國立交通大學

資訊工程系

碩士論文

以特徵為基礎的視訊編碼位元配置結構

A Feature-based Bit Allocation Scheme in Video Coding

研究生：柯鑑洲

指導教授：李素瑛　教授

中華民國九十四年六月

以特徵為基礎的視訊編碼位元配置結構

A Feature-based Bit Allocation Scheme in Video Coding

研 究 生：柯鑑洲　　Student：Chien-Chou Ko

指導教授：李素瑛 教授　　Advisor：Prof. Suh-Yin Lee

國 立 交 通 大 學

資 訊 工 程 系

碩 士 論 文

A Thesis

Submitted to Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science and Information Engineering

June 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年六月

以特徵為基礎的視訊編碼位元配置結構

研究生：柯鑑洲　　指導教授：李素瑛

國立交通大學資訊工程學系

摘要

近年來，在數位視訊通訊的服務，如視訊會議或串流等，已經廣泛地被使用。而由於在一些網路環境下，如 ISDN、PSTN 及網際網路等，受到傳輸頻寬的限制，因此如何在被限制的網路頻寬下提供使用者在觀看影片時能得到更好的視覺效果是一個重要的研究課題。在很多的研究中發現，人類對於畫面中移動的物體是比較敏感的，因此有不少研究便針對這些人眼比較容易注意的部份，在固定的目標傳輸位元率下降低背景的品質來增加前景移動物體的品質，使觀看者在這些吸引人注意的部份得到更好的視覺品質。

在這篇論文中，我們提出了一個以特徵為基礎的位元配置結構。這個結構利用不同區域的大小及動態資訊在固定的位元率下分配位元給這些區塊。另外我們也提出一個簡單而快速的物件切割演算法，利用動態資訊分解出在影片中人眼會被吸引的區域。我們在 H.264 的參考軟體 JM9.5 中實作這個架構並測試了一些影片。實驗結果證明這個方法可以增進影片中移動物體的視覺品質。

A Feature-Based Bit Allocation Scheme in Video Coding

Student: Chien-Chou Ko Advisor: Prof. Suh-Yin Lee

Institute of Computer Science and Information Engineering

National Chiao Tung University

Abstract

In recent years, applications of the digital video communications, such as video conferencing or streaming service are widely used. Since the channel rates of some networks, such as ISDN, PSTN and the Internet are limited, there is an active research topic on how to provide the user a better visual quality under a limited transmission rate.. In many studies, it has shown that human eye is more sensitive to the moving region inside a video scene. Therefore, some researches have concentrated on the human visual system and try to achieve an improved quality of the moving foreground quality as compared to the background to give the viewers a better visual quality on the these attracting regions.

In this thesis, we propose a feature-based bit allocation scheme to distribute bits to different regions in a video scene in a constrained bit-rate by utilizing the feature of size ant motion of different regions. We also propose a simple and fast object segmentation method to extract the interesting regions by using motion information. We implement the scheme into the H.264 reference software JM 9.5 and test with some video sequences. Experimental results prove that we can improve the visual quality of the moving region.

Acknowledgement

I sincerely appreciate the guidance and the encouragement of my advisor, Prof. Suh-Yin Lee. Without her graceful encourage, I would not complete this thesis.

Besides, I would like to extend my thanks to the lab mages in the Information System Laboratory, especially Mr. Ming-Ho Hsiao and Mr. Yi-When Chen.

Finally, I want to express my appreciation to my friends for their consideration and my parents for their supports. The thesis is dedicated to them.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

In recent years, the demand for the applications of digital video communications, such as videoconferencing and streaming service, has increased considerably. The channels, such as ISDN, PSTN and the Internet, provide the ability to transmit at rates sufficient for video transmission, at prices low enough to be within the reach of many consumers. However, the transmission rates over network are limited. In most video sequence, each frame has some few regions, which should be perceptually more pleasing than the rest. Then, improvements in these regions can provide more acceptable results. Therefore, achieving an improved foreground quality as compared to the background within the target bit rate, or channel rate, has been an active research topic. Previous work on region of interest (ROI) processing, which addressed selective facial feature enhancements on low bit rate video sequences, follows two approaches. One is based on a preferential quantization of the foreground regions in the video sequences and the other is based on an enhancement layer encoding to the interesting regions.

In this thesis, we propose a feature-based bit allocation scheme. And for the purpose to distribute bits to different region, we also propose a motion-based object segmentation algorithm. First in the object segmentation algorithm, we utilize the motion information, which had been generated in the motion estimation function of video encoder, to segment the foreground moving objects. Then, in the feature-based bit allocation scheme, we will adopt the characteristics of these segmented objects as

perceptual tuning factors to distribute different amount of the bits among different regions.

1.2 Organization of Thesis

The rest of this thesis is organized as follows. Chapter 2 will introduce the background and the related work of the video object segmentation, rate control and bit allocation algorithm in the H.264 standard. In Chapter 3, we will present the details of our proposed algorithm for object segmentation and strategy for rate control and bit allocation. Chapter 4 will show the experimental results and we will make a conclusion in Chapter 5.

Chapter 2

Background

The purpose of coding region of interest is trying to search regions which human will focus on in the video sequences and improve the visual quality of these areas while sacrificing the quality of background. Thus, video object segmentation is first required to extract the moving objects that might be interesting to the viewers. After the segmentation process, bit allocation scheme will distribute different bits to object regions and background by their characteristics. Finally, in the channel with a limited bitrate, the users will get better visual quality on regions which they are interesting in.

In this chapter, we will introduce the related works of the video object segmentation and rate control/bit allocation. The details of the related works of video object segmentation will be introduced in Section 2.1. Since we are developing the rate control/bit allocation strategy with H.264, we will review the standard of H.264/MPEG-4 Part 10 and its rate control strategy in section 2.2 and 2.3. Then, related works of bit allocation algorithm will be introduced in section 2.4.

2.1 Video Object Segmentation

There are many researches in the literature of object segmentation. Generally, segmentation algorithm can be classified into two categories, change detection based methods and the homogeneity based methods.

The change detection based algorithms [4]-[6] segment objects by taking difference between current frame and previous frame, and then a binary mask indicating the shape and position of the moving objects has been decided with a chosen threshold. Since these methods are suitable for video surveillance or

monitoring by combining background registration [7][8], but they are not suitable for video sequences with camera moving, such as movie.

The other category of segmentation algorithms [9]-[12] are homogeneity based algorithms. These algorithms segment moving objects based on the homogeneity of their color, texture or motion information. Pixels with some similar features are first grouped into small regions, and these regions are then grouped into objects with some other features. However, the primary drawback of these and many other pixel based approaches to object segmentation is the amount of the required computational cost to process the video sequences.

Recently, some fast segmentation algorithms [13]-[15] have been proposed to efficiently segment objects in the video sequence without large amount of computation. These fast algorithms are based on compressed domain and utilize the feature of temporal and spatial information, such as motion vectors and DCT coefficients in MPEG bit-stream. By filtering motion vectors and DCT coefficients, these methods use a watershed algorithm to cluster favorable macroblocks that have similar features.

Since our system is based the idea of the application in real-time video streaming, we will refer the ideas of [13]-[15] and propose a simple and fast algorithm to segment objects by clustering blocks of similar motion vector with region growing algorithm.

2.2 H.264/MPEG 4 Part 10

H.264 is a new standard and promises to outperform the earlier MPEG-4 and H.263 standard, providing better compression of video images. The new standard is entitled “Advanced Video Coding (AVC)” and is published jointly as MPEG-4 Part 10 of MPEG4 and ITU-T Recommendation H.264.

2.2.1. New Features

We list some of the important terminology adopted in the H.264 standard. More features in the H.264 standard are shown in [3].

Variable block-size motion compensation with small block size: This standard support more flexibility in the selection of motion compensation block sizes and shapes than any previous standard. The block sizes may be one 16x16 macroblock partition, two 16x8 partitions, two 8x16 partitions or four 8x8 partitions. If the 8x8 mode is chosen, each of the four 8x8 sub-macroblocks within the macroblock may be split in a further 4 ways, either as one 8x8 sub-macroblock partition, two 8x4 sub-macroblock partitions, two 4x8 sub-macroblock partitions or four 4x4 sub-macroblock partitions.

Multiple reference picture motion compensation: For motion compensation purpose, the encoder can select a large number of pictures, which have been decoded, to be the reference frames.

Weighted Prediction: This feature allows the motion-compensated prediction signal to be weighted and offset by parameters specified by the encoder.

Directional spatial prediction for intra coding: A new technique of extrapolating the edges of the previous-decoded parts of the current picture is applied in regions of pictures that are coded as intra. This improves the quality of the prediction signal, and allows prediction from neighboring areas that were not coded by intra coding.

2.2.2. Introduction to H.264 Encoder

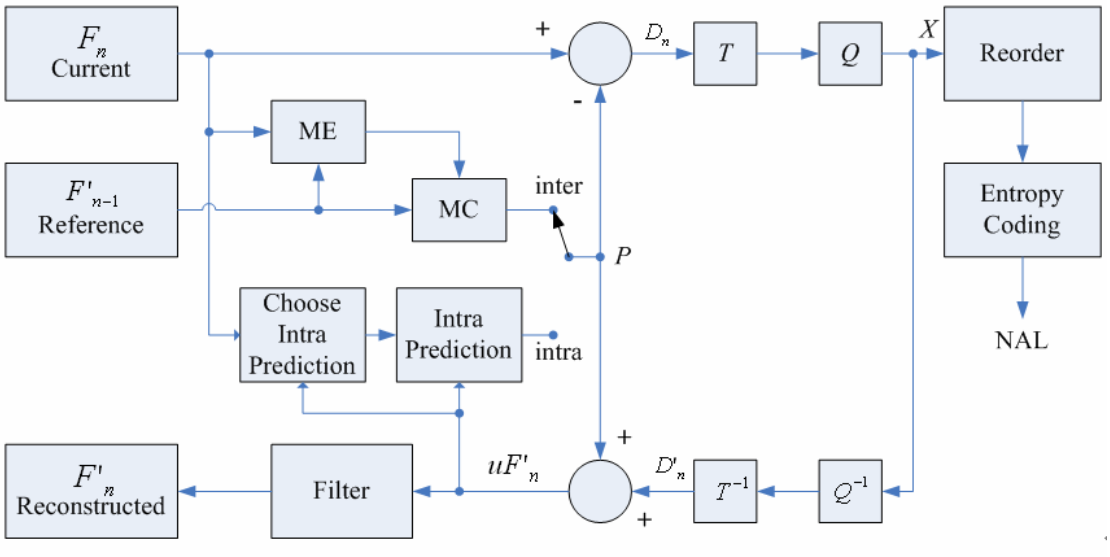


Fig. 2-1 H.264 Encoder [1]

Fig. 2-1 shows the H.264 encoder. An input frame or field F_n is processed in units of a macroblock. Each macroblock is encoded in intra or inter mode and, for each block in the macroblock, a prediction PRED (marked as ***P*** in Fig. 2-1) is formed based on reconstructed picture samples. In intra mode, PRED is formed from samples in the current slice that have previously encoded, decoded and reconstructed (uF'_n). In the inter mode, PRED is formed by motion-compensated prediction from one or two reference frames.

The prediction PRED is subtracted from current block to produce a residual block D_n that is transformed and quantized to give ***X***, a set of quantized transform coefficients which are reordered and entropy encoded.

The encoder decodes a macroblock to provide a reference for further predictions. The coefficients ***X*** are scaled (Q^{-1}) and inverse transformed (T^{-1}) to produce a difference block D'_n. The prediction block PRED is added to D'_n to create a reconstructed block uF'_n. A filter is applied to reduce the effects of blocking distortion and the reconstructed reference frame is created from a series of block F'_n.

2.3 Rate Control for H.264

An encoder employs rate control as a way to regulate varying bit rate characteristics of the coded bit-stream in order to produce high quality decoded frame at a given target bit rate. Rate control is thus a necessary part of an encoder, and has been widely studied in standards, like MPEG 2, MPEG 4, H.263, and so on [18]-[22].

Rate distortion optimization (RDO) is expecting to minimize the decoded distortion under a given rate constraint. The Lagrangian method can find the tradeoff between the rate and distortion efficiently. In H.264, the Lagrangian method is used for mode selection in motion compensation and intra prediction. In other word, it can minimize the distortion and find the optimal motion vector and coding mode of a block at a give rate constraint. However, utilizing Lagrangian method makes the rate control for JVT a more difficult task than those for other standards [23]-[25]. This is because the quantization parameters are used in both rate control algorithm and RDO, which resulted in the following chicken and egg dilemma when the rate control is studied: To perform RDO for macroblocks in the current frame, a quantization parameter (QP) should be first determined for each macroblock by using the mean absolute difference (MAD) of current frame or macroblock [18][19]. However, the MAD of current frame or macroblock is only available after the RDO.

As described above, there is a problem in the implementation of the rate control in H.264 coding. (1) The MAD is unknown before performing RDO. (2) Although we cat get MAD for each coding mode after motion compensation, the best coding mode is still unknown so that we cannot decide which MAD cab be used to estimate the QP.

The H.264 standard uses a single pass rate control algorithm to solve the problem described above. The following sections will describe the H.264 rate control scheme in detail. Fig. 2-2 shows the approach for the rate control in H.264 standard.

