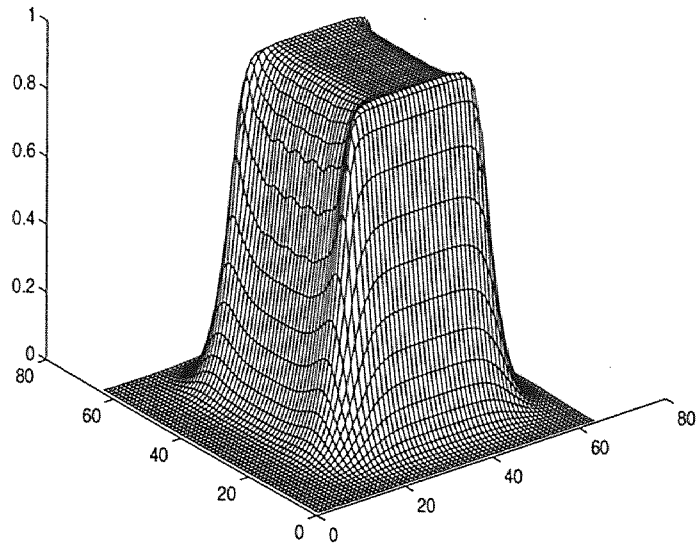


Figure A.5 Optical flow obtained by correlation feedback with Kalman filter

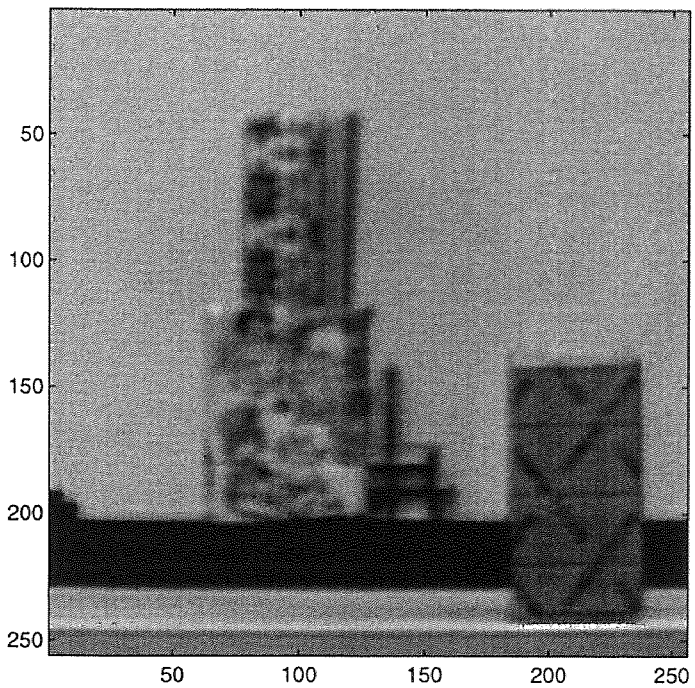


Figure A.6 Real image: three boxes

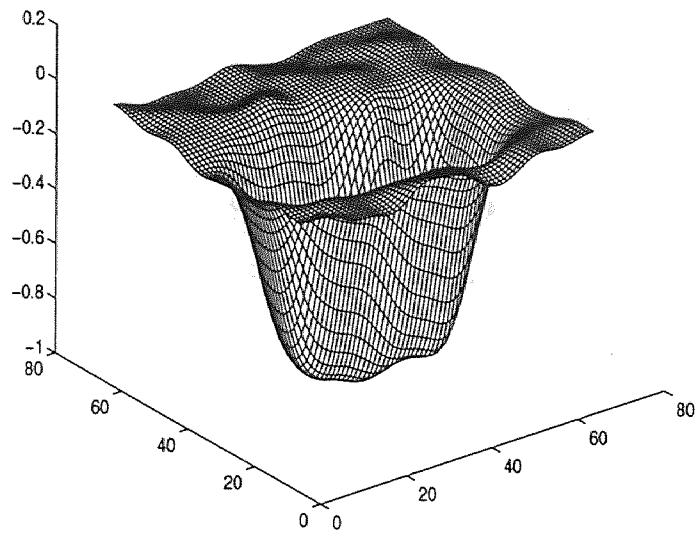


Figure A.7 U^L with correlation-feedback

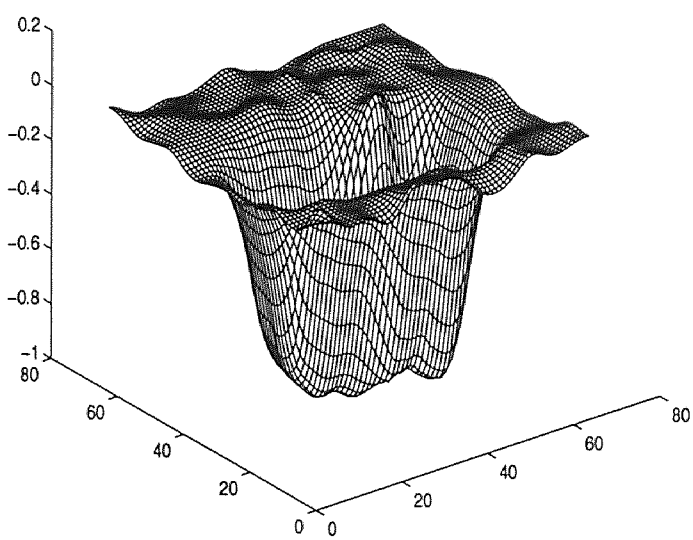


Figure A.8 U^L with correlation-feedback and Kalman filter

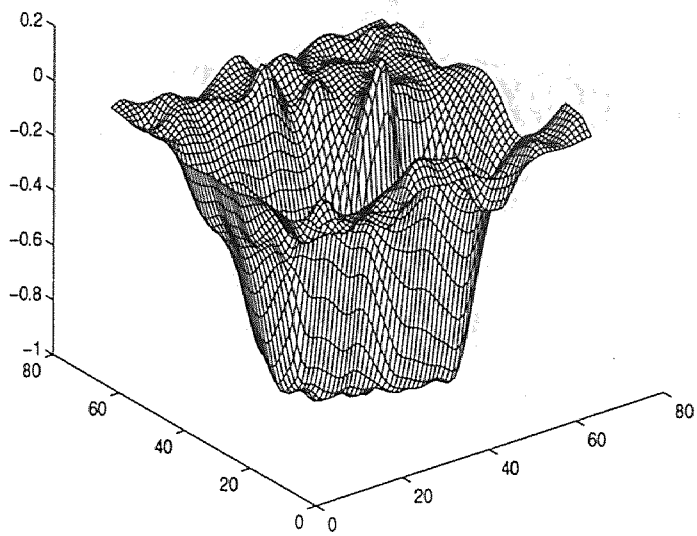


Figure A.9 U^L with Horn-Schunck algorithm

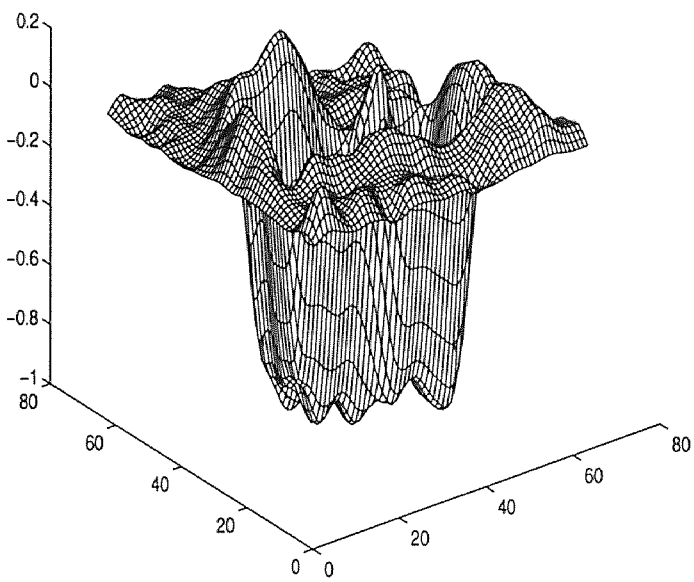


Figure A.10 U^L with Horn-Schunck and Kalman Filter

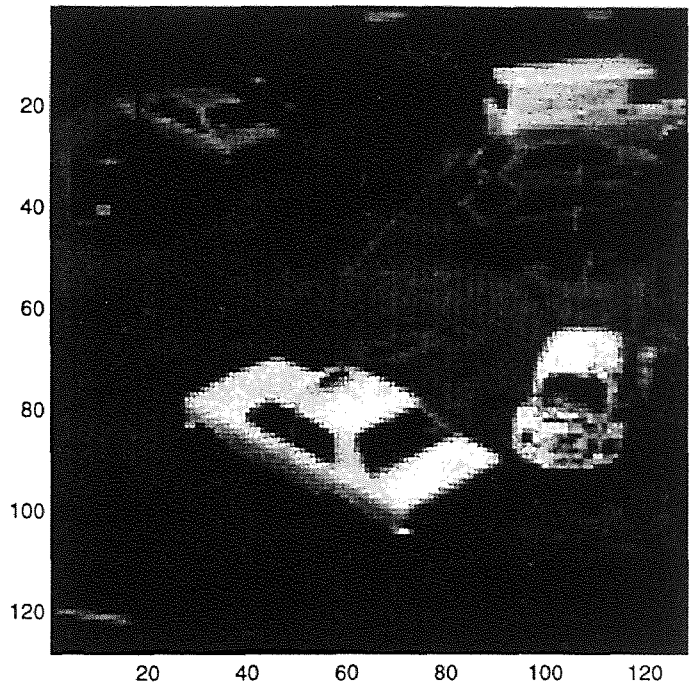


Figure A.11 The image of Hamburg Taxi

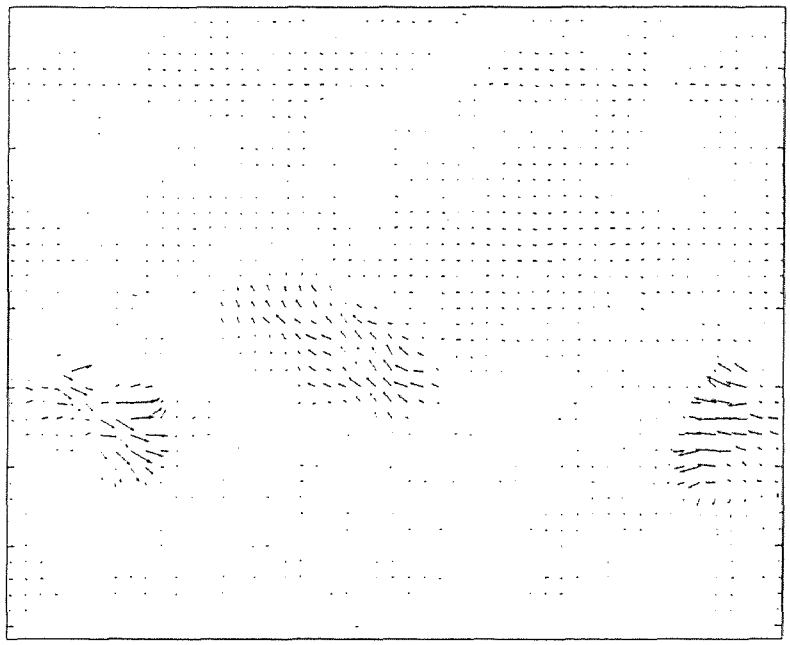


Figure A.12 Needle diagram of optical flow

Figure A.13 Needle diagram of optical flow by Horn and Schunck's algorithm

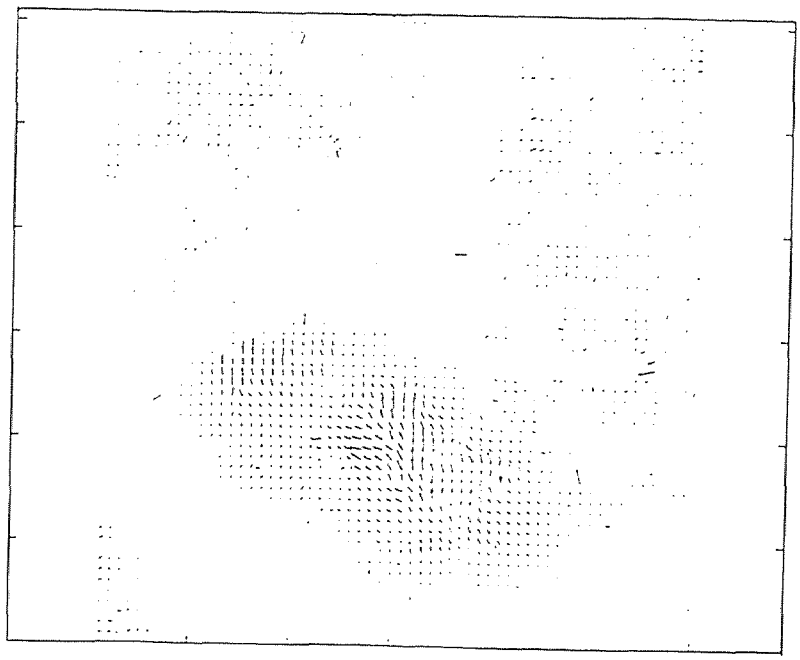


