國立交通大學

資訊科學與工程研究所

碩士論文

利用可變動的影像群大小達成基於影像內容決定的可調性編碼分類

Content-based Classification for Scalable Video Coding Using Adaptive GOP

研 究 生：林俊翰

指導教授：蔡文錦 教授

中華民國九十六年六月

利用可變動的影像群大小達成基於影像內容決定的可調性編碼分類

Content-based Classification for Scalable Video Coding Using Adaptive GOP

研究生：林俊翰 Student : Chun-Han Lin

指導教授：蔡文錦 Advisor : Wen-Jiin Tsai

國立交通大學

資訊科學與工程研究所

碩士論文

A Thesis
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

利用可變動的影像群大小達成基於影像內容決定的可調性編碼分類

學生：林俊翰　　　指導教授：蔡文錦

國立交通大學資訊科學與工程研究所

摘 要

視訊編碼的可調性是視訊標準 H.264/AVC 上新納入的一項技術，其包含時間域，空間域，以及視訊品質三種方式，可以滿足許多視訊傳輸上的需求。可調性的特徵是能夠適應多變的傳輸頻寬，以達到在目前許可的條件之下可達到的最佳視訊品質；而三種可調方式的妥善運用可以使得可調性的編碼方式更具有彈性及全面。本篇論文針對一視訊影像的內容作分類，並根據此分類來決定使用何種可調方式進行編碼；適當的分類將會使得可調性的功能完整地發揮，如此一來當在傳輸過程中發生資料丟棄時，仍能因可調性與影像內容的配合而得到良好的影像回復效果。一視訊影像的內容我們將其分為四種類型，第一種是畫面簡單且物體移動慢，第二種是畫面簡單且物體移動快，第三種是畫面複雜且物體移動慢，第四種是畫面複雜且物體移動快。如此的分類是基於人類視覺上，若針對一個影像作縮放，許多細節將會遺失，因此畫面複雜的影像若使用空間域的可調編碼是不適合的；又，當一影像片段包含快速移動的內容時，若使用時間域的可調編碼將不利於遺失影像的回復。實驗結果中顯示，如此的分類使得傳輸且部分遺失的視訊影像能夠擁有較好的回復品質，且不會因為影像內容的變化而造成視訊品質不穩定的現象。

關鍵字：可調性編碼，變動大小的影像群，影像類別，基於影像內容

Content-based Classification for Scalable Video Coding Using Adaptive GOP

Student: Chun-Han Lin   Advisor: Dr. Wen-Jiin Tsai

Department of Computer Science
National Chiao Tung University

Abstract

Scalable video coding (SVC) is the extension of H.264/AVC standard. In SVC, it provides three types of scalability, including temporal scalability, spatial scalability, and quality scalability, with which great flexibility could be provided to fit different needs of transmission. Scalability is a great feature for transmitting compressed video data adaptively in variant bandwidth. However, manipulating the three types of scalability in different ways to encode the video sequence can yield a wide range of quality and coding efficiency. This paper exploits the characteristics of different scalabilities on the encoding of video sequence with varying features, in order to achieve satisfactory video quality at the desirable bit-rate. This work focuses on how to dynamically decide GOP size and the coding type adaptively according to the features of video content. With adaptive GOP, the most suitable scalable coding type can be applied for each GOP, and the quality under low bandwidth can be satisfying. We classify the content into four types, including simple frame with slow motion (SS), simple frame with high motion (SH), complex frame with slow motion (CS), and complex frame with high motion (CH). The idea is based upon the human perception that simple frame can present better visual quality than complex frame after scaling from low resolution to high one, and thus make it a good choice for coding in spatial scalability; on the other hand, it is easier to reconstruct a lost frame with acceptable quality if the lost frame is in a slow motion sequence than in a high motion one, and thus make it a good choice for using temporal scalability. Under good consideration of the content of the video sequence, a better quality can be obtained in low bandwidth compared to standard scalable coding way. In high bandwidth, the algorithm still can adaptively choose the most suitable scalable coding technique for the video sequence. Therefore it works well in variant bandwidth.

Keywords: scalability, adaptive GOP, content-based, type

Contents

List of Figures

List of Tables

Chapter 1 Introduction

Scalable video coding (SVC) has been a research topic in recent years because of that the requirement for scalable video coding techniques is evident for many application scenarios. In January 2005, MPEG and the Video Coding Experts Group (VCEG) of the ITU-T agreed to jointly finalize the SVC project as an Amendment of their H.264 / MPEG-4 AVC standard, and the scalable coding scheme developed by the image communication group of the HHI was selected as the first Working Draft (WD-1). At the moment that I've started my work, I've used the released document [9, 10] from Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, as well as the referenced software, Joint Scalable Video Model JSVM-5.

The main goal of SVC is to develop a video codec which offers scalability at a bit-stream level with the following features: The extraction of sub-streams with a reduced spatio-temporal resolution and/or a reduced bit-rate shall be possible by simple operations as packet discarding or packet truncation, and the coding efficiency of any possible sub-stream should be comparable to that of a non-scalable codec (H.264 / MPEG-4 AVC). In order to achieve this goal, three types of scalable coding techniques, including spatial scalability, temporal scalability, and quality scalability, have been added to the original structure of H.264 / MPEG-4 AVC, although there are already some concepts of scalability existed in H.264 / MPEG-4 AVC like general temporal scalability. The basic coding scheme for achieving a wide range of spatio-temporal and quality scalability can be classified as layered video codec. The coding structure depends on the scalability space that is required by the application, in other words, we can combine different types of scalability to code an input video, and a great flexibility can be provided.

However, even with the flexibility to use different combinations of scalability to code an input video, we can’t get the idea of how to manipulate them well. In my opinion, we think that applying scalability among a input video without consideration of one’s content would not yield an optimal visual quality after decoding. For example, under the spatial scalability, enhancement layer for higher resolution may be cut off in low bandwidth. In this situation, it may be necessary to scale the frame up to a higher resolution from base layer. According to the human perception, a complex image would lose its details once scaling from an image with lower resolution. Thus it’s definitely not suitable to apply spatial scalability on complex frames. Let’s see another case. Under the temporal scalability, major loss of enhancement layer would directly result in the loss of entire frame, so it’s more suitable to apply it on slow motion video sequence. The reason is because that it’s hard to recover a lost frame if one is in high motion. As you can imagine, the prediction of motion for this lost frame is very hard under a high motion and unpredictable situation.

For the case of quality scalability, it can be applied on an input video with temporal or spatial scalability, or both, to achieve fine granular.

There are few researches which study how to manipulate these scalabilities, and the reason maybe that it’s a new part of H.264 / AVC. Most of the researches relating to coding based on video content are object based. They talked about how to position the region of interest

(ROI) and make the ROI as base layer and the other part of image as enhanced layer, so after the transmission in low bit-rate, the ROI still can be received, which is the most important part in one image [6,13].

In this paper, we will first use spatial prediction error and temporal prediction error as measure for coding type initiating, afterward for some indecisive frames the advanced type defining will process according to an algorithm based on rate distortion. The organization of this work is explained as follows. Chapter 2 explains three different types of scalability techniques defined in SVC as well as the correlation between each type of scalability technique and video content. Chapter 3 gives an overall picture of this work through a block diagram and presents the details of content-based classification. Chapter 4 presents the implementation of this work and the experimental result. Chapter 5 concludes this work.

Chapter 2 Scalability Dimensions

There are two different ways of introducing scalability in a codec, either by using a technique that is intrinsically scalable (such as bitplane arithmetic coding) or by using a layered approach (same concept as the one that is used in many previous standards). In order to illustrate the whole procedure of how SVC works, a joint scalable video model, JSVM-5, is provided for reference. In the JSVM, a combination of the two approaches to enable a full spatio-temporal and quality scalable codec is used. Temporal scalability is enabled by hierarchical B pictures, whereas spatial scalability is provided using a layered approach. For quality (SNR) scalability, two different possibilities are provided; an embedded quantization approach for coarse grain scalability and fine grain scalability (FGS) approach based on the principle of sub-bitplane arithmetic coding. The codec used in JSVM is as shown in Fig. 1 [9]. In the Fig. 1, three levels of spatial scalability are provided through spatial decimation or different resolution of same video content, besides that, in each spatial layer, temporal scalability and quality scalability can be applied. Inter-layer prediction is possible.

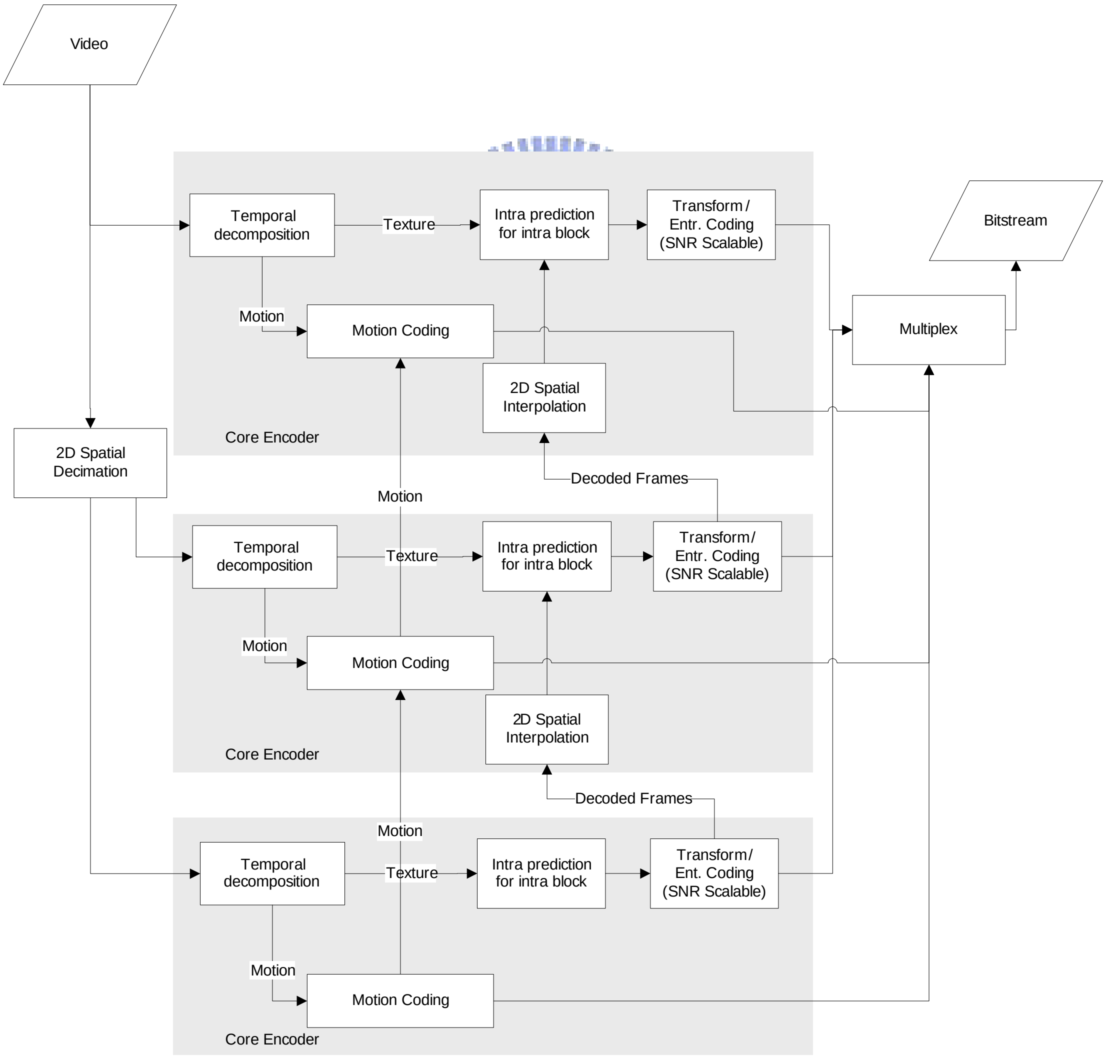


Fig. 1 Scalable encoder using a multi-scale pyramid with 3 levels of spatial scalability [9]

2.1 Temporal Scalability

In AVC, any picture can be marked as reference picture and used for motion-compensated prediction of following pictures independent of the coding types of the corresponding slices. The behavior of the decoded picture buffer (DPB), which can hold up to 16 pictures, can be adaptively controlled by memory management control operation (MMCO) commands, and the reference pictures of the DPB that are used for motion-compensated prediction of another picture can be arbitrarily selected via reference picture list re-ordering (RPLR) commands. These features of AVC allow the selection of arbitrary coding/prediction structures, which are not possible with previous video coding standards. Temporal scalability exploits these features inherent in AVC to achieve the scalable effect on temporal level. Two kinds of methods can be used to achieve temporal scalability, including hierarchical B pictures and motion compensated temporal filtering (MCTF) [3, 4, 5]. The concept of hierarchical B pictures provides a fully predictive structure that is already provided with AVC. Motion compensated temporal filtering (MCTF) can be used as an alternative (and non-normative) encoder configuration. In this case, the encoder structure deviates from the AVC compliant temporal prediction scheme. MCTF can be seen as an encoder-side pre-filtering technique which may provide improved rate-distortion performance on certain input sequences. While in this paper, hierarchical B pictures are used.