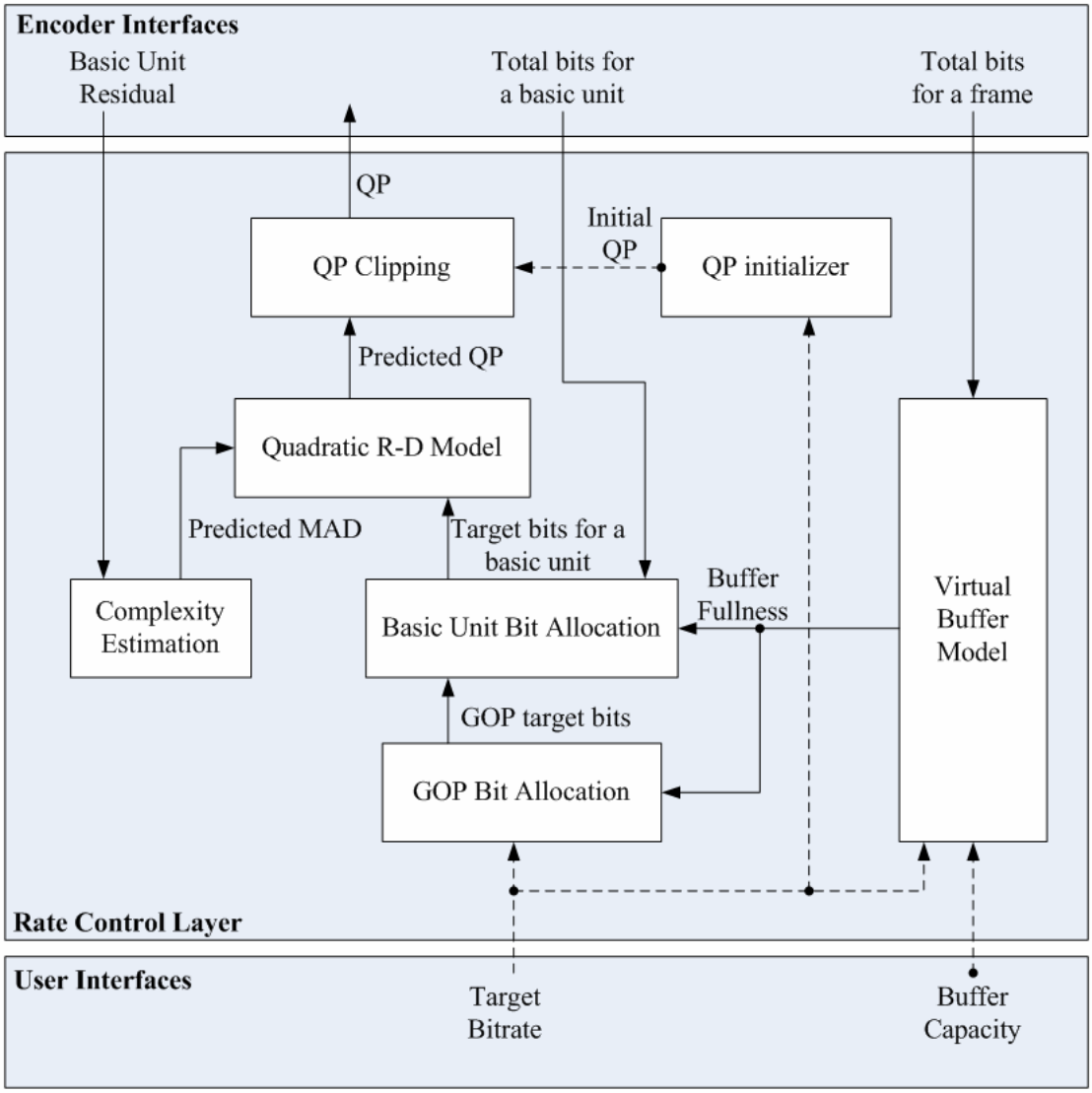


Fig. 2-2 Elements of H.264 Rate Control [25]

2.3.1 Quadratic Rate Distortion Model

Quadratic R-D model is adopted in MPEG-4 and H.264/AVC. To illustrate the rationale of quadratic R-D model, we summarize the results in [18][19].

Assuming that the statistics if input datas are Laplacian distributed:

$$P(x) = \frac{\alpha}{2} e^{-\alpha|x|}, \quad where \quad -\alpha < x < \alpha \tag{1}$$

The distortion measure is defined as $D(x, \tilde{x}) = |x - \tilde{x}|$, then there is a closed-form solution for the R-D functions as derived:

$$R(D) = \ln\left(\frac{1}{\alpha D}\right), \quad where \quad D_{\min} = 0, \quad D_{\max} = \frac{1}{\alpha}, \quad 0 < D < \frac{1}{\alpha} \tag{2}$$

The R-D function is expanded into a Taylor series:

$$\begin{aligned} R(D) &= \left(\frac{1}{\alpha D} - 1\right) - \frac{1}{2}\left(\frac{1}{\alpha D} - 1\right)^2 + R_3(D) \\ &= -\frac{3}{2} + \frac{2}{\alpha} D^{-1} - \frac{1}{2\alpha^2} D^{-2} + R_3(D) \end{aligned} \tag{3}$$

The new model is formulated in the equation as follows:

$$R_i = a_1 \times Q_i^{-1} + a_2 \times Q_i^{-2} \tag{4}$$

where

Q_i : quantization level used for the current frame *i*;

In order to consider the complexity of each frame and the overhead including video/frame syntax and motion vectors, the quadratic R-D model is modified as follows:

$$T_i = \frac{X_1 \cdot MAD_i}{Q_i} + \frac{X_2 \cdot MAD_i}{Q_i^2} \tag{5}$$

where

T_i : total number of texture bits used for encoding the current frame *i*;

MAD_i : MAD of the current frame *i*, computed using motion-compensated residual for the luminance component;

X_1, X_2 : first- and second-order coefficients.

2.3.2 Terminology

A. Definition of Basic Unit

Suppose that a frame is composed of N_{mbpic} macroblocks. A basic unit is defined to be a group of contiguous macroblocks which is composed of N_{mbunit} macroblocks where N_{mbunit} is a fraction of N_{mbpic}. Denote the total number of basic units in a frame by N_{unit}, which is computed by:

$$N_{unit} = \frac{N_{mbpic}}{N_{mbunit}} \tag{6}$$

Examples of a basic unit can be a macroblock, a slice, a field or a frame.

B. A Fluid Flow Traffic Model

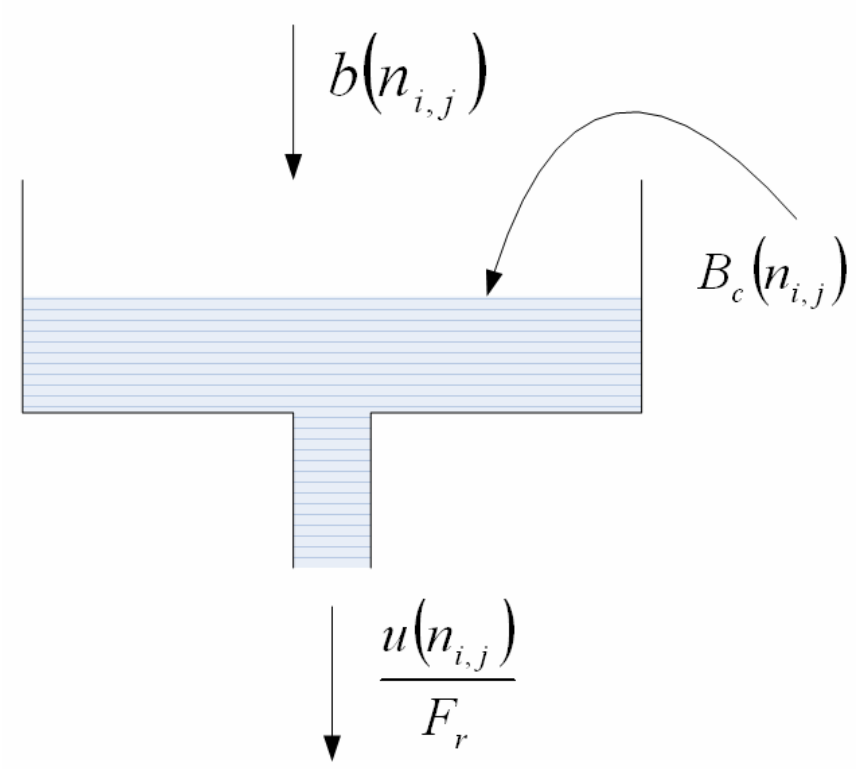


Fig. 2-3 Fluid Flow Traffic Model

We shall now present a fluid flow traffic model to compute the target bit for the current coding frame. Let N_{gop} denote the total number of frames in a group of

picture (GOP), $n_{i,j}(i=1,2,\cdots, j=1,2,\cdots,N_{gop})$ denote the *j*th frame in the *i*th GOP, and $B_c(n_{i,j})$ denote the occupancy of virtual buffer after coding the *j*th frame. We then have:

$$\begin{aligned} &B_c(n_{i,j+1}) = B_c(n_{i,j}) + b(n_{i,j}) - \frac{u(n_{i,j})}{F_r} \\ &B_c(n_{1,1}) = 0 \\ &B_c(n_{i+1,0}) = B_c(n_{i,N_{gop}}) \end{aligned} \tag{7}$$

where b($n_{i,j}$) is the actual number of bits generated by the *j*th frame in the *i*th GOP, u($n_{i,j}$) is the available channel bandwidth which can be either a VBR or a CBR, and F_r is the predefined frame rate.

C. A Linear Model for MAD Prediction

We now introduce a linear model to predict the MAD of current basic unit in the current frame by the actual MAD of the basic unit in the same position of the previous frame. Suppose that the predicted MAD of current basic unit in the current frame and the actual MAD of basic unit in the same position of previous frame are denoted by MAD_{cb} and MAD_{pb}, respectively. The linear prediction model is then given by

$$MAD_{cb} = a_1 \times MAD_{pb} + a_2 \tag{8}$$

where a_1 and a_2 are two coefficients of prediction model. The initial value of a_1 and a_2 are set to 1 and 0, respectively. They are updated after coding each basic unit. The linear model (8) is proposed to solve the chicken and egg dilemma.

D. HRD Consideration

In order to place a practical limit on the size of decoder buffer, a lower bound and an upper bound for the target bits of each frame are determined by considering the hypothetical reference decoder (HRD) [26]. Compliant encoders must generate

bistreams that meet the requirements of the HRD. The lower bound and upper bound for the *n*th frame are bounded by $L(n_{i,j})$ and $U(n_{i,j})$, respectively. It is also shown that HRD consideration is conformed if the actual frame size is always within the range $\lfloor L(n_{i,j}), U(n_{i,j}) \rfloor$.

Let $t_r(n_{i,j})$ denote the removal time of the *j*th frame in the *i*th GOP. Also let $be(t)$ be the bit witch is equivalent of a time *t*, with the conversion factor being the buffer arrival rate [40]. The initial values of the upper and the lower bound are given as follows:

$$\begin{aligned} L(n_{i,1}) &= T_r(n_{i,0}) + \frac{u(n_{i,0})}{F_r} \\ U(n_{i,1}) &= (T_r(n_{i,0}) + be(t_r(n_{1,1}))) \times \varpi \end{aligned} \tag{9}$$

where $T_r(n_{i,0})$ is the remaining bits of the *(i-1)*th GOP and $T_r(n_{1,0}) = 0$. The value of ϖ is 0.9.

$L(n_{i,j})$ and $U(n_{i,j})$ $(i = 1,2,\cdots, j = 2,\cdots, N_{gop})$ are computed iteratively as follows:

$$\begin{aligned} L(n_{i,j}) &= L(n_{i,j-1}) + \frac{u(n_{i,j-1})}{F_r} - b(n_{i,j-1}) \\ U(n_{i,j}) &= U(n_{i,j-1}) + \left(\frac{u(n_{i,j-1})}{F_r} - b(n_{i,j-1}) \right) \times \varpi \end{aligned} \tag{10}$$

2.3.3 Overview of the original H.264 Rate Control Scheme

With the concept of basic unit, models (7) and (8), the steps of the H.264 rate control scheme are given as follows:

1. Compute a target bit for the current frame by using the fluid traffic model (7) and bound it by HRD.
2. Predict the MAD of current basic unit in the current frame by the linear model (8)

using the actual MAD of basic unit in the co-located position of previous frame.

3. Allocate the remaining bits to all non-coded basic units in the current frame equally.

$$Rate_i = T \cdot \frac{BUMAD_i^2}{\sum_i^K BUMAD_i^2}$$
$$Rate_i = \max\left\{R_i, \frac{u(n_{i,j})}{MINVALUE \cdot F_r \cdot K}\right\} \quad (11)$$

where T is the bits allocated for current frame and $BUMAD_i$ is the predicted MAD in the ith basic unit of a frame. $MINVALUE$ is constant, and K is the total number of the basic unit.

4. Compute the quantization parameter by using the quadratic R-D model (5).

5. Perform RDO for each macroblock in the current basic unit by the quantization parameter derived from step 4.

2.3.4 GOP Layer Rate Control

In this layer, we need to compute the total number of remaining bits for all non-coded frames in each GOP and to determine the starting quantization parameter of each GOP. In the beginning of the GOP, the total number of bits allocated for the ith GOP is computed as follows:

$$T_r(n_{i,0}) = \frac{u(n_{i,1})}{F_r} \times N_{gop} - B_c(n_{i-1,N_{gop}}) \quad (12)$$

In CBR case, T_r is updated frame by frame as follows:

$$T_r(n_{i,j}) = T_r(n_{i,j-1}) - b(n_{i,j-1}) \quad (13)$$

The starting quantization parameter of the first GOP is a predefined quantization parameter QP_0. The I-frame and the first P-frame of the GOP are coded by QP_0. QP_0 is predefined based on the available channel bandwidth and the GOP length.