Figure A.14 Needle diagram of optical flow by Singh's algorithm

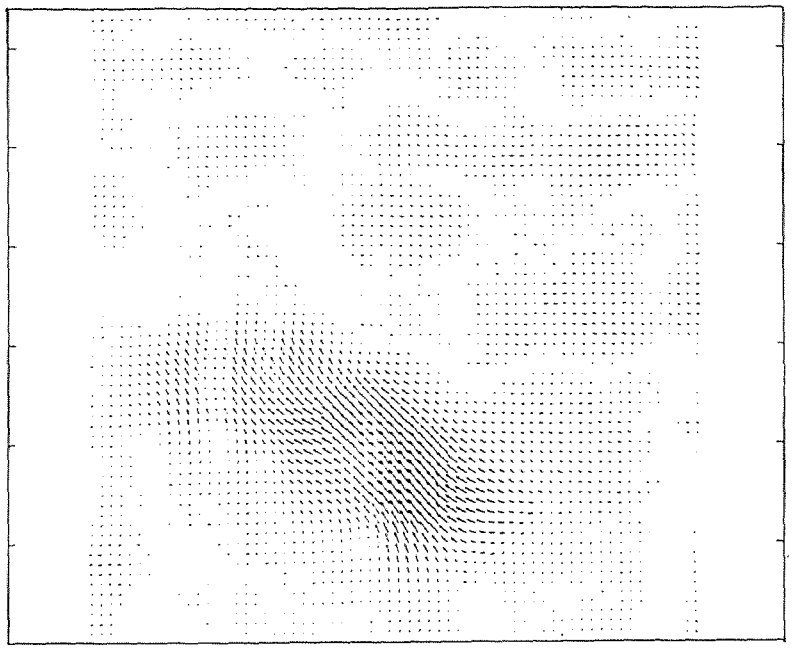


Figure A.15 Needle diagram of optical flow by correlation-feedback

APPENDIX B

EXPERIMENTAL FIGURES IN CHAPTER 4

(a) The 21st original frame of Miss America sequence

(b) The reconstructed 21st frame with proposed algorithm

(c) The reconstructed 21st frame with H.263/N

Figure B.1 Miss America

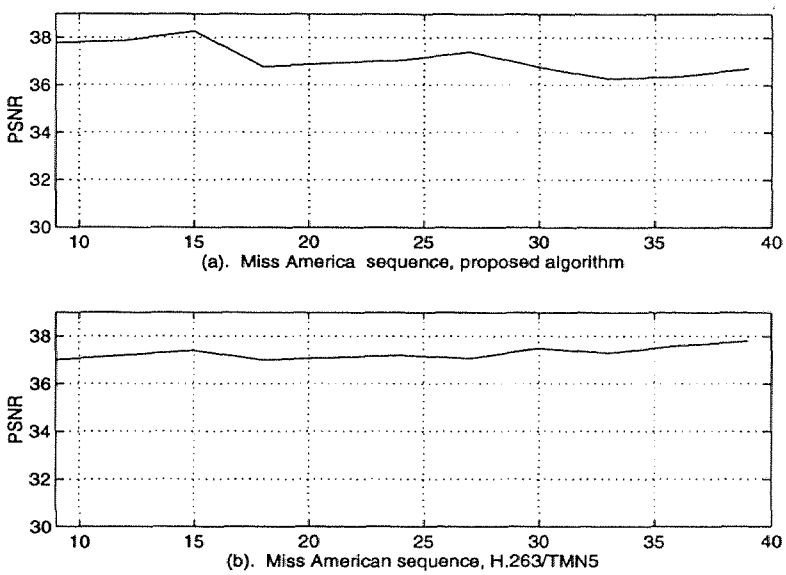


Figure B.2 PSNR of Miss America sequence

(a) The 39th original frame of Claire sequence

(b) The 39th reconstructed frame with proposed algorithm

(c) The 39th reconstructed frame with H.263

Figure B.3 Claire

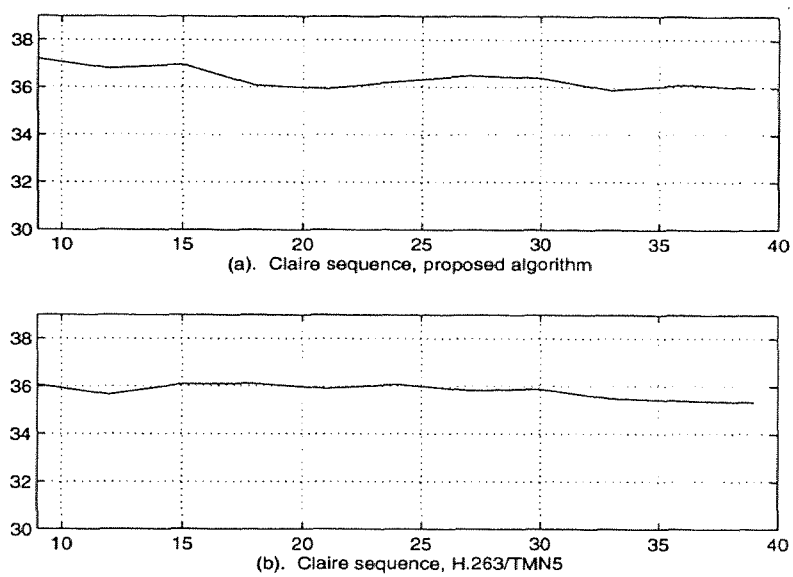


Figure B.4 PSNR of Claire sequence

APPENDIX C

THE MULTIRESOLUTION OF DWT

Figure C.1 The original image

Figure C.2 The level 1 subimages

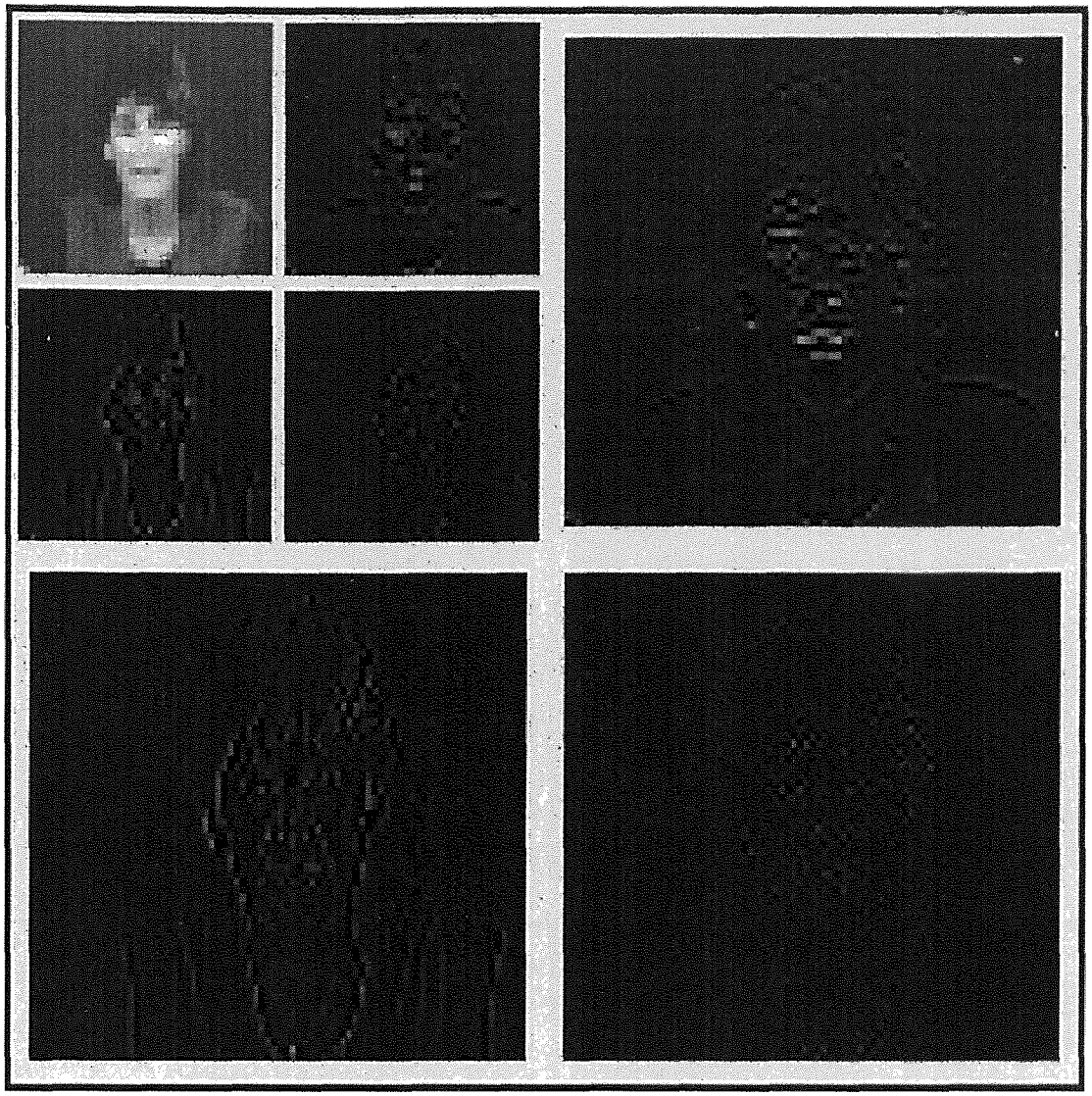


