

RESEARCH

Open Access

An R-D optimized transcoding resilient motion vector selection

Alireza Aminlou^{1*}, Mehdi Semsarzadeh² and Omid Fatemi³

Abstract

Selection of motion vector (MV) has a significant impact on the quality of an encoded, and particularly a transcoded video, in terms of rate-distortion (R-D) performance. The conventional motion estimation process, in most existing video encoders, ignores the rate of residuals by utilizing rate and distortion of motion compensation step. This approach implies that the selected MV depends on the quantization parameter. Hence, the same MV that has been selected for high bit rate compression may not be suitable for low bit rate ones when transcoding the video with motion information reuse technique, resulting in R-D performance degradation. In this paper, we propose an R-D optimized motion selection criterion that takes into account the effect of residual rate in MV selection process. Based on the proposed criterion, a new two-piece Lagrange multiplier selection is introduced for motion estimation process. Analytical evaluations indicate that our proposed scheme results in MVs that are less sensitive to changes in bit rate or quantization parameter. As a result, MVs in the encoded bitstream may be used even after the encoded sequence has been transcoded to a lower bit rate one using re-quantization. Simulation results indicate that the proposed technique improves the quality performance of coding and transcoding without any computational overhead.

Keywords: Motion estimation; Rate-distortion model; Rate-distortion optimization; H.264/AVC; Lagrange multiplier; Video transcoding

1 Introduction

In ubiquitous multimedia access, the same content is transferred between different users with various devices and applications; each has different capabilities in their resources such as network bandwidth, display resolution, frame rate, and decoding bit rate. Therefore, there is an urgent need to use adaptation techniques to support moving from one specification to another. One of such technique is SNR adaptation or bit rate reduction which targets network bandwidth and/or decoding bit rate limitations of different users.

Scalable video coding (SVC) is a viable solution to address the content adaptation. However, as further discussed in Subsection 2.3, deployment of SVC decoders has been limited in practical applications. On the other hand, transcoding is a suitable alternative to accommodate users' requirements by modifying the compressed

bitstream. The most straightforward realization of a transcoding architecture is to cascade a decoder and an encoder. But cascading scheme imposes a considerable computational complexity. Transcoding is normally performed at the intermediate network nodes where there is usually a limited computational power; hence, some methods were proposed in the literature to meet the computational constraint. One of such method for bit rate reduction is typically implemented by simple re-quantization [1]. Among existing methods for re-quantization [1,2], the cascaded pixel-domain transcoding (CPDT) that cascades a decoder and a simplified encoder is computationally effective and has no drift problem.

In CPDT architecture, considering the significant computational complexity of motion estimation (ME) and mode decision (MD) processes [3], the cascaded encoder is simplified to further reduce the transcoding complexity. This simplification is realized by either performing a fast ME and mode refinement [4,5], or skipping ME and MD altogether by reusing the original motion vectors (MV) and modes. Hence, as shown in Figure 1, MVs and

* Correspondence: a.aminlou@ece.ut.ac.ir

¹School of Electrical and Computer Engineering, University of Tehran, North Kargar Street, Tehran, P.O. Box: 14395-515, Iran

Full list of author information is available at the end of the article

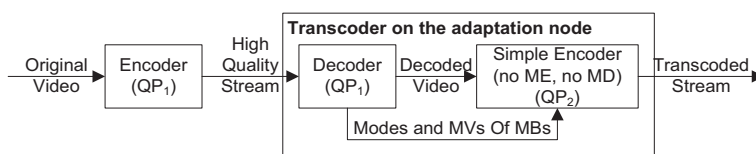


Figure 1 CPDT transcoding architecture for re-quantization with MV and mode reuse.

modes are extracted from the high quality bitstream and reused in the transcoded bitstream. However, new residuals are calculated by motion compensation, re-quantized by new quantization parameter (QP_2), and then re-encoded into the transcoded bitstream. Hence, in this CPDT architecture, the rate-distortion (R-D) performance of the transcoded bitstream is significantly affected by MVs in the original bitstream.

Motion vectors in most practical encoders are selected by conventional motion estimation methods, using simplified Lagrangian cost function [6]. This in turn may have a negative effect on the performance of video coding. Furthermore, the selected MV is dependent of QP; hence, the selected MV for high bit rate compression is not suitable for low bit rate ones. This feature has a negative impact on the quality performance when a video bitstream is transcoded using CPDT with MV and mode reuse technique. The quality degradation is more critical when simplified mode decision methods are used in the encoder side to achieve low complexity encoding.

The joint multi-rate optimization method [7] is the most recent R-D optimized encoder that is designed to be used with CPDT transcoding system. In this method, the video is coded with three different QP values at the same time, and the best modes are selected based on a joint optimization. Although this method improves the quality performance of the transcoded bitstream, it suffers from high computational overhead at the encoder side.

In this paper, we propose a new R-D optimized criterion to select MVs that are suitable for both high and low bit rate compressions. We firstly propose a two-piece model for the R-D curve of a block. Then, based on two-piece R-D model, a two-piece Lagrange multiplier selection is introduced for motion estimation process. We show that our proposed scheme results in motion vectors that are less sensitive to changes in bit rate or QP. As a result, MVs which are selected for high bit rate bitstream may be reused even after the bitstream has been transcoded. Compared to the conventional motion estimation method, the proposed motion estimation scheme not only improves the quality of the encoder but also results in a far better R-D performance when the bitstream is transcoded using CPDT with MV reuse technique.

The rest of the paper is organized as follows: Section 2 reviews the background on rate-distortion optimization,

the existing R-D models, and the SVC deployment problems. The proposed two-piece R-D model is presented in Section 3. Section 4 describes the proposed motion estimation method and two-piece Lagrange multiplier selection. The application of the proposed motion estimation in a transcoding system is presented in Section 5. The simulation results for R-D performance and computational complexity are reported in Section 6, followed by concluding remarks in Section 7.

2 Background and literature review

In this section, we first summarize the R-D optimization tool in a video encoder and exiting R-D models. Then, we highlight the limitations of the SVC deployment in real applications.

2.1 Rate-distortion optimization

A video encoder should optimally determine the coding parameters for each block in order to generate an R-D optimized bitstream. These coding parameters (P_B) mainly include QP, prediction mode, and MV, which are optimally selected for each block or macroblock by minimizing the Lagrangian cost function [8]. In a basic method, the optimization is performed at the block level by checking all the combinations of all the coding parameters using (1).

$$P_B = \arg \min \{J_B = D_B(P_B) + \lambda R_B(P_B)\} \quad (1)$$

where R_B , D_B , and P_B are the rate, distortion, and coding parameters of a block, respectively [9]. The parameter of λ is Lagrange multiplier. But there is a large number of combinations of parameters (mode and MV) that should be considered. Although the R-D models, presented in Subsection 2.2 below, can be used for fast optimization, in a typical encoder such as the H.264/AVC [10,11], the selection of QP, MV, and mode is performed separately in three consecutive stages of rate control (RC), motion estimation, and mode decision, respectively.