Normally, a small QP_0 should be chosen if the available channel bandwidth is high and a large QP_0 should be used if it is low.

The starting quantization parameter of other GOPs, QP_{st}, is computed by

$$QP_{st} = \frac{Sum_{PQP}}{N_p} + \frac{8 \times T_r\left(n_{i-1,N_{gop}}\right)}{T_r\left(n_{i,0}\right)} - \min\left\{2, \frac{N_{gop}}{15}\right\} \quad (14)$$

where N_p is the total number of P frames in the previous GOP and Sum_{PQP} is the sum of quantization parameters for all P frames in the previous GOP. Same as QP_0, QP_{st} is adaptive to the GOP length and the available channel bandwidth.

2.3.5 Frame Layer Rate Control

The frame layer rate control scheme consists of two stages: pre-encoding and post-encoding.

2.3.5.1. Pre-Encoding Stage

A. Quantization parameters of B frames

Since B frames are not used to predict any other frame, the quantization parameters can be greater than those of their adjacent P or I frames such that the bits could be saved for I and P frames. On the other hand, to maintain the smoothness of visual quality, the difference between the quantization parameters of two adjacent frames should not be greater than 2.

Suppose that the number of successive B frames between two P frames is L and the quantization parameters of the two P frames are QP_1 and QP_2, respectively. The quantization parameter of the *i*th B frame is calculated according to the following two cases:

Case 1: L=1. In other words, there is only one B frame between two P frames. The

quantization parameter of the B frame is computed by

$$\widetilde{QB}_1 = \begin{cases} \dfrac{QP_1 + QP_2 + 2}{2} & \text{if } QP_1 \neq QP_2 \\ QP_1 + 2 & \text{Otherwise} \end{cases} \quad (15)$$

Case 2: L>1. In other words, there are more than one B frame between two P frames. The quantization parameters of *i*th B frame between two P frames are computed by

$$\widetilde{QB}_i = QP_1 + \alpha + \max\{\min\{\frac{(QP_2 - QP_1)}{L-1}, 2\times(i-1)\}, -2\times(i-1)\} \quad (16)$$

where α is the difference between the quantization parameter of the first B frame and QP_1, and is given by

$$\alpha = \begin{cases} -3 & QP_2 - QP_1 \le -2\times L - 3 \\ -2 & QP_2 - QP_1 = -2\times L - 2 \\ -1 & QP_2 - QP_1 = -2\times L - 1 \\ 0 & QP_2 - QP_1 = -2\times L \\ 1 & \mathrm{QP}_2 - QP_1 = -2\times L + 1 \\ 2 & \text{Otherwise} \end{cases} \quad (17)$$

The case that $QP_2 - QP_1 < -2\times L + 1$ can only occur at the time instant where the video sequence switches from one GOP to another GOP.

B. Quantization parameters of P frames

The quantization parameters of P frames are computed via the following two steps:

Step 1 Determine a target bit for each P frame.

Step 1.1 Determination of target buffer occupancy.

We predefine a target buffer level for each frame according to the frame sizes of the first I frame and the first P frame, and the average complexity of previous coded frames. The function of the target buffer level is to compute a target bit for each P frame, which is then used to compute the quantization parameter. Since the quantization parameter of the first P frame is given at the GOP layer, we only need to

predefine target buffer levels for other P frames in each GOP.

After coding the first P frame in the *i*th GOP, we reset the initial value of target buffer level as

$$Tbl(n_{i,2}) = B_c(n_{i,2}) \tag{18}$$

where $B_c(n_{i,2})$ is the actual buffer occupancy after coding the first P frame in the *i*th GOP.

The target buffer level for the subsequent P frames is determined by

$$Tbl(n_{i,j+1}) = Tbl(n_{i,j}) - \frac{Tbl(n_{i,2})}{N_p - 1} + \frac{\tilde{W}_p(n_{i,j}) \times (L+1) \times u(n_{i,j})}{F_r \times (\tilde{W}_p(n_{i,j}) + \tilde{W}_b(n_{i,j}) \times L)} - \frac{u(n_{i,j})}{F_r} \tag{19}$$

where $\tilde{W}_p(n_{i,j})$ is the average complexity weight of P pictures, respectively.

In the case that there is no B frame between two P frames, Equation (19) can be simplified as

$$Tbl(n_{i,j+1}) = Tbl(n_{i,j}) - \frac{Tbl(n_{i,2})}{N_p - 1} \tag{20}$$

It can be easily shown that $Tbl(n_{i,N_{gop}})$ is about 0. Thus, if the actual buffer fullness is exactly the same as the predefined target buffer level, it can be ensured that each GOP uses its own budget. However, since the rate-distortion (R-D) model and the MAD prediction model are not accurate [18][19], there usually exists a difference between the actual buffer fullness and the target buffer level. We therefore need to compute a target bit for each frame to reduce the difference between the actual buffer fullness and the target buffer level.

Step 1.2 Microscopic Control (target bit rate computation).

The target bits allocated for the *j*th frame in the *i*th GOP is determined based on

the target buffer level, the frame rate, the available channel bandwidth and the actual buffer occupancy as follows:

$$\tilde{f}(n_{i,j}) = \frac{u(n_{i,j})}{F_r} + \gamma \times (Tbl(n_{i,j}) - B_c(n_{i,j})) \quad (21)$$

where γ is a constant.

The number of remaining bits should also be considered when the target bit is computed.

$$\hat{f}(n_{i,j}) = \frac{W_p(n_{i,j-1}) \times T_r(n_{i,j})}{W_p(n_{i,j-1}) \times N_{p,r}(j-1) + W_b(n_{i,j-1}) \times N_{b,r}(j-1)} \quad (22)$$

If the last frame is complex and uses excessive bits, more bits should be assigned to this frame. The target bit is a weighted combination of $\tilde{f}(n_{i,j})$ and $\hat{f}(n_{i,j})$:

$$f(n_{i,j}) = \beta \times \hat{f}(n_{i,j}) + (1-\beta) \times \tilde{f}(n_{i,j}) \quad (23)$$

where β is a constant.

Step 2 Compute the quantization parameter and perform RDO.

The MAD of current P frame is predicted by the linear model (8) using the actual MAD of previous P frame. Then, the quantization parameter $\hat{Q}_{pc}$ corresponding to the target bit is computed by using the quadratic model (5).

The quantization parameter is then used to perform RDO for each macroblock in the current frame by using the method.

2.3.5.2. Post-Encoding Stage

Finally, there are three major tasks in this stage: update the parameters a_1 and a_2 of linear model (8), the parameters X_1 and X_2 of quadratic R-D model (5), and determine the number of frames needed to be skipped.

2.3.6 Basic Unit Layer Rate Control

If the basic unit is not selected as a frame (a macroblock, a slice, or a group of

macroblocks) , an additional basic unit layer rate control should be added in the scheme.

Same as the frame layer, we shall first determine the target bit for each P frame. The process is the same in that at the frame layer. The bits are then allocated to each basic unit. First, the MADs of all non-coded basic units in the current frame are predicted by linear model (8) using actual MAD of bask unit in the same position of previous frame, and we allocate the remaining bits to all non-coded basic units in the current frame by function (11) using these predicted MADs.

Then, we compute the quantization parameter of current basic unit by using quadratic R-D model (5). But, we need to consider the following three cases:

Case 1: The quantization parameter for first basic unit in the current frame is assigned to the average value of quantization parameters for all basic units in the previous frame.

Case 2: If the number of remaining bits for all non-coded basic units in the current frame is less than zero, the quantization parameter should be greater than that of previous basic unit.

Case 3: Otherwise, we shall compute quantization parameter by using the quadratic model.

After all, the RDO process and updating for parameters of linear model and quadratic model is done by the same way as the frame layer.

2.4 Bit Allocation Strategy

In the previous section, we have introduced the rate control strategy in H.264. And there are many other schemes proposed to improve it.

Pan et al. [28] proposed a new scheme for the bit allocation of each P frame to further improve the perceptual quality of the reconstructed video. A new

least-mean-square estimation method of the R-D model parameters was developed by Nagn et al. [29]. However, these target bit estimation schemes, as an important factor in determining the quantization parameter (QP), are distributing bits to every basic unit equally without considering the complexity of the frame, and it results in poor target bit estimation for different frames.

In [30][31], Ling et al. had proposed a modified algorithm using more accurate frame complexity to allocate bits. While the predicted MAD calculated in linear model (8) is not very accurate, Yu et al. [32] have used a measure named motion complexity of the frame to distribute more bits to high motion scenes. However, these methods only try to allocate more bits to complex frames, and it only results in a general better quality to whole frame.

Since the human visual system (HVS) is more sensitive to the moving regions, it is worthwhile to sacrifice quality of the background regions while enhancing that of the moving regions. Some research works on region/content-based rate-control have been reported [33][34]. They adopted a heuristic approach to decide the quantization parameters for different regions in a frame. Region of Interest (ROI) will obtain a finer quantizer and a coarser quantizer will be used for non-ROI. These methods [33][34] just set quantizers with constants and do not take the contents of region into consideration, and this may cause improper QPs and unreasonable bits used for different regions. So, there are some other improved algorithms trying to adaptively adjust these factors. Lai et al. [35] proposed a scheme which uses a region-weighted rate-distortion model to calculate different QPs for different regions. Sun et al. [36] also proposed a scheme to allocate bits to foreground and background by utilizing a weighting function for different regions. However, these algorithm [33]-[36] only use fixed values or simple region-based weighting scheme to assign quantization parameters to foreground and background without considering the characteristics of

these regions.

In [37-38], the algorithms that take account of size, motion and priority of the foreground and background regions has been proposed. But these methods adjust the quality of foreground/background by taking the whole foreground as one part. Since there may be multiple objects in the foreground region, we propose an algorithm utilizing the features of different objects to further adjust different quality of these object regions.

Chapter 3

Motion-based Object Segmentation and Feature-based Bit Allocation Scheme

In this chapter, we present our methods for video object segmentation and rate control. In section 3.1, we first go through the whole scheme and give an overview quickly. In section 3.2, we present the object segmentation algorithm. And in section 3.3, the bit allocation strategy for background and foreground objects is presented.

3.1 Overview

Our proposed scheme contains two parts, video object segmentation parts and the bit allocation parts. Since we are focusing on uncompressed video input sources, the object segmentation algorithm is only used with inter-coding frames. In the beginning, we use a multi-resolution algorithm to find the motion vector. In the coarsest level, we establish a object mask and a object set by using coarse motion vectors generated in the motion estimation modules. While in every finer level the multi-resolution algorithm refines the motion vectors, we also use these finer motion vectors to update our objects mask and object set. Then, the object set is then used by bit allocation module. The bit allocation strategy uses the information of objects to judge the importance of foreground objects and background, and then different coding bits will be allocated to these regions to keep the visual quality of foreground objects. The flow of the whole system is illustrated in Fig. 3-1.

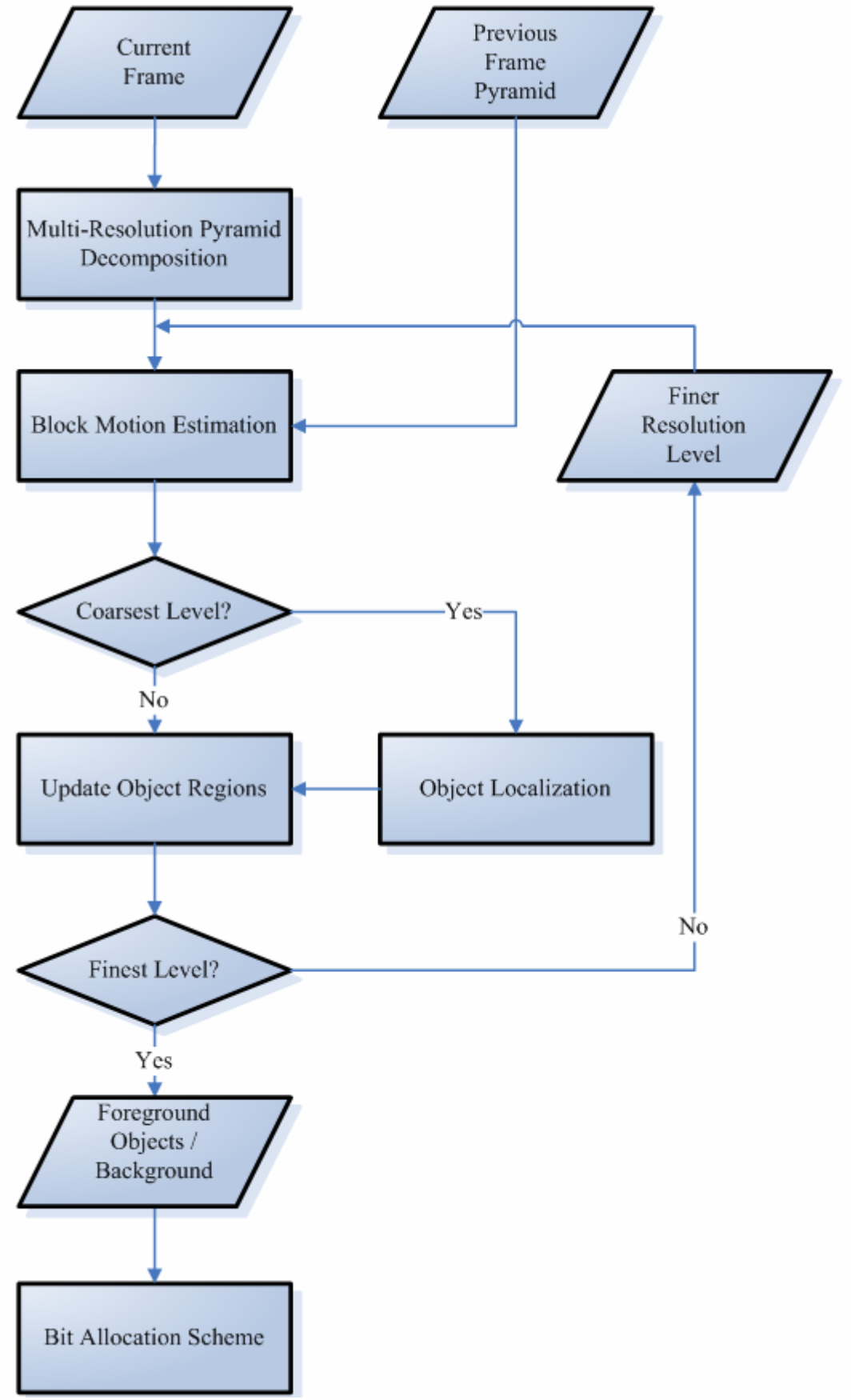


Fig. 3-1 System Overview

3.2 Motion-based Video Segmentation Algorithm

The video segmentation algorithm directly takes the raw video data as input to segment the object regions and extracts the object mask for proceeding processing. A multi-resolution pyramid structure has been adopted to find motion vectors and to segment objects by utilizing the motion vectors iteratively. In section 3.2.1, we will present the multi-resolution motion estimation algorithm, and in section 3.2.2, the object localization algorithm will be proposed. The algorithm of updating object regions and morphological operation will be proposed in section 3.2.3 and 3.2.4, respectively.