2.1.1 Structure of Temporal Scalability

A typical hierarchical prediction structure with 4 dyadic hierarchy stages is depicted in Fig. 2. This structure provides 4 temporal scalability levels. The first picture of a video sequence is intra-coded as IDR picture; so-called key pictures (black in Fig. 2) are coded in regular (or even irregular) intervals. At this, a picture is called a key picture when all previously coded pictures precede this picture in display order. A key picture and all pictures that are temporally located between the key picture and the previous key picture (the IDR picture at the beginning of a video sequence is also a key picture) are considered to build a group of pictures (GOP). The key pictures are either intra-coded (e.g. in order to enable random access) or inter-coded using previous (key) pictures as reference for motion compensated prediction. The remaining pictures of a GOP are hierarchically predicted as illustrated in Fig. 2.

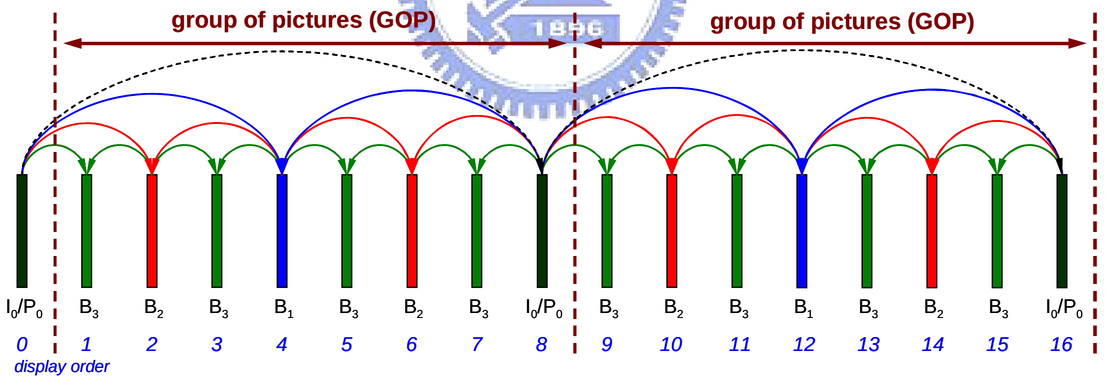


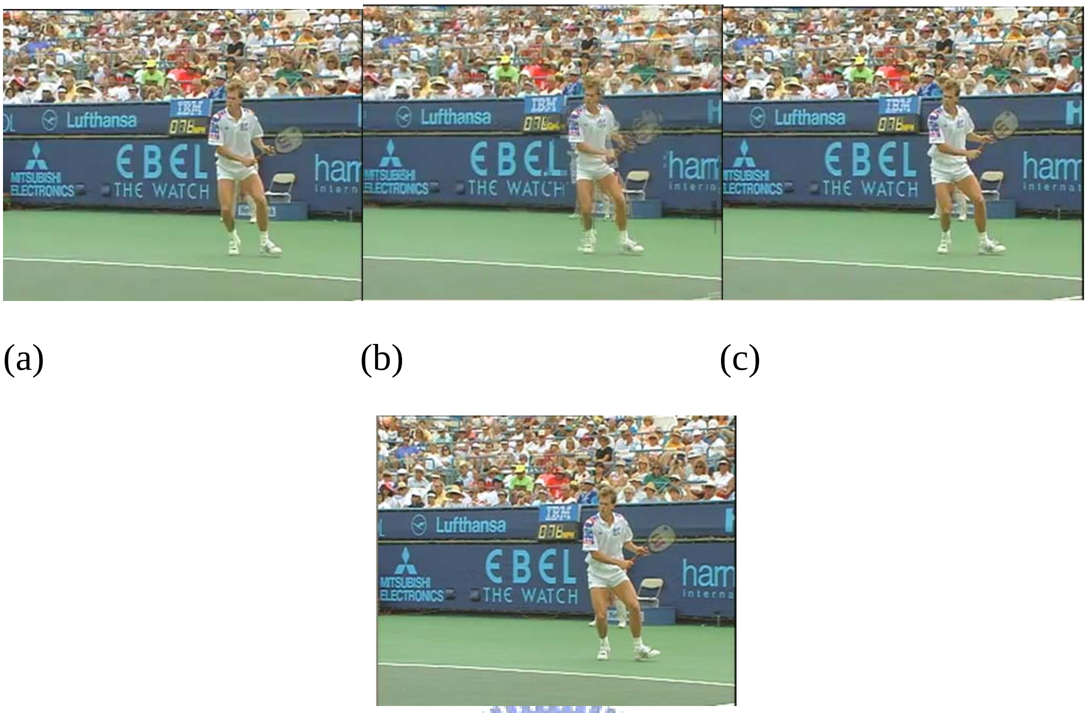
Fig. 2 Dyadic hierarchical coding structure with 4 temporal levels and a GOP size of 8. Each B pictures is predicted using 2 reference pictures, which are the nearest pictures of the lower temporal level from the past and the future [9]

In the hierarchical coding structure, the key picture provides the lowest resolution of temporal scalability, in this case the bandwidth is very restricted and the receiver can only watch the key pictures. There is one important prediction rule of temporal scalability – for the current temporal level of the hierarchical structure; only the pictures with lower temporal level can be used as the reference pictures. It's because the higher temporal level of the hierarchical coding structure will be dropped of truncated first if the bandwidth is restricted. For example, in the first GOP of Fig. 2, the lowest temporal level is composed of picture 0 and 8, and the highest temporal level is composed of picture 1, 3, 5, and 7. For picture 4, it can only use picture 0 and 8 as reference pictures.

2.1.2 Video content v.s. Temporal Scalability

The problem of temporal scalability would happen when some pictures are dropped during transmission because of the limited bandwidth. In this situation, the receiver may just play the incomplete video or try to use some techniques of frame loss recovery to rebuild the whole video. If we want to recover the whole video well, the better result would happen when this video sequence is in slow motion not high motion because of the nature of temporal scalability. There are some frame loss recovering method are discussed [2, 12, 15], but it's really difficult to recover a picture in high motion well especially when the number of adjacent lost pictures is more than one, which happens commonly in temporal scalability.

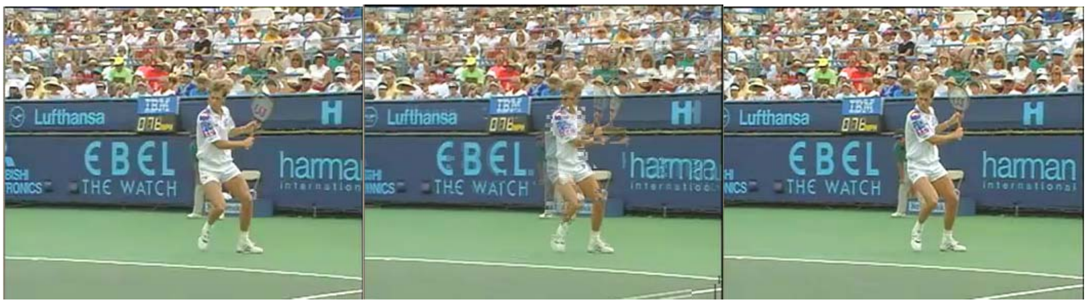
Therefore, a post-checking of suitable coding modes for a video is needed. For a part of video with slow motion, we can say that it's appropriate to code this part as temporal scalability, and thus result in better recovering while any loss of frames exist, as shown in Fig. 3, 4.



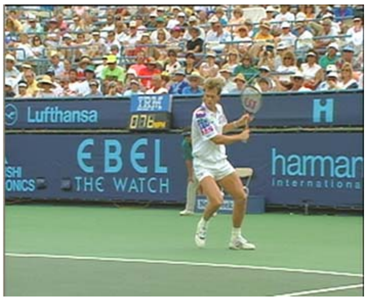
(a) (b) (c)

(d)

Fig. 3 illustration of the relation between slow motion and TSC (a) Frame(i-1) (b) recovered Frame(i) (c) Frame(i+1) (d) Frame(i) From (b) and (d) we can see that slow motion makes the result of recovery better.



(a) (b) (c)



(d)

Fig. 4 illustration of the relation between high motion and TSC (a) Frame(i-1) (b) recovered Frame(i) (c) Frame(i+1) (d) Frame(i). From (b) and (d) we can see that high motion results in worse recovering quality

In Fig. 3 and 4 respectively, three successive frames, frame(i-1), frame(i), frame(i+1), as well as one recovered frame(i) are shown under the situation that frame(i) is lost. Fig. 3 is in slow motion while Fig. 4 is in high motion.

2.2 Spatial Scalability

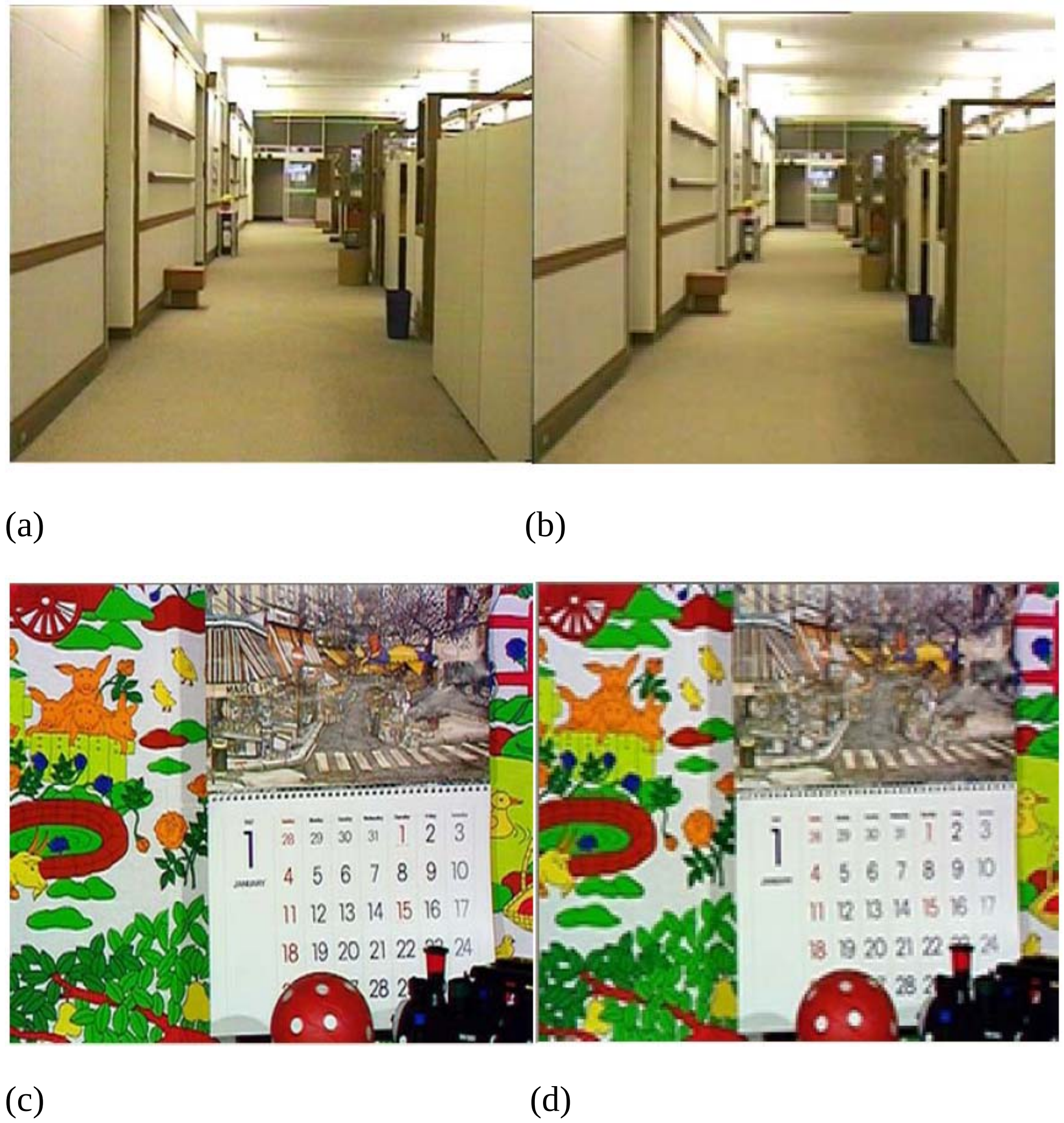
In SVC, it also provides the scalability on 2D spatial domain. The basic purpose of this scalability is that if the current bandwidth is restricted and insufficient for complete transmission of encoded stream, it may truncate the part of enhancement layer and thus result in a decoded video with lower resolution like qcif format; however, if the current bandwidth is sufficient, it may have the chance to transmit a truncated stream with higher resolution like cif format. The application of spatial scalability can be the case when the bandwidth is restricted, the receiver can still receive the whole video with lower resolution and scale it to higher resolution to watch.

2.2.1 Structure of Spatial Scalability

Spatial scalability is a layered approach that provides the scalability on spatial domain, which is the resolution of input video. The coding of spatial scalability is also straightforward. Use the lowest resolution video as the base layer, and code the difference between lower resolution video and higher resolution ones as enhancement layer to provide higher video resolution if allowed. The problem of calculating the difference on different resolution videos can be solved by interpolation on lower resolution one.

2.2.2 Video Content v.s. Spatial Scalability

The problem is obvious to see that when we scale the low-resolution video to higher resolution, how is the blurry degree of the video content? From the point of human perception, a image with many details or we can say, complex content, would result in very bad visual and measurable quality after scaling from low resolution. However, a image with smooth change or we can say, simple content, could yield much better visual and measurable quality after scaling from low resolution video. Fig. 5 illustrates this situation.



(a) (b)

(c) (d)

Fig. 5 illustration of the relation between texture and SSC (a) original 352 x 288 image (hallway.yuv) (b) scaled image from 176 x 144 (c) original 352 x 288 image (mobile.yuv) (c) scaled image from 176 x 144.