The residual coding process and coding a block for all MV candidates is still computationally intensive [9]; hence, practical motion estimation methods ignore the residual coding and determine the best MV using the available data during motion estimation process [12,13]. The early motion estimation techniques only minimized the motion compensation distortion using a block matching algorithm [12]. The more recent motion estimation algorithms, such

as the one adopted in the joint model (JM) of the H.264/AVC video encoder, are based on the methods proposed in [6] and [13]. In this method, MV is selected by minimizing the motion compensation cost, as defined in (2),

$$MV = \arg \min_{mv \in SA} \{J_{MC} = D_{MC}(mv) + \lambda R_{MC}(mv)\} \quad (2)$$

where R_{MC} and D_{MC} are the number of bits required to code the motion vector and the distortion of motion compensation, respectively, λ is the Lagrange multiplier which is determined based on QP, and SA is the search area. It should be noted that R_{MC} includes the number of bits to code the motion vector difference, the difference between the actual motion vector (mv) and motion vector predictor and the reference frame index. But only mv is presented in the notation to shorten the abbreviation.

According to (2), the selected MV is dependent on λ (and QP). Particularly, when λ (and corresponding QP) is small, a bigger MV with a larger number of bits (R_{MC}) is selected. On the other hand, when λ (and the corresponding QP) is big, a smaller MV is selected.

2.2 Existing R-D models

Several models have been proposed in the literature in order to estimate the R-D curve of a block. These models are used either in rate control [14-25], or in mode decision [26-35], or as a study for the behavior of R-D curves [36-40]. In [14,15,22,23,34,36,39], and [40], various distribution models have been used to model different types of data such as residuals, DCT coefficients, quantized coefficients in order to analytically extract R-D curves. While in [16-21,24-33,35,37], and [38], rate and distortion are empirically estimated based on the different parameters like the number of zero coefficients or tokens of the context adaptive variable length coding (CAVLC).

These models estimate rate or distortion based on residual, DCT, quantized, or scanned data, which requires a large amount of computation when used in motion estimation process as they should be evaluated for all motion vector candidates. Furthermore, all the existing R-D models have been proposed for the residual coding part, without considering the motion compensation process. In the following sections, we propose a two-piece R-D model which considers both the prediction and residual coding processes.

2.3 SVC limitations

SVC suffers from several shortcomings in practical applications; the most important are mentioned next. There are limited practical software or hardware implementations of SVC encoder and decoder for real-time applications. SVC decoder also requires more decoding power

which is more critical in mobile applications. Not only the bit rate changes are coarse and limited to the number of the layers but also the bit rate cannot be lower than the bit rate of the base layer. It should be noted that the bit rate of the base layer cannot be very small because of its negative effect on the R-D performance of enhancement layers. Finally, compared to single layer coding, SVC imposes about 10% bit rate overhead for the enhancement layer [41]. Because of these problems, transrating is a suitable alternative to accommodate each user's requirements in a ubiquitous multimedia by modifying a compressed bitstream.

3 The proposed two-piece R-D model

We analyze the R-D optimization in a video encoding process in two steps to extract the new optimization criterion for motion vector selection. For this purpose, we first defined motion compensation and residual coding R-D curves for a block. Then, using a simple model for R-D curve of residuals, we analyze the R-D optimization within a block. As a result of this step which is referred to as local optimization, we propose a new two-piece R-D model for the block. This model is later used in Section 4 to analyze the R-D optimization among different blocks in the video sequence. The result of this step, referred to as global optimization, is the new motion estimation criterion and the two-piece Lagrange multiplier selection.

3.1 R-D curve of motion compensation

During motion estimation, a search area with a typical size of $(2s + 1) \times (2s + 1)$ is considered, where s is the search range. Each mv candidate is coded with $R_{MC}(mv)$ bits, and motion compensation with this mv results in a prediction distortion value of $D_{MC}(mv)$. This distortion can be represented as the energy of residuals, for example, with sum of square difference (SSD), sum of absolute difference (SAD), or sum of absolute transformed difference (SATD) metrics. Figure 2 shows all the R-D points of the motion search for one block. Each point on the convex curve may be selected as the best MV based on the value of the Lagrange multiplier. Hence, we define them as the motion compensation R-D curve for that block. It should be noted that the motion compensation R-D curve of a block is independent of its corresponding residual coding process. The motion compensation R-D curves of few other blocks are illustrated in Figure 3. In this figure, the bold part (B0_MC and B1_MC) represents the motion compensation R-D curve of the corresponding block.

3.2 R-D curve of residual coding

A typical residual coder, such as the one used in the H.264/AVC, transforms the residual values, quantizes by a QP, and then uses an entropy coder. By applying different

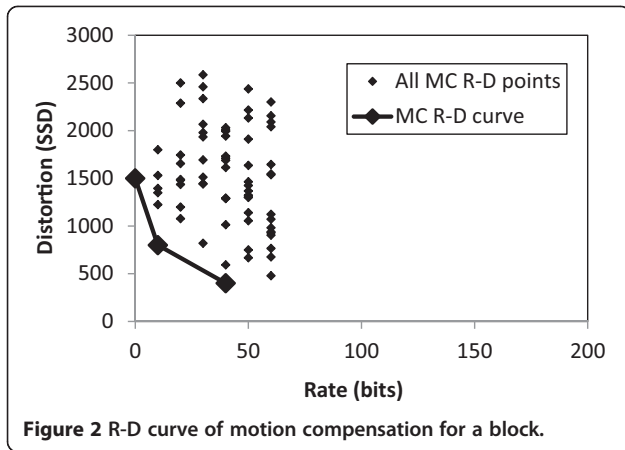


Figure 2 R-D curve of motion compensation for a block.

values of QP, several R-D points are obtained which form the R-D curve of residual coding. The residual R-D curve can be represented by an exponential form [6] with a time constant of TC, as expressed in (3),

$$D_{Res} = D_0 \exp\left(-\frac{R_{Res}}{TC}\right) \quad (3)$$

where R_{Res} is the number of bits representing the quantized residuals, D_{Res} is the corresponding quantization distortion which equals to the distortion between the original and the reconstructed residuals [42], and D_0 is the energy of residuals. As described in Subsection 4.3, the same model is also valid for SAD and SATD but with a doubled TC.

Several examples of residual R-D curve are shown in Figure 3 for two blocks (B0 and B1). Each curve corresponds to a different MV of a certain block. For example, B0_Res2 is the residual R-D curve for B0 when its second MV is selected.

As it will be described in Subsection 4.3, we have assumed that TC is only related to the size of the block and is independent of the residual values. Hence, the

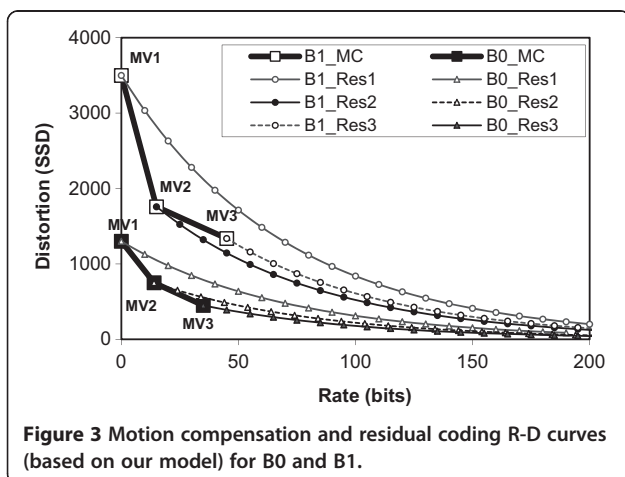


Figure 3 Motion compensation and residual coding R-D curves (based on our model) for B0 and B1.

second part of the model of (3), representing the residual coding behavior of that block by an exponential function, is independent of the motion compensation process of that block. The first part of the model of (3) reflects the motion compensation process of the block which is independent of residual coding of that block as well, because D_0 is related only to the original pixels values of the current block and the reconstructed pixels values of the reference frame.

