Video Codec Identification

By

Ahmed Allam

749285

School of Engineering

Politecnico Di Milano

Polo Regional Di Como

2010-2011

Video Codec Identification

By

Ahmed Allam

Master Thesis

Submitted to the School of Engineering

Through Computer Engineering Program, Communication Track

In Partial Fulfilment of the Requirements for

The Degree of Master of Science at the

Politecnico Di Milano

Polo Regional Di Como

Como, Italy

2010-2011

Approved By:

Dr. Marco Tagliasacchi (Advisor), Computer Science Engineering Department (Polimi)

Dr. Enrico Magli (Co-Advisor) Computer Science Engineering Department (PoliTo)

DECLARATION

I hereby declare that I am the sole author of this document. I authorize the Politecnico Di Milano University to lend this document to other institutions or individuals for the purpose of scholarly research.

I further authorize the Politecnico Di Milano University to reproduce the document by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Ahmed Allam

749285

ABSTRACT

The popularity and the presence of acquisition and storage devices have been increasing rapidly over the past few years. Hence, the concept of a multimedia object in the multimedia world started to formulate and exist as independent entity. This object that is like many others in the world possesses properties that identify and describe the various elements it comprises and more importantly artifacts that characterize its uniqueness.

The birth of multimedia objects does not come from void; in fact it passes into process of construction which includes processing steps that may include different stages that leave their characterization that is known especially in multimedia forensics by footprints. The latest researches in the field of forensics exploit the fact of the birth stages to investigate more for discovering artifacts that will be a very valuable source to better understand the formulation and the past of these objects. These artifacts or footprints that are left play big role in shaping the understanding of forensic analysts of the history of multimedia objects.

Forensic analysis in the multimedia field is a very essential nowadays; many researches are focused on the multimedia processing including images, audio and video. Because of the artifacts imposed in the acquisition or due to lossy coding compression phase and others lead to the development of methods for removing and fighting these “footprints”.

Instead in our work we consider these footprints as the trace that we will follow to reach the origin of the multimedia object and the process it passed to reach to its state.

For this reason, the goal of this thesis is to exploit the artifacts and the footprints left by video encoders to detect the type of encoding of a sequence. That means, given an uncompressed sequence which is “raw video”, the algorithm should understand if the sequence passed through encoding and decoding process and which type of encoding focusing on the most spread types of encoding (AVC/H.264, MPEG4 visual part 2, MEG2). We leverage the challenge into introducing noise or second intermediate encoding with different types and our goal is to see to which extent the algorithm can still detect the type of encoding and specifically for which quantization parameter. Moreover, we explore the estimation of GOP structure and identification of I-frames.

DEDICATION

I want to express my sincere gratitude and feelings to all people who supported and stood beside me from my instructors, parents, my family and friends to produce this master thesis project.

ACKNOWLEGEMENTS

On the behalf of my colleagues, I, the author of this project wish to express my deep and fond appreciation and gratitude to my advisor Dr. Marco Tagliasacchi. He is the man who supported and guided my efforts to achieve this master thesis project. His advice, encouragement and help, boosted my work and motivated me to strive hard in my endeavour to produce this master thesis project. It is my honour to be student under his supervision and his wise guidance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

1.1 Introduction

Investigating the history of objects is of such importance since it gives us the possibility to understand and discover the origin. It is the quest of all sciences from geology, life sciences and recently multimedia and computer science field. The clues and hints that we start with our investigation varies from one field to another but what is more relevant for us in the multimedia processing field are the artifacts and the footprints left by the process and the stages of developing each multimedia object.

To understand the past of objects, we have to understand the artifacts of the past. These unique "left-over" are the pattern that forms which gives us the chance and the possibility to challenge ourselves to enhance our understanding of the past. Embracing these "remainders" will open for us new way and perspective to look to things in deeper way that widens our vision for analyzing complex relation and composite formulation of objects.

For example, the forensics research field is still used to investigate and study artifacts that occur due to lossy coding compression of images and videos or due to acquisition procedure and other types of artifacts. The current state of the art targets type of device detection like type of cameras used or tampering identification which this opens another developing field which is the anti-forensics. Basically most of the work targets still images where in the case of video detection techniques are realized through frame by frame processing analysis. Currently State-of-the-art video coding techniques adopt motion-compensated prediction along time and, considering also the recent 3D video coding architectures, across different views. Video encoders supporting recent standards (e.g. H.264/AVC) enable several coding tools, each responsible for specific footprints left in the coded material. Note that many tools are not described in the normative part of the standards, thus the corresponding footprint might be related to the specific encoder implementation. Transmission of the video content over a noisy channel might suffer from losses. In these cases, the reconstruction of the video content at the decoder is also implementation-dependent, thus leaving a characteristic footprint. Understanding the side-effects of recent coding tools (e.g. intra-prediction, in-loop deblocking filter) and handling motion properly is necessary to design effective coding-based footprint detectors.

For many reason in forensics applications the key idea is to understand the origin of the object. As mentioned above, knowing the acquisition device type and which type of encoding is something challenging to know and understand.

Therefore the studies of artifacts that result due to noise, sensor acquisition, lossy coding and other source of artifacts are studied in order to know how to eliminate them for having better image or video which could be considered as an enhancement or a step ahead toward better result or better performance. Although, recent researches in the forensics field are not only to detect artifacts for elimination or improving by minimizing their effect instead, these artifacts are treated as footprints for detecting stages of processing of multimedia objects. For example evaluating the distortion introduced by video coding is part of the forensics field where the study of quality assessment metrics for encoded images and videos. Like in the work of [Liu and Heynderickx, 2008] that perceptually quantify the blocking artifacts of the DCT coded macro blocks. Another footprint due to low bit rate JPEG 200 coding which is an overall blurring of the compressed image.

This was noticed and quantified in the work of [Marziliano et al., 2002], by analyzing the average width of image edges. On the opposite side in the work of [Caviedes et al., 2006] the observation of sharpness which is the opposite phenomenon of the blurring was done by inferring the degree of sharpness through analyzing the kurtosis of DCT coefficients that indicates the peakdness of the coefficient distribution. And many other relevant work for detecting footprints and formulating quality assessment metrics. Moreover, not only coding footprints but also footprints due to acquisition was studied and many articles are worth mentioned. Since many techniques that are applied on images are still valid and applicable for video when it comes to simple frame processing, works of [Bayram et al., 2008] and [vanHouten and Geradts, 2009] focus on extracting the PRNU footprints from video clips. In addition, similar approach was used in [Caldelli et al., 2007] to detect malevolent changes in video sequences, whose presence is detected by recognizing a change in the PRNU pattern underlying the sequence. Moving from acquisition based footprints to explore another type of footprints due to coding is another interesting field. In the simplest coding architectures, each video frame is encoded as a still image (intra-frame coding), exploiting only spatial redundancy by means of transform coding, e.g. adopting JPEG-like coding tools. As such, footprint detection techniques designed to detect blockiness, blurring or sharpness can be applied without modifications to video sequences. The overall MSE distortion due to coding can be evaluated in a no-reference fashion leveraging statistical characteristics of the DCT coefficients in coded frames. For example, in [Ichigaya et al., 2006] and [Brandao and Queluz, 2008] the blind estimation of PSNR is carried out by first estimating the original distribution of DCT coefficients in each subband, starting from the quantized coefficients. In [Li and Forchhammer, 2009] the quantization parameter of MPEG Intra frames is computed from quantized coefficients and used to tune a de-blocking algorithm. The work in [Wang and Farid, 2009] describes the detection of double MPEG compression, when only I-frames are used. However, conventional video coding standards leverage motion-compensated prediction in order to tackle temporal redundancy. Each group of pictures (GOP) contains frames of different kind (e.g. I-, P- and B-frames), depending on the reference frames used for prediction. The GOP structure is detected in [Luo et al., 2008] based on the strength of spatial blocking artifacts, which exhibit a characteristic temporal pattern. Not to mention many other types of footprints that are related to channels errors when coded videos are transmitted through error-prone channels. As it is obvious and evident how many work has been done in the field of detection of footprints that results from different procedures and processes of multimedia objects. After all this discussion what can be noticed is the emphasis on the footprints related to video and on images since the argument will not finish due to enormous amount of researches in the field of forensics field. But what has not been investigated yet is the detection of type of video encoding for raw video sequences. The idea of understanding or knowing what is the type of encoding, a sequence has passed through is challenging. Many websites like (Youtube, Facebook, etc) have their own process for treating and processing videos after they are uploaded by users to the website. By having an algorithm or technique to identify the type of encoding sequences passed through can help in further studying the origin of the video. Since some pattern or footprints can be detected to identify if the video is downloaded from specific website like Youtube or other sites and whether it has been encoded and which type of encoding specifically. Stressing on the idea of detecting the most spread three types of encoding which are (H.264/AVC, MPEG 4 part 2 visual and MPEG2). In this case for future studies this can be investigated more to see if some videos could be traced back and understand where they were downloaded or originated from which site so that we know the users uploaded them which help in detecting the origin of pirated movies. Another useful idea is by knowing type of encoding technically we can understand how to encode again if needed the videos and which type of encoding to use. Further, in this master thesis project, detection is not the only outcome where also we can estimate and identify the I-frames of each group of picture GOP. No matter which is the structure of the GOP of the video sequence, the detection of I-frames is identified with the estimation of the Quantization factor for each I-frame in the sequence. In addition, not only I frames will be detected but also P and B frames especially in the case of AVC/H.264 encoding. Starting from small scenarios for testing and detecting fixed quantization of video sequence and escalading to more complex scenarios and relaxing hypothesis and constraints of having fixed quantization or knowledge of the GOP structure of the video sequences, the thesis goes through all these by exploring algorithms created and implemented with tests that will be more discussed in chapter three and four. The thesis will be explored in four chapters which are divided in this manner:

Chapter 1 introduces to the topic of the master thesis project; what are the main points and goals, motivations and high level of statement of the problem. In addition, it contains a brief description for each chapter in the master thesis project.

Chapter 2 will be introducing the current state of the art and what has been done similar or near to this project. This section will explore detection of MPEG compression, estimation of quantization parameters, estimating the GOP structure, detection of footprints in video coding and video forensics tools in general.

Chapter 3 talks about the technique developed for this master thesis project. The type of video encoding detection and the idea behind it will be illustrated more. The development of the algorithms and addressing different stages form simple to more complex ones. Relaxing the constraints and taking into account broader scenarios to test the robustness and scalability of the algorithm or the technique. The stage of preparation of the sequences or the training set and the validation stage will be discussed in details. Addressing specific scenarios and having different approaches to test for obtaining results and validating the simulation. In general, profound and deep explanation for the technique proposed will be presented in this chapter.

Chapter 4 will contain the results of the test and the simulation with the analysis of the results. For each scenario, test results will be obtained and explained elaborating on the way for developing techniques for identification from the simple to complex scenarios. In this chapter we will comment on the results of the technique proposed for the detection of the type of video encoding the video sequence passed by (AVC/H.264, MPEG4 part 2 visual, MPEG2) and the detection of I-frames with the estimation of quantization parameter. Moreover, we exceed to detect also the P and B frames and the estimation of quantization in the case of H.264/AVC that is done in a more robust and particular method that differs from any other method in the current state of the art. Finally, a conclusion and future work will be mentioned as future continuity of this work.

Chapter 2

Video forensics

In this chapter we will see various techniques and methods in Video forensics field which span many aspects of detections and parameter estimation of video encoding. Therefore what will be handled in this part will be divided into three parts. First one will be about the PSNR (peak signal to noise ratio). PSNR is very important metric or indicator in the field of signals and evaluating multimedia objects from picture, video and audio. In this context, we will see the importance of PSNR in video coding showing two methods for PSNR estimation, one for estimating coding PSNR using quantized DCT coefficients that is in the work of [Ichigaya et al., 2006]. And the other method is non-reference PSNR estimation algorithm for H.264/AVC encoded video sequences which is in [Brandao and Queluz, 2008]. The second part of this chapter will introduce methods used to detect MPEG 2 video parameter and PSNR estimation in the work of [Li and Forchhammer, 2009] and MPEG recompression detection and GOP structure estimation in the work of [Luo et al., 2008]. At the end of this chapter we reach to H.264/AVC encoding where in this section we deal with estimating QP and motion vectors from decoded pixels that is done by [Tagliasacchi, Valensize and Tubaro]. Moreover, we address the blind estimation of the parameter in H.264/AVC decoded video that is found in [Tagliasacchi and Tubaro, 2010].

2.1 PSNR

Peak signal to noise ratio abbreviated PSNR is the ratio between the maximum possible power of a signal and the noise power that affects or disturb the content of the signal. It is used as a measure of quality especially for the reconstruction of the lossy compression codecs. In our case here the signal is the data or the video sequence and the noise is the error introduced because of compression. It is expressed in logarithmic scale decibel. What is more important is the interpretation of this metric. In other words, what does it mean to have high or low PSNR? The answer will having higher PSNR means something good that is the reconstruction is of higher quality and more equivalent or similar to the original data or video sequence in our case. Now after introducing the PSNR term, we will investigate some of the works that have been done in video coding related to estimation or detection of PSNR. Starting from the work of [Ichigaya et al., 2006] we explore a method for estimating coding peak signal to noise ratio (PSNR) without the use of reference signal by using quantized DCT coefficients.

2.1.1 Estimation of PSNR Using Quantized DCT Coefficients

The idea can be summarized in this way: The method in [Ichigaya et al., 2006] paper allows for the PSNR estimation depending on the probability density function of the quantized DCT cosine transform coefficients that are extracted from MPEG-2 bit stream. “The experiment was held on MPEG-2 video coding bit streams under varying quantization scheme and evaluate the method with comparing PSNRs with the actual PSNRs” [Ichigaya et al., 2006]. There are two types for measuring the video quality without the original or reference sequence: first method using decoded video signals and the second method is by using information contained in a bit stream. Where the second method assumes noise model for several spatial or temporal features including blockiness, edge energy etc [Ichigaya et al., 2006]. Therefore the method proposed in this paper is of the second type since estimation of PSNR is done from MPEG-2 bit streams under rate control scheme. In this method DCT quantization noises from statistical properties of quantized DCT coefficients and utilizes several coding parameters in a coded bit stream. By analyzing the statistical properties of decoded pictures, the information for calculating PSNR is derived. The method supports PSNR estimation for each frame of every picture type without preliminary experiments for calibration.

Framework of Estimation Explained:

The degradation of quality of the picture in MPEG-2 coding occurs due to quantization of DCT coefficients. Where PSNR represents the amount of coding error that is the difference between the source and decoded signal and it is given by this formula [Ichigaya et al., 2006]:

$$PSNR = 20\log_{10}\left(\frac{S_{p-p}}{\sqrt{MSE}}\right)$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - x'_i)^2 \qquad (1)$$

x_i : sample values of the source signal

x'_i : Corresponding decoded signal at sample number i.

S_{p-p}: Peak signal Amplitude which is equal to 255 in case of 8 bit representation.

Parseval's theorem states that mean square error (MSE) in the pixel domain is equivalent to mean square quantization error (MSQE) in the DCT domain because DCT is a normalized orthogonal transformation. Hence using the following equation (2), it is possible to measure PSNR from quantization error in DCT domain.

$$PSNR = 20\log_{10}\left(\frac{S_{p-p}}{\sqrt{MSQE}}\right)$$

$$MSQE = \frac{1}{N}\sum_{i=1}^{N}(X_i - X'_i)^2 \qquad (2)$$

X_i : DCT coefficient of the source signal

X'_i : DCT coefficient of the decoded signal

Given the amplitude distribution of DCT coefficient of the source signal, the value of the quantization error will be:

$$MSQE = \frac{1}{N}.\sum_{v=0}^{7}\sum_{u=0}^{7}\sum_{q\_code=1}^{31}(n_{(u,v,q\_code)} \times \sum_i \int_{\alpha_i}^{\beta_i} f_i(X)dX)$$

$$f_i(X) = p_{(u,v,q\_code)}(X).(X - q_i)^2$$

$$N = \sum_{v=0}^{7}\sum_{u=0}^{7}\sum_{q\_code=1}^{31} n_{(u,v,q\_code)} \qquad (3)$$

$p_{(u,v,q\_code)}$: Probability Density Function (pdf) of DCT coefficient value X at (u,v) for each quantized step size.

q_i: Quantization representative value for the quantization interval number i.