3.2.1. Multi-Resolution Motion Estimation

For the sake of reducing the computation load segmentation, a multi-resolution motion estimation algorithm has been applied. The multi-resolution algorithm is chosen due to its pyramid structure, robustness and improvements in comparison to the one-level schemes. Since motion clustering is time-consuming, we can utilize the iterative pyramid structure to decrease the complexity by generating a rough mask at the coarsest level and refining it at each finer level.

In the following, we will present the details of the multi-resolution motion estimation scheme that has been used in our system.

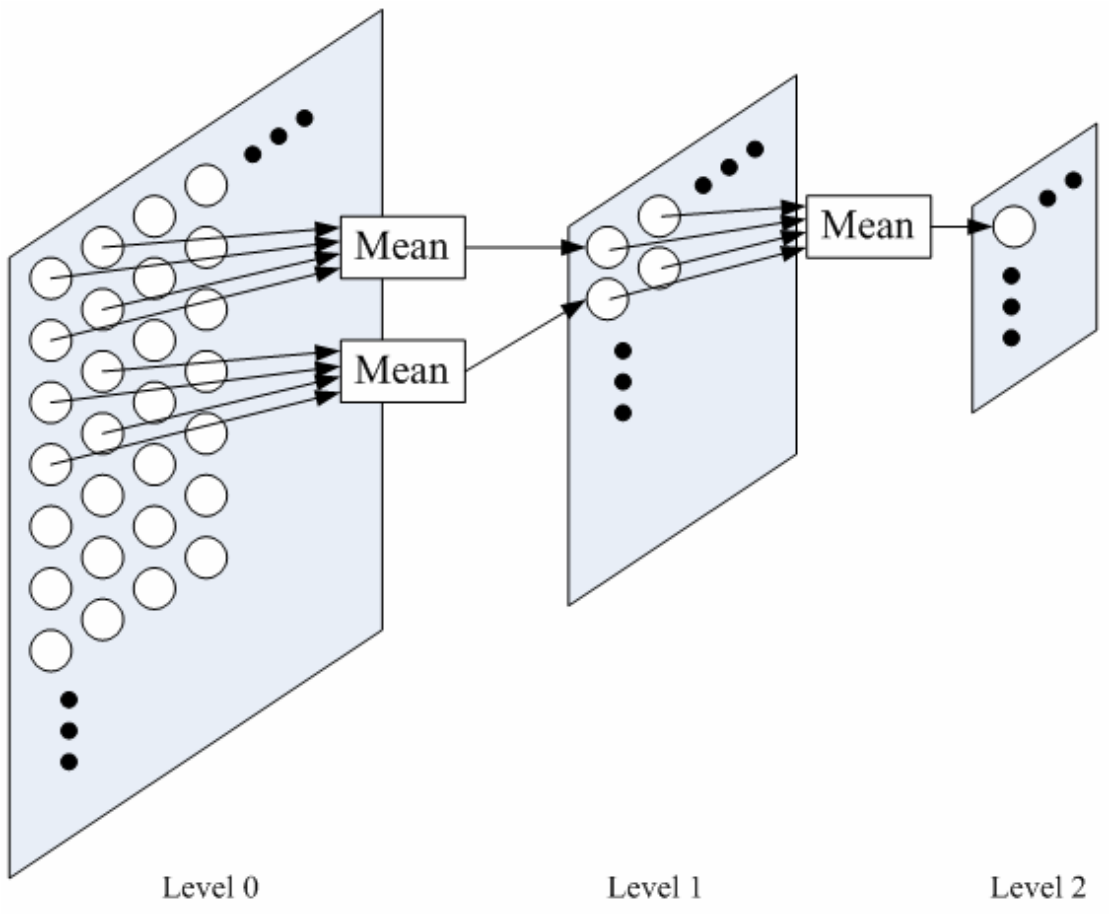


Fig. 3-2 Multi-Resolution frame structure

3.2.1.1 Multi-Resolution Frame Structure

The multi-resolution motion estimation we applied is a simple method. First we decompose the input frame into a three layer pyramid by the following sub-sampling function:

$$I_k^{(l+1)}(i, j) = \frac{1}{4}\sum_{m=0}^{1}\sum_{n=0}^{1} I_k^{(l)}(i+m, j+n) \qquad (24)$$

where $I_k^{(l+1)}(i, j)$ represents the intensity value at the position (i, j) of the kth frame at level $l + 1$. The number of pixels in the next upper level is reduced to on fourth of the lower level. The multi-resolution frame structure is illustrated in Fig. 3-2. The MB size becomes 16 × 16, 8 × 8 and 4 × 4 at levels as 0, 1, and 2, respectively.

The sum of absolute difference (SAD) is widely used as the matching criterion for BMA due to its low computational cost. For a 16 × 16 MB, SAD at level *l* can be defined as:

$$SAD_{MB}^{(l)}(p,q) = \sum_{i=0}^{\left(16/2^l-1\right)}\sum_{j=0}^{\left(16/2^l-1\right)}\left|I_k^{(l)}(i,j)-I_{k-1}^{(l)}(i+p,j+q)\right| \quad (25)$$

where *l* is the level number and *l* = 0, 1, 2, In Eq. (25), (p,q) denotes a motion vector in a given search range.

3.2.1.2 Motion Search Framework

1) Search at Level 2: We choose two candidates, i.e., $\left\{MV_1^{(1)}, MV_2^{(1)}\right\}$, based on the spatial correlation in motion vector fields as well as minimum SAD, and employ them as initial search centers at level 1. $MV_1^{(1)}$ having the minimum SAD are found by full search within a search range SR_2:

$$MV_1^{(1)} = 2\cdot\left(\arg\min_{(p,q)\in SR_2} SAD_{MB}^{(2)}(p,q)\right) \quad (26)$$

where $SR_2 = \left\{(p,q)\,|\,-\frac{w}{4}\le p\le\frac{w}{4},-\frac{w}{4}\le q\le\frac{w}{4}\right\}$ and *w* is the predefined search range by encoder.

$MV_2^{(1)}$ is predicted from adjacent motion vectors at level 0 via a component-based median predictor.

2) Search at Level 1: Local search are performed around the two candidates in order to find a motion vector candidate for the search at level 0.

$$MV^{(0)} = 2\cdot\left(\arg\min_{(p,q)\in SR_1} SAD_{MB}^{(1)}(p,q)\right) \quad (27)$$

where

$$SR_1 = \{(p,q) \mid -2 \le (p - p_n^{(1)}) \le 2, -2 \le (q - q_n^{(1)}) \le 2\}$$
$$(p_n^{(1)}, q_n^{(1)}) = MV_n^{(1)}, n = 1,2$$

3) Search at Level 0: A final motion vector is found from a local search around $MV^{(0)}$ as follows:

$$MV_{MB} = \left(\arg \min_{(p,q) \in SR_0} SAD_{MB}^{(0)}(p,q) \right) \quad (28)$$

where

$$SR_0 = \{(p,q) \mid -2 \le (p - p^{(0)}) \le 2, -2 \le (q - q^{(0)}) \le 2\}$$
$$(p^{(0)}, q^{(0)}) = MV^{(0)}$$

3.2.2. Object Localization

At the coarsest level, after multi-resolution motion estimation, object localization algorithm is used to locate potential objects in a video sequence for subsequent object based bit allocation. Initially, we check if there is any camera motion of each frame and compensate motion vectors with global motion if camera motion happens. Otherwise, noisy motion vectors are eliminated directly without motion compensation. Subsequently, motion vectors that have similar magnitude and direction are clustered together and this group of associated macroblocks of similar motion vectors is regarded as an object. The overview of the algorithm of object localization is shown in Fig 3-3.

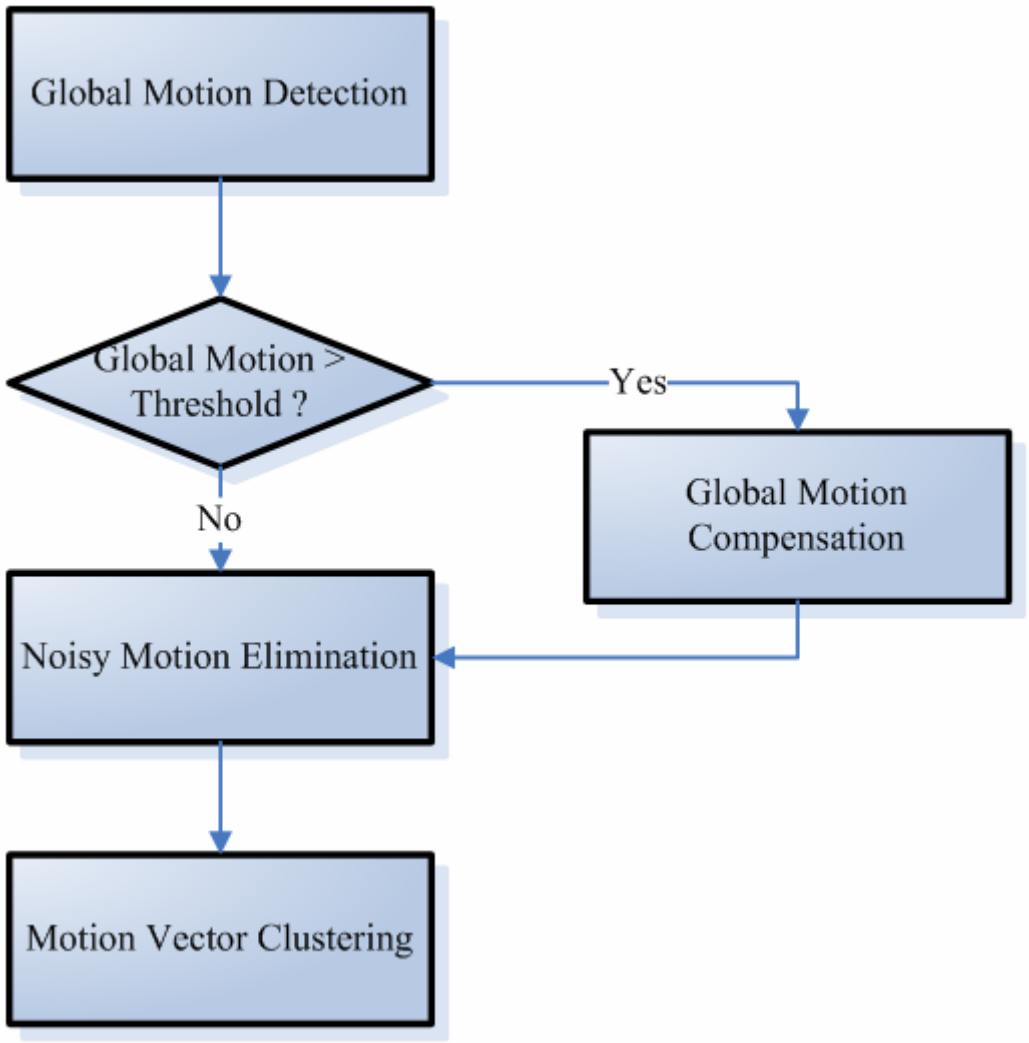


Fig. 3-3 Object localization algorithm

3.2.2.1. Global Motion Estimation

To correctly locate the position of objects, global motion (camera motion) such as panning, zooming, and rotation, should be estimated for compensation. In this section, a fast and simplified global motion detection algorithm is proposed.

Many global motion estimation algorithms have been proposed, and are based on the motion model of two (translation mode), four (isotropic model), six (affine model), eight (perspective model), or twelve parameters (parabolic model). They can be classified into three types: frame matching, differential technique, and feature points based algorithm.

Since all the method based on motion model need heavy computation, we propose a simple algorithm to calculate global motion by using histogram to reduce the complexity.