3.3 R-D curve of a block

A block rate (R_B) is the sum of the rates of MVs and residuals as presented in (4). The rate of prediction mode has not been considered in (4), since it is fixed during ME step of each block. The block distortion (D_B) is the distortion between the original and the reconstructed block. It can be easily shown that the block distortion (D_B) is equal to the corresponding distortion of the residuals (D_{Res}) as presented in (5).

$$R_B = R_{MC} + R_{Res} \quad (4)$$

$$D_B = D_{Res} = D_0 \exp\left(-\frac{R_{Res}}{TC}\right) \quad (5)$$

We recall that we have used the exponential model of (3) for the R-D curve of residuals. Setting R_{Res} to zero (e.g., by quantizing all residual values to zero) in (5), D_B equals D_0 . In this case, on the other hand, block distortion equals to the energy of the residual, in other words $D_{MC}(mv)$. Hence, D_0 equals $D_{MC}(mv)$, and we express the distortion of the block as (6). This model separates the effects of motion compensation and residual coding processes on the distortion of the block.

$$D_B(mv, R_{Res}) = D_{MC}(mv) \exp\left(-\frac{R_{Res}}{TC}\right) \quad (6)$$

3.4 Two-piece model for R-D curve of a block

We formulate the formation of a block R-D curve as a constrained optimization problem for that block as presented in (7). This formulation expresses that for any value of R_B , MV, and R_{Res} should be optimally determined to minimize the distortion value of the block. As mentioned before, this step is referred to as local optimization, because it is the optimum bit allocation between the motion compensation (R_{MC}) and residual coding (R_{Res}) processes of each block.

$$\begin{cases} R_{MC}(mv) + R_{Res} \leq R_B \\ D_B(mv, R_{Res}) = \min \end{cases} \quad (7)$$

MV for the block can be found by minimizing (6), subject to the constraint on R_B . We minimize (6) in logarithmic domain, as presented in (8). Using (4) in (8), and

according to the derivation in (9), the initial MV is determined by (10) which minimize D_B subject to the constraint on R_B . We refer to this MV as local MV (LMV) as it is selected based on the local optimization within the block.

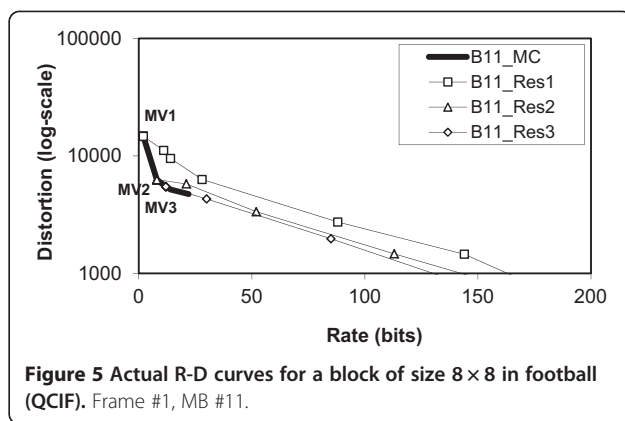
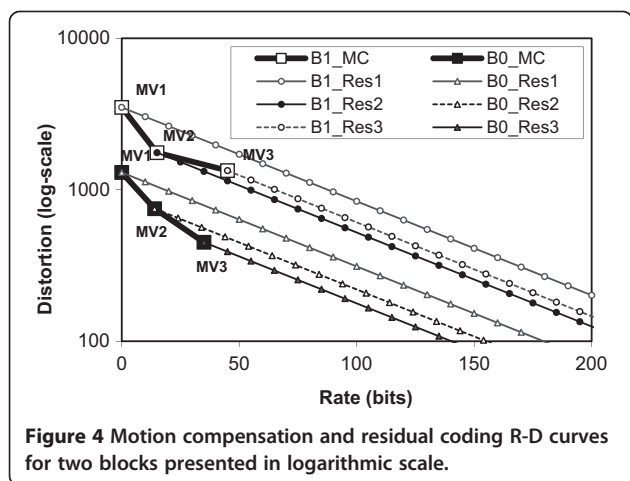
$$\ln(D_B) = \ln(D_{MC}(mv)) - \frac{R_{Res}}{TC} \quad (8)$$

$$\ln(D_B) = \left(\ln(D_{MC}(mv)) + \frac{R_{MC}}{TC} \right) - \frac{R_B}{TC} \quad (9)$$

$$LMV = \arg \min_{mv \in SA} \left\{ \ln(D_{MC}(mv)) + \frac{R_{MC}(mv)}{TC} \right\} \quad (10)$$

Figure 4 displays a graphical representation for the cost function in (10). In this figure, the distortion axis is in logarithmic scale which corresponds to the logarithm function in (10). In this graphical representation, residual R-D curves, modeled as exponential form, are seen as several lines with the equal slopes of $-1/TC$, where TC is the time constant of the R-D curves which is used as a parameter in the model of (3). Therefore, different residual R-D curves of a block are parallel lines, and do not cross each other. Hence, according to (10), the residual R-D curve corresponding to LMV will definitely fall below all remaining residual R-D curves of that block. This implies once again that LMV is the best MV for the block in all R_{Res} values.

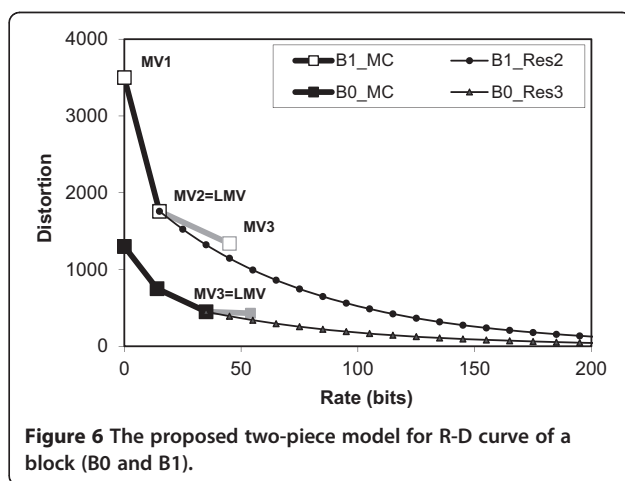
As shown in Figure 4, LMV of B0 is the third motion vector (MV3) and B0_Res3 is its corresponding residual R-D curve, falling below other residual curves (B0_Res1 and B0_Res2). But for block B1 as another example, LMV is second motion vector (MV2). Hence, B1_Res2 falls below the other R-D curves of B1. In Figure 5, an actual R-D curve of a block of size 8×8 is presented for football (quarter common intermediate format (QCIF)) which is quite similar to our model in Figure 4.