$n_{(u,v,q\_code)}$: Total sample number of DCT coefficient at (u,v) for each quantized step size.

$[\alpha_i, \beta_i)$: Quantization interval

According to the results of the preliminary experiments for the analysis of the quantized DCT coefficients indicates: The Laplacian distribution is more relevant than Generalized Gaussian function since the kurtosis cannot be used for estimating distribution. Laplacian distribution is identified by standard deviation, and this allows estimating source distribution after quantization distribution. In this method scheme the distribution of DCT coefficients is modeled in the form of laplacian distribution.

One important note that since the standard deviation depends on quantizer step size then it is necessary to calculate MSQE for each quantized step size.

In MEG-2 there are different prediction methods that are used by the three types of pictures. I-picture is coded using its own information; P and B-picture are coded using motion compensated prediction from reference frames.

The quantization of intra-dc coefficients is linear and the step size is constant in a frame. Since the shape of dc component distribution is strongly dependent on the picture contents, intra-dc coefficients are estimated using the difference value for quantized coefficients between adjacent blocks. In the intra-coding, the *dct_diff* value is analyzed instead of the dc coefficient itself. The distribution of the *dct_diff* value can be assumed to be Laplacian distribution. [Ichigaya et al., 2006]

For the case of non-intra coding the following expression represents the decoded picture:

$\hat{s}_k + w(s_k - \hat{s}_k)$ (s_k : input image; $\hat{s}_k$: prediction image at instant k)

$w(*)$: DCT quantization and their inverse processes

Therefore the coding error e_k is represented by:

$$(s_k - \hat{s}_k) - w(s_k - \hat{s}_k) \quad (4)$$

According to equation (4), MSQE for P and B-Pictures can be calculated for prediction error signals that are similar to decoded image signals. Due to the presence of noncoded blocks in P and B-Pictures which means the absence of information for estimating the MSQE in the coded bit stream, the MSQE is calculated using coded blocks:

$$MSQE = aMSQE_{INTRA} + bMSQE_{Non\mathrm{int}ra} \quad (5)$$

(a and b are ratios of intra-coded blocks and non-intra-coded blocks among all coded blocks.

The quantization interval $[\alpha_i, \beta_i)$ is required to estimate coding PSNR even though coded bit stream does not contain information about it. Where the quantization representative value q_i is assumed to be located in the midpoint of quantization interval:

$$\alpha_i = q_i - 0.5 \times SS_{q\_code}$$

$$\beta_i = q_i - 0.5 \times SS_{q\_code}$$

Now we reach to the procedure for obtaining the PSNR:

Using the probability function of the laplacian distribution that approximates the DCT Coefficients of the ac components in the following equation:

$$p(x) = \frac{1}{\sqrt{2\sigma}} \exp\left(-\frac{\sqrt{2}\,|\,x\,|}{\sigma}\right) \quad (6)$$

And using the equations (2) ,(3) and (5) we will obtain PSNR by these steps:

1) Extract quantized DCT coefficients and decode quantization step size SS_{q_code}

2) For each DCT component (u,v) and quantization step size SS_{q_code}, calculate the standard deviation σ

3) Estimate pdf $p_{(u,v,q\_code)}(X)$ of source DCT coefficients by equation above (6)

4) Calculate MSQE by using equation (3)

5) Calculate PSNR by (2) and (5)

Another extension of the method is the compensation method for estimating the Distribution. In some cases most of the coefficients becomes zero after quantization which results in smaller estimated standard deviation of the distribution much smaller than that of source distribution. Therefore MSQE is estimated smaller than the actual value which leads to the reduction of the estimation accuracy of DCT quantization error. For this reason to improve the estimation accuracy, non-zero coefficients should be handled carefully. Hence it comes the need to introduce the extended laplacian function that is represented in the following equation:

$$P(x) = \frac{1}{\sqrt{2\sigma}} \exp(-\frac{\sqrt{2}}{\sigma}(|X| - C)$$

C: is parameter to control the spread of the function.

This function will be used for estimating the distribution in this way:

1) Measure the standard deviation σ for the distribution of quantized DCT coefficients by frequency $f(q_i)$ of the quantization representative value q_i and estimate pdf $p_1(X)$, as in the estimation method described above.

$$\sigma^2 = \frac{\sum_i \{ f(q_i) \times q_i^2 \}}{N}$$

$$p_1(x) = \frac{1}{\sqrt{2\sigma_1}} \exp(-\frac{\sqrt{2}\,|x|}{\sigma_1})$$

2) Measure the Measure the standard deviation σ_2 for the distribution of quantized DCT coefficients excluding the zero value coefficients and estimate probability distribution $F(X)$ which is a extended Laplacian distribution.

$$\sigma^2 = \frac{\sum_i \{ f(q_i) \times (|q_i| \ - SS_{q\_code})^2 \}}{N} \quad (q_i \neq 0)$$

$$P(x) = \frac{1}{\sqrt{2\sigma_2}} \exp(-\frac{\sqrt{2}}{\sigma_2}(|X| - \frac{SS_{q\_code}}{2}))$$

$$= p_2(X) \times n_2$$

$$n_2 = \int p_2(X)dX = \exp(\frac{SS_{q\_code}}{\sqrt{2}\sigma_2})$$

3) Obtain pdf $p(X)$ as a weighted average of $p_1(X)$ and $p_2(X)$ which are the Laplacian pdfs

$$p(X) = \frac{p_1(X) + n_2 \times p_2(X)}{1 + n_2}$$

The results of the two experiments for estimating PSNR with and without compensation, show that the determination coefficient between estimated PSNR and actual PSNR are higher than 0.9. Moreover, the average estimation errors are within ± 1.0 dB except for HDTV 18 Mbit/s in B picture. Some of the results are in the following tables:

	I-picture [dB]	P-picture [dB]	B-picture [dB]
SDTV, 5 Mbit/s	-0.479	0.364	-0.702
HDTV, 60 Mbit/s	-0.157	0.724	0.708
HDTV, 18 Mbit/s	-0.838	-0.305	-1.458

Table 2-1 Average error between estimated without compensation and actual PSNR [Ichigaya et al., 2006].

	I-picture	P-picture	B-picture
SDTV, 5 Mbit/s	0.985	0.985	0.955
HDTV, 60 Mbit/s	0.987	0.995	0.987
HDTV, 18 Mbit/s	0.982	0.973	0.916

Table 2-2 Coefficients of Determination R^2 between estimated without compensation and actual PSNR [Ichigaya et al., 2006].

Table 2-3 Average error between estimated with compensation and actual PSNR [Ichigaya et al., 2006].

Table 2-4 Coefficients of Determination R^2 between estimated with compensation and actual PSNR [Ichigaya et al., 2006].

2.1.2 Non-Reference PSNR Estimation for H.264 Video Sequences

In this section, the discussion will be mainly on the second technique under the PSNR section which explores the work of [Brandao and Queluz, 2008]. This paper proposes algorithm for non-reference PSNR estimation for video sequences that are encoded by H.264 encoding. Statistical properties of the transformed coefficients that are modeled by Cauchy or Laplace probability density function are exploited by this method. “Where the distribution’s parameters are computed from quantized coefficient received at the decoder, combining maximum likelihood with linear prediction estimates.” [Brandao and Queluz, 2008]. It can be considered as a non-reference metric for evaluating the coded sequences because it does

	I-picture [dB]	P-picture [dB]	B-picture [dB]
SDTV, 5 Mbit/s	-0.718	-1.134	-0.709
HDTV, 60 Mbit/s	-1.252	-1.429	-2.029
HDTV, 18 Mbit/s	-1.423	-1.513	-0.549

	I-picture	P-picture	B-picture
SDTV, 5 Mbit/s	0.980	0.978	0.918
HDTV, 60 Mbit/s	0.979	0.992	0.983
HDTV, 18 Mbit/s	0.969	0.969	0.887

not depend on the original sequence. In this paper, the algorithm proposed estimates errors due to lossy compression in block wise DCT based video encoding schemes. It is easily adaptable to other DCT-based video encoding schemes.