2.3 Quality Scalability

Quality scalability provides the scalability on video quality basically through the concept of quantization. Coarse grain scalability is a layered scalable coding technique, for the reason that it is composed of more than one coding layer and the basic one is called the base layer, while the others are called enhanced layers. The base layer forms the limit of available bit-rate, and the enhanced layers provides more details of video and thus presents the enhancement of video quality. To achieve coarse grain scalability one can apply different quantization factors in descending value to different layers from base layer to highest enhanced layer successively. For the details of FGS the reader can refer to [11, 14]. Fine grain scalability is much similar like the coarse grain one but with the different coding of enhanced layers. Base layer still exists for the basic bit-rate requirement, but enhanced layers are coded in bit-plane method, which provides well adaptability to bandwidth and thus achieves smooth scalability for video transmission, as shown in Fig. 6 [14]. In this paper, we choose to use fine granularity scalability as the scalability of quality simply because of the better adaptability than coarse granularity one.

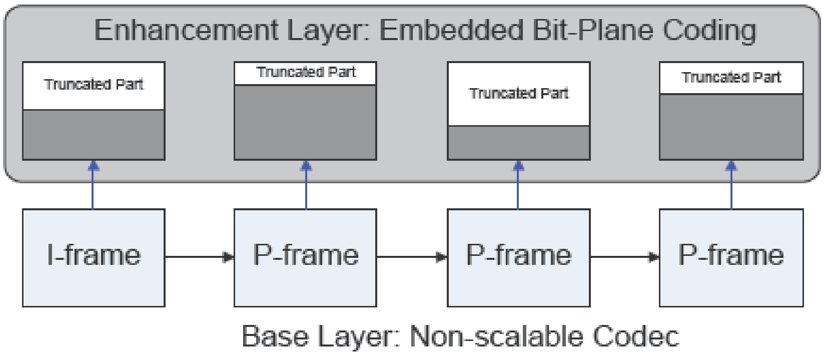


Fig. 6 Basic structure of fine grain scalability [14]

2.4 Extraction of Encoded Stream with Scalability

The encoded stream can be represented in layers. It’s usually the case that the operation of extraction begins from the lowest layer, which represents the limit of bandwidth for the current stream, but for the combination of temporal, spatial, and quality scalability, the extraction is different and we can’t only consider the order of layers. We use the application, extractor, provided in JSVM-5 to simulate the operation of extraction, and the following sub sections illustrate the ways of extraction for different types of scalability. All of the following sub sections use the case of encoding for 5 pictures, and the spatial layers use two kinds of formats, qcif(176x144) and cif(352x288), as the options for multiple resolution.

There are some parameters which need to be explained:

D: dependency level, or spatial level

T: temporal level

Q: quality level

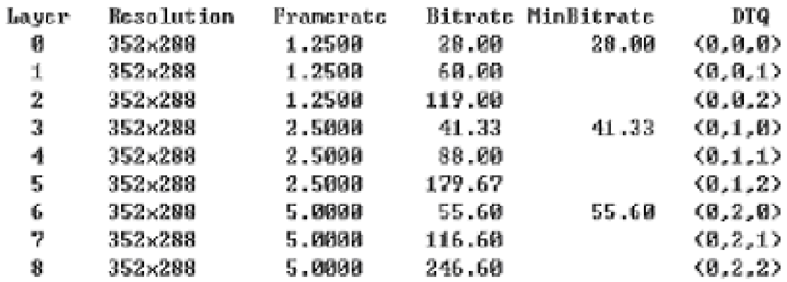
D represents the level of video resolution, for example, if a stream is encoded in qcif format for base layer and cif format for enhancement layer, then the value of D for base layer would be 0 while for enhancement layer it would be 1.

T represents the level of temporal structure. Take Fig. 2 in section 2.1 for example, frame 0 and 8 belongs to temporal level 0; frame 4 belongs to temporal level 1; frame 2 and 6 belongs to temporal level 2; frame 1, 3, 5, and 7 belongs to temporal level 3.

Q represents the quality level. Q=0 represents base layer, Q=1 represents enhancement layer1, Q=2 represents enhancement layer2, etc.

2.4.1 Temporal and Quality Scalability

This sub section illustrates how the extraction software works for the stream with the combination of temporal and quality scalability. Fig. 7 shows the layers and the corresponding bit-rates of the encoded stream. For a given bit-rate, the extraction software will discard stream from high to low layers until hitting the allowed bit-rate.



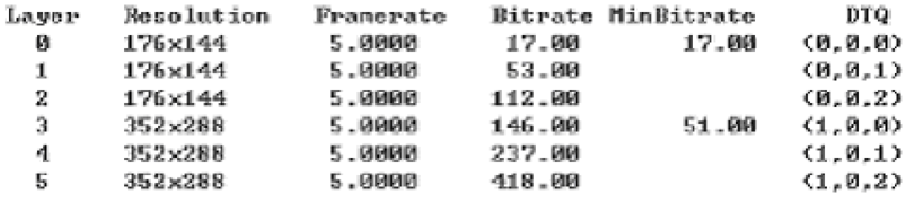
```
Layer   Resolution   Framerate   Bitrate MinBitrate    DTQ
  0     352x288       1.2500      28.00     28.00   (0,0,0)
  1     352x288       1.2500      60.00             (0,0,1)
  2     352x288       1.2500     119.00             (0,0,2)
  3     352x288       2.5000      41.33     41.33   (0,1,0)
  4     352x288       2.5000      88.00             (0,1,1)
  5     352x288       2.5000     179.67             (0,1,2)
  6     352x288       5.0000      55.60     55.60   (0,2,0)
  7     352x288       5.0000     116.60             (0,2,1)
  8     352x288       5.0000     246.60             (0,2,2)
```

Fig. 7 The encoded stream in layers for temporal and quality scalability

As the figure shows, there are 9 layers with accumulated bit-rate for each layer ranging from 28.0 kbps to 246 kbps.

2.4.2 Spatial and Quality Scalability

This sub section illustrates how the extraction software works for the stream with the combination of spatial and quality scalability. Fig. 8 shows the layers and the corresponding bit-rates of the encoded stream. For a given bit-rate, the extraction software will discard stream from high to low layers until hitting the allowed bit-rate.



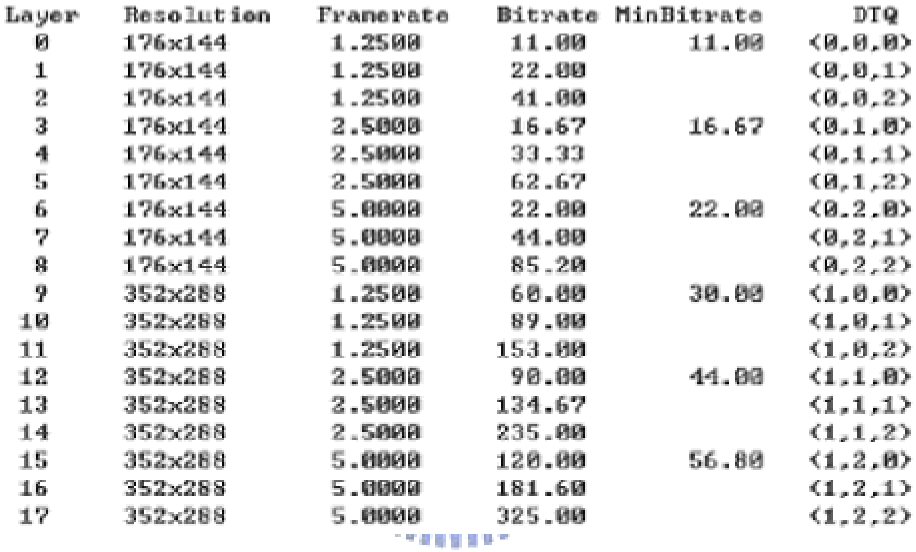
```
Layer   Resolution   Framerate   Bitrate MinBitrate    DTQ
  0     176x144       5.0000      17.00     17.00   (0,0,0)
  1     176x144       5.0000      53.00             (0,0,1)
  2     176x144       5.0000     112.00             (0,0,2)
  3     352x288       5.0000     146.00     51.00   (1,0,0)
  4     352x288       5.0000     237.00             (1,0,1)
  5     352x288       5.0000     418.00             (1,0,2)
```

Fig. 8 The encoded stream in layers for spatial and quality scalability

As the figure shows, there are 6 layers with accumulated bit-rate for each layer ranging from 17.0 kbps to 418 kbps.

2.4.3 Temporal, Spatial and Quality Scalability

This sub section illustrates how the extraction software works for the stream with the combination of temporal, spatial and quality scalability. Fig. 9 shows the layers and the corresponding bit-rates of the encoded stream. The extraction is different for this combination from others. If the given bit-rate is not greater than the one which can allow full extraction of low-resolution data, then the operation of extraction is as usual; otherwise, the extraction will process based on the same extracting way of section 2.4.1 and regard only resolution 352x288, cif format. For example, given a bit-rate 85.2 kbps, 5 pictures would be decoded with resolution as 176 x 144; given a bit-rate 134.67 kbps, only 3 pictures would be decoded with resolution as 352 x 288.



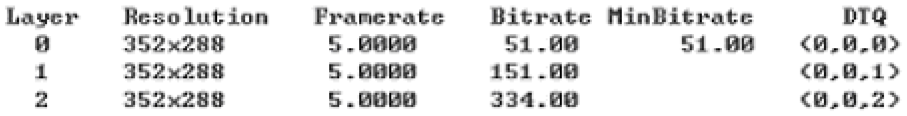
```
Layer   Resolution   Framerate   Bitrate MinBitrate     DTQ
  0     176x144       1.2500      11.00      11.00   <0,0,0>
  1     176x144       1.2500      22.00              <0,0,1>
  2     176x144       1.2500      41.00              <0,0,2>
  3     176x144       2.5000      16.67      16.67   <0,1,0>
  4     176x144       2.5000      33.33              <0,1,1>
  5     176x144       2.5000      62.67              <0,1,2>
  6     176x144       5.0000      22.00      22.00   <0,2,0>
  7     176x144       5.0000      44.00              <0,2,1>
  8     176x144       5.0000      85.20              <0,2,2>
  9     352x288       1.2500      60.00      30.00   <1,0,0>
 10     352x288       1.2500      89.00              <1,0,1>
 11     352x288       1.2500     153.00              <1,0,2>
 12     352x288       2.5000      90.00      44.00   <1,1,0>
 13     352x288       2.5000     134.67              <1,1,1>
 14     352x288       2.5000     235.00              <1,1,2>
 15     352x288       5.0000     120.00      56.00   <1,2,0>
 16     352x288       5.0000     181.60              <1,2,1>
 17     352x288       5.0000     325.00              <1,2,2>
```

Fig. 9 The encoded stream in layers for temporal, spatial and quality scalability

As the figure shows, there are 18 layers with accumulated bit-rate for each layer ranging from 11.0 kbps to 325 kbps.

2.4.4 Quality Scalability

This sub section illustrates how the extraction software works for the stream with quality scalability. Fig. 10 shows the layers and the corresponding bit-rates of the encoded stream. For a given bit-rate, the extraction software will discard stream from high to low layers until hitting the allowed bit-rate.



```
Layer    Resolution    Framerate    Bitrate  MinBitrate     DTQ
  0       352x288       5.0000       51.00      51.00    (0,0,0)
  1       352x288       5.0000      151.00               (0,0,1)
  2       352x288       5.0000      334.00               (0,0,2)
```

Fig. 10 The encoded stream in layers for quality scalability

As the figure shows, there are 3 layers with accumulated bit-rate for each layer ranging from 51.0 kbps to 334 kbps.

2.5 Recovery Methods

In this section, the recovery method used in the proposed method is discussed. Scaling up method as shown in Fig. 11 is used for recovering the incomplete stream coded with spatial scalability, while temporal direct as shown in Fig. 12 is used for recovering the incomplete stream coded with temporal scalability.

Scaling up method expands each pixel in a lower resolution picture to four in square and thus forms a higher resolution picture with twice the height and twice the width.

The idea of temporal direct-mode can be found in [1, 8]. This method is one of the MB encoding types and can be used on the recovery of frame loss. It uses the MVs of a certain picture to predict the MVs of the lost frame. The MVs used for prediction will be divided into two directions (list0 and list1) according to the relative picture distance. The following rule gives an explicit explanation.