The histograms of magnitude and direction of motion vectors are computed to acquire dominant motion direction and dominant motion magnitude to further identify whether global motion, pan and tilt, happens or not. Using the approach of histogram-based dominant motion computation, we can avoid matrix multiplications, which are computationally inefficient when motion vectors are fit to motion model. The magnitude and direction of camera motion are obtained by using the equations below:

$$SDMH_i = Num(Bin_{DMH-1,i}) + Num(Bin_{DMH,i}) + Num(Bin_{DMH+1,i}) \quad (29)$$

$$SDAH_i = Num(Bin_{DAH-1,i}) + Num(Bin_{DAH,i}) + Num(Bin_{DAH+1,i}) \quad (30)$$

where *DMH* and *DAH* are the dominant magnitude and dominant direction of motion vector histogram, respectively, $SDMH_i$ is the summation of three bins ($Bin_{DMH-1,i}$, $Bin_{DMH,i}$, $Bin_{DMH+1,i}$) of the magnitude histogram of the i^{th} frame, $SDAH_i$ is the

summation of three bins ($Bin_{DAH-1,i}$, $Bin_{DAH,i}$, $Bin_{DAH+1,i}$) of direction histogram of the i^{th} frame, and $N(Bin_{j,i})$ means the value of the j^{th} bin in the i^{th} frame.

In the ideal situations, macroblocks in an object would have the same motion magnitude and direction. However, although the entire objects moves toward the same direction, some regions in the object might have different but similar motion magnitude and direction because objects in real world are not rigid in their shape and size. Consequently, to tolerate the error of motion estimations, the values of $Bin_{DMH-1,i}$, $Bin_{DMH,i}$ and $Bin_{DMH+1,i}$ of magnitude histogram are summed to examine whether the summation $SDMH_i$ is larger than the threshold or not, and the values of $Bin_{DAH-1,i}$, $Bin_{DAH,i}$ and $Bin_{DAH+1,i}$ of direction histogram are summed to examine $SDAH_i$. If $SDMH_i$ and $SDAH_i$ are both larger than the threshold T_{global}, global motion happened, and *DMH* and *DAH* are identified as magnitude and direction of camera motion in i^{th} frame. Moreover, motion vectors are compensated with the magnitude and direction of global motion for further processing.

3.2.2.2. Object Clustering

We use region growing approach to cluster macroblocks that have motion vectors with similar magnitude and direction together and this group of associated macroblocks of similar motion vector is regarded as an object,. Detailed algorithm is presented in the following.

Object Localization Algorithm

Input: Coarsest layer of the input frame

Output: Object sets { Obj_1, Obj_2, …, Obj_n }, where *n* is the total number of objects in frame. Each object size is measured in terms of the number of

macroblocks, and the centroid of the object is also calculated by averaging the coordinates of all macroblocks inside the object region.

Step 1. Analyze motion vector of inter-coded macroblocks in a frame to see if there is any camera motion.

Step 2. If there is no global motion, go to step 3. If glolbal motion is detected, motion vectors that are not noisy are compensated with camera motion magnitude and direction.

Step 3. Cluster motion vectors that are of similar magnitude and direction into the same group with region growing approach.

Step 3.1 Set search windows (*W*) size 3x3 macroblocks.

MV_1	MV_2	MV_3
MV_4	*Center*	MV_5
MV_6	MV_7	MV_8

Step 3.2 Search all macroblocks within *W*, and compute the difference ($diffMag_k$ and $diffAng_k$) of motion vector magnitude ($|MV|$) and direction ($\angle MV$) between center MV_{center} and its neighboring eight motion vectors MV_k within *W*.

$$\begin{aligned} diffMag_k &= abs(|MV_{center}| - |MV_k|) \\ diffAng_k &= abs(\angle MV_{center} - \angle MV_k) \end{aligned} \qquad (31)$$

where MV_{center} is the motion vector in the center position of *W* and $MV_k \in$ motion vectors within *W* except MV_{center}, $k \in [1,8]$

For all $k \in [1,8]$, $flag\, F_k = \begin{cases} 1, & diffMag_k < T_{Mag} \text{ and } diffAng_k < T_{Ang} \\ 0, & otherwise \end{cases}$

where T_{Mag} is the predefined threshold for motion vector magnitude and T_{Ang} is the threshold for motion vector direction.

If $\sum_{k=1}^{8} F_k \geq 6$, mark F_{center} of MV_{center} as 1, where F_{center} is the flag of the center motion vector within *W*. Otherwise, set all flags within *W* to 0.

Step 3.3 Go to step 3.2 until all macroblocks are processed.

Step 3.4 Group macroblocks that are marked as 1 into the same cluster.

Step 3.5 Compute each object center and record its associated macroblocks.

Step 3.6 Generate one object set for each P-frame.

3.2.3. Update Object Regions in Finer Level

While the multi resolution motion estimation algorithm is iteratively refining motion vectors of every macroblocks in a frame at each finer level, the rough object mask generated at coarsest level is also refined by these refined motion vectors. Details of the refining algorithm is presented as follows.

Object Sets Refining Algorithm

Input: Object set $\{Obj_1, Obj_2, \ldots, Obj_n\}$.

Output: Refined object set $\{Obj_1, Obj_2, \ldots, Obj_n\}$ where n is the total number of refined objects in a frame. The object size, dominant motion vector magnitude/direction and centroid are measured by the number of macroblocks within the object, average value of motion vector magnitude/direction and average value of coordinates, repectively.

Step 1. Calculate the motion vector magnitude and direction of the centroid macroblock.

Step 2. Search all macroblocks within the object region, and compute the difference *diffMag* and *diffAng* of motion vector magnitude and direction between centroid and these macroblocks. The block will be excluded from the object

if both $diffMag > T_{Mag}$ and $diffAng > T_{Ang}$ where T_{Mag} and T_{Ang} are predefined thresholds

Step 3. Go to step 2 until all macroblocks are processed.

Step 4. Generate the object mask with the reformed object set, then refining the object mask by employing morphological operation and regenerating the object set with the fined mask.

3.2.4. Morphological Operation

To smooth the boundaries of regions of interest and remove the noisy blocks, two kinds of morphological operations are frequently used. The closing operation is first used to fill the block holes inside the objects mask and the opening operation is the used to remove the small noise blocks that do not belong to the moving objects. In our algorithm, the structure element of size 3×3 is selected for closing and opening operations respectively.

After the morphological operations, the object mask is refined and indicates the shapes and the positions of all the moving objects in the current frame. Then, the individual objects can be extracted to generate the new object set.

3.3 Feature-based Bit Allocation Strategy

Our proposed bit allocation method is based on the characteristics of the object regions, which include size and motion. In order to make a more accurate bit distribution, we will allocate the bits in the frame level first.

3.3.1. Frame Level Rate Control

It is well known that MAD of the residual component can be a good indication of encoding complexity. In the quadratic R-D model, the encoding complexity is usually substituted by MAD. In order to solve this problem of distributing the bits to different frames, we refer to the scheme in [30] and adopted here.

3.3.1.1. Measure of Frame Encoding Complexity

A MAD ratio is used to measure the complexity of a frame, which is the ratio of the predicted MAD of current frame to the average MAD of all previous encoded frames. The MAD ratio of *i*th frame is calculated as the following:

$$MAD_{ratio}(i) = \frac{MAD_i}{\left(\sum_{J}^{i-1} MAD_{avg}^{j}\right) \Big/ (i-1)} \qquad (32)$$

where MAD_i is calculated by linear model (8), MAD_{avg}^{j} is the average MAD of *j*th previous coded frame and (*i-1*) is the total number of previous coded frames.

3.3.1.2. Adaptive Target Bit Estimation Control

We use the MAD ratio to simply control the target bits estimation for the frame. The distribution of the bit count is scaled by a function of MAD_{ratio}. Initial target bits T_r for a frame can be adjusted as shown in the following pseudo code:

Calculate the *average MAD* **of all previously inter-coded frames;**

Calculate the MAD_{ratio} **using** *predicted MAD of the current frame / average MAD;*

IF (MAD_{ratio} < 0.9) **THEN**

$T_r = T_r * 0.5$

ELSE IF (MAD_{ratio} < 1.0) **THEN**

$T_r = T_r * MAD_{ratio} * 0.6$

ELSE IF (MAD_{ratio} < 1.8) **THEN**

$T_r = T_r * MAD_{ratio} * 0.7$

ELSE IF (MAD_{ratio} >= 1.8) **THEN**

$T_r = T_r * 1.8$

The basic idea is to set T_r smaller if the current frame complexity is low and set T_r larger if the current frame complexity is high. The objective of the improvement is to save bits from those frames with relatively less complexity and allocate more bits to frames with higher complexity duo to high motion or scene changes.

3.3.2. Macroblock Rate Control

In the macroblock level, a content-based bit allocation strategy has been used in our scheme. We have proposed an approach whereby bit allocation to every region is determined based on the characteristics of different image regions. These characteristics include object region size and object dominant motion.

• **Size:** First, bit allocation is governed by the size of the object region and background region. The normalized size of each object regions is determined by $S_f^i = N_f^i / N_f$ and for background, $S_b = N_b / N$ where N_f^i is the total number of macroblocks in the *i*th foreground object, N_b is the total number of macroblocks

in the background, N_b is is the total number of macroblocks in the foreground and N is the number of macroblocks in a frame.

• **Motion:** Bit allocation is also performed according to the activity of each object region, which can be measured by its motion. The normalized motion parameters for each object are derived as

$$M_f^i = \frac{\left|MV_{dominant}^i\right|}{\sum_j^k \left|MV_{dominant}^j\right|}$$

where $\left|MV_{dominant}^i\right|$ is the dominant motion magnitude of the *i*th object.

Based on the above characteristics, the amount of bits can be assigned to foreground objects and background region as follows:

Method 1: If $S_b > TH_b$ and TH_b is the threshold for the ratio of number of macroblocks in the background to the number of macroblocks in the frame, then the bits allocation is done as follows:

$$\begin{aligned} B_f &= \frac{X_1 \cdot MAD_f}{Q_{best}} + \frac{X_2 \cdot MAD_f}{Q_{best}^2} \\ B_b &= T_r - B_f \end{aligned} \tag{33}$$

where B_f is the bits allocated to the foreground, B_b is the bits allocated to a macroblock of the background, MAD_f are the predicted MAD of the frame and Q_{best} is the quantization level determined by the *QP* of this frame that is calculated in section 2.3.5.1.

Method 2: If $S_b < TH_b$, then bits are distributed as follows:

$$\begin{aligned} B_b &= T_r \cdot S_b \cdot \alpha_p \\ B_f^i &= \left(T_r - B_b\right) \cdot \left(\omega_M M_f^i + \omega_S S_f^i\right) \end{aligned} \tag{34}$$

where α_p means the portion of the bits that the background will transfer to foreground, ω_M, ω_S are the respective weighting functions of the size and motion parameters and $\omega_M + \omega_S = 1$.

3.3.3. Post-Encoding Process

After encoding, the encoder updates the R-D model based on the encoding results. The first and second model parameters X_1 and X_2 are updated by using the linear regression technique [40]. And the buffer fullness is updated after encoding by fluid flow traffic model (7).

Chapter 4

Experimental Results

In this chapter, we will show the experimental results and give some discussions. We will describe the experiment environment first in section 4.1. And we will list some results in section 4.2

4.1 Experiment Environment

In this thesis, we implemented the object segmentation algorithm and bit allocation method by modifying the H.264 reference software JM 9.5[39] and the original version was used to comparison purposes. All experiments were conducted on the PC with an Intel Pentium 4 CPU 2.4 GHz and 256 MB of RAM.

Our experimental work uses the following approach:

1) First frame is intra-coded and others are P-frames.
2) Macroblock type only use 16 × 16
3) Original version is adopted the multi-resolution motion estimation.

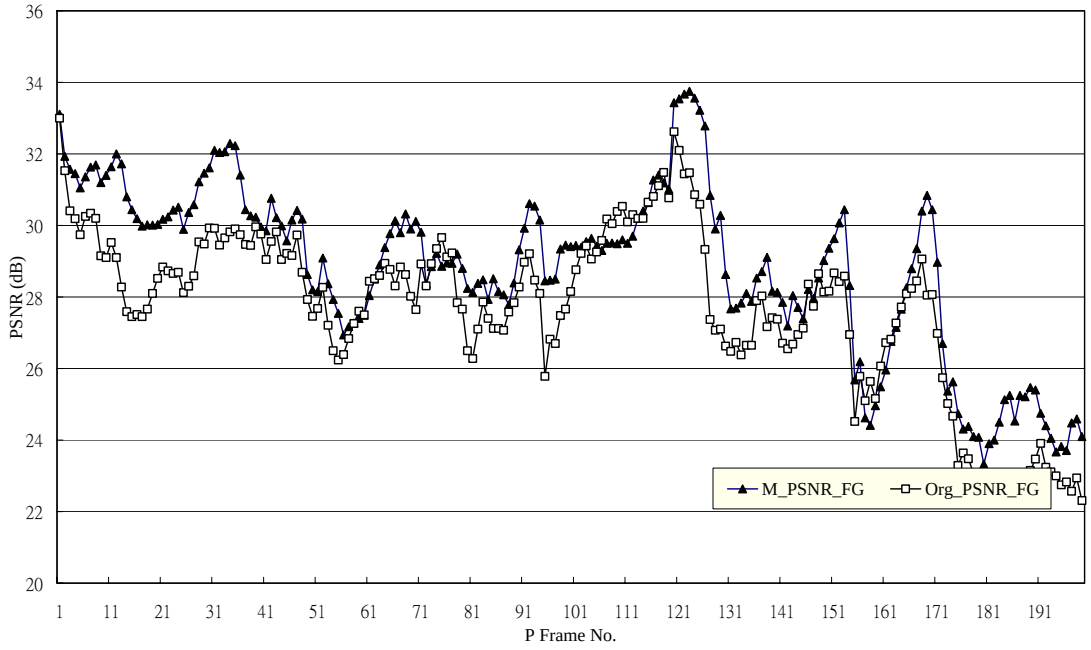
4.2 Experimental Results

We had experimented with five sequences: “Football” and “Stefan” of SIF format (352 × 240) and “Foreman”, “Mother and daughter”, and “Hall” of CIF format (352 × 288). According to the sequence type, we encoded the Football and Stefan sequence with 500 kbps because their high motion. Foreman and Mother were encoded with 100 kbps because their obvious foreground region. And we had compressed the Hall sequence with 50 kbps because its static scene with small foreground objects.