2.2 MPEG Recompression Detection Based on Block artifacts

In this section we move from estimation to detection in the MPEG 1/2 encoding. In the work proposed by [Luo et al., 2008] there is a method for detecting tampering that is done to MPEG videos in which some frames are removed and then the rest of the video is re-encoded. The method that detects this type of tampering works on the temporal patterns of block artifacts in video sequences. The idea is due to the MPEG compression, block artifacts are introduced to different types of frames. Where there is a pattern for a given GOP that shows the strength of block artifacts as function of time. The observation comes from the relation between the block artifacts introduced by the first compression before the removal of frames and the effect on the average strength of the block artifact of the second compression that is related to number of frames deleted and the type of GOP used previously. First starting by defining important metrics or symbols for the technique proposed, we have Block Artifact Strength (BAS). It is a score to quantify the varying levels of block artifact for various types of frames. Note that video sequences with same GOP, their temporal patterns of BAS are similar. Moreover, in MPEG compression intra and inter frame coding are used to reduce spatial and temporal redundancy. And since it is block based coding in the spatial domain, this will result in block artifacts for each frame after MPEG compression. The raw frames are compressed into two types: intra-frames (I-frame) and inter-frames that consist of predicatively coded frames (P-frames) and bidirectional predicatively coded frames (B-frames). I-frames are encoded without depending on other frames by applying DCT based coding for non-overlapping 8x8 block. While Inter-frames, are encoded predicatively through motion estimation and compensation and this tries to remove temporal redundancy in the video. Motion estimation is done on macro-block basis of size 16x16, where motion compensation errors are encoded for each 8x8 block like I-frames. The arrangement of intra and inter frames is specified by using the structure of GOP (group of pictures) where I-frame is the key frames for GOP.

Figure 2-1 GOP (N=12 M=3) [Luo et al., 2008]

Block artifact detection for P and B-frames are observed along the 8x8 block boundaries that are due to motion compensation errors. Hence the block artifact of 8x8 is measured by following these steps: Given a frame fr, we divide it into 16x16 non-overlapping blocks. Then the BAS score is computed for the frame fr as the percentage of blocks that satisfy the following relation |E+H-F-G|>|A +D-B-C| that is demonstrated by the figure 2-2:

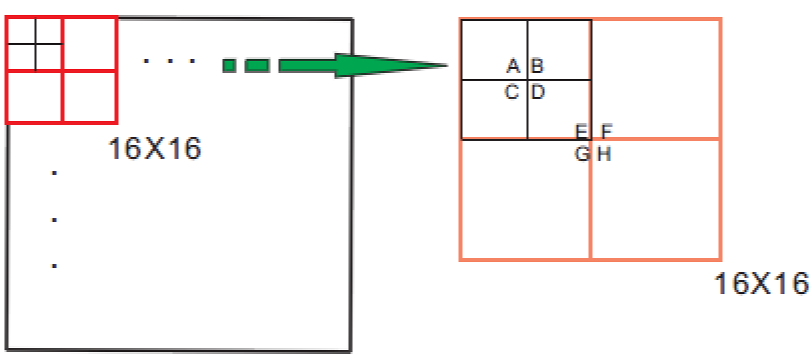


Figure 2-2 Block artifacts detection [Luo et al., 2008]

The results of the observations of the tests and observations came out with two properties for BAS:

1) The average level of BAS increases after MPEG compression
2) BAS of each frame is at similar level of raw video sequence; instead it fluctuates over time for different frame encoding for an MPEG sequence. The distance of P-frame from I-frame may also affect the BAS.

Detection Methodology:
The feature curve is the key to detect tampering associated with recompression. The following procedure will be explained step by step and the figure 2-3 illustrates visually the technique:
1) For a given MPEG video encoded at GOP equal to 12 frames, we remove 1 to 11 frames from the video respectively
2) Recompress these frames with specific GOP structure. The outcome will be 11 video streams denoted as $MPEG_i$ where i=1…11.
3) Calculate the average value of BAS for all the frames in each video stream and denote the result as Ave_i where i=1…..11.
The feature vector will be constructed [Ave_1, Ave_2 ,….,Ave_{11}] where it can be visualized as feature curve.

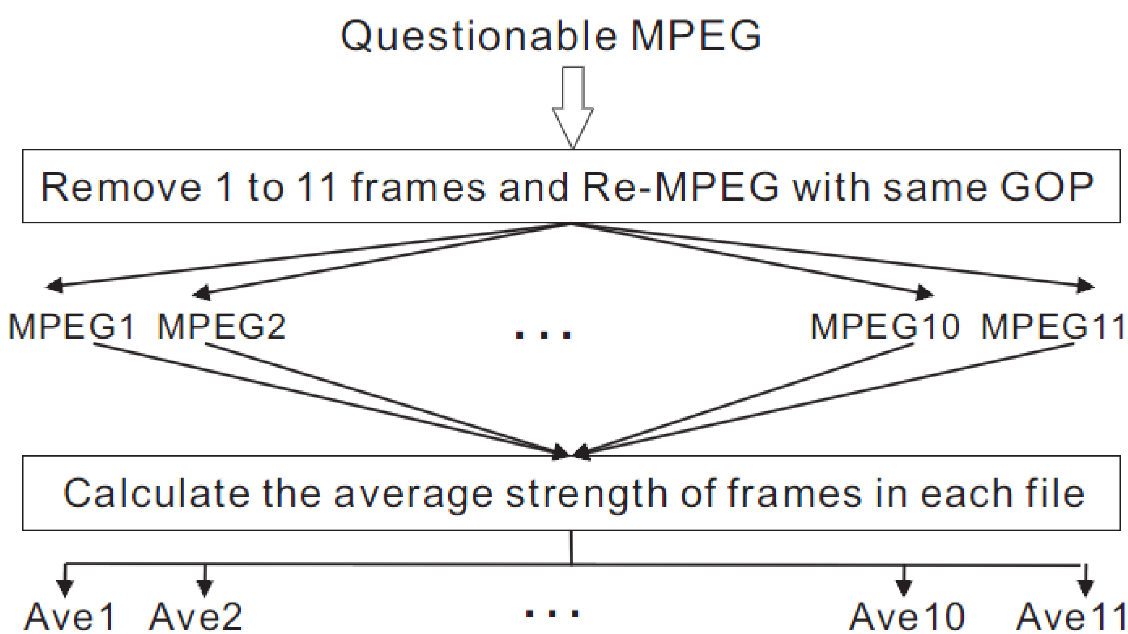


Figure 2-3 The process of extracting feature curve [Luo et al., 2008]

To see the results and the reliability of the technique, tests for frame removal and recompression occurred in this following procedure:
An original video is compressed with GOP (N=12 and M=3). Then we remove some frames and re-save with the same GOP structure. In figure 2-4, we can observer the change occurred.

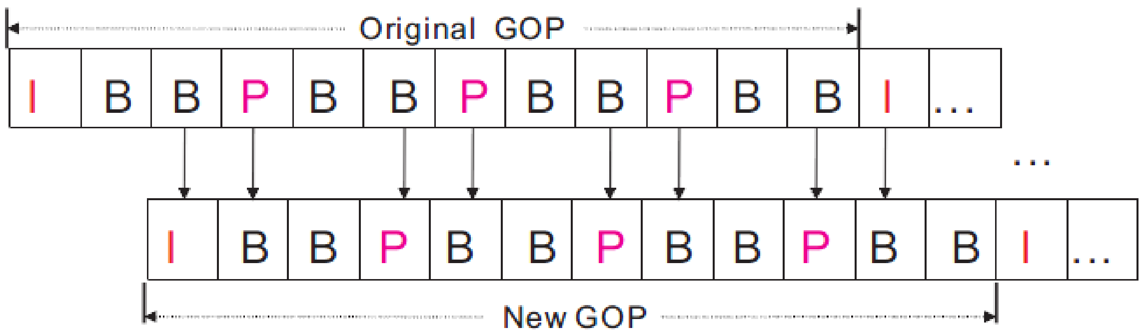


Figure 2-4 Frame removal and recompression [Luo et al., 2008]

Tampering operation happens by:
1) Encoding YUV sequence into MPEG with GOP (N=12 and M=3)
2) Remove 1 to 11 frames from the video respectively
3) Recompress the resulting sequence by same GOP which will give 11 tampered video.