Let MV(i,j,p) denotes the past MV in frame(i), MB(j); MV(i,j,f) denotes the future MV in frame(i), MB(j); POC(i) denotes the picture order count of frame(i); Rp(i) denotes the past reference picture of frame(i) and Rf(i) denotes the future reference picture of frame(i). Assume frame(i) is lost (discarded), the MVs of MB(j) on it can be obtained as following

$$MV(i,j,p) = MV(Rf(i),j,p)*((POC(i)-POC(Rp(i)))/(POC(Rf(i))-POC(i)))$$

$$MV(i,j,f) = MV(Rf(i),j,p)*((POC(i)-POC(Rf(i)))/(POC(Rf(i))-POC(i)))$$

………………………..Rule.2.5.1

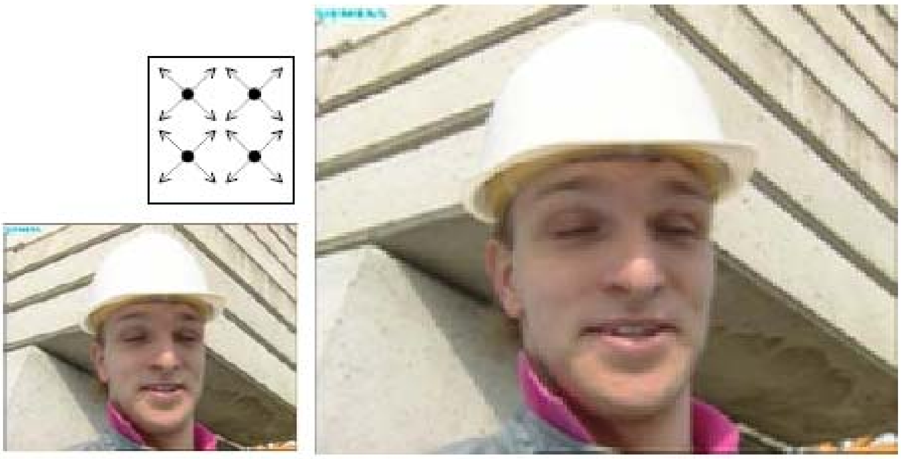


Fig. 11 Scaling up from 176x144 image to 352 x 288 image

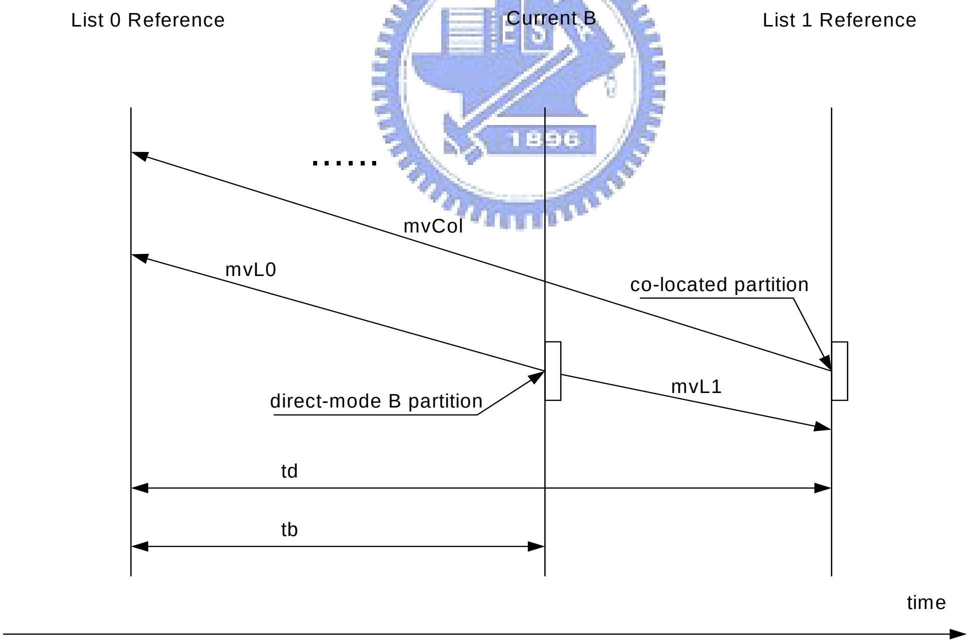


Fig. 12 Example for temporal direct-mode motion vector inference [8]

Chapter 3 Content-based Classification

A video sequence may be composed of variant video contents because of object moving, camera moving, scene change, shot switching, etc, a single scalable coding type won’t be sufficient to apply to the whole video sequence. In this section, we present a method, which exploits video content characteristics to decide scalable coding type for each frame, thus the best received video quality could be achieved in variant bandwidth.

One simple and straightforward way to find the optimal scalable coding combination for a video sequence is to measure the distortion of every possible combination of scalability. In the encoder end, this can be achieved by encoding with different combination of scalable coding types, simulating the extraction with the desired bandwidth condition, decoding finally to compare their quality through PSNR measurement and pick the one with least distortion. It is obvious to see that it spends a lot of time to find the best scalable coding combination although the result could be optimal. On the other hand, using the GOP as the unit to decide and apply the coding type seems to be reasonable because temporal scalable coding is operated based on the GOP. In the current video standard, every GOP in a video sequence can be applied only one scalable coding type or the combination of quality scalability and one of the other two scalabilities. But there’s some problem if fixed-sized GOP is used. Using a fixed-size GOP with large size to decide the scalable coding type won’t result in good effect, because the GOP would consist of variant types of video content. Using a fixed-size GOP with small size would be better in deciding suitable scalable coding type, however, the resulting coding efficiency would be terrible. Thus in this paper, a novel method to quickly decide the scalable coding combination with adaptive GOP is presented.

A block diagram of the proposed method is depicted in Fig. 13. A default sized GOP is used initially for type initiating, and once the current default sized GOP is done, the next GOP with default size will be read in and continue the procedure till the end of the stream. During type initiating, some pictures within the default sized GOP will be assigned certain scalable coding type, while some others won’t be. The group adjusting procedure is then used to adjust the GOP to sub-GOPs on which temporal scalable coding can be applied. Each sub-GOP is sent to sub-GOP encoding procedure to encode, but for those pictures that are not assigned with a scalable coding type in type initiating stage, the distortion measurement procedure is used to find out the most suitable scalable coding type, then the encoding could be done.

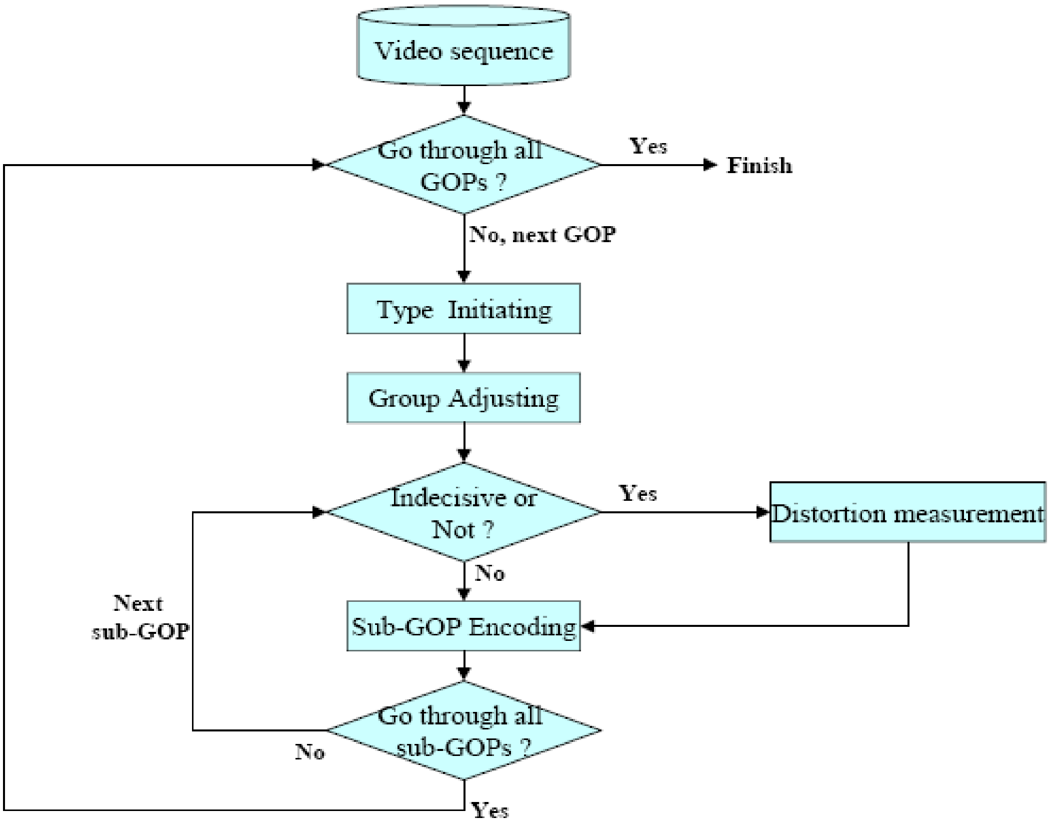


Fig. 13 the block diagram of the proposed method

3.1 Type Initiating

We classify the variant video content into four basic types: simple texture with slow motion (SS), simple texture with high motion (SH), complex texture with slow motion (CS), and complex texture with high motion (CH). For simple texture with slow motion, both spatial scalable coding and temporal scalable coding can be used. For simple texture with high motion, only spatial scalable coding is suitable for the reason that high motion results in worse recovering quality if temporal scalable coding is used as discussed in section 2.1. For complex texture with slow motion, temporal scalable coding is a better choice than spatial scalable coding which may lose many details of the texture as discussed in section 2.2. For complex texture with high motion, both temporal and spatial scalable coding techniques are not appropriate and thus quality scalable coding is applied. Let C(fi) denote the classification for frame i, and S(fi) denote the scalable coding type for frame i, then we will have the following representation:

If C(fi) == SS then S(fi) = SSC or TSC

If C(fi) == SH then S(fi) = SSC

If C(fi) == CS then S(fi) = TSC

If C(fi) == CH then S(fi) = QSC………..Rule.3.1.1

, where SSC represents spatial scalable coding along with FGS, TSC represents temporal scalable coding along with FGS, and QSC represents quality scalable coding which is FGS here.

Given the concept of classification, now the key issue is how to distinguish the kinds of texture and the moving degree of motion.

3.1.1 Simple texture v.s. Complex texture

In the stage of type initiating, when each picture is entering the encoder, the texture of each picture will be examined first. The examination is done by first applying decimation on the input picture, and then calculates the difference between the scaled up version of decimated picture and the original one. The procedure is as shown in Appendix A.

The difference is calculated as

$$Diff_{fi} = \frac{\sum_{h=0}^{H}\sum_{w=0}^{W}(I_{hw} - I'_{hw})^2}{W \times H} \qquad \text{……….Rule.3.1.2}$$

, where W represents the width of frame i, while H presents the height. I_{hw} denotes the original frame i, while I'_{hw} denotes the scaled up version of the decimated frame i. Let γ be the complexity threshold of a frame

$$T(fi) = \begin{cases} \text{C}, & \text{if } Diff_{fi} \geq \gamma_i \\ \text{S}, & \text{otherwise} \end{cases} \qquad \text{……….Rule.3.1.3}$$

, where T(fi) denotes the texture of frame i, C represents complex texture and S represents simple texture. For a frame with Diff greater than or equal to γ, we will classify it as a complex texture, otherwise a simple texture. According to the experiments, the value of threshold γ is set to be 150, which is a suitable boundary for distinguishing two different types of texture.

3.1.2 Slow motion v.s. High motion

After the examination of texture, the degree of motion in one picture will be measured. Motion estimation (ME) and motion compensation (MC) are used for this purpose. The operation is processed by taking each input picture as a B type frame, which means the MC residual of both past and future directions for one picture will be calculated. Let Dp(fi) denote the residual when past frame is used as the reference frame for frame fi, and Df(fi) when future frame is used. There are three conditions that one picture will be taken as a high motion picture(the threshold value, β, is set to 280.368 in the experiment):

1. Dp(fi) $> \beta$ and Df(fi) $> \beta$.
2. Dp(fi) $> \beta$, Df(fi) $< \beta$, and frame fi is a recovered-by-past MV picture.
3. Df(fi) $> \beta$, Dp(fi) $< \beta$, and frame fi is a recovered-by-future MV picture.

The first situation means that if the residuals of both directions are big, this picture is a high motion picture. The recovered-by-past MV in condition 2 means that, when frame fi is lost, its past MV and future MV are both predicted from a past MV of another frame; while recovered-by-future MV in condition 3 means that the past MV and future MV of fi are both predicted from a future MV of another frame. In temporal coding hierarchy, each B frame has two MVs; one (called past MV) uses a reference frame earlier than it, and the other one (called future MV) uses a reference frame after it. Fig. 12 shows a temporal scalable coding hierarchy, where the second picture is a recovered-by-past MV picture.

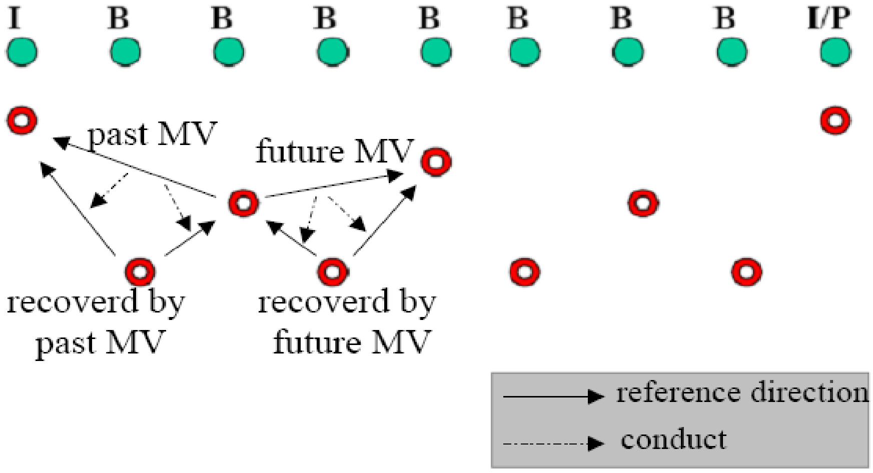


Fig. 14 the hierarchical coding structure marked with some recovery directions with GOP size = 8

Because, once it is lost, its past MV and future MV both come from past MV of the third picture. On the other hand, for the forth picture, it's a case of recovered-by-future MV picture. According to Rule.2.5.1, the MV of lost second picture in Fig. 14 can be obtained as following:

MV(2,j,p) = MV(3,j,p)/2

MV(2,j,f) = -MV(3,j,p)/2

For every picture in a hierarchical coding structure shown in Fig. 15, we use ← to indicate that it is a recovered-by-past MV picture; while → to indicate that it is a recovered-by-future MV picture.