As a conclusion, the R-D curve of each block can be formed based on a two-piece model. As shown in Figure 6, the first piece corresponds to the R-D curves of motion compensation of that block up to LMV. From that point on, the block R-D curve continues on the residual R-D curve corresponding to this LMV. As described in the next section, we have extended the use of this two-piece model to the bit allocation between different blocks of a frame to derive a new optimization condition for motion vector selection.

4 The proposed motion estimation method

To develop a new optimization condition for motion estimation, we analyze the R-D optimization when blocks of a video are coded using inter prediction. We first analyzed the optimization within each block in the previous section which resulted in a local motion vector (i.e., LMV) and a two-piece R-D model for each block. In this section, we analyze the optimization among different blocks in a video. We refer to this step as global bit allocation.



4.1 The proposed optimization condition for MV selection

To determine the optimum point for each block, we should consider the optimization among all the blocks in a video. The optimum point for each block is found by applying the Lagrange theorem of (1) to the two-piece R-D model of each block. Based on the value of λ , the optimum point may be either on the motion compensation piece or on the residual coding piece. As shown in Figure 7 for B0, when the value of λ is smaller than a threshold ($\lambda < \lambda_{Thr}$), the optimum point falls on the residual R-D curve. In this case, LMV is selected as the best MV for this block.

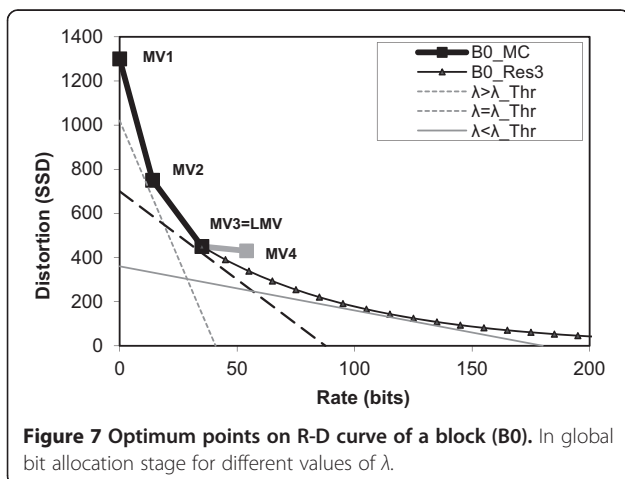
In the other case ($\lambda > \lambda_{Thr}$), the optimum point falls on the motion compensation R-D curve, and consequently, for the example, shown in Figure 7, MV2 is selected as the best MV for this block. We refer to this MV as global MV (GMV), hereafter, because it is found in global bit allocation step. GMV is found by applying the Lagrange theorem of (1) to the motion compensation R-D curve of the block as presented in (11).

$$GMV = \arg \min_{mv \in SA} \{D_{MC}(mv) + \lambda R_{MC}(mv)\} \quad (11)$$

In this case, as shown in Figure 7, we noted that the rate of the GMV of the block is smaller than that of LMV. This means that the GMV selected by (11) is the final MV of the block only when its rate is smaller than that of LMV. On the contrary, when its rate is larger than that of LMV, GMV is not the final MV of the block and instead LMV is selected as the final MV for the block. Consequently, we select the final MV of the block as the minimum of LMV and GMV using (12).

$$MV = \begin{cases} LMV & R_{MC}(LMV) < R_{MC}(GMV) \\ GMV & R_{MC}(LMV) > R_{MC}(GMV) \end{cases} \quad (12)$$

The proposed motion estimation algorithm that finds the optimum MV for a block with the proposed optimization condition can be summarized as in Figure 8. In this



algorithm, by checking all mv candidates in the search area, LMV and GMV are found using (10) and (11), respectively. Then, the MV of the block is selected using (12).

4.2 The proposed two-piece Lagrange multiplier

The proposed motion estimation should find LMV for each block using (10), where logarithm function should be evaluated for each motion vector candidate. Based on the simulation results, the execution time of the encoder is increased by about 20% to 30% for different configurations. Based on the proposed two-piece model, we introduce a new two-piece-based Lagrange multiplier for motion estimation process to eliminate the overhead of the computational complexity. The aim is to introduce a Lagrange multiplier that when used with the conventional motion estimation algorithm selects the same motion vector that is selected by the proposed optimization condition.

According to the proposed criterion in (12), the final MV of a block should not go beyond the LMV of that block. On the other hand, as show in Figure 7, LMV of a block is also selected when the value of the Lagrange multiplier is equal to λ_{Thr} . The value of λ_{Thr} can be found for each block using the negative slope of residual R-D curve that corresponds to LMV, in other words, when the residual rate is zero. Based on the model in (5) and (6) for the residual coding curve, λ_{Thr} is calculated using (13)

$$\lambda_{Thr} = - \left. \frac{\partial D_{Res}}{\partial R_{Res}} \right|_{\substack{MV = LMV \\ R_{Res} = 0}} = \frac{D_{MC}(LMV)}{TC} \quad (13)$$

It should be noted that LMV of the current block is found by the motion search process, hence LMV and consequently $D_{MC}(LMV)$ in (13) is not available yet. In this research, we proposed to estimate $D_{MC}(LMV)$ using the information of the collocated block in the previously coded frame. Hence, we estimate the value of λ_{Thr} using (14),

$$\lambda_{Thr} \cong \frac{D'_{MC}(MV')}{TC} \quad (14)$$

where MV' and D'_{MC} are the motion vector and the corresponding motion compensation distortion of the collocated block in the previously coded frame.

As shown in Figure 7, when $\lambda > \lambda_{Thr}$ the selected GMV is used as the final MV of the block. On the other hand, when $\lambda < \lambda_{Thr}$ the final MV of the block is LMV which is selected by λ_{Thr} . Hence, we propose to define the two-piece Lagrange multiplier ($\lambda_{2-piece}$) as in (15).

$$\lambda_{2-piece} = \begin{cases} \lambda_{Thr} & \lambda_{Thr} > \lambda \\ \lambda & \lambda_{Thr} < \lambda \end{cases} \quad (15)$$

```

Imin=Jmin=MAX

// find LMV and GMV

for all mv in the search area (SA){

    D=DMC(mv), R=RMC(mv)

    IME=log(D)+R/TC // (11)

    if (IME<Imin) then {LMV=mv, Imin=IME}

    JME=D+λR // (12)

    if (JME<Jmin) then {GMV=mv, Jmin=JME}

}

// Find the MV using (13)

if (RMC(GMV)<RMC(LMV)) then MV=GMV else MV=LMV
    
```

Figure 8 The proposed motion estimation algorithm.

It is worth nothing that each case in the model of (15) corresponds to one case in the model of (12). Then, the final MV of the block is selected using (16).

$$MV = \arg \min_{mv \in SA} \{D_{MC}(mv) + \lambda_{2\text{-piece}} R_{MC}(mv)\} \quad (16)$$

Based on the proposed two-piece Lagrange multiplier, before starting the motion search process for a block, the value of the λ_{Thr} is estimated using (14). Then, the value of the two-piece Lagrange multiplier is determined by (15). Then, the conventional motion estimation process is executed.