The feature curves for these videos are shown in figure 2-5

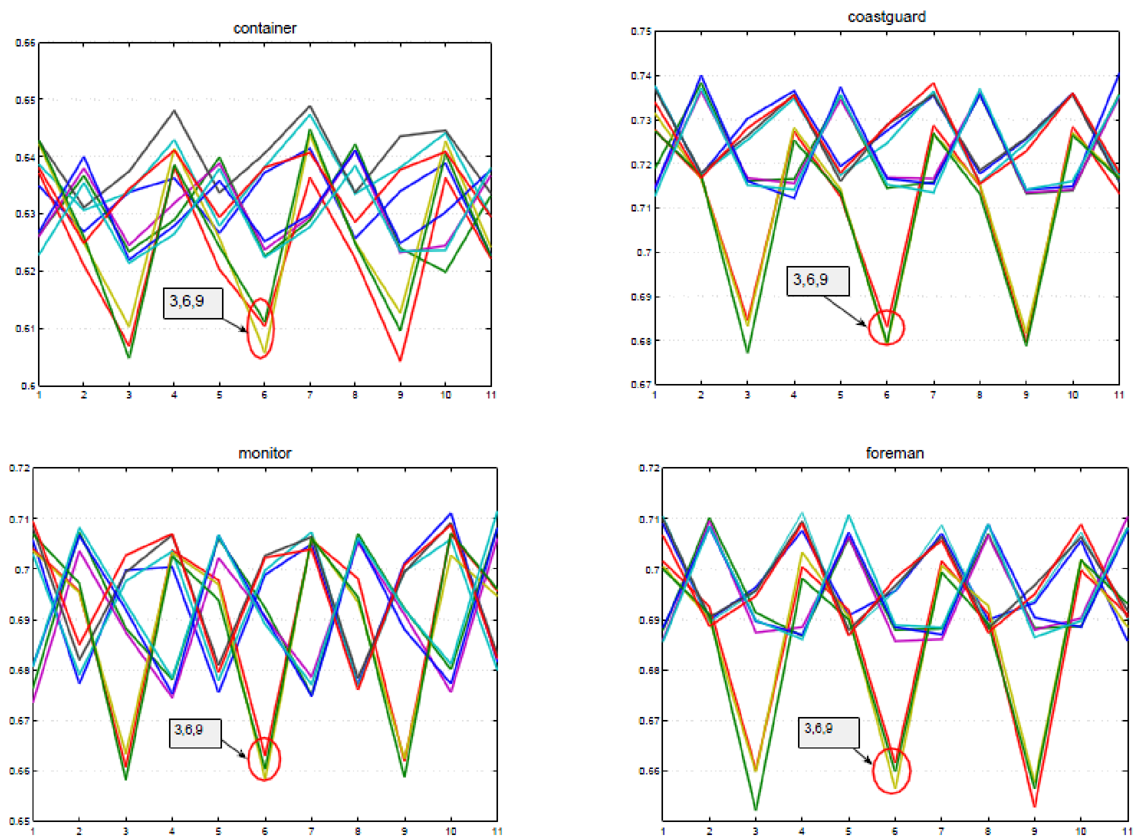


Figure 2-5 feature curves in tampered videos (frame removal and recompression) [Luo et al., 2008]

In the above figure, what can be noticed is each subplot contains 11 features curves. The curve for frame deletions of m and n have similar shapes when m=n(mod M), where M=3 in the example proposed [Luo et al., 2008].

In case of GOP conversion which means to re-save tampered video with different GOP structure from the initial compression one. The results show inconsistencies with the feature curve of the original one. Sample of the results is shown in figure 2-6. The red circles mark the inconsistencies with the original feature curve in each figure.

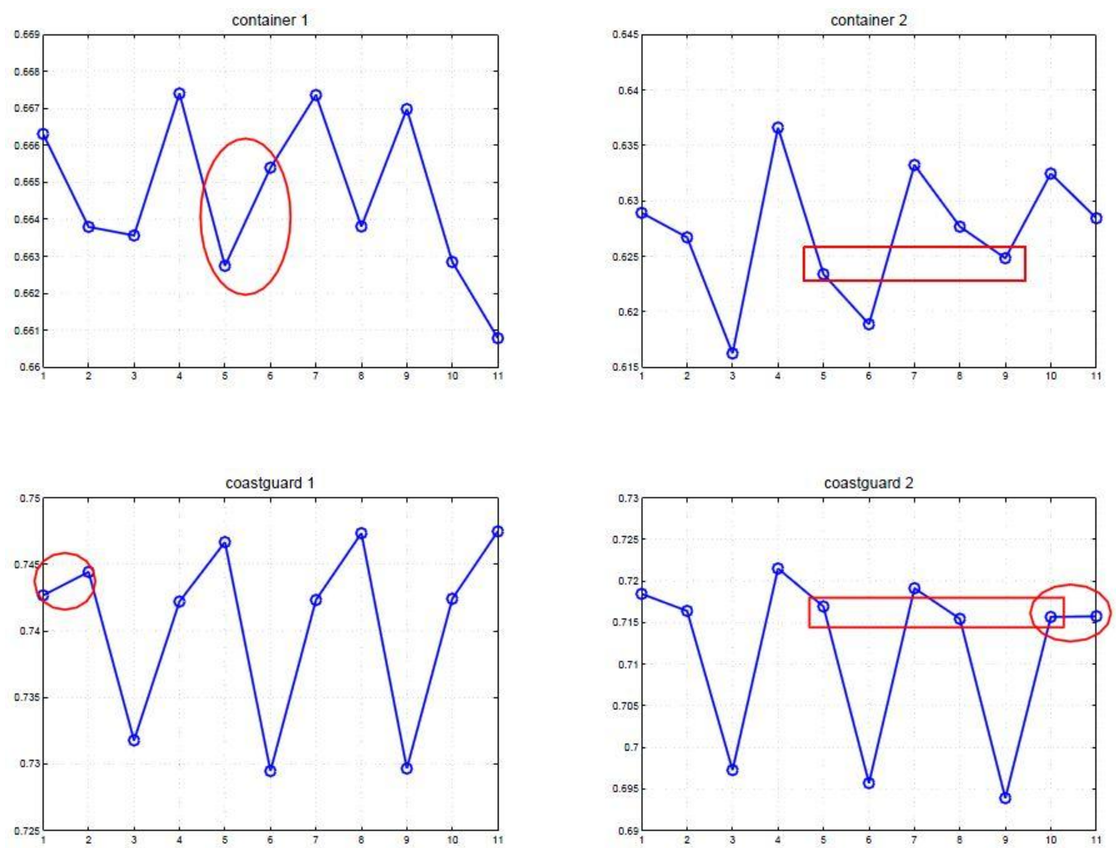


Figure 2-6 feature curve of tampered videos (different GOP conversions) [Luo et al., 2008].

As result the paper of [Luo et al., 2008] proves the reliability and the efficiency of the technique proposed for detection of tampered and recompressed MPEG video sequences.

2.3 MPEG 2 video Parameter and non-reference PSNR estimation

In this section we will explore the work of [Li and Forchhammer, 2009] where it emphasizes on estimating MPEG parameter information from decoded video stream without the access to MPEG stream. The techniques presented in [Li and Forchhammer, 2009] will be:

1) Detection of MPEG I-frames and DCT block size is presented
2) Quantization parameters for I-frames are estimated
3) PSNR is estimated from the decoded video without reference images

It is worth mentioning that all the distortion and artifacts originates from the DCT domain quantization. Moreover,“The distortion and the strength of the artifacts are correlated with the values of the quantization step sizes which are given by MPEG2 parameters quantization scale (Qs) and quantization matrix (Q_M)” [Li and Forchhammer, 2009].