Figure C.3 The level 2 subimages

APPENDIX D

EXPERIMENTAL FIGURES IN CHAPTER 6

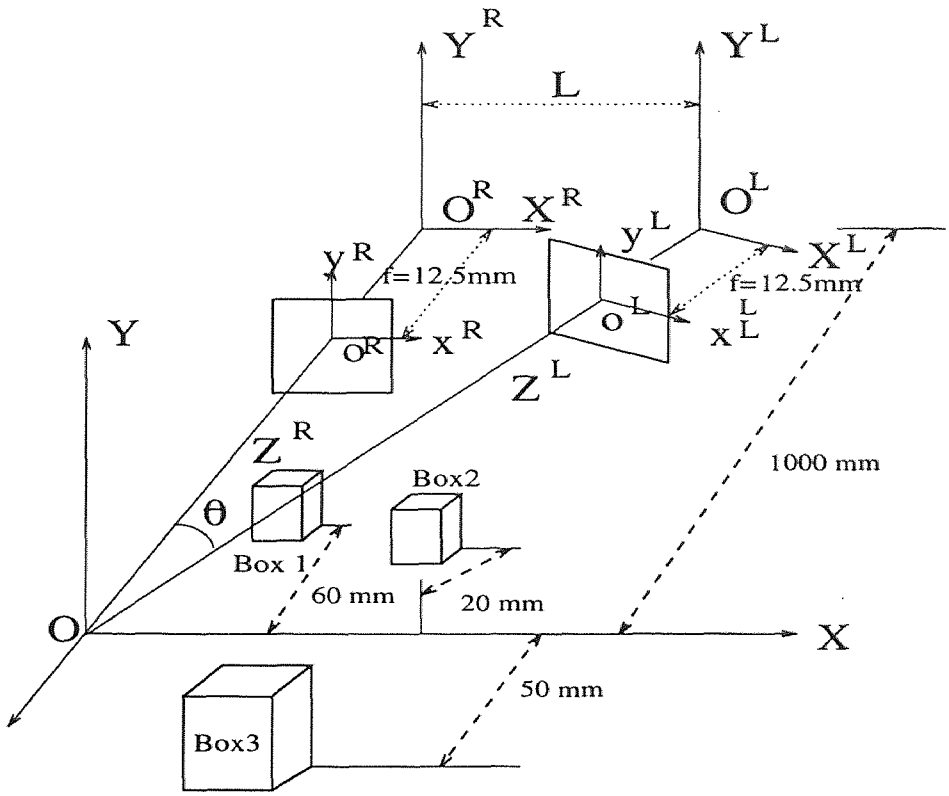


Figure D.1 The camera setting for simulation sequence

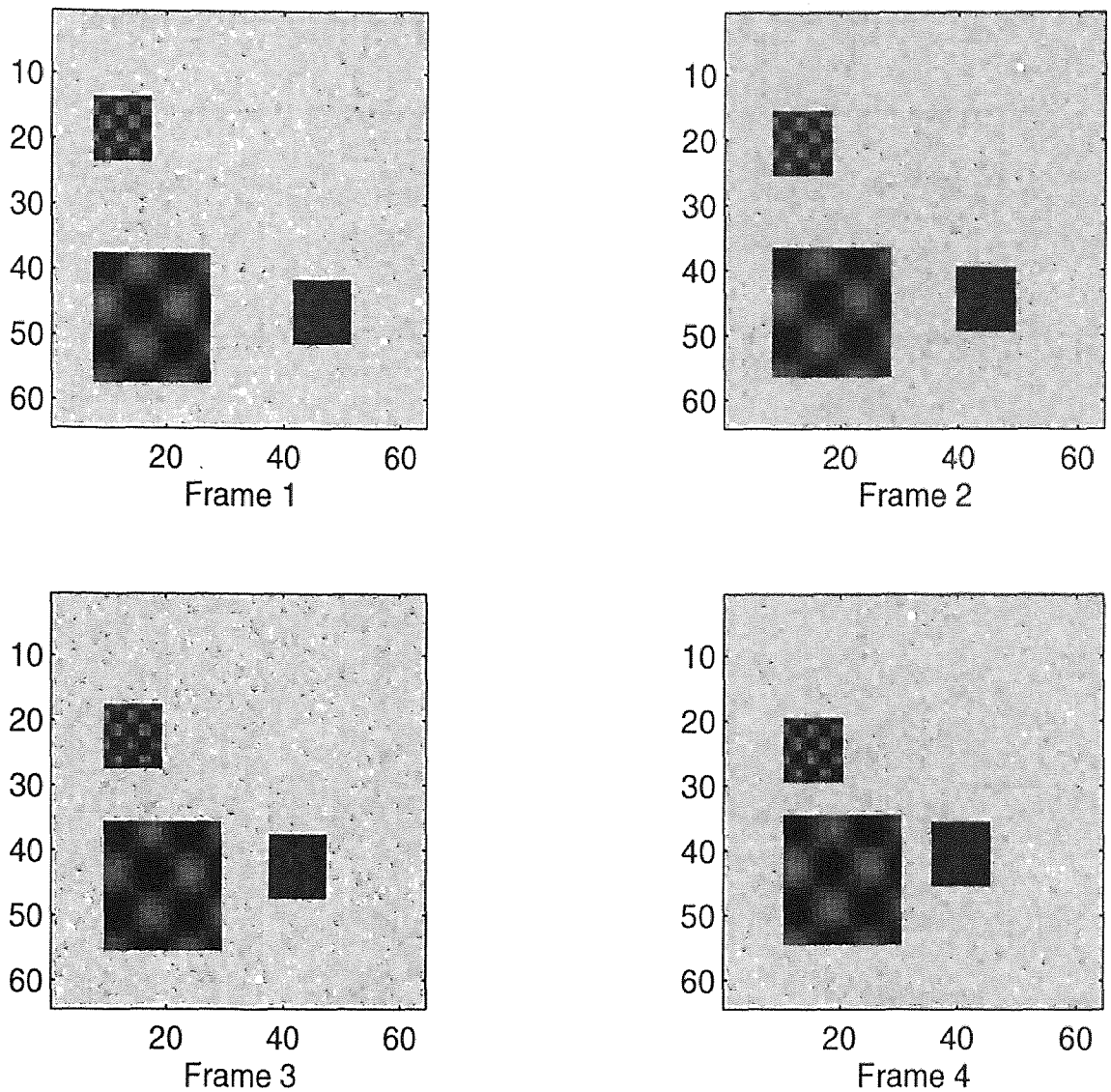


Figure D.2 The first four frames of the simulation sequence

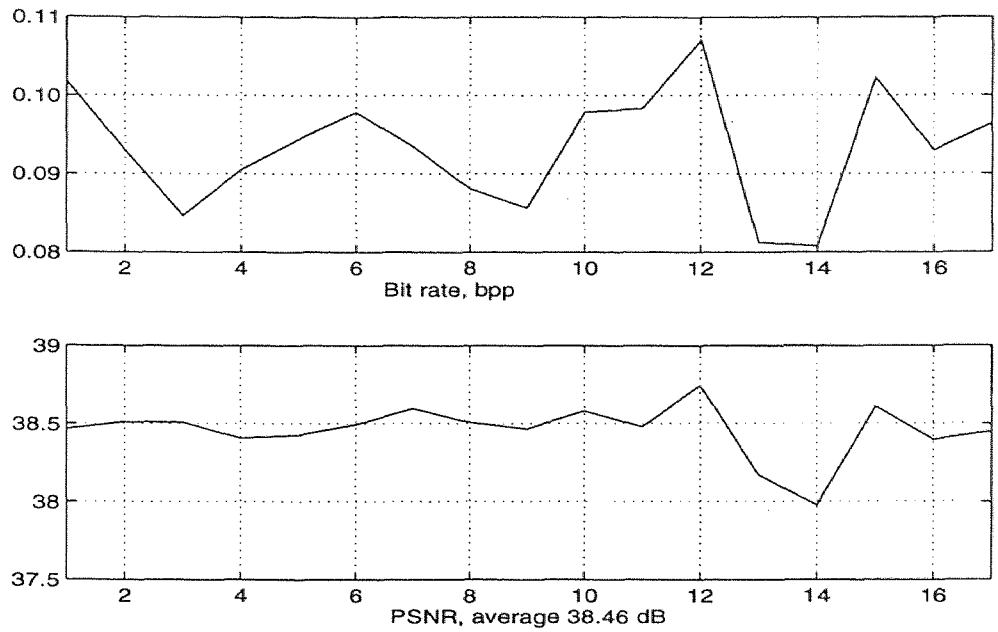


Figure D.3 The result of the simulation sequence

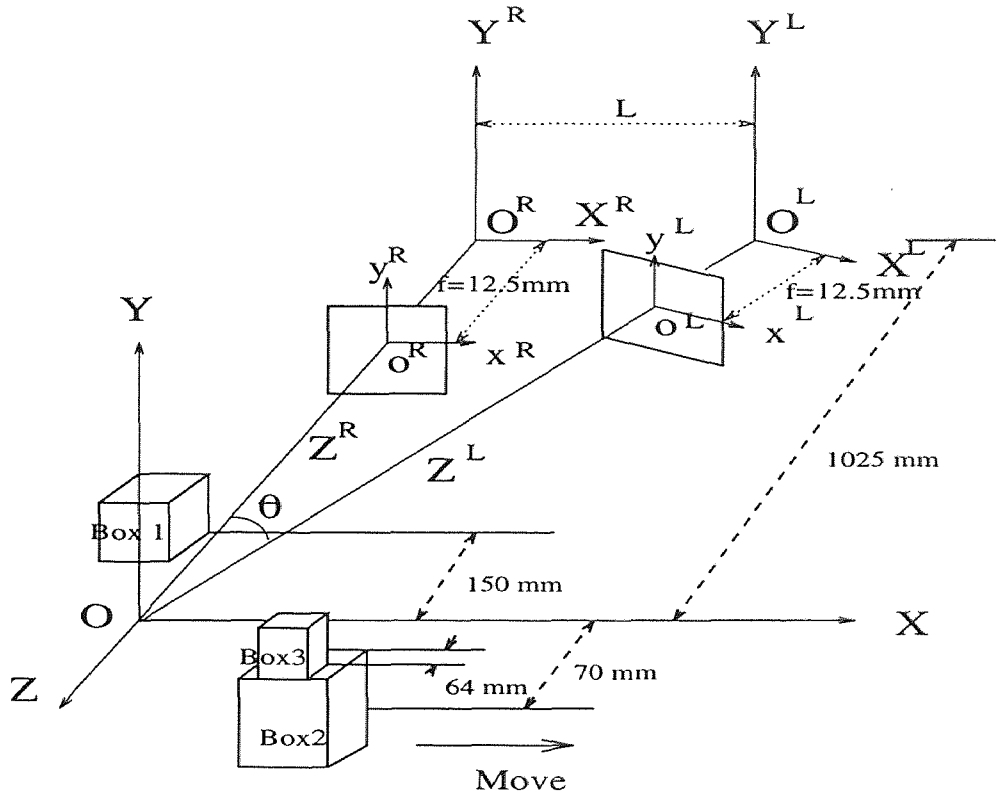


Figure D.4 The camera setting for the real image sequence

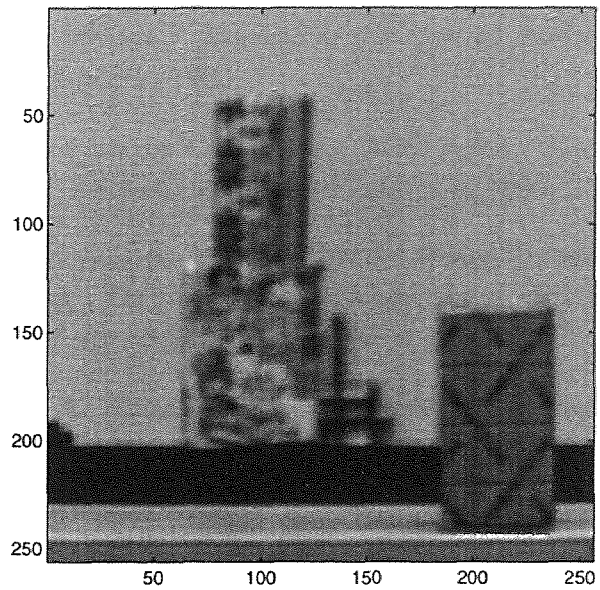


Figure D.5 The first frame of the real image sequence

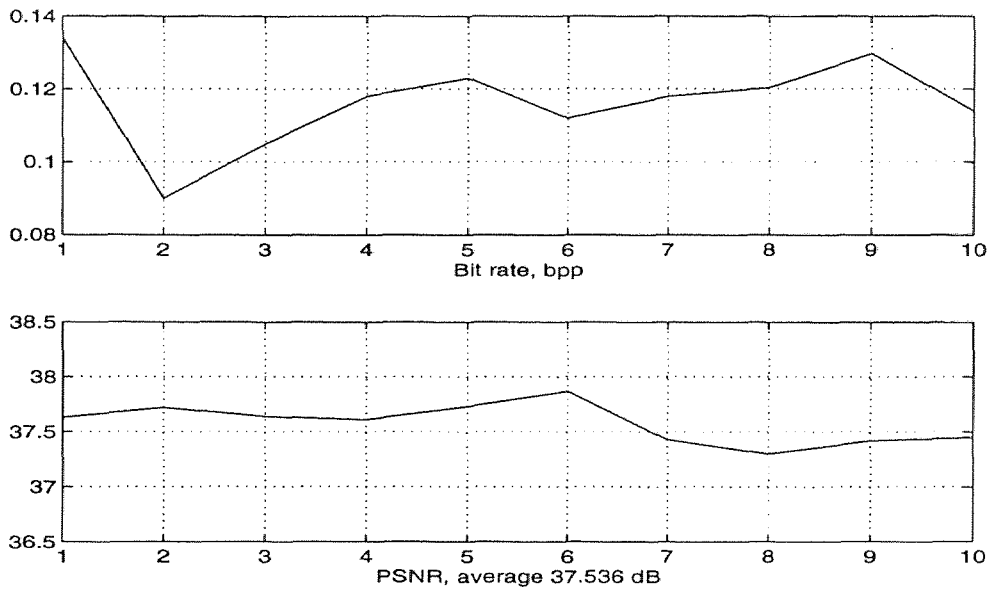


Figure D.6 The results of the real image sequence

REFERENCES

1. Adelson, E. H., and J. R. Bergen, "Spatiotemporal Energy Models for the Perception of Motion," *Journal of the Optical Society of America A*, vol. 2, no. 2, pp. 284-299, 1985.