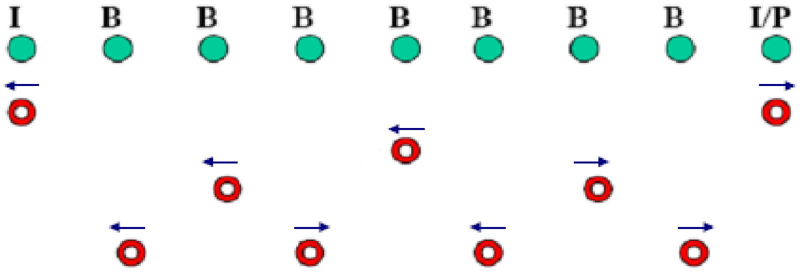
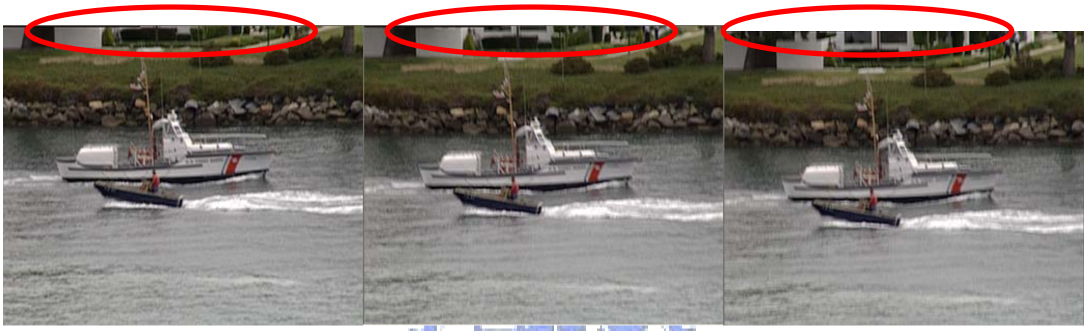


Fig. 15 the recovery direction of each picture in a hierarchical coding structure of GOP=8

The idea behind conditions 2 and 3 is to consider the case that something

1896

appearing to or disappearing from a video sequence. As an example in Fig. 16, there are something appearing on top of (b) and (c), compared to (a). This means that Dp(frame(b)) and Dp(frame(c)) would be large, and the past MV of both frame (b) and (c) won't be reliable (something cannot be found in reference frame (a)). Therefore, using past MV of frame (c) to predict both MVs of frame (b) (if (b) is lost) won't be a good idea.



(a) (b) (c)

Fig. 16 successive frames for illustrating condition of high motion (a) in the same position of picture 1 in Fig. 14 (b) in the same position of picture 2 in Fig. 14 (c) in the same position of picture 3 in Fig. 14

After texture and motion estimation described above, the final step in type initiating stage is to mark pictures with different types according to Rule.3.1.1. if a picture with simple texture as well as slow motion, it will be marked as an indecisive frame, because in this situation both spatial and temporal scalable coding can be applied. The decision of scalable coding type for these pictures will rely on the distortion measurement algorithm as discussed in section 3.4.

3.2 Group Adjusting

During type initiating, a default sized GOP is initially used. For the reason that a GOP may consist of variant types of video content, the frames in the GOP could be assigned with different scalable coding types and form different sub-GOPs. There will be four possible types for sub-GOPs: TSC, SSC, QSC, and indecisive sub-GOPs. For example, after type initiating, a GOP with default size 16 could be assigned like this:

T: TSC S: SSC Q: QSC ?: indecisive

Frame number:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Coding type:	I	S	S	S	S	T	T	T	?	?	?	?	?	?	T	T	Q

; Thus 5 groups would be formed, including frames 0~4, 5~7, 8~13, 14~15, and 16, each of which contains one scalable coding type or indecisive type.

However, the size of these groups may need to be further adjusted because temporal scalable coding type cannot be applied to arbitrary size of sub-GOPs. Only the sub-GOP size equaling to 2^n or $2^n + 1$ is allowed. For example, a GOP with default size equaling to 16, the size of its TSC sub-GOPs can only be one of the cases, 2, 3, 4, 5, 8, 9, 16; where the size 2, 4, 8, 16 will be encoded using closed GOP, while the others will be encoded using open GOP. The indecisive group also needs to be adjusted because they may be assigned as temporal scalable coding type later during the step of distortion measurement. To be general, the following rules are applied for TSC and indecisive groups.

(1) If group size = 2^n, n>0, close sub-GOP is applied and no adjusting

(2) Else if group size = $2^n + 1$ and n>0, open sub-GOP is applied and no adjusting

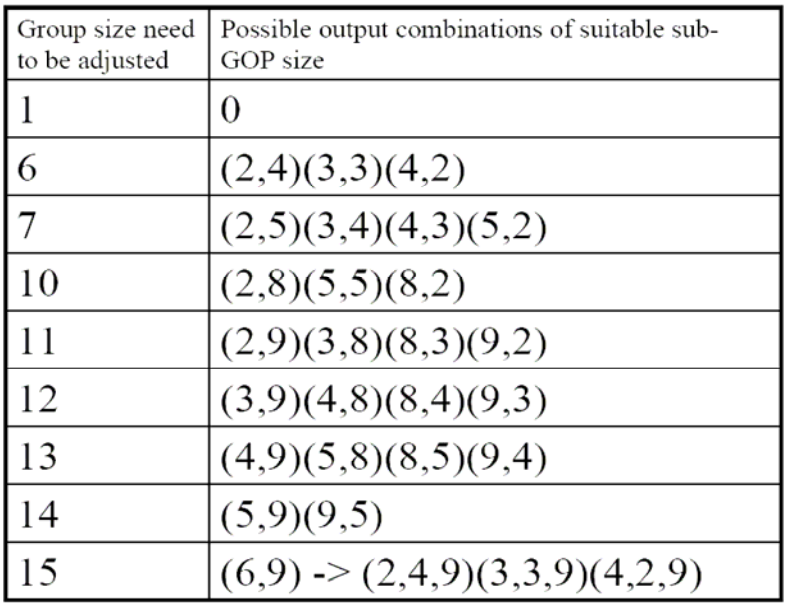
(3) Else, apply the group adjusting

…………………..Rule.3-2-1

For the group whose group size needs to be adjusted, the following rule is applied to further divide it into multiple sub-GOPs with appropriate sizes.

```
If current group contains the first frame of default GOP THEN
  First_grp = 1
END
FOR I = 2 to group_size-2 DO
  IF I and group_size-I both satisfy Rule.3-2-1-(1)or(2) THEN
     IF First_grp AND I satisfies Rule.3-2-1-(2) THEN
          GET new sub-GOP pair (I, group_size-I)
     ELSE IF NOT First_grp THEN
          GET new sub-GOP pair (I, group_size-I)
     END
  END
END
…………………..Rule.3-2-2
```

The following table 1 shows the example of a GOP with default size 16. There are nine cases of group size that need to be adjusted. Assume that all group sizes listed are not located in the first group of default GOP, where there’s no past reference picture for I frame and thus open sub-GOP is not allowed. For the case of group size 6, it could be divided into two sub-GOPs with size 2 and 4, 4 and 2, or 3 and 3; for the case of 15, its division can be obtained by applying the above algorithm two times. The first time divides it into (6, 9), the second time divides the 6 into three pairs and get (2, 4, 9), (4, 2, 9), (3, 3, 9) finally.



Group size need to be adjusted	Possible output combinations of suitable sub-GOP size
1	0
6	(2,4)(3,3)(4,2)
7	(2,5)(3,4)(4,3)(5,2)
10	(2,8)(5,5)(8,2)
11	(2,9)(3,8)(8,3)(9,2)
12	(3,9)(4,8)(8,4)(9,3)
13	(4,9)(5,8)(8,5)(9,4)
14	(5,9)(9,5)
15	(6,9) -> (2,4,9)(3,3,9)(4,2,9)

Table 1 different combinations of size for original group with unsuitable size for temporal scalable coding

Be noted that if there is a single picture of indecisive type or temporal scalable coding type among others with different types, this picture will be changed to use the same coding type as its neighbors'. In other words, it would be taken as an error judgment in the type initiating stage.

3.3 Distortion Measurement Algorithm

Distortion measurement algorithm is applied on the indecisive groups. Recall that a picture will be assigned as indecisive type in the type initiating stage only if it is judged as slow texture and slow motion (SS). That is, it is suitable to be coded using spatial scalable coding (SSC) as well as temporal scalable coding (TSC). The main idea of this distortion measurement algorithm is to estimate the distortions after different scalable coding types are applied at the desirable bit-rate, and then a comparison is made to decide the best coding type for these indecisive pictures. In order to estimate the distortion, the indecisive group must be encoded using SSC and TSC. For each of the encoded stream, extract partial stream according to the desirable bit-rate, decoded it from truncated bit-stream, recover it from incomplete video sequence, and then calculate the PSNR value between original pictures and recovered ones. The flow of this distortion measurement step is as shown in Fig. 17, where linear scaling up and temporal direct are used to recover the incomplete SSC and TSC streams, respectively.

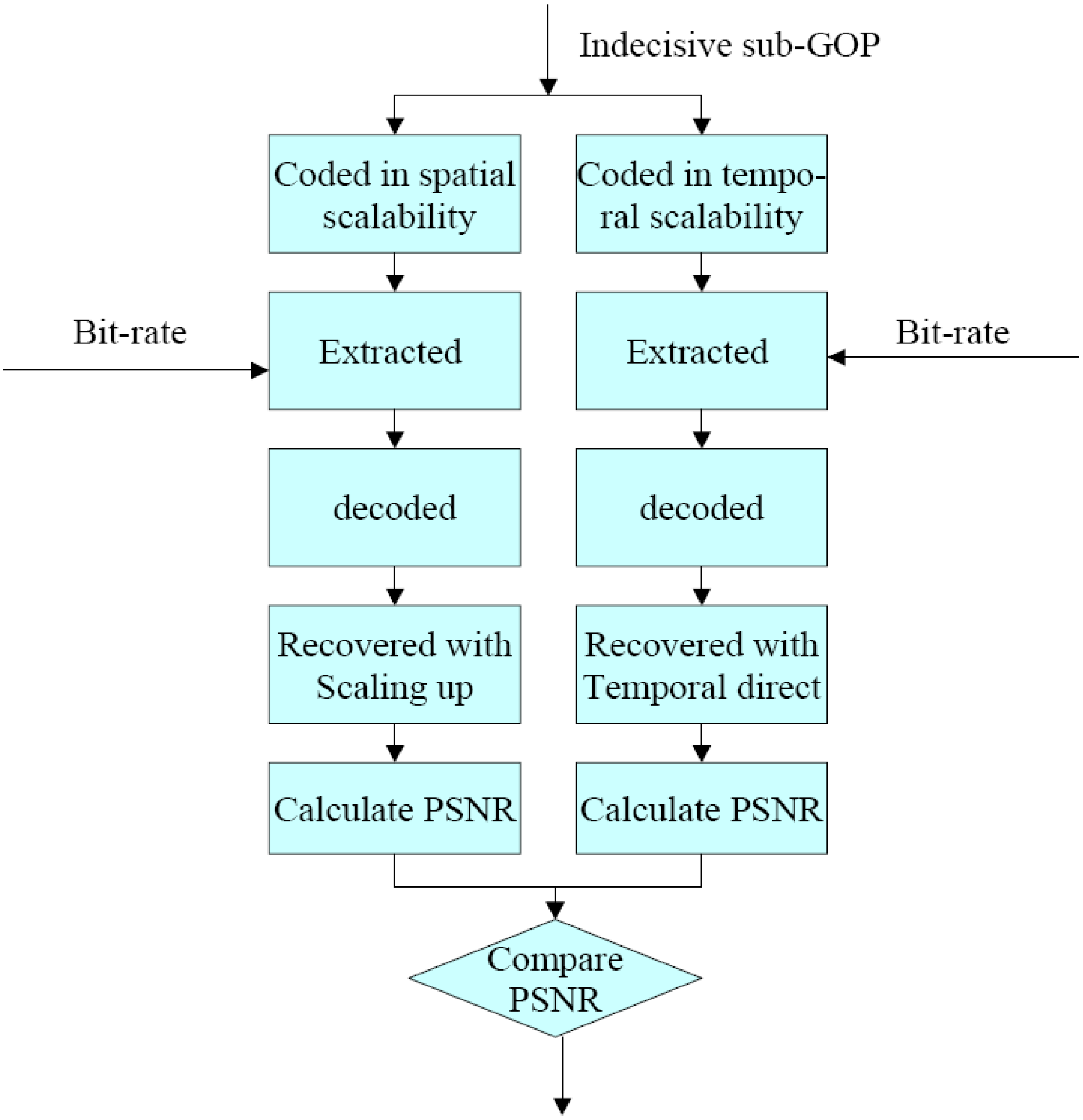

Indecisive sub-GOP
Coded in spatial scalability
Coded in tempo-ral scalability
Bit-rate
Extracted
Extracted
Bit-rate
decoded
decoded
Recovered with Scaling up
Recovered with Temporal direct
Calculate PSNR
Calculate PSNR
Compare PSNR

Fig. 17 the flow of distortion measurement step

Although linear scaling up and temporal direct are used, they can be replaced with any other recovery approaches. Using different techniques of recovery for spatial domain [7] and temporal domain may result in different selection of scalable coding types, the framework proposed in this paper still work to pick the most suitable scalable coding combination in a stream to obtain better visual quality in the receiver end. For a group with n pictures, there are 2^n combinations that these pictures could be encoded (each can be SSC or TSC). The distortion measurement for all the coding combination will suffer from large computation overhead. Fortunately, decisive frames during type initiating stage and lots of inappropriate group size excluded in the group adjusting stage can reduce the workload significantly. Fig. 18 shows the appropriate sub-GOP size in a tree structure, assuming that the original indecisive size is 8.