Starting by brief background of MPEG-2 decoding:
First of all in MPEG2, the 16x16 pixel (luminance) Macro Block (MB) is the fundamental unit for processing that is divided into four 8x8 blocks that are transformed using DCT. Then the quantization of the transformed coefficients occurs that is controlled by one quantizer scale value Q_s per MB. The following figure 2-7 shows the scheme for the decoding process in MPEG2.

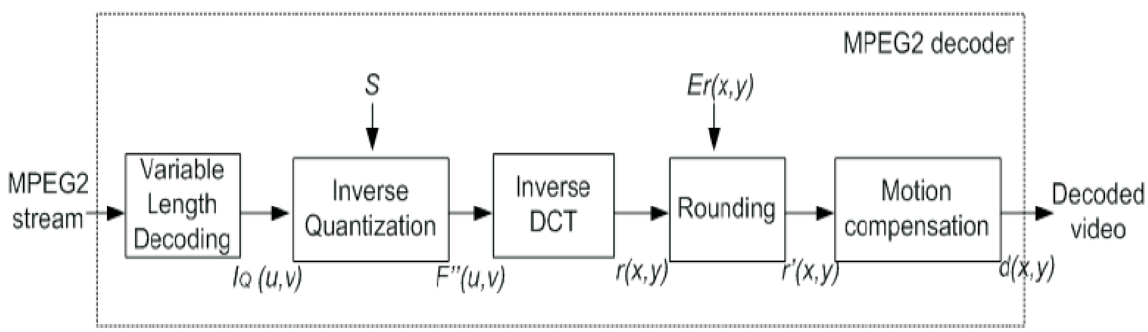


Figure 2-7 MPEG2 decoding process [Li and Forchhammer, 2009]

As we can see from the process above we start from variable length decoding and the output obtained is $I_Q(u,v)$ which is an integer value that represents the index of the quantization interval of DCT coefficient at (u,v). Then we pass to inverse quantization which will give as outcome $F^{''}(u,v)$

which is the DCT coefficients that are used for the reconstruction. For the intra MB we have:

$$|F''(u,v)| = \text{floor}\left(\frac{I_Q(u,v) \times Q_M(u,v) \times Q_s}{16}\right) \qquad (1)$$

$Q_M(u,v)$: Frequency dependent quantization matrix values

All the symbols or parameter in equation (1) are integers. Then after the reconstruction of F''(u,v), the inverse DCT will transform F''(u,v) to r(x,y) that by itself passes through rounding block and clipped to obtain r'(x,y) that is integer value in the range [0 255].Finally, we will get the output d(x,y) which is the decoded video with treating intra-blocks r'(x,y) with motion compensated data for the output d(x,y).

Estimation process:

The process provided below in the figure 2-8 aims into estimating and detecting three main things: DCT Block size and position, Quantization step size and I frames.

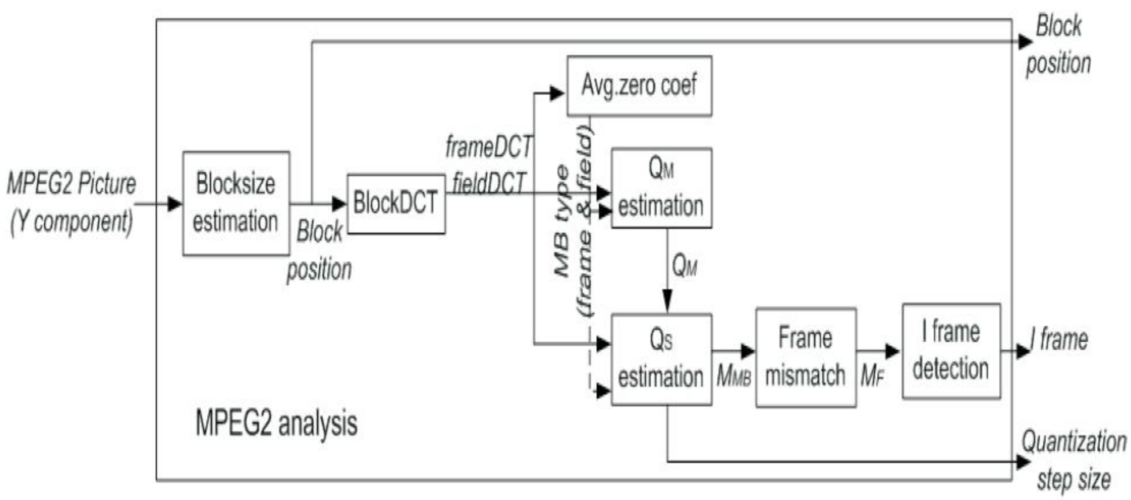


Figure 2-8 MPEG2 frame level analysis [Li and Forchhammer, 2009]

The procedure in [Li and Forchhammer, 2009] can be summarized as follows:

1) DCT Block boundary positions are estimated horizontally and vertically.

2) 8x8 block DCT is applied on each DCT block. In case the block size is not 8x8, DCT blocks should be rescaled into 8x8, after that DCT transformed.

3) Frame and field MB are estimated by taking minimum number of the zero DCT coefficients for MB.

4) Q_M and Q_s are estimated depending on reconstructed DCT coefficients F' (u,v). The quantization step sizes for each DCT coefficient are recovered based on Q_M and Q_s.

5) MB level mismatch (M_{MB}) is obtained

6) The frame level average of M_{MB} permits the calculation of frame level mismatch M_F

7) In the end, I-frames are detected by M_F

For the block size estimation, to perform a blocking artifact analysis, absolute difference between adjoining pixels is used. Even more, the calculation goes beyond this to calculate the differences of the differences and then project the values on the horizontal and vertical axis by summation. Using FFT to the projected values shows visible peaks related to block structure.

When it comes to quantization step size estimation, general case is considered where inttra/non-intra frame and MB type information is not known. First the method treats all MBs as intra. After processing the decision about intra or non-intra frames is validated. Delta is the intra frame quantization step size which is function of DCT frequency (u,v)

$$\Delta(u,v) = \frac{Q_S \times Q_M(u,v)}{16} \qquad (2)$$

It is essential for recovering Q_S and $Q_M(u,v)$ on I-frames is to apply DCT on 8x8 blocks of the decoded video r'(x,y) to recover DCT coefficients F'(u,v), and based on estimate of equation (1) for each DCT coefficient (u,v) where Q_M is fixed at frame level and Q_s at MB level [Li and Forchhammer, 2009]. In addition, having Q_M , $I_Q(u,v)$ x Q_S recovered then estimating Q_S using greatest common divisor approach. Some assumption should be taken into consideration like the distribution of the rounding error can be approximated by Laplace distribution. Therefore, Q_S becomes maximum likelihood problem. This results in this following equation or relation:

$$\min(|F'(u,v) - n\Delta(u,v)|), n > 0 \qquad (3)$$

The argument of equation (3) explains the distance between reconstructed DCT value $F'(u,v)$ and the nearest possible reconstructed value of the MPEG2 decoder.

At this point we reach to Q_M matrix estimation where all AC coefficients of given frequency (u,v) are quantized using the same Q_M. Since we have four DCT blocks within the same MB, they have the same Q_S. After perfomring testing, the Q_S having the smallest MB mismatch value, M_{MB} is selected:

$$Q_M = \hat{Q}_M(p) : \arg \min_{p\varepsilon\{Q_M\}} \sum_{MB} \min_{q\varepsilon\{Q_S\}} (M_{MB}(p,q)) ,$$

Where $\mathrm{MMB}(Q_M, Q_S) = \sum_{(u,v)\in \mathrm{MB}} |\,\mathrm{round}\,(\frac{F'(u,v)\times 16}{Q_M(u,v)\times Q_S}) - \frac{F'(u,v)\times 16}{Q_M(u,v)\times Q_S}\,|$

Modifying the log-likelihood by normalizing by the quantization step to make the mismatch independent of Q_S and Q_M.

After this we continue to Q_S quantizer scale estimation and we have that Q_S is unique for every MB. The decoded DCT coefficients within one MB will be distributed on integer multiples of the quantization step size [Turaga,Chen,Caviedes,2004].