2. Advanced television systems Committee, DOC A/54, "Guide to the Use of the ATSC Digital Television Standard", Oct. 4, 1995.

3. Aizawa, K., and Thomas S. Huang, "Model-Based Image Coding: Advanced Video Coding Techniques for Very Low Bit-Rate Applications", *Proceedings of the IEEE*, vol. 83, No. 2, pp. 259-271, Feb. 1995.

4. Altunbasak, Yucel, A. Murat Tekalp, and Gozde Bozdagi, "Two-Dimensional Object-Based Coding Using a Content-Based Mesh and Affine Motion Parameterization", *Proceedings of 1995 IEEE International Conference on Image Processing*, Hyatt Regency Crystal City, Washington, D.C., vol. 2, pp. 394-397. Oct. 23-26.

5. Anandan, P., "A Computational Framework and an Algorithm for the Measurement of Visual Motion," *International Journal of Computer Vision*, vol. 2, pp. 283-310, 1989.

6. Anastassiou, D., "Digital Television,", *Proceedings of IEEE*, Vol. 82, No. 4, pp. 510-519, April 1994.

7. Apostolopoulos, John G., and Jae S. Lim, "Representing Arbitrarily-Shaped Regions: A Case Study of Overcomplete Representations", *Proceedings of 1995 IEEE International Conference on Image Processing*, Hyatt Regency Crystal City, Washington, D.C., vol. 2, pp. 390-393. Oct. 23-26.

8. Aravind, R., G. L. Cash, D. C. Duttweller, H.-M. Hang, B. G. Haskel, and A. Puri, " Image and Video Coding Standards", *AT&T Tech. J.*, vol. 72, January/February 1993, pp. 67-88.

9. Auyeung, C., J. Kosmach, M. Orchard and T. Kalafatis, "Overlapped Block Motion Compensation," *SPIE Proceedings of Visual Communications and Image Processing '92*, vol. 1818, pp. 561-571, Boston, MA, Nov. 1992.

10. Barron, J.L., D.J. Fleet and S.S. Beauchemin, "Systems and Experiment Performance of Optical Flow Techniques", *International Journal of Computer Vision*, vol. 12, No. 1, pp. 43-77, 1994.

11. Bigun, J., G. Granlund, and J. Wiklund, "Multidimensional Orientation Estimation with Applications to Texture Analysis and Optical Flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 775-790, 1991.

12. CCITT recommendation T.81, "Information Technology-Digital Compression and Coding of Continuous-tone Still Images Requirements and Guidelines", Sep. 1992.

13. C-Cube Microsystems, Inc., "The JPEG File Interchange Format", and available in *ftp://ftp.uu.net/graphics/jpeg/jfif.ps.gz.*

14. Chiariglione, L., " The Development of an Integrated Audiovisual Coding Standard: MPEG", *Proceedings of the IEEE*, vol.83, No. 2, pp. 151-157, Feb. 1995.

15. Choi, C. S., T. Takebe, " Analysis and Synthesis of Facial Expressions in Knowledge-Based Coding System for Facial Image Sequence", *ICASSP 91*, Toronto, 1991.