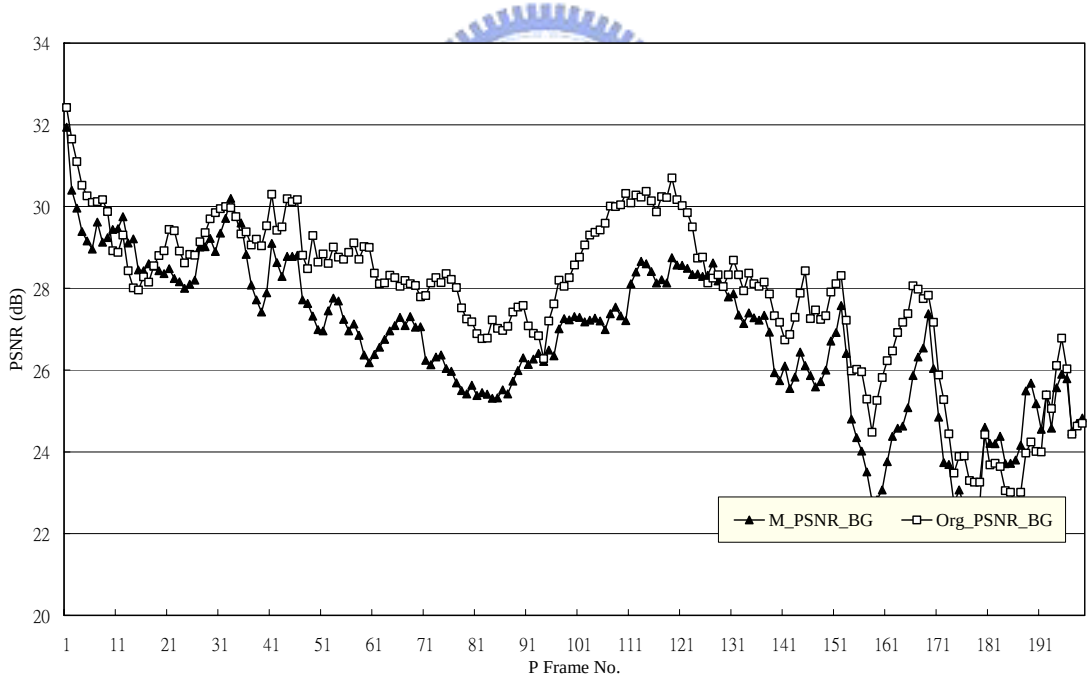
	Original Version			Modified Version		
	AVG PSNR FG (dB)	AVG PSNR BG (dB)	Bitrates (kbps)	AVG PSNR FG (dB)	AVG PSNR BG (Db)	Bitrates (kbps)
Football	23.73	25.52	513.33	25.73	24.44	557.58
Stefan	28.77	29.94	510.48	30.17	28.47	541.58
Foreman	27.73	27.87	111.01	28.85	26.82	114.47
Mother and daughter	33.53	36.48	109.23	34.47	35.66	110.48
Hall	24.94	32.30	51.37	26.05	30.99	51.26

Table 4-1 Encoded Results of five sequence of JM original version and original version

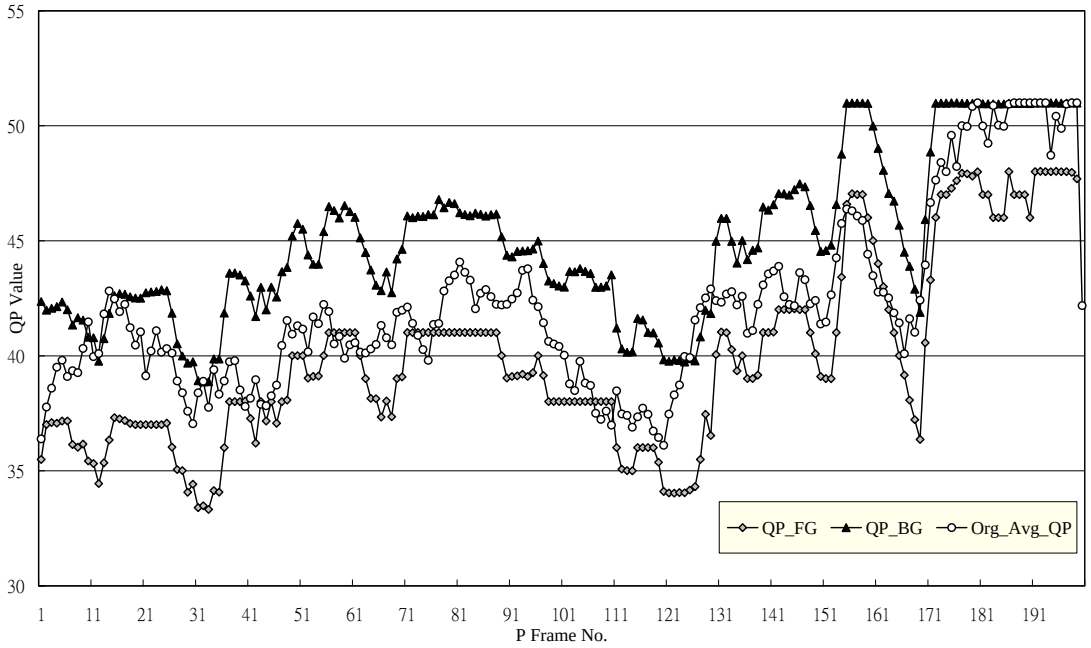
First, with encoding a sequence which has only one obvious object, we use the CIF format sequence “Foreman” for example. We had encoded the sequence with 100 kbps and compare the original version without bit allocation and the modified version with our proposed method. First, by comparing the object quality, we can see that our method have improved the average quality of foreground object region In Fig.4-1(a). Second, by comparing the generally quality, the average PSNR of the foreground was improved by 1.12 dB in Table 4-1, whereas the background quality was degraded by 1.05 dB. Finally, by comparing the two encoded images shown in Fig. 4-2, we can clearly see that the quality of facial region was much improved and its bit-rate only increase 0.03%.


PSNR (dB)
P Frame No.
M_PSNR_FG
Org_PSNR_FG

(a)

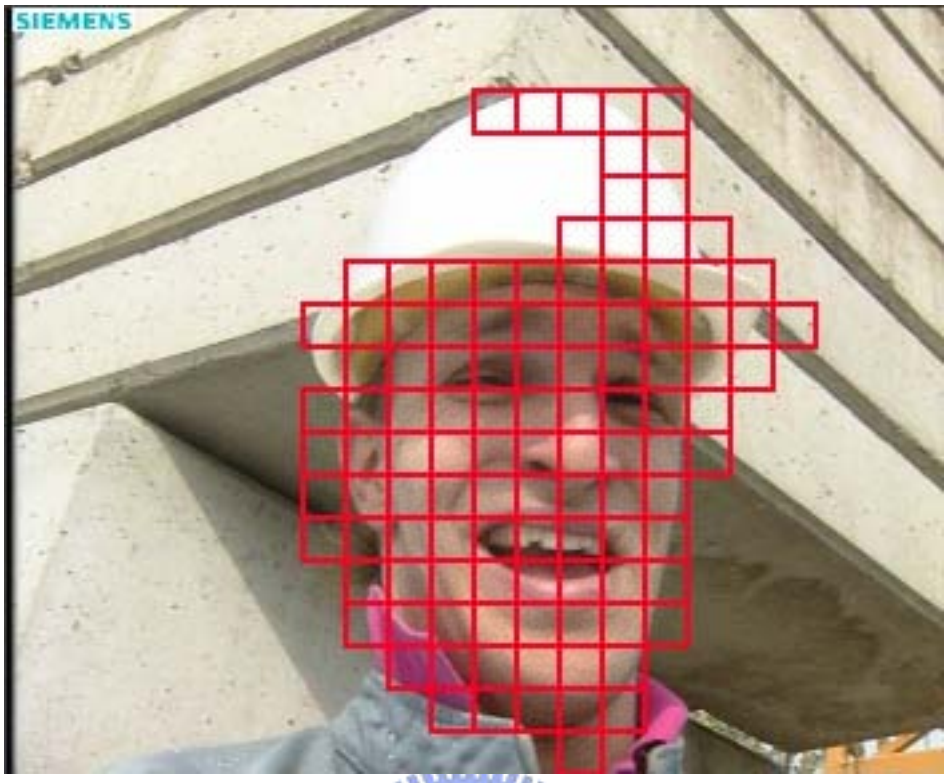

PSNR (dB)
P Frame No.
M_PSNR_BG
Org_PSNR_BG

(b)



(c)

Fig. 4-1 Foreman sequence encoded by original JM software and modified version: (a) Average PSNR of foreground region in original and modified version, (b) Average PSNR of background region in original and modified version, (c) Average QP value of foreground/background in modified version and average QP value in original version.



(a) Segmented Result

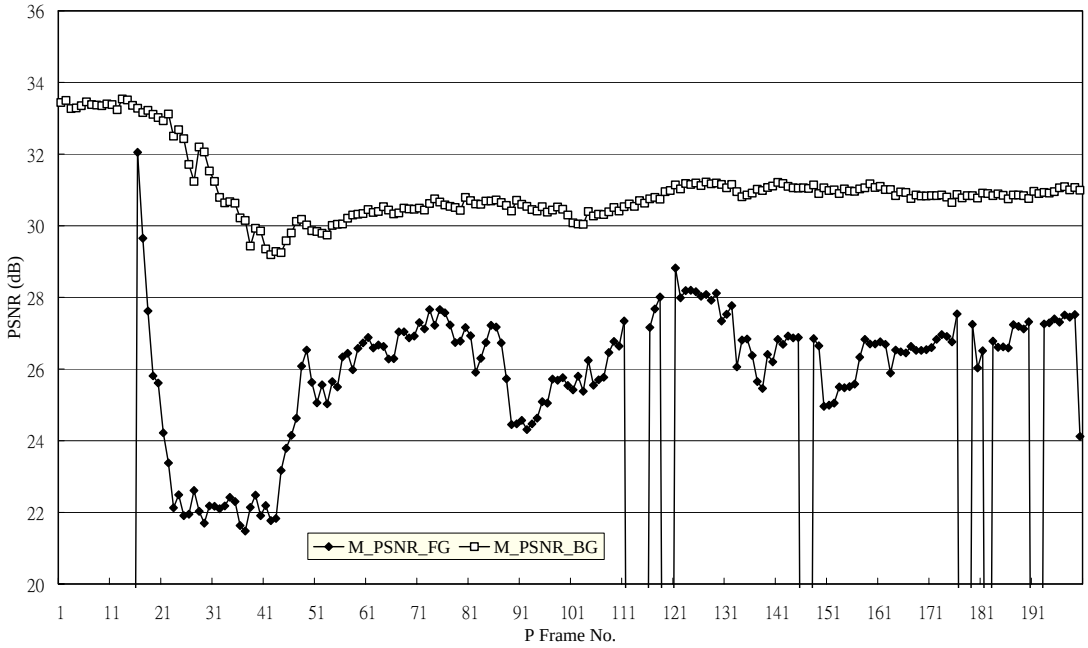
(b) Original version.

(c) Modified version.

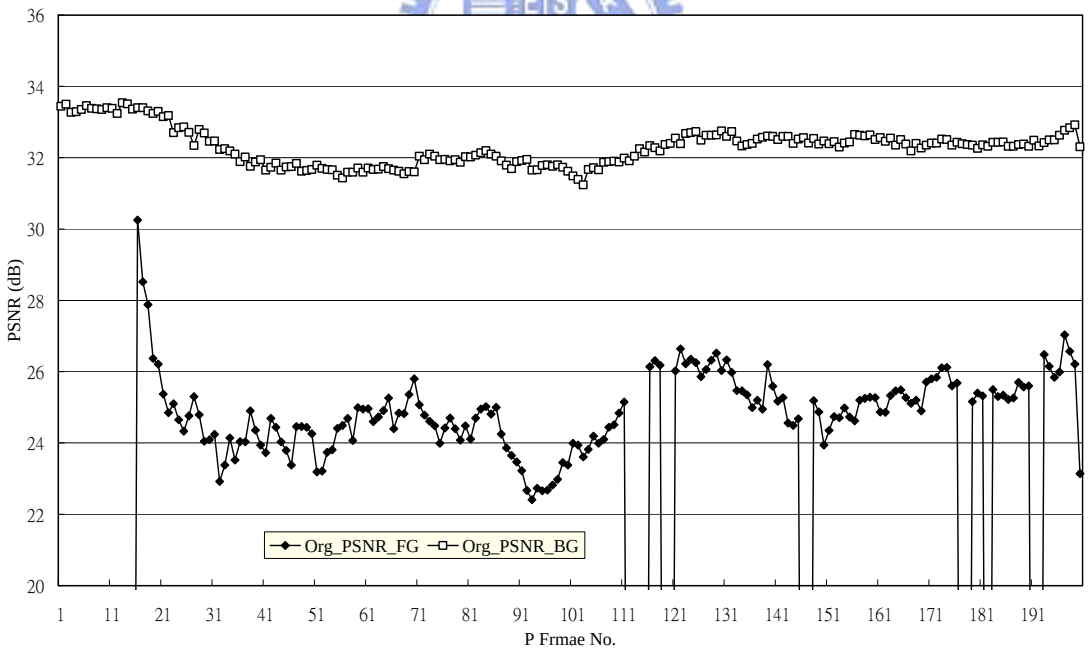
Fig. 4-2 Results of Foremen sequence encoded by (a) segmented result (b) original version and (c) Modified version JM.