4.3 Optimization parameters selection

The proposed motion estimation algorithm has two optimization parameters, namely λ and time constant (TC). The parameter of λ is the Lagrange multiplier which is used in mode decision and is defined as (17) for H.264/AVC,

$$\lambda = c \times 2^{\frac{QP-12}{3}} \quad (17)$$

where c is about 0.85 according to experimental results [43].

In this section, we discuss the value of TC in the model of (3). The R-D function for a source with normal distribution is modeled as (18) [42],

$$d = \delta^2 2^{-2r} = \delta^2 e^{-2 \ln(2)r} \quad (18)$$

where r and d are average rate and distortion (SSD) per symbol, expressed by (19) and n is the number of symbols in the residual block. Using (19) in (18), the R-D curve of the residual is extracted as (20). Comparing (20) to the model we used in (3), TC can be expressed as (21) for each block.

$$\begin{cases} r = \frac{R_{Res}}{n} \\ d = \frac{D_{Res}}{n} \end{cases} \quad (19)$$

$$D_{Res} = n \delta^2 \exp\left(-\frac{R_{Res}}{n/2 \ln(2)}\right) \quad (20)$$

$$TC = \frac{n}{2 \ln(2)} \quad (21)$$

Hence, the value of TC is calculated for each block based on the number of coefficients in that block. As the value of n is related to the size of block, TC can be calculated based on the size of residual block as (22),

$$TC = \frac{W \times H}{16 \times 16} TC_{MB} \quad (22)$$

where W and H are width and height of the block, and TC_{MB} is the time constant of MB with the size of 16×16 .

For the case that the distortion of motion compensation is calculated in terms of SAD, the values of λ and TC can also be calculated as below. The relation between SAD and SSD can be expressed as (23) where α is approximately 1.25 for the residuals with zero mean Gaussian distribution. Hence, the Lagrange multiplier for SAD can be expressed as (24) which has also been noted in [43].

$$D_{SAD} = \alpha \sqrt{D_{SSD}}, \quad \alpha = \sqrt{\pi/2} \approx 1.25 \quad (23)$$

$$\lambda_{SAD} = -\frac{\partial D_{SAD}}{\partial R} = \alpha \frac{\sqrt{3c}}{2} \sqrt{\lambda_{SSD}} = \sqrt{\lambda_{SSD}} \quad (24)$$

Substituting the model of (3) in (23), SAD can be expressed as (25) which implies that the TC for SAD distortion metric can be expressed as (26).

$$D_{SAD} = \alpha \sqrt{D_0} \exp\left(-\frac{R_{Res}}{2 \times TC}\right) \quad (25)$$

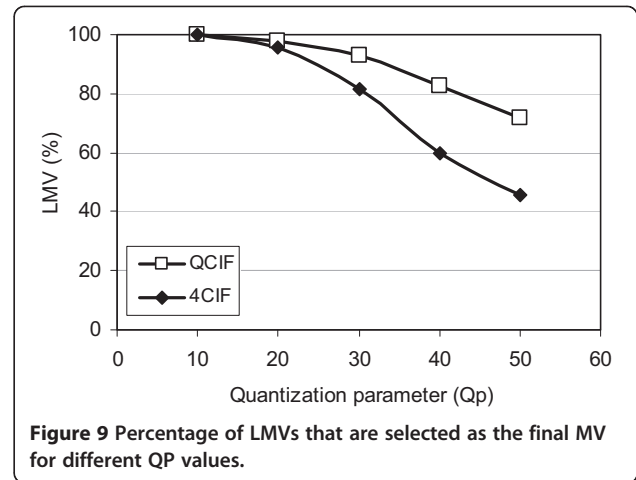
$$TC_{SAD} = 2 \times TC_{SSD} \quad (26)$$

5 Transcoding-resilient motion estimation

As described in the Introduction section, the motion vector selected by conventional motion estimation using (2) is influenced by λ , and consequently by QP. Hence, the selected MV for high bit rate compression may not be suitable for low bit rate when transcoding the video with reusing motion information, resulting in quality degradation.

In contrast, based on the theoretical analyses and empirical results presented below, we demonstrate that the MV, selected by the proposed motion estimation process, is less affected by QP. In other words, it is suitable for both high bit rate and low bit rate compressions. Hence, the selected MV for high bit rate compression can also be used for low bit rate when transcoding the video, without degrading the quality significantly. Thus, the motion re-estimation stage during transcoding is no longer required, leading to a much simpler and faster adaptation, while at the same time providing better quality.

The proposed motion estimation uses (12) to select the motion vector. In high bit rate compression, QP and its corresponding λ value are small, leading to a large $R_{MC}(GMV)$ according to (11). As a result, the LMV is the most dominant motion vector selected by the proposed motion estimation according to (12). This observation is supported by the experimental results performed over a wide range of QPs, for a variety of video contents and resolutions (city, crew, harbor, soccer in 4CIF and football, garden, mobile, Paris, silent, Stefan, students, tennis in QCIF). Based on the experimental analysis, presented in Figure 9, for QPs in the range of



20 to 30, near 90% of MVs selected by the proposed motion estimation method are LMVs.

Furthermore, the LMV of a block is not significantly affected by changes in QP because, as presented in (10), LMV only depends on the motion compensation rate (R_{MC}) and the distortion (D_{MC}). As described below in more details, R_{MC} is unchanged, and D_{MC} is almost unchanged when transcoding a video.

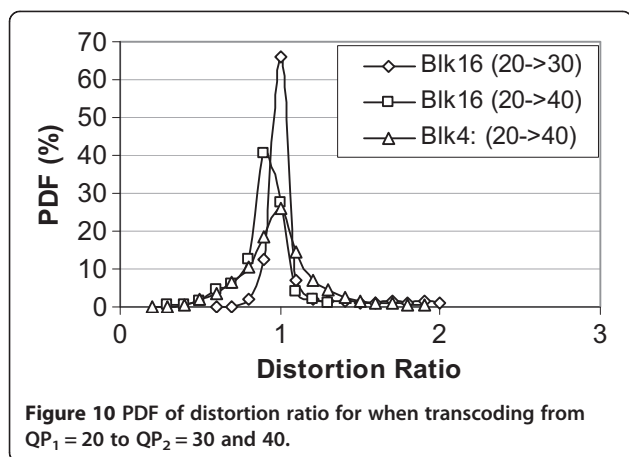
If the selected motion vectors remain unchanged for neighboring blocks (which is valid when transcoding with mode and MV reuse), the predicted motion vectors and consequently the rate of the candidate motion vectors (R_{MC}) remain unaffected for the current block.