16. Daubechies, I., "Orthonormal Bases of Compactly Supported Wavelets," *Comm. Pure Applied Mathematics*, vol. 41, pp. 909-996, 1988.

17. Dufaux, F., and F. Mescheni, "Motion Estimation Techniques for Digital TV: A Review and a New Contribution," *Proceedings of the IEEE*, vol. 83, no. 6, pp. 858-876, 1995.

18. Ebrahimi, Touradj, Emmanuel Resusens and Wei Li, " New Trends in Very Low Bitrate Video Coding", *Proceedings of the IEEE*, vol. 83, No. 6, pp. 877-891, June 1995.

19. Fleet, D. J., and A. D. Jepson, "Computation of Component Image Velocity from Local Phase Information," *International Journal of Computer Vision*, vol. 5, pp. 77-104, 1990.

20. Fuldseth, Arild, and Tor A. Ramstad, "A New Error Criterion for Block Based Motion Estimation", *Proceedings of 1995 IEEE International Conference on Image Processing*, Hyatt Regency Crystal City, Washington, D.C., vol. 3, pp. 188-191. Oct. 23-26.

21. Ghanbari, M., " An Adapted H.261 Two-Layer Video Codec for ATM Networks", *IEEE Trans. Communications*, vol. 40, No. 9, pp. 1481-1490, September 1992.

22. Gonzalez, Rafael C., and Richard E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, New York, 1992.

23. Guillemain, Philippe, and Richard Kronland-Martinet, " Characterization of Acoustic Signals Through Continuous Linear Time-Frequency Representations", *Proceedings of the IEEE*, vol. 84, No. 4, pp. 561-585, April 1996.

24. Hoogendorn, A., "Digital Compact Cassette," *Proceedings of IEEE*, vol. 82, pp. 1479-1489, Oct. 1994.

25. Hopkins, R., "Choosing an American Digital HDTV Terrestrial Broadcasting System," *Proceedings of IEEE*, vol. 82, pp. 554-563, April 1994.

26. Huang, H.-C., and J.-L. Wu, " Real-Time Software-Based Video Coder for Multimedia Communication Systems," *J. Multimedia Systems*, vol.1, pp.110-119, 1993.

27. Hung, Yan, Xinhua Zhuang, "Motion-Partitioned Adaptive Block Matching for Video Compression". *Proceedings of 1995 IEEE International Conference on Image Processing*, Hyatt Regency Crystal City, Washington, D.C., vol. 1, pp. 554-557. Oct. 23-26.

28. Horn, Berthold K. P., and Brian G. Schunck, "Determining Optical Flow", *Artificial Intelligence*, No. 17, pp. 185-203, 1981.

29. Illgner, Klaus, and Frank Muller, "Hierarchical Coding of Motion Vector Fields", *Proceedings of 1995 IEEE International Conference on Image Processing*, Hyatt Regency Crystal City, Washington, D.C., vol. 1, pp. 566-569. Oct. 23-26.

30. ISO DIS 10918-1, "Digital Compression and Continuous-tone Still Image (JPEG)", CCITT Recommendation T.81.

31. ITU-T Recommendation H.261, " Video Codec for Audiovisual Services at px64 kbits", March 1993.

32. ITU-T Recommendation H.263, "Video Coding for Narrow Telecommunication Channels at Less Than 64kbits/s", (Draft) April 1995.

33. ITU-T Recommendation H.324 (draft), "Terminal for Low Bitrate Multimedia Communications", Nov. 22, 1995.

34. Jain, J.R., and A.K. Jain, "Displacement Measurement and Its Application in Interframe Image Coding", *IEEE Trans. Commun.*, vol. 29, pp. 1799-1806, Dec. 1981.

35. Jozawa, Hirohisa, "Segment-Based Video Coding Using an Affine Motion Model", *Proceedings SPIE, Visual Communications and Image Processing '94*, Chicago, Illinois, pp. 1605-1614, September 25-29.

36. Kearney, Joseph K., B. William Thompson, Daniel L. Boley, " Optical Flow Estimation: Error Analysis of Gradient-Based Methods with Local Optimization", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 9, No. 2, March 1987.

37. Koga, T., K. Linuma, A. Hirano, Y. Lijima, and T. Ishiguro, "Motion-Compensated Interframe Coding for Conferencing", in *NTC 81, Proceeding*, pp. G 5.3.1 - G 5.3.5, New Orleans, LA, Dec. 1981.

38. Krishnamurthy, Ravi, Pierre Moulin and John Woods, "Optical Flow Techniques Applied to Video Coding", *Proceedings of 1995 IEEE International Conference on Image Processing*, Hyatt Regency Crystal City, Washington, D.C., vol. 1, pp. 570-573. Oct. 23-26.

39. Kunt, M., A. Ikonomopoulos, and M. Kocher, "Second Generation Image Coding Techniques", *Proceedings of the IEEE*, vol. 73, pp. 549-575, April 1985.

40. Kunt, M., M. Benard, R. Leonardi, " Recent Results in High Compression Image Coding", *IEEE Trans. on Circuits and Systems*, vol. 34, pp. 1306-1336, November 1987.

41. Lane, Tom et. al., "The Independent JPEG Group software JPEG codec". Source code available in *ftp://ftp.uu.net/graphics/jpeg/jpegsrc.v5.tar.gz.*

42. Laplante, Phillip A., and Alexander D. Stoyenko, " Real-Time Imaging", *IEEE Press*, The Institute of Electrical and Electronics Engineers, INC., New York. pp.125- 142, 1992.

43. LeGall, D., "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM*, Vol. 34, pp. 46-63, April 1991.

44. Li, Haibo, Pertti Roivainen and Robert Forchheimer, *Report 1991- 10-30*, Electrical Engineering Department, Linkoping University, Sweden.

45. Li, Haibo, Pertti Roivainen and Robert Forchheimer, "3-D Motion Estimation in Model-Based Facial Image Coding", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol 15, No. 6, pp. 545-554, June 1993.

46. Lin, Shu, and Y. Q. Shi, " A new Approach to Video Teleconferencing Coding - Another Look at Wireframe Model in Facial Coding", *Proceedings of the Ninth Workshop on Image and Multidimensional Signal Processing*, Belize City, Belize, IMDSP Belize, pp. 172-173, March 3-6, 1996.

47. Lin, Shu, Yun Q. Shi, and Ya-Qin Zhang, "An Optical Flow Based Motion Compensation Algorithm for Very Low Bit-Rate Video Coding," *IEEE ICASSP97*, Groebenzell, Germany, 1997. (Accepted)

48. Lin, Shu, Yun Q. Shi, "Region-Based Adaptive DWT Video Coding Using Dense Motion Field," *The First IEEE Signal Processing Society Workshop on Multimedia Signal Processing*, Princeton, New Jersey, June 23-25, 1997 (Submitted).