Assumptions and observation for the algorithm proposed:

1) Set Q_S for all possible values are given by MPEG2 Q_S table

2) For single MB, Q_S upper bound Q_S^{up} can be calculated from $\min((I_Q \times Q_S)_{up})$ where min is over frequencies (u,v) for all non-zero DCT coefficients.

3) MPEG2 has bias towards maintaining same Q_S as previous MB. Hence previous Q_S can be used for estimation.

Estimation Algorithm:

The Q_S algorithm For each MB do

1) For all AC DCT values $F'(u,v)$ for (u,v) $\neq 0$ within current MB, calculate $F_{QS}(u,v) = F'(u,v) \times \frac{16}{Q_M}$, $(|\,I_Q(u,v)\,| \times Q_S)_{up}$ and $(|\,I_Q(u,v)\,| \times Q_S)_{down}$

2) Round F_{QS} (u,v) to the nearest even integer value K(u,v)

3) set all K(u,v) less than 4 to zero (All DC values are set to 0)

4) Calculate Q_S upper bound Q_S^{up} by $\min((I_Q \times Q_S)_{up})$ (min is over non-zero DCT coefficients)

5) For j $j \in \{Q_S\}$ and $4 \le j \le Q_S^{up}$:

$$\hat{Q}_S = \arg\max_j [N_1(j) + N_2(j)]$$

Where $N_1(Q_S)$ is the number of DCT coefficients for which

$K(u,v) = Q_S$ and $N_2(Q_S)$ is the number of $K(u,v)$ which are divisible by Q_S

6) For MBs, which do not contain any non-zero AC coefficients, the steps above do not provide a result.Instead the estimated $\hat{Q}_S$ value from the previous MB is used for the current MB **[Li and Forchhammer, 2009]**

Validation and I frame detection:

The method proposed in **[Li and Forchhammer, 2009]** for I frame detection at frame level can be also used for adaptive GOP structures. The mismatch measure M_F is introduced to measure the accuracy of the estimated step size

$$M_F = \sum_{MB} M_{MB}(\hat{Q}_M, \hat{Q}_S)$$

Small values of M_F will be obtained for correctly estimated Q_S of the frame.High percentage of Q_S values are estimated correctly thought the mismatch is due to rounding error. The other frame types will have wrong quantization step size estimate because of incorrect contribution to DCT coefficients from motion compensated contributions, hence a threshold is applied to M_F.

PSNR Estimation:

The estimation of PSNR of I frames can be done depending on the predicted value of quantization step size. Methods were proposed by [3] for non-reference PSNR estimation for MPEG2 using fixed Q_S. Moreover, in the same paper [3], the DC and AC overall mean square error distortion was calculated. The variables or parameters that play role in the estimation are Δ, α and λ where Δ is the quantization step size, α is the shift factor in MPEG-2 quantization scheme and λ is the Laplacian parameter for each DCT coefficient. Now what is important is the estimate of λ that is realized in the method in [7] where the estimation depends on the number of zero coefficients that provider simpler and faster estimation. The formula that governs the relation between λ and other parameters is: $\lambda(Q_S, Q_M) = -\dfrac{\frac{\Delta}{2} + \alpha}{\ln(1 - p_0(Q_S, Q_M))}$

$p_0(Q_S, Q_M)$ is the ratio number of zero coefficients for all the coefficients quantized by Q_S and Q_M

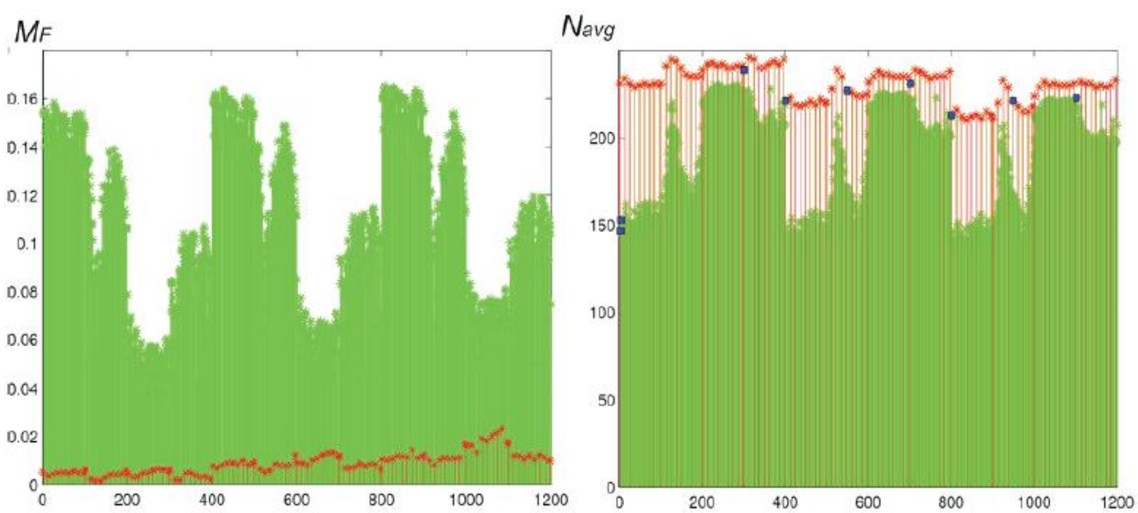
$p_0 = \frac{N_0}{N}$; N_0 is estimated number of zero coefficients estimated by the number of reconstructed values in the interval $[-\frac{\Delta}{2} - \alpha, \frac{\Delta}{2} + \alpha]$

N is total number of coefficients

The testing of the methods proposed was done on 12 sequences and the results confirmed the accuracy and the efficiency of the proposed methods.Sample of the results are shown below in table 2-5 and figure 2-9:

	CITY2	SOC 2	ICE 2	CREW2	CITY3	SOC 3	ICE 3	CREW3	CITY4	SOC 4	ICE 4	CREW4
avg Q_S	24.70 (24.70)	29.29 (29.19)	11.11 (11.21)	21.53 (21.35)	16.67 (16.67)	18.02 (18.04)	7.50 (7.34)	12.79 (12.77)	13.12 (13.12)	13.90 (13.92)	6.24 (5.69)	10.00 (10.00)
PSNR	31.76	31.52	39.82	35.91	33.96	33.96	41.56	38.01	35.38	35.47	42.75	39.08
M1	31.35 (31.36)	30.52 (30.55)	38.25 (38.26)	33.26 (33.39)	34.59 (34.59)	34.19 (34.19)	41.48 (41.63)	37.12 (37.17)	36.62 (36.62)	36.31 (36.31)	43.12 (43.78)	39.11 (39.13)
M2	31.88 (31.88)	32.09 (32.10)	37.95 (37.95)	35.32 (35.36)	33.95 (33.95)	34.28 (34.27)	40.07 (40.17)	37.57 (37.57)	35.24 (35.34)	35.68 (35.67)	41.19 (41.66)	38.81 (38.80)

2-5 I frame, Q_S and PSNR estimation results ($\lambda(Q_S, Q_M)$) [Li and Forchhammer, 2009]



2-9 I frame detection results for frames of the concatenated sequence [Li and Forchhammer, 2009]

From the left, detection results by M_F and from the right is by N_{avg} the average number of zero coefficients for each MB

2.4 Blind estimation of Q_P parameter in H.264/AVC decoded video

The estimation of Q_P in H.264/AVC encoding is proposed in the work of [Tagliasacchi, Tubaro,2010] where in this paper they prove and show the possibility for reverse engineering of the decoded video sequences to reveal part of its coding history. The benefits from the proposed algorithm can be realized or categorized in the video quality assessment and an outlook for video forensics tools like detecting temporal cropping and merging processes. What is different in this work from others is the scenario that handles or investigates in which we only have the accessibility for the pixel values of the decoded video.From here comes the assumption that each encoding process will leave footprints that can be detected to find out the encoding process the video sequence passed through.Moreover, we will see two methods for the estimation of quantization parameters in H.264/AVC encoding that account for spatial and temporal prediction tools enabled by the standard.