To analyze the behavior D_{MC} , we have compared D_{MC} values of a block for the same candidate motion vectors when different QP values are used to code the reference frame. Let the distortion ratio (DR) to be defined as (27) which represents the ratio of motion compensation distortions of a block for the same candidate motion vector (mv) when the reference frame is encoded with two different QP values of QP_1 and QP_2 .

$$DR(mv) = \frac{D_{MC}(mv, QP_{ref} = QP_2)}{D_{MC}(mv, QP_{ref} = QP_1)} \quad (27)$$

The probability distribution function (PDF) of the distortion ratio is depicted in Figure 10 for block sizes of 16×16 and 4×4 when the QP value for the reference data is changed from 20 to 30 and from 20 to 40 for blocks of football and foreman sequences of size QCIF. It can be seen that most of the ratios are near 1.0, which implies that the distortion value of a candidate motion vector takes nearly the same value when QP changes. As a result, the motion compensation rate (R_{MC}) and distortion (D_{MC}), and consequently the LMV of the block are not significantly affected when QP changes.

To summarize, as indicated by the analyses and the simulation results, LMV is the most dominant MV that



is selected by the proposed motion estimation method. Furthermore, LMV is unaffected by changes in QP. This suggests that the MV selected by the proposed method is not significantly affected by QP for all practical purposes.

It should be noted that the two-piece Lagrange multiplier, proposed in Subsection 4.2, is designed to select the same motion vector that is selected using the proposed condition of (12). Hence, the property described in this section is also valid for the MV which is selected by the proposed two-piece Lagrange multiplier. This property can also be justified based on the concept of the two-piece Lagrange multiplier. When coding a video with a low QP, and using the conventional condition of (2), the Lagrange multiplier (λ) is small, and a large MV is selected which is not suitable for large QP values. But according to (15), the two-piece Lagrange multiplier ($\lambda_{2\text{-piece}}$) is limited to be larger than λ_{Thp} and hence it is larger than λ . As a result, the MV selected by $\lambda_{2\text{-piece}}$ is smaller and also suitable for encoding with larger QP.

6 Simulation results

We analyze the quality performance of the proposed motion estimation for an encoding and a transcoding system in Subsections 6.1 and 6.2, respectively. The complexity of the proposed algorithm and quality-complexity trade-off are presented in Subsections 6.3 and 6.4, respectively. Motion estimation with the proposed two-piece Lagrange multiplier in (15) has the same R-D performance of when using the proposed condition in (12), while it imposes no computational overhead to the encoder. Hence, we present only the results for the proposed two-piece Lagrange multiplier method, which is referred to as the PropME hereafter.

PropME has been evaluated with JM16.2 [44], the reference software of H.264/AVC, and compared with its motion estimation algorithm referred to as the conventional motion estimation (ConvME) hereafter. The baseline profile and its corresponding configuration file have

been used for the simulations. IntraPeriod option is set to 50. UMHEx motion search method with the search range of 32 and number of reference frames of 1 and 5 have been used in analyses.

Different test sequences have been coded when RC is enabled and disabled. In disabled RC scenario (RC = OFF), constant QP values of 20, 24, 32, 36, and 40 are used for encoding. In enabled RC scenario (RC = ON), initial QP is set to different values, mentioned above, and the target bit rate is set to the corresponding bit rate produced in disabled RC scenario. The average coding efficiency gain for all QP values has been calculated using the Bjøntegaard delta rate (BDR) [45] that presents the bit rate expansion of a method with respect to an anchor method. As a negative value of BDR means coding efficiency gain, we have presented the negative of BDR ($-BDR\%$) in the tables and figures. When comparing the coding efficiency for a specific QP value, BDR method cannot be used, because it calculates the overall coding gain when having several R-D points. Instead, we have used the bit rate (BR) and PSNR values of the method and the anchor for that specific QP to calculate the coding efficiency. It should be noted that the PSNR values of different methods are very close for the same QP value. For a fair comparison, however, the slight change in PSNR has been compensated for in the bit rate, hence the coding efficiency is measured by calculating the percentage of bit rate reduction (BRR) at the same quality using (28).

$$BRR = 1 - \frac{BR}{BR_{\text{Anchor}}} + \frac{PSNR - PSNR_{\text{Anchor}}}{PsnrBrRatio} \quad (28)$$

where $PsnrBrRatio$ is the ratio of PSNR change to the percentage of bit rate change. The value of $PsnrBrRatio$ is calculated for each test sequence, test configuration, and QP value using PSNR and bit rate values of two consecutive QP (QP and QP') as shown in (29).

$$PsnrBrRatio = \left(\frac{PSNR(QP) - PSNR(QP')}{\frac{BR(QP)}{BR(QP')} - 1} \right) \quad (29)$$

The value of $PsnrBrRatio$ is about 5 which means a 0.5-dB difference in PSNR corresponds to a 10% difference in bit rate [45]. As presented in tables below, the average values of BRR over different QP values (Avr.) are almost the same as the values of BDR. This indicates that the BRR method of (28) accurately estimates the coding efficiency.

We have tested the R-D performance of PropME in combination with several well-known MD approaches in the reference software, referred to as MD0, MD1, MD2, and MD4. MD1 is a high complexity full search mode decision, also known as RDO-ON in JM software, where

Table 1 Coding efficiency of PropME over ConvME for D1MD0 configuration in video coding application