49. Liou, M., " Overview of the $p \times 64$ kbit/s Video Coding Standard", *Communications of the ACM*, vol. 34, No. 4, pp. 59-63, April 1994.

50. Mallat, S., "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Transactions on Patter Analysis and Machine Intelligence*, vol.11, pp.674-693, Nov. 1989.

51. Mallat, Stephane, " Wavelets for a Vision", *Proceedings of the IEEE*, vol. 84, No. 4, pp. 604-614, April 1996.

52. Matthews, Kristine E., and Nader M. Namazi, "Simultaneous Motion Parameter Estimation and Image Segmentation Using the EM Algorithm", *Proceedings of 1995 IEEE International Conference on Image Processing*, Hyatt Regency Crystal City, Washington, D.C., vol. 1, pp. 542-545. Oct. 23-26.

53. Monaco, J. W., and M. J. T. Smith, "Video Coding Using Image Warping Within Variable Size Blocks," *Proceeding of IEEE International Symposium on Circuits and Systems*, vol. 2 pp. 794-797, Atlanta, GA, May 1996.

54. Muraayama, J., T. Miyauchi, N. Shirota, " Image Sequence Coding Using a Contour-Based Method", *Proceedings of 1995 IEEE International Conference on Image Processing*, Hyatt Regency Crystal City, Washington, D.C, vol. 1, pp. 546-549, Oct. 23-26.

55. Musmann, H.G., M. Hotler and J. Ostermann, "Object-Oriented Analysis-Synthesis Coding of Moving Images", *Signal Processing: Image Communication*, vol. 1, pp. 117-132, 1989.

56. Nagel, H. H., "On a Constraint Equation for the Estimation of Displacement Rates in Image Sequence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 13-30, 1989.

57. Nogahi, S., and M. Ohta, "An Overlapped Block Motion Compensation for High Quality Motion Picture Coding," *Proceedings of IEEE International Symposium on Circuit and Systems*, pp. 184-187, San Diego, CA, May 1992.

58. Okubo, S., " Reference Model Methodology - A Tool for the Collaborative Creation of Video Coding Standards", *Proceedings of the IEEE*, vol.83, No. 2, pp. 139-150, February 1995.

59. Orchard, M.T., "Prediction Motion Field Segmentation for Image Sequence Coding", *IEEE Trans. Circuits and Systems for Video Technology*, vol. 3, No. 1, pp. 54-70, Feb.1993.

60. Ostermann, Jorn, "Differences Between an Object-Based Analysis-Synthesis Coder and Block-Based Hybrid Coder", *Proceedings of 1995 IEEE International Conference on Image Processing*, Hyatt Regency Crystal City, Washington, D.C., vol. 2, pp. 398-401. Oct. 23-26.

61. Pan, Jingning, Yun Q. Shi, and Shu Lin, " A Kalman Filter for Improving Optical Flow Accuracy on Moving Boundaries", (submitted to IEEE trans. on Image Processing).

62. Pan, Jingning, "Motion Estimation Using Optical Flow Field", *Ph.D. Dissertation*, Electrical and Computer Engineering Department, NJIT, Newark, May 1994.

63. Pennenbaker, W. B., and J. L. Mitchell, *JPEG Still Image Data Compression Standard*, New York, Van Nostrand Reinhold, 1993.

64. Ramchandran, Kannan, Martin Vetterli, and Cormac Herley, " Wavelets, Subband Coding, and Best Bases", *Proceedings of the IEEE*, vol. 84, No. 4, pp. 541-560, April 1996,

65. Ran, Xiaonong, and Chang Y.Choo, "Syntax-Based Arithmetic Video Coding for Very Low Bitrate Visual Telephony", *Proceedings of 1995 IEEE International Conference on Image Processing*, Hyatt Regency Crystal City, Washington, D.C., vol. 2, pp. 410-413. Oct. 23-26.

66. Salembier, P., " Morphological Multiscale Segmentation for Image Coding", *IEEE Signal Processing*, vol. 38, No. 3, pp. 359-384, 1994.

67. Schroder, Karsten, and Roland Mech, "Combined Description of Shape and Motion in an Object Based Coding Scheme Using Curved Triangles", *Proceedings of 1995 IEEE International Conference on Image Processing*, Hyatt Regency Crystal City, Washington, D.C., vol. 2, pp. 390-393. Oct. 23-26.

68. Schroder, Peter, " Wavelets in Computer Graphics", *Proceedings of the IEEE*, vol. 84, No. 4, pp. 615-625, April 1996.

69. Shapiro, Jerome M., "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", *IEEE Transactions on Signal Processing*, vol 41, No.12, Dec. 1993.

70. Singh, A., "An Estimation Theoretic Framework for Image-Flow Computation", *Proceedings of the 3rd Conference on Computer Vision*, Osaka, Japan, Dec. 4-6, 1990.

71. Shi, Y. Q., C. Q. Shu, and J. N. Pan, " Unified Optical Flow Field Approach to Motion Analysis From s a Sequence of Stereo Images", *Patter Recognition*, vol. 27, No. 12, pp. 1577-1590, 1994.

72. Soundararajan, Aravind, "Digital Image and Video Standards", *Communications of the ACM*, vol. 34, No. 4, pp. 47-58, April 1991.

73. Unser, Michael, and Akram Aldroubi, " A Review of Wavelets in Biomedical Applications", *Proceedings of the IEEE*, vol. 84, No. 4, pp. 626-638, April 1996.

74. Wallace, Gregory K., "The JPEG Still Picture Compression Standard," *Communications of the ACM*, Vol.34, No.1, pp.31-44, April 1991.

75. Wang, J.Y.A., and E.H. Adelson, "Spation-Temporal Segmentation of Video Data", *SPIE Image & Video Processing II*, vol. 2182, (San Jose, USA), Feb.1994.

76. Wang, Xiangwen, and Defu Cai, "Fast Automatic Face Feature Points Extraction for Model-Based Image Coding", *Proceedings of 1994 SPIE's Visual Communications and Image Processing*, Chicago, Illinois. pp. 545-553, Sep. 25-29.

77. Watanabe, H., and S. Singhal, "Windowed Motion Compensation," *SPIE Proceedings of Visual Communications and Image Processing '91*, vol. 1065, pp. 582-589. Boston, MA, Nov. 1991.

78. Wornell, Gregory W., " Emerging Applications of Multirate Signal Processing and Wavelets in Digital Communications", *Proceedings of the IEEE*, vol. 84, No. 4, pp. 586-603, April 1996.

79. Yeh, Joseph, Martin Vetterli, and Masoud Khansari, "Motion Compensation of Motion Vectors" *Proceedings of 1995 IEEE International Conference on Image Processing*, Hyatt Regency Crystal City, Washington, D.C., vol. 1, pp. 574-577. Oct. 23-26.

80. http://www.mot.com/MIMS/ISG/Papers/v34tutorial/v34_tutorial.html.