Second, with the sequence which has static scene and small foreground objects, such as CIF format source “Hall” sequence, we had encoded the sequence with 50 kpbs. First, we see the PSNR that in our method of the foreground region is still worse than the background in Fig. 4-3 (a). But in Fig.4-3 (b), we can see that in the original JM, PSNR of the foreground is already worse than background. And in Fig. 4-3 (c), (d) and Table 4-1, we clearly see that we had improved the quality of the foreground region with 1.11 dB whereas the background quality was only degraded by 1.31 dB. In Fig 4-3 (a), (b) and (c), the value zero of the PSNR of foreground is because there has no foreground objects had been segmented. In frames 1 to 16, there are really no objects in the scene, but in other frames, objects are not segmented because these frames are too similar to previous frame and there are no motion information can be

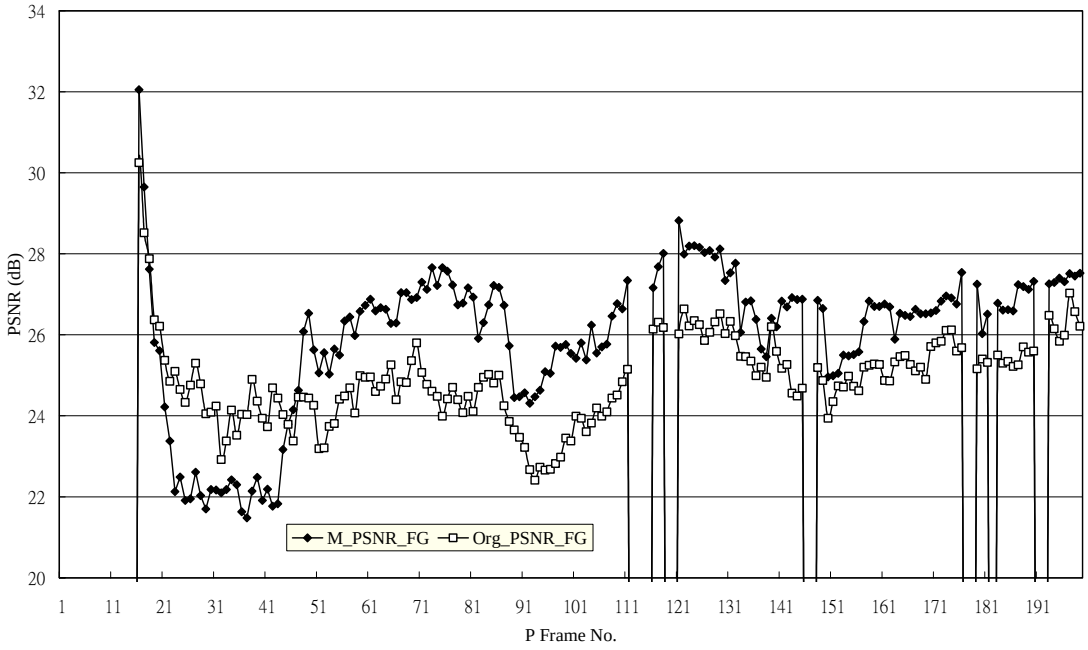
used in the object regions. The subjective quality is shown as Fig. 4-4.



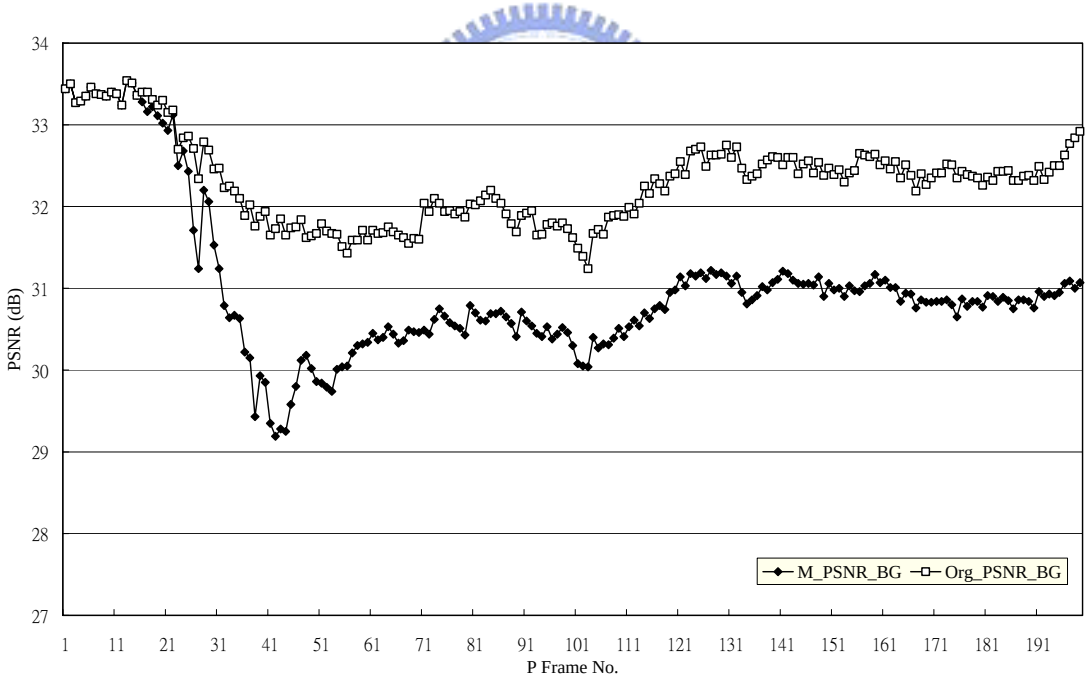
(a)



(b)

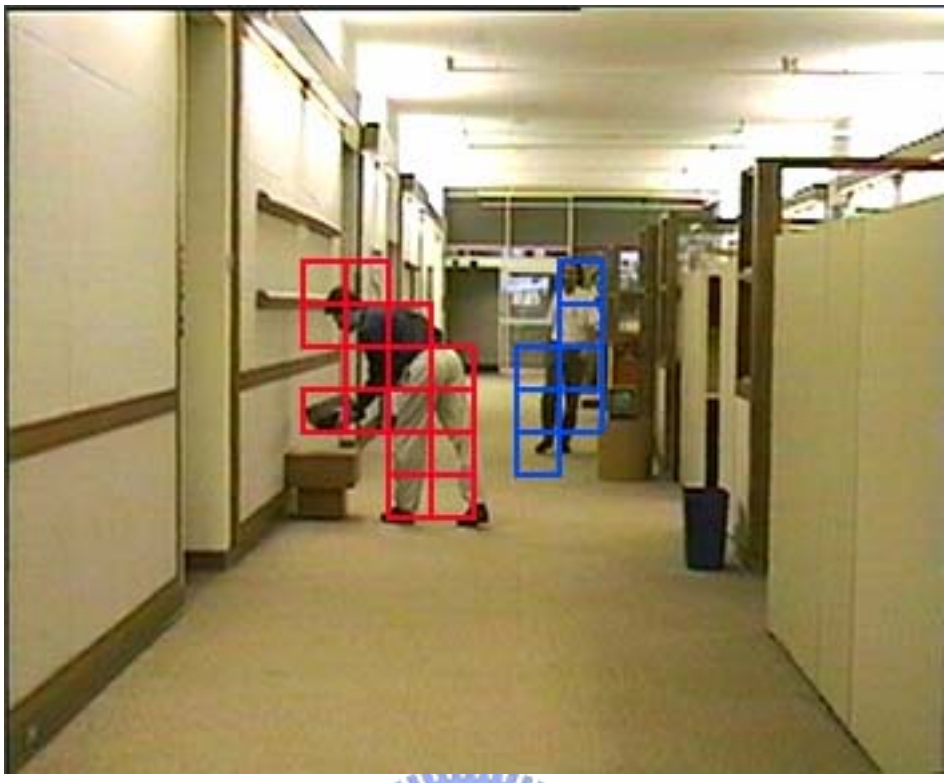


(c)



(d)

Fig. 4-3 Hall sequence encoded by original JM software and modified version: (a) PSNR of foreground/background region of modified version, (b) PSNR of foreground/background region of original version, (c) PSNR of foreground region in original and modified version, (d) PSNR of background region in original and modified version.



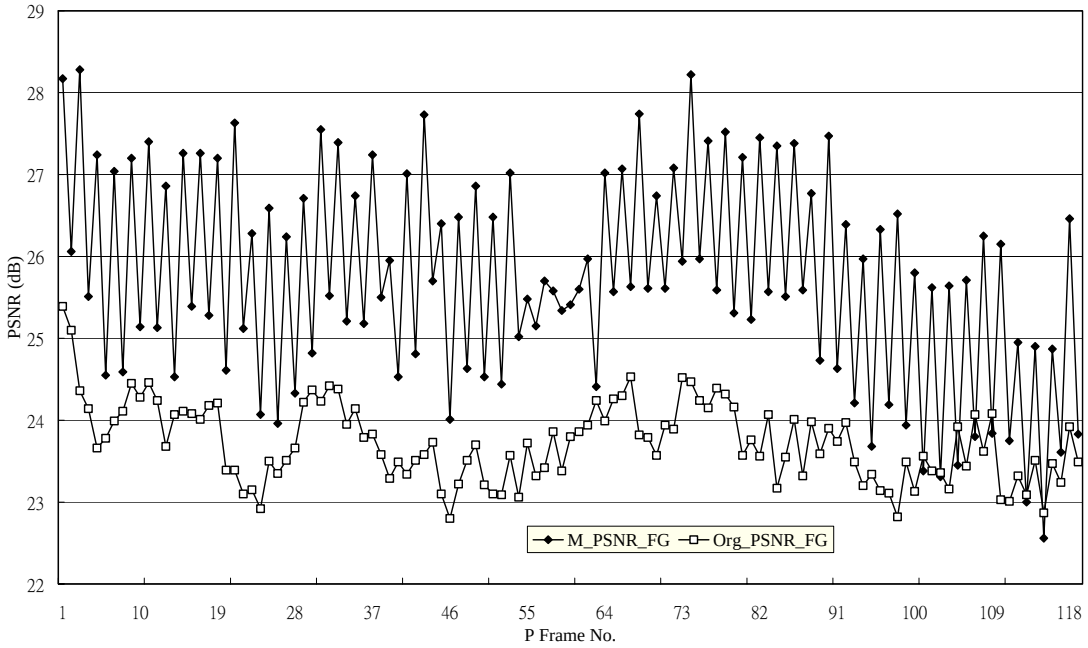
(a) Segmented Result

(b) Original version

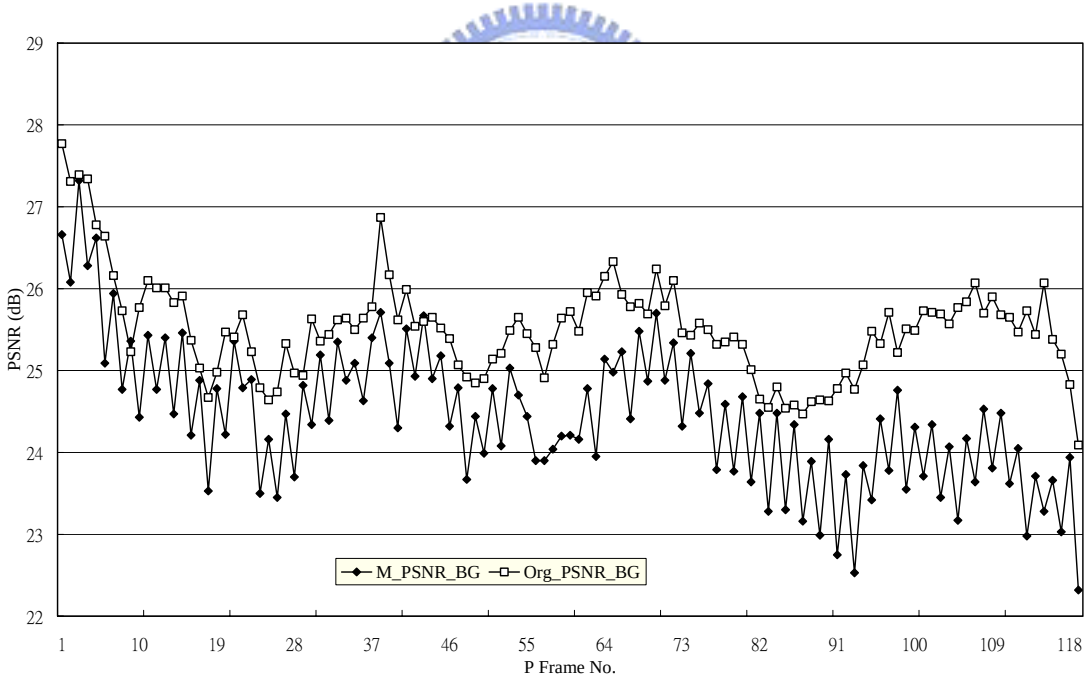
(c) Modified version

Fig. 4-4 Results of Hall sequence encoded by (a) segmented result (b) original version and (c) modified version JM.