Brief Background

We start by the analysis of the transformed prediction residuals in H.264/AVC. Referring to 4x4 transform in baseline, extended and main profile where it also applied on 8x8 transform for high profile. Consider ***E*** as 4x4 block transform, the DCT approximation transform that is given by H.264/AVC is governed by this formula

$Y = Z \circ S$ and $Z = TET^T$

T and S are defined in [12] and $\circ$ denoted element multiplication.

T comprises transform operation that consists of add and shift operations where S is post pre-scaling operation.The quantization coefficient Y_j is given by:

$$\hat{Y}_j = I_j \times q = sign(Y_j)[\frac{|Y_j|}{q} + 1 - \alpha] \times q$$

q: quantization step

α: parameter that controls width of dead zone around 0

I_j : quantization index that is transmitted actually

The quantization parameter Q_p is defined through the computation of q:

$$q = q_B(\mathrm{mod}(Q_p, 6))2^{[Q_p/6]}$$

q_B is defined in the following table 2-6:

$mod(QP,6)$	q_B
0	0.625
1	0.6875
2	0.8125
3	0.8750
4	1.0000
5	1.125

2-6 Base quantization steps[Tagliasacchi, Tubaro,2010]

Now we come to the distribution of quantized coefficients that is modeled by this equation:

$$p(\hat{Y};q) = \sum_k w_k \delta(\tilde{Y} - kq) \qquad [11]$$

Applying the same transform and quantization process at the decoder we can model the result using this distribution:

$$p(\hat{Y};q) = \sum_k w_k N(\tilde{Y}, kq, \sigma^2)$$

$\sum_k N(\tilde{Y}, kq, \sigma^2)$: Gaussian probability density function with mean μ and variance σ^2

Algorithm:

We consider the following assumptions:

1)The sequence is coded by H.264/AVC (baseline,extended, main profile)

2) All macro-blocks share same QP

The algorithm is described for P slices and can be applied for I and B slices.

Algorithm in **[Tagliasacchi, Tubaro,2010]**:

For each frame:

1) Perform motion estimation to compute motion vectors for each 4x4 block. Any motion estimation algorithm can be used for this purpose. Let (mvx^i,mvy^i) denote the motion vector of i^{th} 4x4 block.

2) Compute the motion compensated prediction residuals for each 4x4 block. Let $\hat{X}^i$ represents the 4x4 block in pixel domain and $\hat{E}^i$ its prediction residual.

3) Discard blocks that satisfy this condition

$$\sum_{x=1}^{4}\sum_{y=1}^{4} |\hat{X}^i(x,y)|^2 < \sum_{x=1}^{4}\sum_{y=1}^{4} |\hat{E}^i(x,y)|^2$$

This retain blocks that are likely to be inter-predicted

4) Transform the prediction residuals $\hat{E}^i$ according to (1) to obtain $\hat{Y}^i$

5) Collect the transformed prediction residuals from all retained blocks and estimated QP :

$$\hat{QP} = \arg\max_{QP} \sum_{j=1}^{N} \log p(\hat{Y}_j; q)$$

q is the quantization step corresponding to QP according to (3)

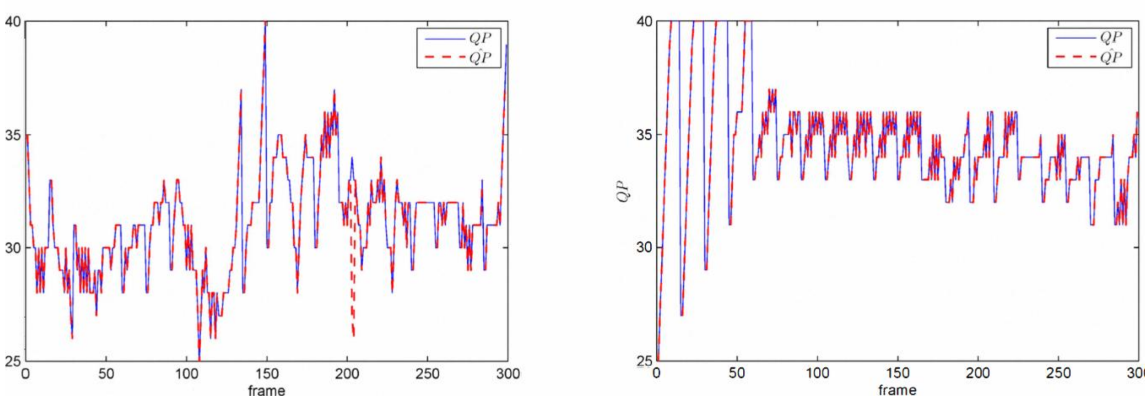
in practical encoding scenario the expression in (7) simplifies to :

$$\hat{QP} = \arg\max_{QP} \sum_{j=1}^{N} (\min_k (\hat{Y}_j - kq)^2 - \log(w_k))$$

$$\hat{k} = \arg\min_k (\hat{Y}_j - kq)^2$$

Therefore each quantized transform coefficient we determine distance to closest quantization reconstruction level kq and select QP that minimizes sum of distances over all coefficients **[Tagliasacchi, Tubaro,2010].**

After the testing of the algorithm, sample of the results represented in figure 2-10 that shows the actual and estimated QP on a frame by frame basis of two encoded video sequence foreman at 500kbps and mobile at 1000kbps.



2-10 Estimated $\hat{QP}$ for foreman and mobile video sequences [Tagliasacchi, Tubaro,2010].

This concludes the algorithm and method proposed by **[Tagliasacchi, Tubaro, 2010]** but it continues with the above technique in [**Tagliasachi, Valesize, Tubaro,2010]** where the proposed algorithm to estimate QP will be used for the motion vector refinement using QP consistency constraints.

Motion vector refinement using QP consistency constraints:

The continuity of the above procedure can be explored by exploiting the algorithm in refining motion vectors estimated at the decoder. We start by assuming that QP is constant over the whole frame and $\hat{QP}$ is the estimated QP for a given frame.

The method in [**Tagliasachi, Valesize, Tubaro,2010]** is as follows:

1) The mismatch of the current prediction residual obtained with mv_i with respect to $\hat{QP}$ is computed by evaluating the cost function:

$$C(mv_i) = \sum_{j=1}^{N} (\min_k (\hat{Y}_j(mv_i) - k\hat{q})^2 - \log(w_k))$$

$\hat{Y}_j(mv_i)$: transformed prediction residuals

$\hat{q}$: quantization step corresponding to $\hat{QP}$

If C=0, there is no need for refinement and the search is stopped.

2) Let WxW to be the size of a search window centered around mv_i. Start to evaluate the mismatch with the expected coefficient distribution imposed by $\hat{QP}$ for each motion vector in the search using the objective function above. The refined motion vector mv_i is the one that minimizes the model mismatch (10) and gives the maximum likelihood predictor of that block according to (5). Select the refined motion vector which results lowest SAD of prediction residual [**Tagliasachi, Valesize, Tubaro,2010]**.

One note should be considered that full or fast search could be adopted to find the refined motion vector.

Results obtained for testing the algorithm is shown in table 2-7 below:

		R [kbps]	PSNR [dB]	*MeanAE*	*MaxAE*	*WrongQP*	*ImprPred*	*OrigPred*	*Improv*
Hall monitor	L	125	31.58	0.121	7	4.3%	68%	66%	62%
	M	250	35.37	0.007	2	0.3%	43%	41%	83%
	H	500	37.89	2	2	0.7%	39%	36%	91%
Foreman	L	250	31.27	0.301	11	6.8%	70%	60%	84%
	M	500	34.85	0.05	7	1.0%	49%	44%	87%
	H	750	36.88	0.065	2	4.6%	49%	47%	92%
Mobile	L	500	26.49	0.021	6	0.3%	43%	41%	96%
	M	750	28.28	0	0	0.0%	39%	37%	96%
	H	1000	29.79	0	0	0.0%	39%	37%	96%

2-7 QP estimation and motion vector refinement results of three video sequences [Tagliasachi, Valesize, Tubaro,2010]

The results show and confirm the efficiency of the algorithm proposed where the accuracy of the results increase or improve at higher bit rates and decrease if deblocking filter is enabled. And after exploring methods and algorithms from various encoding type we conclude chapter 2 to start speaking about our technique in chapter 3.

Conclusion

Summary

References