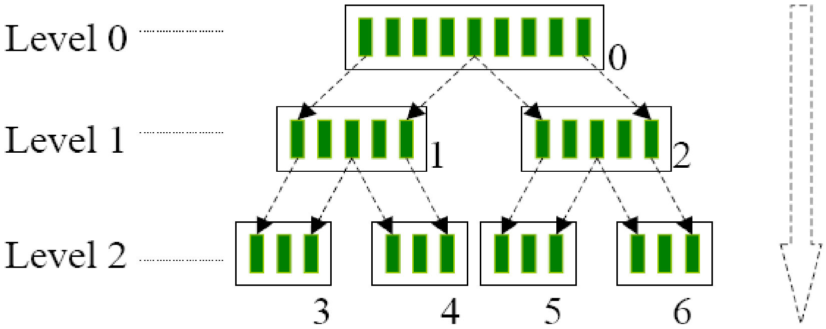


Fig. 18 the tree-like processing structure for a indecisive group with size equaling to 8 plus an extra picture.

In Fig. 18, the root node represents the original indecisive group. The distortion estimation order is from top to bottom, left to right. Let Dssc(Ni) and Dtsc(Ni) denote the distortion estimation after applying SSC and TSC on Ni, node i, respectively. Let Db(Ni) denote the distortion estimation for the coding type which results smaller distortion on Ni, while Ni1 and Ni2 denote the two child nodes derived from Ni. The distortion measurement process will go through the nodes by the estimation order until hitting the two termination conditions as shown below:

1. The conditions that the process terminates on node i with SSC are

(a) Dssc(Ni) < Dtsc(Ni) and
Dssc(Ni1) < Dtsc(Ni1) and
Dssc(Ni2) < Dtsc(Ni2)

(b) Dssc(Ni) < Dtsc(Ni) and
[Dtsc(Ni1) < Dssc(Ni1) or Dtsc(Ni2) < Dssc(Ni2)]
Average(Db(Ni1), Db(Ni2)) > Dssc(Ni)

2. The conditions that the process terminates on node i with TSC are

(a) Dtsc(Ni) < Dssc(Ni) and
Dtsc(Ni1) < Dssc(Ni1) and
Dtsc(Ni2) < Dssc(Ni2) and
Average(Dtsc(Ni1), Dtsc(Ni2)) > Dtsc(Ni)-0.5

(b) Dtsc(Ni) < Dssc(Ni) and
[Dssc(Ni1) < Dtsc(Ni1) or Dssc(Ni2) < Dtsc(Ni2)]
Average(Db(Ni1), Db(Ni2)) > Dtsc(Ni)

The condition 1.(a) above is satisfied when the current node i as well as its two child nodes has smaller distortion estimation in spatial scalable coding (SSC); the reason of terminating on node i under this condition is because that the averaged distortion conducted from child nodes usually similar to the parent node and, therefore, there is no need to further divide it into multiple smaller sub-GOPs.

The condition 1.(b) above is satisfied when the current node i has smaller distortion estimation with SSC, but not all of its child nodes do. The "Average(Db(Ni1), Db(Ni2)) > Dssc(Ni)" means that further division will result in larger distortion estimation than current node I, no matter which coding type is used, so the process stops. The following Fig. 19 gives an example of the above condition:

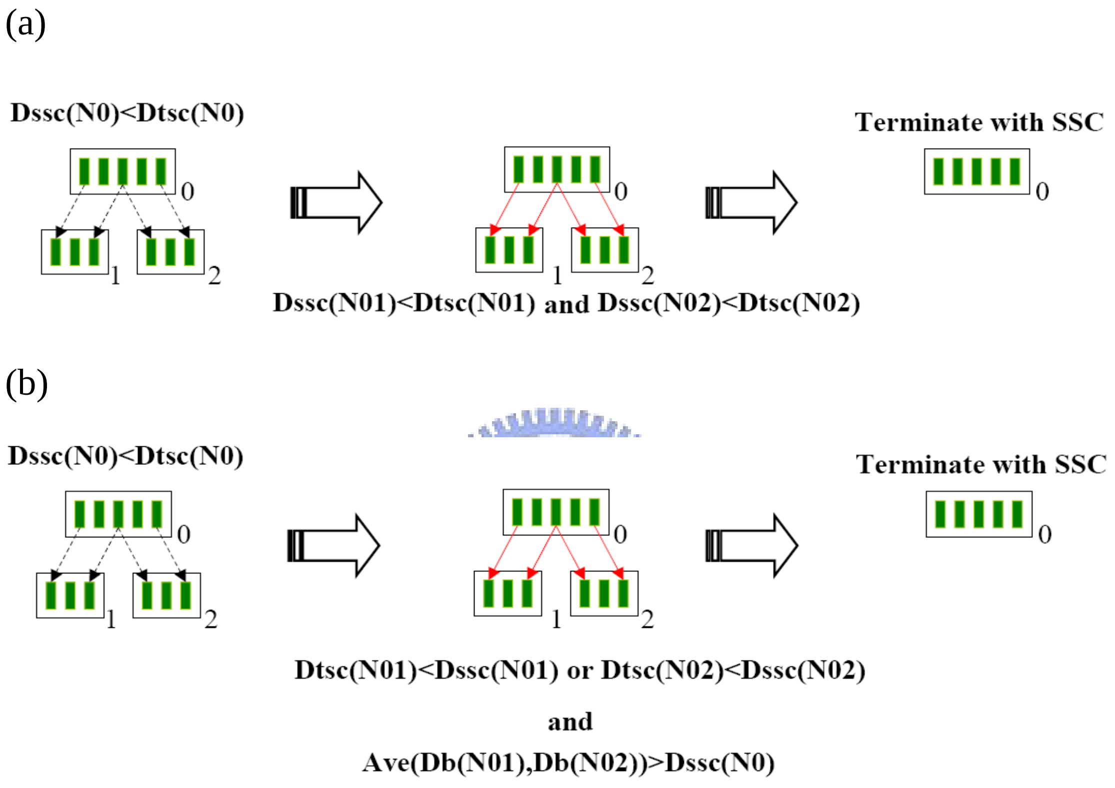

(a)
Dssc(N0)<Dtsc(N0)
0
1
2
0
1
2
Dssc(N01)<Dtsc(N01) and Dssc(N02)<Dtsc(N02)
Terminate with SSC
0
(b)
Dssc(N0)<Dtsc(N0)
0
1
2
0
1
2
Dtsc(N01)<Dssc(N01) or Dtsc(N02)<Dssc(N02)
and
Ave(Db(N01),Db(N02))>Dssc(N0)
Terminate with SSC
0

Fig. 19 The example of termination condition 1 (a) condition 1.a (b) condition 1.b

The condition 2.(a) is similar to condition 1.(a) except that a threshold is set. Because the smaller sub-GOPs a GOP divides in TSC, the smaller distortion estimation it should have. The threshold is set for that if the averaged distortion of further divided sub-GOPs is not less than that of current node to a certain extent, then terminates on current node with TSC.

The condition 2.(b) is also similar to 1.(b). The following Fig. 20 gives an example of the above condition:

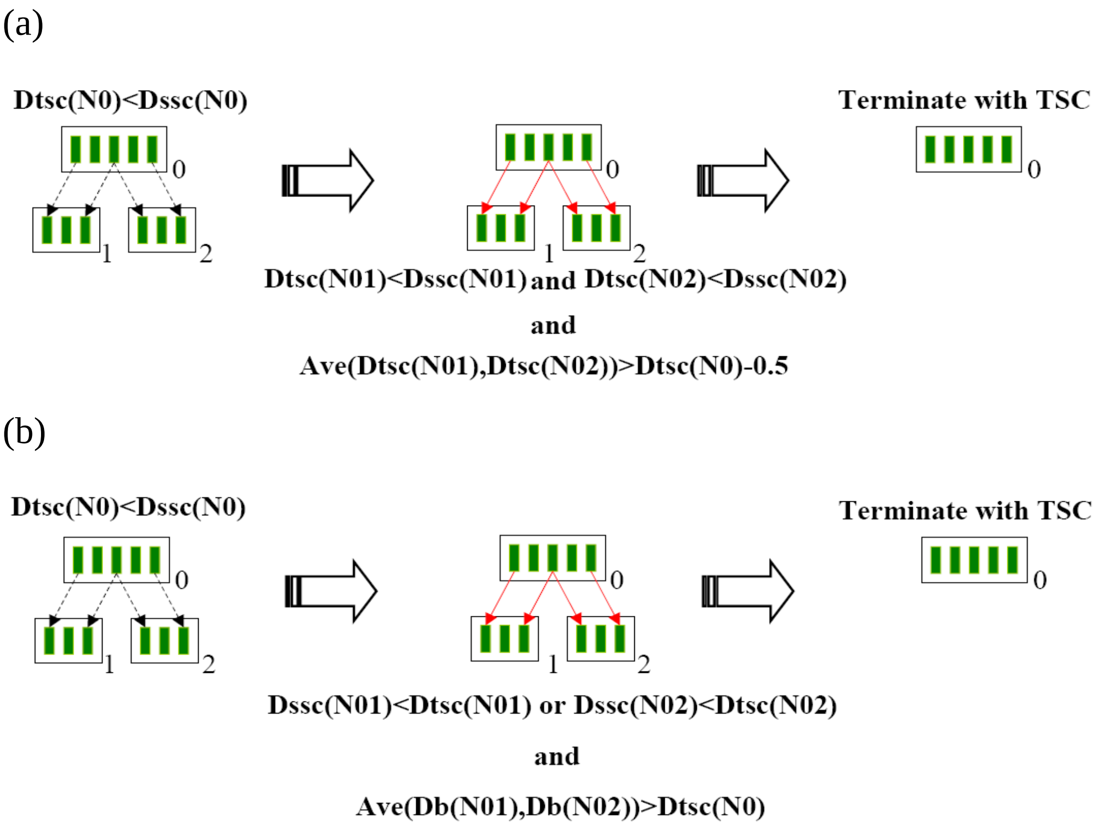

(a)
Dtsc(N0)<Dssc(N0)
0
1
2
0
1
2
Terminate with TSC
0
Dtsc(N01)<Dssc(N01) and Dtsc(N02)<Dssc(N02)
and
Ave(Dtsc(N01),Dtsc(N02))>Dtsc(N0)-0.5
(b)
Dtsc(N0)<Dssc(N0)
0
1
2
0
1
2
Terminate with TSC
0
Dssc(N01)<Dtsc(N01) or Dssc(N02)<Dtsc(N02)
and
Ave(Db(N01),Db(N02))>Dtsc(N0)

Fig. 20 The example of termination condition 2 (a) condition 2.a (b) condition 2.b

Chapter 4 Experiment and Results

In this chapter, we first illustrate the experimental environment, including the platform, software tools, testing video samples and some important parameters used for experiments, then show the experimental results. The experimental result shows that the purposed content-based classification method can result in better video quality than original scalable coding techniques under low bit-rate. For the case of higher bit-rate, the temporal scalability, due to its great coding efficiency, makes the encoded stream can be transmitted without truncation and thus need no applying content-based classification.

4.1 Experimental environment

The experiments are proceeding on window XP platform and the coding environment is Microsoft Visual C++. The tools used are the encoder, extractor, and decoder included in reference software model JSVM-5 and some modifications are made in these tools to begin the experiments.

Some parameters in the configuration file are set common for all experiments. The quantization parameter is set as 33, the quality or fine granularity scalability is layered as 3 including base layer, the spatial scalability is layered as 2 including base layer, the default GOP size is set as 16, and the symbol mode is set as CABAC.

The PSNR is measured by using following equation

$$PSNR = 10 \times \log\left(\frac{255^2}{MSE}\right)$$

$$MSE = \frac{\sum_{n=1}^{FrameSize} (I_n - P_n)^2}{FrameSize}$$

4.2 Experiments

We have done many video samples to test the purposed method. Because of the reason that not many testing samples are composed of variant video content, we’ve chosen some representative video samples to show the experimental results; most of the other samples are classified as pure temporal scalable coding with FGS or pure spatial scalable coding with FGS because of the consistency within one video sample.

The following experiments are performed to make a comparison among three types of scalable coding techniques and the purposed classification method. *Spatial+FGS* represents spatial scalability with FGS, *temporal+FGS* represents temporal scalability with FGS, *mix* represents the combination of temporal, spatial scalability with FGS, and *CBC* represents the purposed method. All but CBC use fixed GOP to encode the video sequence in each experiment, while as we've known that CBC uses adaptive GOP.

4.2.1 Exp. 1

Experiment 1 uses a yuv sample, coastguard, with 162 frames as testing. The spatial scalability is achieved by using two kinds of resolution format, cif(352 x 288) and qcif(176 x 144). The testing bit-rate is set as 618 kbps.