Video size	Rate control QP	Fixed QP (RC = OFF)							RC = ON	
		Bit rate reduction (BRR%)							-BDR%	-BDR%
		20	24	28	32	36	40	Avr.		
QCIF	Avr. PSNR	41.9	38.8	35.9	33	30.3	27.7	-	-	-
	Akiyo	0.9	2.4	4.2	2.4	0.4	0.5	1.8	1.5	1.5
	Carphone	2.7	3.4	3.1	2.0	1.4	0.2	2.1	2.5	3.7
	City	5.8	6.7	5.8	4.7	2.9	0.1	4.3	5.2	3.7
	Claire	3.8	4.3	4.5	2.9	0.8	0.1	2.7	2.3	5.3
	Coastguard	6.8	6.6	5.1	5.3	3.6	1.7	4.9	5.2	5.0
	Container	-3.5	-1.5	0.4	3.2	0.3	-0.2	-0.2	-1.1	0.5
	Crew	9.8	9.8	7.1	3.9	0.7	-0.4	5.2	5.7	5.4
	Football	2.6	2.9	3.2	4.3	5.7	3.6	3.7	4.4	3.7
	Foreman	2.9	3.8	4.1	3.2	1.5	-0.3	2.5	2.6	2.0
	Garden	1.7	2.3	2.7	2.4	1.7	1.2	2.0	2.2	2.6
	Grandma	3.2	3.6	5.2	2.9	1.2	0.1	2.7	2.5	0.9
	Hall monitor	2.1	3.0	2.7	2.3	1.5	-0.1	1.9	2.0	0.5
	Harbor	7.4	8.4	8.4	7.9	5.4	1.8	6.6	7.1	7.0
	Highway	0.9	3.1	3.2	2.1	0.2	0.1	1.6	0.4	2.8
	Ice	-1.6	-0.5	1.3	1.7	1.5	1.7	0.7	1.4	1.2
	Miss am	2.4	3.7	2.7	0.4	0.0	0.0	1.5	1.4	11.9
	Mobile	1.1	0.9	1.8	2.7	3.9	2.5	2.1	2.4	2.2
	Mother15fps	5.7	6.3	4.9	1.4	0.9	-0.3	3.2	3.8	0.0
	Mother30fps	3.3	5.4	5.1	4.5	1.1	0.3	3.3	3.3	2.4
	News	0.8	1.6	2.4	3.0	2.8	0.4	1.8	2.5	1.9
	Paris	-0.4	0.2	1.4	2.6	3.8	1.8	1.5	2.4	1.1
	Salesman	2.5	2.9	4.0	3.7	1.6	0.1	2.5	2.5	3.5
	Silent	4.5	5.1	4.8	3.6	1.9	0.0	3.3	3.9	2.8
	Stefan	2.0	2.7	3.3	4.8	5.5	3.3	3.6	4.6	4.1
	Students	1.0	1.8	2.8	3.5	3.2	1.0	2.2	2.6	-0.4
	Suzie	6.5	6.7	5.7	2.4	0.3	0.0	3.6	3.6	3.2
Tennis	3.7	3.0	1.6	2.4	2.8	1.7	2.5	2.9	2.0	
Trevor	3.4	4.8	4.5	3.2	1.8	0.6	3.0	3.6	3.6	
Walk	7.7	7.7	7.6	6.5	4.2	1.0	5.8	6.3	4.8	
<i>Average</i>	3.1	3.8	3.9	3.3	2.2	0.8	2.8	3.1	3.1	
CIF	Avr. PSNR	42.1	38.9	36	33.1	30.5	28.1	-	-	-
	Akiyo	4.8	4.7	3.5	0.9	0.0	-0.1	2.3	2.1	1.5
	Bus	6.8	7.5	7.0	6.9	5.5	2.9	6.1	6.7	7.4
	City	6.0	6.0	5.3	4.2	0.9	0.1	3.7	3.6	5.8
	Coastguard	11.3	12.0	11.0	8.5	4.7	1.1	8.1	8.7	8.7
	Container	0.4	0.5	1.2	2.2	3.1	1.4	1.5	1.3	1.2
	Crew	11.5	9.7	6.2	3.0	0.8	0.2	5.2	5.4	5.1
	Crew	9.8	7.9	4.8	2.1	0.4	-0.5	4.1	4.2	3.5
	Foreman	5.3	5.6	3.3	1.7	0.1	-0.4	2.6	2.6	2.5
	Hall monitor	4.0	5.0	4.3	3.6	1.5	0.4	3.1	2.5	2.2
	Harbour15fps	10.3	11.9	12.1	10.8	6.3	1.8	8.9	9.4	9.5

Table 1 Coding efficiency of PropME over ConvME for D1MD0 configuration in video coding application (Continued)

	Harbour30fps	9.0	9.9	9.8	8.6	5.0	1.2	7.2	7.6	8.5
	Ice15fps	2.1	2.3	2.2	1.9	0.6	0.2	1.5	1.7	1.7
	Ice30fps	1.1	1.7	1.5	1.6	0.8	-0.3	1.1	1.5	2.0
	Mobile	1.6	3.0	4.5	5.6	5.4	2.5	3.8	4.2	4.2
	Mother	7.9	5.8	2.9	1.0	0.5	0.0	3.0	2.7	2.2
	News	2.8	3.3	3.3	2.7	1.4	0.5	2.3	2.4	2.0
	Paris	1.6	2.3	3.0	3.5	3.2	1.8	2.6	3.0	2.3
	Sign Irene	6.3	7.6	7.4	5.8	2.8	0.2	5.0	5.3	4.5
	Silent	5.8	5.7	5.0	3.0	1.0	0.2	3.4	3.6	2.2
	Stefan	5.2	6.4	7.3	7.7	7.0	3.3	6.1	6.8	7.2
	Student	4.1	4.7	3.5	1.8	1.5	-0.2	2.6	3.2	1.3
	Tempete	4.2	5.4	5.6	5.7	4.6	1.3	4.5	5.1	4.6
	Walk	11.0	10.8	9.4	6.7	2.9	0.8	6.9	7.1	7.5
	<i>Average</i>	<i>5.8</i>	<i>6.1</i>	<i>5.4</i>	<i>4.3</i>	<i>2.6</i>	<i>0.8</i>	<i>4.2</i>	<i>4.4</i>	<i>4.2</i>
4CIF	Avr. PSNR	42	39.3	36.9	34.4	32.2	30.2	-	-	-
	City	10.1	8.9	6.5	3.8	1.1	0.1	5.1	5.1	6.3
	Crew	12.8	9.3	4.9	1.5	0.5	0.0	4.8	4.8	4.9
	Harbor	14.4	15.6	14.0	9.2	2.6	0.1	9.3	9.5	10.5
	Ice	4.6	4.1	3.0	1.6	0.4	0.2	2.3	2.0	2.1
	Soccer	10.1	8.7	6.4	4.4	1.7	0.0	5.2	5.6	5.7
	<i>Average</i>	<i>10.4</i>	<i>9.3</i>	<i>6.9</i>	<i>4.1</i>	<i>1.3</i>	<i>0.1</i>	<i>5.4</i>	<i>5.4</i>	<i>5.9</i>
HD	Avr. PSNR	40.9	37	34.3	31.7	29.3	27.4	-	-	-
	Mobcal	5.3	5.6	7.1	7.3	4.1	0.6	5.0	4.7	4.6
	Parkrun	6.3	6.6	6.1	6.2	6.4	4.7	6.0	6.5	5.1
	Shields	7.2	6.2	7.0	4.1	1.3	-0.1	4.3	3.3	3.0
	Stockholm	7.6	3.2	4.8	3.3	0.4	-0.2	3.2	1.6	3.0
	<i>Average</i>	<i>6.6</i>	<i>5.4</i>	<i>6.3</i>	<i>5.2</i>	<i>3.0</i>	<i>1.2</i>	<i>4.6</i>	<i>4.0</i>	<i>3.9</i>

a macroblock is fully coded for each mode in order to determine its final rate and distortion. Then, the best mode is selected by the Lagrange multiplier of (1). On the other hand, MD0 is a low complexity simplified mode decision method, also known as RDO-OFF in JM software, where the residual is not coded and the motion compensation cost in (2) is used to determine the best mode. MD2 and MD4 are similar to MD1 and MD0, respectively, but they use early skip selection method to reduce the computational cost especially at low bit rate compression. MD0 (and MD4) approach, however imposes a quality performance degradation, has much lower computational cost, and is mostly used in practical hardware and real-time software implementations of H.264/AVC to meet power consumption, gate count, and/or real-time constraints.

For a comprehensive analysis, we also compared PropME when different distortion metrics, referred to as D0 and D1 hereafter, are used for motion estimation. In both cases of

D0 and D1, SAD is used in integer motion estimation (IME), because SSD and SATD impose an intolerable complexity when used with IME. In D0 and D1 cases, the distortion metric of fractional motion estimation is set to SATD and SSD, respectively. It should be noted that SATD requires the Hadamard transform, hence its computational complexity is higher than that of SSD. According to the simulation results, the overall encoder complexity when using D1 is 15% to 35% less than when using D0.

Based on the above abbreviations, we present different test configurations with D_xMD_y notation. For example, in D1MD0 configuration, D1 means that the distortion metric of fractional motion estimation is set to SSD, and MD0 means that mode decision method is the simplified mode decision.