With the sequence which has high motion and multi objects, such as SIF format source “Football” sequence, we had improve the foreground objects’ quality as shown in Fig. 4-5 (a). Since the motion is complex in the “Football” Sequence, the bits allocated to the frame is changed not so stably, so the quality of the foreground region is changed a lot in every frame in our modified method. But it is still better than the original method. From Table 4-1, we can know that we had improved the foreground regions’ quality with 2dB whereas the background quality was only degraded by 1.08 dB. And the bit-rate of our method increases only 0.09% due to the complex foreground of the sequence. The subjective quality is shown in Fig. 4-6.



(a)

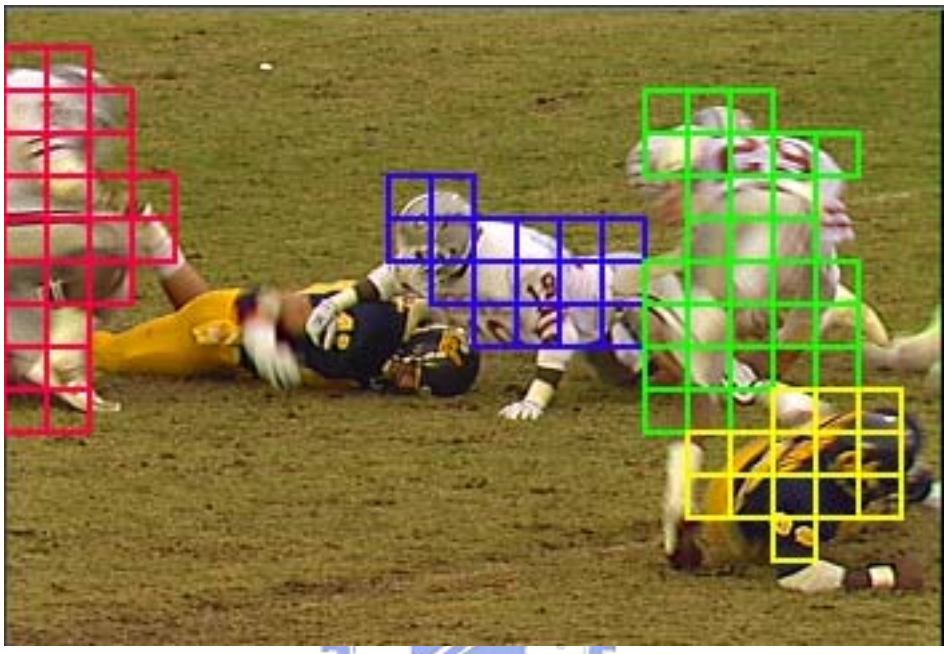


(b)

Fig. 4-5 Football sequence encoded by original JM software and modified version: (a) PSNR of foreground region in original and modified version, (b) PSNR of background region in original and modified version.

In the above experimental results, we know that our method can successfully

improve the quality of foreground moving objects with degradation the quality of the background. In high motion and complex objects scenes, it will cause a certain increasing bitrates to trade the quality of the moving objects. In the sequence that has obvious objects or static scene with small objects, our method is still working well.



(a) Segmented Result

(b) Original version

(c) Modified version

Fig. 4-6 Results of Football sequence encoded by (a) segmented result (b) original version and (c) modified version JM.

Chapter 5

Conclusion and Feature Work

In this thesis, we have presented a motion-based object segmentation algorithm and an object-based rate control scheme. In order to improve the quality of the regions that people are interested in as compared to the background within a limited bit rate, the object segmentation part is first used to segment foreground objects with similar motion activity. When objects have been segmented, the characteristics of these objects, such as size and motion activity, are used to measure the importance of these objects. Then, in the rate control scheme that integrates the feature-based bit allocation we distribute bits to different regions according to its importance.

To improve the performance and the robustness of the system, some enhancements can be worked on:

- Improving the segmentation algorithm so that it is more robust to lighting variation and complex scenes by using other features, such as color information.
- Considering human visual system, the bits which are allocated to the background region can be distributed perceptually by the distance to the foreground region.
- The rate control scheme at the frame level can be improved by considering different complexity of foreground and background.

Video coding with achieving a better foreground quality as compared to the background within a limited transferring rate is an important research topic. We construct the object segmentation and bit allocation scheme for the purpose. For

future enhancement, this scheme can fit to achieve better visual quality for human eye in a limited channel rate.

Reference

[1] "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14486-10 AVC)", in Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG Document, JVT-G050, March 2003.

[2] Iain E. G. Richardson, H.264 and MPEG-4 Video Compression, Wiley, 2003

[3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Transactions on Circuits and System for Video Technology, Vol. 13, Issue 7, pp. 560-676, July 2003.

[4] T. Aach, A. Kaup and R. Mester, "Statistical Model-Based Change Detection in Moving Video", Signal Processing, Vol.31, No. 2, pp.203-217, 1993.

[5] A. Neri, S. Colonnese, G. Russo, and P. Talone, "Automatic moving object and background separation", Signal Processing, Vol.66, pp.219-232, 1998.

[6] D. D. Giusto, F. Massidda, and C. Perra, "A Fast Algorithm for Video Segmentation and Object Tracking", International Conference on Digital Signal Processing, Vol.2, pp.697 – 700, 2002.

[7] Shao-Yi Chien, Shyh-Yih Ma, and Liang-Gee Chen, "Efficient Moving Object Segmentation Algorithm Using Background Registration Technique", IEEE Transactions on Circuits and Systems for Video Technology, Vol.12, No. 7, pp. 577 – 586, 2002.

[8] Jinhui Pan, Chia-Wen Lin, Chuang Gu, and Ming-Ting Sun, "A Robust Video Object Segmentation Scheme with Pre-stored Background Information", IEEE International Symposium on Circuits and Systems, Vol.3, pp.803 – 806, 2002.

[9] Yaakov Tsaig and Amir AverBuch, "Automatic Segmentation of Moving Objects in Video Sequence: A Region Labeling Approach", IEEE Transactions on

Circuits and Systems for Video Technology, Vol.12, No. 7, pp.597 – 612, 2002.

[10] J.C Choi, S.-W Lee, and S. –D. Kim, “Spatio-Temporal Video Segmentation Using a Joint Similarity Measure”, IEEE Transactions on Circuits and Systems for Video Technology, Vol.7, No. 2, pp. 279 – 286, 1997.

[11] D. Wang, “Unsupervised Video Segmentation Based on Watersheds and Temporal Tracking”, IEEE Transactions on Circuits and Systems for Video Technology, Vol.8, No. 5, pp. 539 – 546, 1998.

[12] Hieu T. Nguyen, Marcel Worring, and Anuj Dev, “Detection of Moving Objects in Video Using a Robust Similarity Measure”, IEEE Transactions on Image Processing, Vol.9, No. 1, pp.137 – 141, 2000.

[13] M. L. Jamrozik and M.H. Hayes, “A Compressed Domain Video Object Segmentation System”, Proceedings of 2002 International Conference on Image Processing, Vol. 1, pp. 113-116, Sept. 2002.

[14] G. Agarwal, A. Anbu and A. Sinha, “A Fast Algorithm To Find The Region-Of-Interest In The Compressed MPEG Domain”, Proceedings of 2003 International Conference on Multimedia and Expo, Vol. 2, pp. 133-136, July 2003.

[15] A. Anbu, G. Agarwal and G. Srivastava, “A Fast Object Detection Algorithm Using Motion-Based Region-Of-Interest Determination”, 14th International Conference on Digital Signal Processing, Vol. 2, pp. 1105-1108, July 2002.

[16] Hui-Ping Kuo, “Object-based Video Tracking and Abstraction on Surveillance videos”, NCTU CSIE, June 2004.

[17] Yi-Wen Chen, Duan-Yu Chen and Suh-Yin Lee, “Moving Object Tracking for video Surveillance in Compressed Videos”, in The 7th International Conference on Internet and Multimedia Applications and Systems, pp. 695-698, Aug. 2003.

[18] Hng-Ju Lee and Tihao Chiang and Ya-Qin Zhang, “Scalable Rate Control for

MPEG-4 Video", IEEE Transactions on Circuits and System for Video Technology, Vol. 10, Issue 6, pp. 878-894, Sept. 2000

[19] Vetro, A, Huifang Sun and Yao Wang, "MPEG-4 rate control for multiple video objects", IEEE Transactions on Circuits and System for Video Technology, Vol. 9, Issue 1, pp. 186-199, Feb. 1999.

[20] Ribas-Corbera J. and S. Lei, "Rate control in DCT video coding for low-delay communications", IEEE Transactions on Circuits and System for Video Technology, Vol. 9, Issue 1, pp. 172-185, Feb. 1999.

[21] MPEG-2 Test Model 5, Doc. ISO/IEC JTC1/SC29 WG11/93-400, Apr. 1993.

[22] Z. G. Li, X. Lin, C. Zhu and F. Pan, "A Novel Rate Control Scheme for Video Over the Internet", Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing 2002, Vol. 2, pp. 2065-2068, May 2002.

[23] S. W. Ma, W. Gao, Y. Lu and H. Q. Lu, "Proposed draft description of rate control on JVT standard", JVT-F086, 6th meeting, Awaji, Japan, Dec. 2002.

[24] S.W. Ma, W. Gao, P. Gao and Y. Lu, "Rate Control for Joint Video Team (JVT) Standard", JVT-D030, in 4th meeting: Klagenfurt, July 2003.

[25] Z. G. Li, F. Pan, K. P. Lim, G. N. Feng, X. Lin and R. Susanto, "Adaptive basic unit layer rate control for JVT", JVT-G012, in 7th meeting: Pattaya, March 2003.

[26] Z. G. Li, W. Gao, F. Pan, S. W. Ma, K. P. Lim, G. N. Feng, X. Lin, R. Susanto, Y. Lu and H. Q. Lu, "Adaptive rate control with HRD consideration", JVT-H014, in 8th meeting: Geneva, May 2003.

[27] Z. G. Li, F. Pan, K. P. Lim, X. Lin and S. Rahardja, "Adaptive Rate Control For H.264", 2004 IEEE International Conference on Image Processing (ICIP), Vol. 2, pp. 745-748, Oct. 2004.

[28] F. Pan, Z. Li, K. Lim, and G. Feng, "A study of MPEG-4 rate control scheme and its improvements", IEEE Transactions on Circuits and System for Video

Technology, Vol. 13, Issue 5, pp. 440-446, May 2003.

[29] K. Ngan, T. Meier and Z. Chen, “Improved Single-Video-Object Rate Control for MPEG-4”, IEEE Transactions on Circuits and System for Video Technology, Vol. 13, Issue 5, pp. 385-393, May 2003.

[30] M. Jiang, X. Li and N. Ling, “Improved Frame-Layer Rate Control For H.264 Using MAD Ratio”, Proceedings of the 2004 International Symposium on Circuits and Systems, Vol. 3, pp. 813-816, May 2004.

[31] X. Yi and N. Ling, “Rate Control Using Enhanced Frame Complexity Measure For H.264 Video”, 2004 IEEE Workshop on Signal Processing Systems, pp. 263-268, Oct. 2004.

[32] H. Yu, F. Pen and Z. Lin, “A New Bit Estimation Scheme for H.264 Rate Control”, 2004 IEEE International Symposium on Consumer Electronics, pp. 396-399, Sept. 2004.

[33] Chun-Huang Lin and Ja-Ling Wu, “Content-Based Rate Control Scheme for Very Low Bit-Rate Video Coding”, IEEE Transactions on Consumer Electronics, Vol. 43, No. 2, May 1997.

[34] S. Aramvith, H. Kortrakulkij, et al., ”Joint Source-Channel Coding using Simplified Block-Based Segmentation and Content-based Rate-Control for wireless Video Transport”, Proceeding of International Conference on Information Technology: Coding and Computing 2002, Las Vegas, pp. 71-76, April 2002.

[35] W. Lai, X. D. Gu, R. H. Wang, W. Y. Ma and H. J. Zhang, “A Content-based Bit Allocation Model for Video Streaming”, 2004 IEEE International Conference on Multimedia and Expo (ICME), Vol. 2, pp. 1315-1318. June 2004.

[36] Y. Sun, D. Li, I. Ahmad and J. Luo, “A Rate Control Algorithm for Wireless Video Transmission Using Perceptual Tuning”, International Conference on

Information Technology: Coding and Computing (ITCC), Vol. 1, pp.109-114, April 2005.

[37] D. Chai, K. N. Ngan and A. Bouzedoum, “Foreground/Background Bit Allocation For Region-Of-Interest Coding”, Proceedings of 2000 International Conference on Image Processing, Vol.2, pp. 923-926, Sept. 2000.

[38] S. Sengupta, S. K. Gupta and J. M. Hannah, “Perceptually Motivated Bit Allocation for H.264 Encoded Video Sequences”, 2003 International Conference on Image Processing, Vol. 2, pp. 797-800, Sept. 2003.

[39] H.264 reference software JM 9.5, http://iphome.hhi.de/suehring/tml/, 2005

[40] Chi-Tsong Chen. Linear system theory and design. Rinehart and Winston, New York, 1984.