The initial type after the step of content type initiating, the adjusted type after group adjusting, and the decided type after the step of distortion measurement for each GOP is as shown in table 2

	Content type initiating	**Group adjusting**	**Distortion measurement**
GOP1	**I????????????????**	**None**	**I t t t t t t t t t t t t t t t t**
GOP2	**I????????????????**	**None**	**I t t t t t t t t t t t t t t t t**
GOP3	**I ???? t t t t t ? t t t t ? t**	**(I????)(t t t)(t t t t t t t t)**	**(I t t t t)(t t t t)(t t t t t t t t)**
GOP4	**I t ? t t t ??? t t t t t t t t**	**(I t t)(t t t)(???)(t t t t t t t t)**	**(I t t)(t t t)(t t t)(t t t t t t t t)**
GOP5	**I t ???????? t t t t t t t**	**(I????)(?????)(t t t)(t t t t)**	**(I t t)(t t)(t t t)(t t)(t t t)(t t t t)**
GOP6	**I t t t t t t t t t t t t t t t t**	**None**	**None**
GOP7	**I t t t t t t t t t t t t t t t t**	**None**	**None**
GOP8	**I t t t t t t t t t t t t t t t t**	**None**	**None**

GOP9	**I t t t t t t t t t t ??????**	**(I t t)(t t t t t t t t)(???)(???)**	**(I t t)(t t t t t t t t)(ttt)(t t t)**
GOP10	**I?????? t t ?? t t t t t t**	**(I??)(????)(t t)(??)(t t t) (t t t)**	**(I t t)(t t t t)(t t)(t t)(t t t)(t t t)**

Table 2 procedure of type deciding for each GOP

The experimental result is as shown in Fig. 21

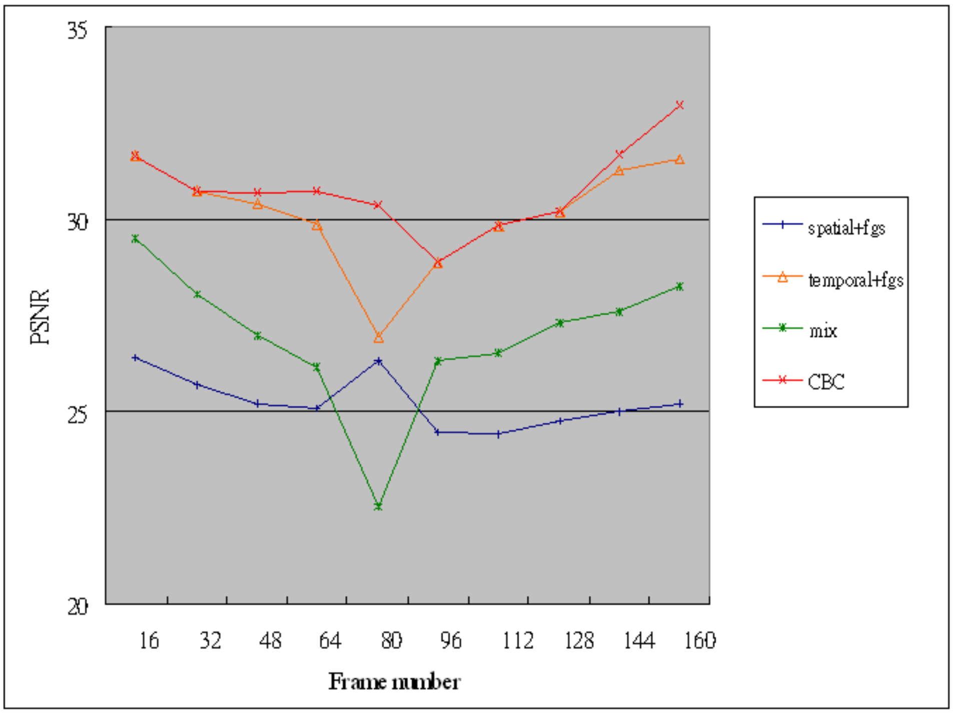


Fig. 21 Experimental result of coastguard

This video sequence is basically a slow motion video but not the case during frame 48~96. The video content of frame 48~96 is about drastically moving the camera upward and therefore, not suitable for using TSC which leads to a obvious decrease on quality. In most of the case, cbsvc results in better quality or equal to the best standard coding for the current GOP.

4.2.2 Exp. 2

Experiment 2 is using a yuv sample, stefan, with 81 frames as testing. The spatial scalability is achieved by using two kinds of resolution format, cif(352 x 288) and qcif(176 x 144). The testing bit-rate is set as 718 kbps.

The scalable type assignment for each step of the proposed method is shown in table 3 below

	Content type initiating	**Group adjusting**	**Distortion measurement**
GOP1	**I t t t t t q q q q q t t t t t t**	**(I t t)(t t t)(q q q q q)(t t t)(t t t)**	**None**
GOP2	**I t t t t t t t t t t t t t t t t**	**None**	**None**
GOP3	**I t t t t t t t t t t t t t t t t**	**None**	**None**
GOP4	**I t t t t t t t t t t t t t t t t**	**None**	**None**
GOP5	**I t t t t t t t t t t t t t t t t**	**None**	**None**

Table 3 procedure of type deciding for each GOP

The experimental result is as shown in Fig. 22

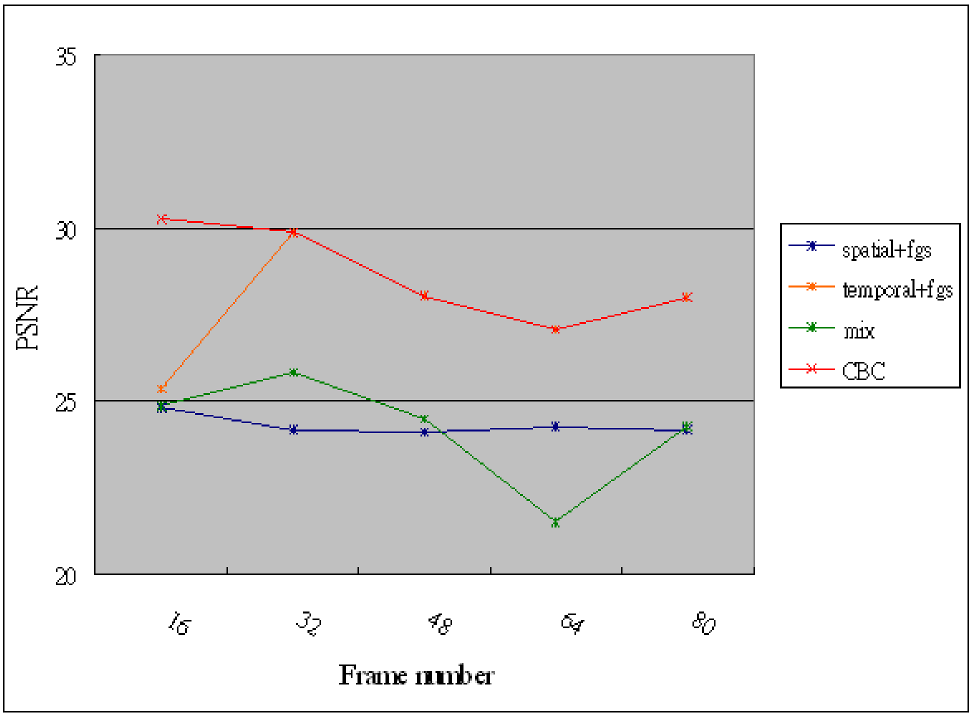


Fig. 22 Experimental result of Stefan

Stefan is a complex video sequence. During frame 0~16, the video content contains high motion and thus, QSC only is used. The resulting quality is far better than that of the others.

4.2.3 Exp. 3

Experiment 3 is using a yuv sample, dancer, with 81 frames as testing. The spatial scalability is achieved by using two kinds of resolution format, cif(352 x 288) and qcif(176 x 144). The testing bit-rate is set as 350 kbps.

The scalable type assignment for each step of the proposed method is shown in table 4 below

	Content type initiating	**Group adjusting**	**Distortion measurement**
GOP1	**I????????????????**	**None**	**I t t t t t t t t t t t t t t t t**
GOP2	**I????????????????**	**None**	**(I t t t t t t t t)(s s s s s s s s)**
GOP3	**I????????????????**	**None**	**I s s s s s s s s s s s s s s s s**
GOP4	**I????????????????**	**None**	**I s s s s s s s s s s s s s s s s**
GOP5	**I????????????????**	**None**	**(I t t t t t t t t)(s s s s s s s s)**

Table 4 procedure of type deciding for each GOP

The experimental result is as shown in Fig. 23

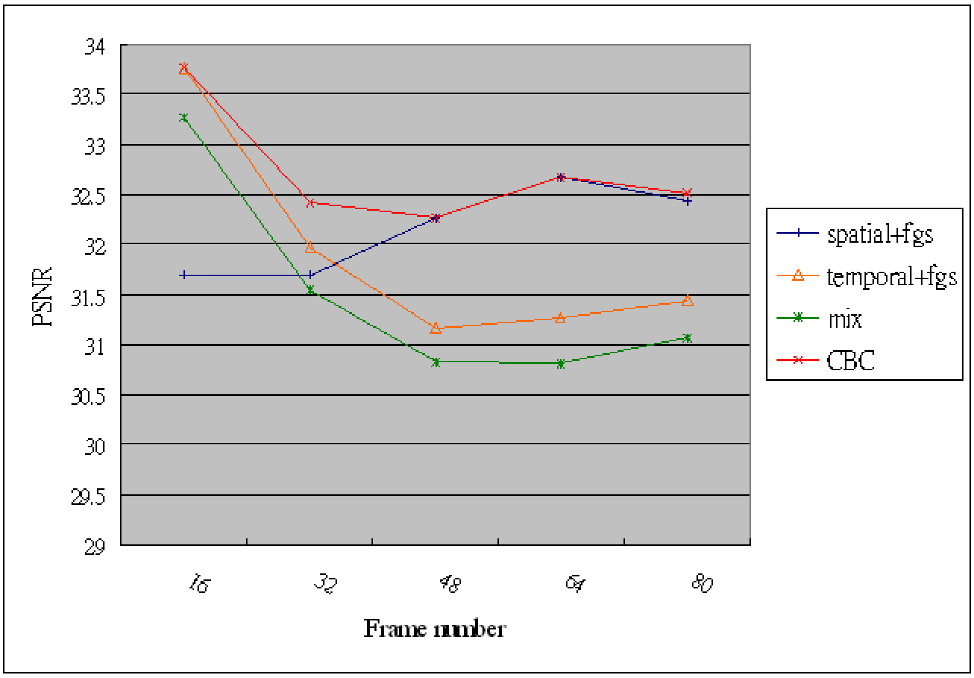


Fig. 23 Experimental result of dancer

Dancer is a case of simple texture with slow motion and thus, it needs to be applied distortion measurement on each GOP. The resulting video quality performs better than that of the others in most case, and equal to some other standard coding which is best among others. The result tells the success of applying distortion measurement.

4.2.4 Exp. 4

Experiment 4 is using a yuv sample, football, with 81 frames as testing. The spatial scalability is achieved by using two kinds of resolution format, cif(352 x 288) and qcif(176 x 144). The testing bit-rate is set as 570 kbps.

The scalable type assignment for each step of the proposed method is shown in table 5 below

	Content type initiating	**Group adjusting**	**Distortion measurement**

GOP1	**I????? s s s s s s s s s s ?**	**(I??)(???)s s s s s s s s s s s**	**(I t t)(t t t)s s s s s s s s s s s**
GOP2	**I? s s ? s s s s s s s s s s ??**	**I s s s s s s s s s s s s s s ??**	**I s s s s s s s s s s s s s s s s**
GOP3	**I????????????????**	**None**	**I t t t t t t t t s s s s s s s s**
GOP4	**I????????????????**	**None**	**I s s s s s s s s s s s s s s s s**
GOP5	**I????????????????**	**None**	**I s s s s s s s s s s s s s s s s**

Table 5 procedure of type deciding for each GOP

The experimental result is as shown in Fig. 24

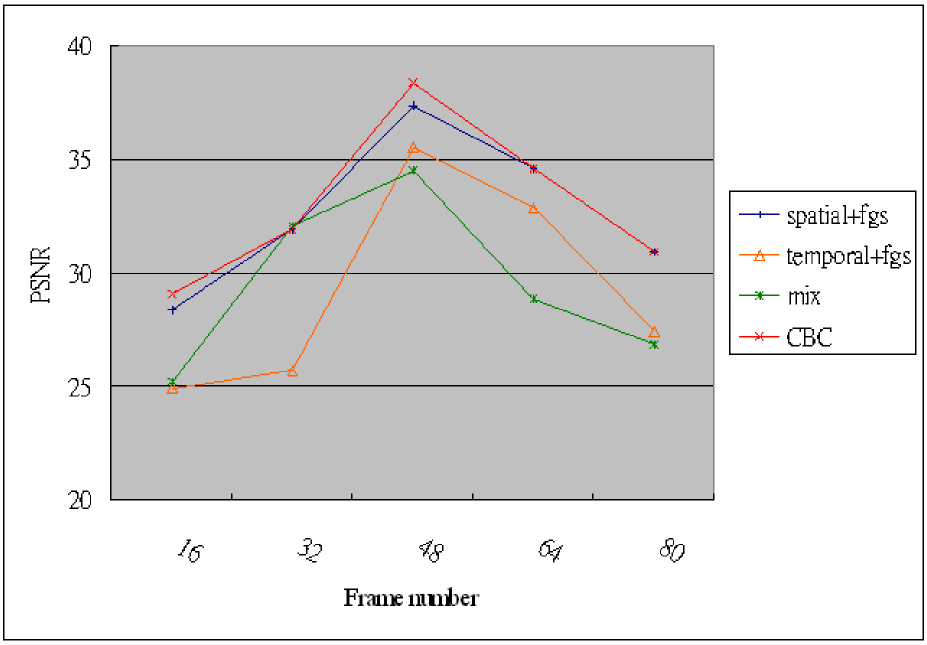


Fig. 24 Experimental result of football

Football is a simple-texture video sequence. The stage of type initiating well determines the type for some high motion pictures during frame 0~32, which makes the resulting quality equal to that of the best standard coding or perform better than that of the others. For other GOPs, distortion measurement works well for choosing the most suitable coding type for each frame.

4.2.5 Exp. 5

Experiment 5 is using a real-world sample, Madagascar, with 209 frames as testing. Madagascar is an animation movie, and the reason of using this as sample is that we want to examine the effect by applying the proposed method on a real-world video sequence. The spatial scalability is achieved by using two kinds of resolution format, 704 x 576 and 352 x 288. The testing bit-rate is set as 1100 kbps.

The video content type for this sample contains SS and SH type. Because the procedure of similar classification has been shown in previous section, we only show the experimental result.

The experimental result is as shown in Fig. 25

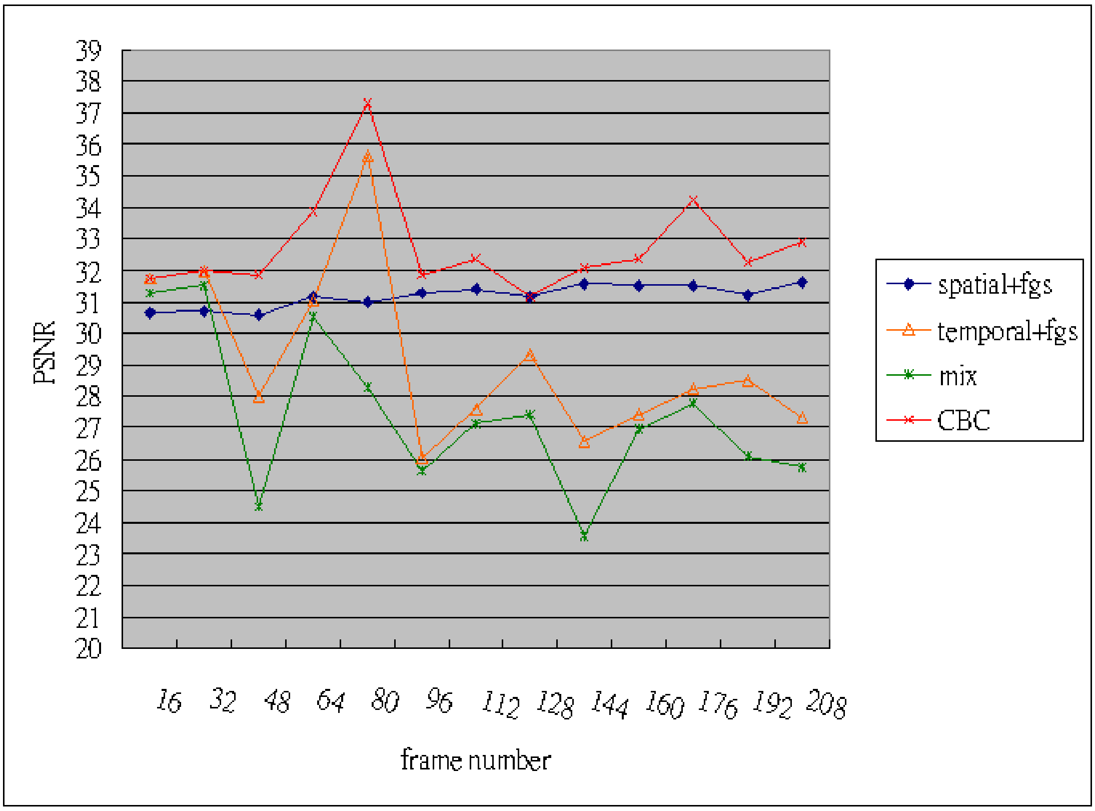


Fig. 25 Experimental result of Madagascar

Madagascar is a testing sample transformed from AVI file to video sequence. The content of Madagascar contains many scene changes and exaggerate object movement, including deformation. We can observe the result in Fig. 25 that the proposed CBC method performs well with a real-world video compared to the standard.

4.2.6 Complexity analysis

This section discusses the comparison of complexity among the proposed method and standard coding ways. The complexity is measured in terms of number of motion estimation and compensation (ME and MC), because ME and MC take the major amount of time during encoding process. Before discussing the complexity, we first address the concept of reusing ME and MC, which means that the result of ME and MC used in the type initiating stage to estimate the degree of motion can be reused in the distortion measurement stage if the reference frames are the same.

Assuming that after type initiating and group adjusting of a GOP, the number of picture marked with S is Ns, the number of picture marked with T is Nt, the number of picture marked with Q is Nq, the number of picture marked with ? is N?, and the size of the GOP is N. For the sub-GOP with type S and Q, no more ME and MC is needed because distortion measurement is not required by them; However, for the sub-GOP of type T, all the pictures except those at the highest enhancement layer need one extra ME and MC because the corresponding reference frames used by these pictures are different from those used in the type initiating stage. That is, Nt – floor(Nt/2) – 1 pictures require extra ME and MC. Similarly, for the sub-GOP of type indecisive, N? – floor(N?/2) – 1 pictures require extra ME and MC.

Except the first and the last frame, considering bi-prediction to be used for each picture in a GOP, and let C be the number of ME and MC, we'll have the following comparison of complexity:

Proposed method: C = (2N+1) + [(Nt – floor(Nt/2) – 1)x2-1] + [(N? – floor(N?/2) – 1)x2-1]

SSC: C = 2N+1

TSC: C = 2N+1

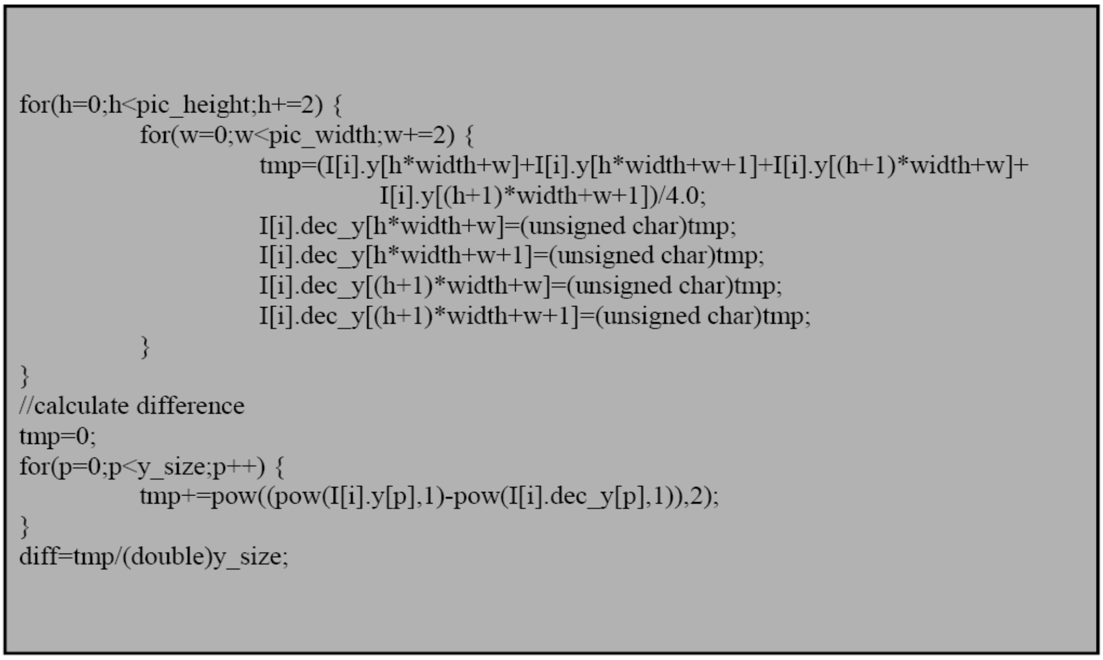
The equation of the proposed method contains three terms. The first term represents the number of ME and MC during type initiating, and the remaining terms represent the number of extra ME for sub-GOPs marked with T or indecisive. The worse case will happen when Nt = N+1 or Ns = N+1, in this case C will be about 3N-1, which is about N-2 times more than the standard coding ways.

Chapter 5 Conclusion

After examining on a mount of video samples, we usually have two situations. One is that the video content in a video sequence is consistent; in this case, the result of the proposed CBC would choose to use the most suitable scalable coding method (SSC, TSC, or mix) for the corresponding sequence. The other is that there exists some inconsistency among a video sequence, like sudden high moving of camera; in this case, the result of CBC would give a better video quality over the other fixed GOP scalable coding methods adopted by the standard.

From the experimental results, we observe that CBC provides not only a usually better video quality, but also a stable quality. Not like the other scalable coding methods, which often suffer from dramatically quality drop at the occurrence of significant change in video content, the CBC uses adaptive GOP and applies hybrid SVC to avoid a burst of quality degradation. Therefore, CBC provides a customized scalable coding way for the input video sequence to achieve the best manipulation of scalability.

Appendix A



```
for(h=0;h<pic_height;h+=2) {
        for(w=0;w<pic_width;w+=2) {
                tmp=(I[i].y[h*width+w]+I[i].y[h*width+w+1]+I[i].y[(h+1)*width+w]+
                        I[i].y[(h+1)*width+w+1])/4.0;
                I[i].dec_y[h*width+w]=(unsigned char)tmp;
                I[i].dec_y[h*width+w+1]=(unsigned char)tmp;
                I[i].dec_y[(h+1)*width+w]=(unsigned char)tmp;
                I[i].dec_y[(h+1)*width+w+1]=(unsigned char)tmp;
        }
}
//calculate difference
tmp=0;
for(p=0;p<y_size;p++) {
        tmp+=pow((pow(I[i].y[p],1)-pow(I[i].dec_y[p],1)),2);
}
diff=tmp/(double)y_size;
```

Fig. 26 procedure of texture detecting

References

[1] Alexis Michael Tourapis, Member, IEEE, Feng Wu, Member, IEEE, and Shipeng Li, Member, IEEE, “Direct Mode Coding Bipredictive Slices in the H.264 standard”, IEEE transaction on circuits and systems for video technology, vol. 15, no.1, January 2005

[2] Chen Ying, Xie Kai, Zhang Feng, Pandit Purvin, Boyce Jill, “Frame loss error concealment for SVC”, Thomson Corporate Research, Technology Fortune Center, Beijing 100085, China

[3] Emrah Akyol, A. Murat Tekalp, M. Reha Civanlar, “motion-compensated temporal filtering within the H.264/AVC standard”, College of Engineering, Koc University, Istanbul, Turkey, Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627, 2004 International Conference on Image Processing (ICIP)

[4] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand, “analysis of hierarchical B pictures and MCTF”, Fraunhofer Institute for Telecommunications – Heinrich Hertz Institute, Image Processing Department, Einsteinufer 37, 10587 Berlin, Germany

[5] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand, “MCTF and scalability extension of H.264/AVC”, Fraunhofer Institute for Telecommunications – Heinrich Hertz Institute, Image Processing Department, Einsteinufer 37, 10587 Berlin, Germany

[6] Hiroyuki Katata, Norio Ito, Member, IEEE, and Hiroshi Kusao, “Temporal-Scalable Coding Based on Image Content”, IEEE transaction on circuits and systems for video technology, vol. 7, no. 1, February 1997

[7] Hong Chang, Dit-Yan Yeung, Yimin Xiong, Department of Computer Science, Hong Kong University of Science and Technology, “Super-Resolution Through Neighbor Embedding”, Proceedings of the 2004 IEEE computer Society Conference on Computer Vision and Pattern Recognition (CVPR’04), 1063-6919/04, 2004

[8] ISO/IEC 14496-10, International standard, Information technology – coding of audio-visual objects – Part 10: Advanced Video Coding, Second edition 2004-10-1

[9] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), JVT-R202 of Joint Scalable Video Model JSVM-5, Draft Output Document from JVT, 18th Meeting: Bangkok, Thailand, 14-20 January, 2006.

[10] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), Joint Draft 5: Scalable Video Coding (in integrated form with ITU-T Rec. H.264 | ISO/IEC 14996-10), 18th Meeting: Bangkok, Thailand, 14-20 January,

2006

[11] K. Ugur and P. Nasiopoulos, “Design Issues and a Proposal for H.264-based FGS”, contribution MPEG03/M9505, ISO/IEC JTC1/SC29/WG11, Pattaya, Thailand, March 2003

[12] Stefano Belfiore, Macro Grangetto, Member, IEEE, Enrico Magli, Member, IEEE, and Gabriella Olmo, Member, IEEE, “Concealment of Whole-Frame Losses for Wireless Low Bit-Rate Video Based on Multiframe Optical Flow Estimation”, IEEE transaction on multimedia, vol. 7, no. 2, April 2005.

[13] Wallace Kai-Hong Ho, Wai-Kong Cheuk, and Daniel Pak-Kong Lun, Member, IEEE, “Content-based Scalable H.263 Video Coding for Road Traffic Monitoring”, IEEE transaction on multimedia, vol. 7, no. 4, August 2005

[14] Wen-Hsiao Peng, Scalable Video Coding-Advanced Fine Granularity Scalability, a thesis submitted to institute of electronics college of electrical and computer engineering National Chiao Tung university in partial fulfillment of the requirements for the degree of Doctor of philosophy in electronics engineering, December 2005.

[15] Yu Chen, Keman Yu, Jiang Li and Shipeng Li, “an error concealment algorithm for entire frame loss in video transmission”, Department of Electronic Engineering, Tsinghua University Microsoft Research Asia