The anchor method, also reported for each result, is the corresponding configuration with ConvME when reporting the gain of the PropME over ConvME. The anchor method is D1MD0 configuration with ConvME when

Table 2 Coding efficiency (–BDR%) of PropME over ConvME for different configurations in video coding application

Ref	Configuration	Rate control = OFF					Rate control = ON				
		QCIF	CIF	4CIF	HD	Avr.	QCIF	CIF	4CIF	HD	Avr.
1	D1MD0	3.1	4.4	5.4	4.0	4.2	3.1	4.2	5.9	3.9	4.3
	D1MD4	1.4	2.8	3.8	1.1	2.3	1.3	2.4	3.9	0.6	2.0
	D0MD0	2.6	1.3	1.8	0.8	1.6	0.8	1.1	1.5	3.9	1.8
	D0MD4	1.5	0.7	1.1	0.0	0.8	0.4	0.6	1.0	0.6	0.6
5	D1MD0	2.3	3.6	4.7	3.3	3.5	2.7	3.7	7.3	2.7	4.1
	D1MD4	0.3	0.6	0.9	0.2	0.5	0.0	0.4	0.8	0.2	0.4
	D0MD0	0.3	0.6	1.0	0.5	0.6	0.3	0.6	1.2	0.1	0.5
	D0MD4	0.0	0.0	0.1	–0.2	0.0	–0.2	0.0	–0.1	0.0	–0.1

comparing the quality-complexity performance of various combinations of motion estimation, mode decision, and distortion metric together.

6.1 Quality performance for video encoding

In the simplified mode decision (MD0 or MD4), which uses (2), R-D costs of different modes of a macroblock are calculated by accumulating all motion compensation R-D costs of all the corresponding sub-blocks. These R-D costs are then compared to select the best mode for the macroblock. When using ConvME, a large MV may inappropriately be selected specially for small block sizes (e.g., Inter 8×8). As a result, small R-D cost may be calculated for small block size modes, which are then mostly selected as the best mode of the macroblock. This results in inappropriate best mode selection and R-D performance degradation. On the other hand, PropME limits MV to be smaller than LMV, and hence it prevents resulting in inappropriately small motion compensation R-D cost. Hence, the simplified mode decision is not trapped in selecting small block size modes, resulting in better R-D performance as reported in this section below.

In the full search mode decision (MD1 or MD2), R-D costs of different modes of the macroblock are instead calculated using actual bit rate and final distortion of all

the corresponding sub-blocks. In this case, the best mode which also has the best MVs, is selected as the final mode. Hence, full search mode decision methods compensate for most of the shortcomings of motion estimation process. This has been confirmed by the simulations, where PropME with MD1 (and MD2) resulted in only about 0.1% coding efficiency for video coding applications. Consequently, there was no point in reporting them in details.

The coding efficiency results of PropME over ConvME have been presented for all the test sequences in Table 1 for the configuration of D1MD0. As it will be presented below in more details, PropME with D1MD0 configuration shows the best quality and computational performance in both video coding and transcoding applications. It has been noticed that the bit rate reduction is more significant for the sequences with higher motion activity. It is about 2.5 times of the average value. Furthermore, the R-D performance improvement is more noticeable in high PSNR ranges (low QPs). This is mostly because in high PSNR, where QP and consequently λ are very small, the ConvME results in a large MV, whereas the PropME limits the rate of MV to the rate of LMV instead. With less rate for the MV, there is an improvement in R-D performance of the coded video. On the other hand, in low PSNR values (larger QPs), λ has a larger value and the optimum point mostly lies on the motion compensation R-D curve. Hence, PropME normally selects the same MV as ConvME does, hence they show almost the same R-D performance.

The bit rate reduction is higher for larger video sequences (4CIF and HD). The reason is that the MV values are larger in high resolution video sequences and they consume more bits. Furthermore, each block corresponds to a smaller area in the real world, hence the motion compensation is more accurate, residual values are small, and residual coding generates fewer bits. As a result, the share of the MV bits is higher in the total bits, and optimum selection of the MV leads to higher bit rate reduction.

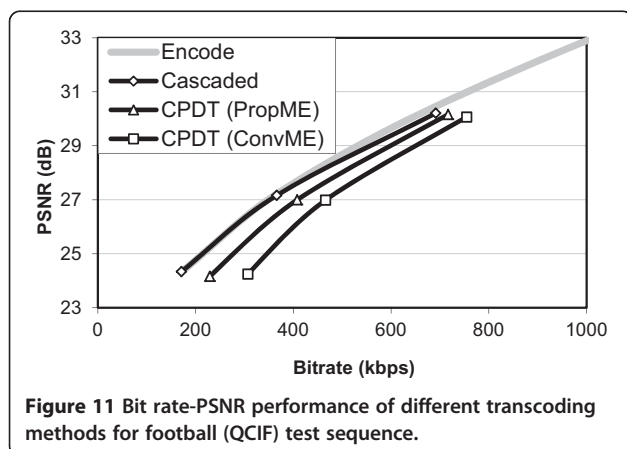


Figure 11 Bit rate-PSNR performance of different transcoding methods for football (QCIF) test sequence.

Table 3 Coding efficiency of PropME over ConvME for D1MD0 configuration in video transcoding applicaiton

Rate control		Fixed QP (RC = OFF)							RC = ON	
Video size	Test sequence	Bit rate reduction (BRR%)						-BDR%	-BDR%	
		Enc	Transcoding							Avr.
			QP ₁	QP ₂						
				20	24	28	32	36	40	
QCIF	Akiyo	0.9	9.6	17.6	25.2	30.2	33.0	19.4	19.7	16.2
	Carphone	2.7	13.1	21.8	29.5	33.7	36.4	22.9	23.9	20.7
	City	5.8	17.1	28.4	38.9	46.3	49.7	31.0	30.1	26.2
	Claire	3.8	11.3	16.8	22.3	24.4	25.1	17.3	17.7	12.8
	Coastguard	6.8	15.9	29.6	46.7	59.5	66.2	37.4	36.2	36.8
	Container	-3.5	7.6	17.5	28.1	40.1	47.9	23.0	22.7	22.6
	Crew	9.8	18.0	27.2	36.3	44.4	48.9	30.7	29.5	27.6
	Football	2.6	6.6	13.8	24.8	37.5	47.9	22.2	19.2	23.1
	Foreman	2.9	14.0	24.9	33.9	39.5	41.6	26.1	26.3	23.7
	Garden	1.7	3.7	7.8	15.5	28.9	44.2	17.0	15.1	20.0
	Grandma	3.2	14.1	23.8	33.0	37.8	38.7	25.1	23.4	18.7
	Hall monitor	2.1	20.2	29.2	35.5	40.8	44.0	28.6	30.4	27.1
	Harbor	7.4	13.1	22.9	37.7	51.7	60.9	32.3	31.1	34.4
	Highway	0.9	19.3	31.6	40.0	43.6	42.5	29.7	31.1	30.1
	Ice	-1.6	4.5	10.1	16.3	22.3	28.7	13.4	14.2	13.5
	Miss am	2.4	11.0	17.5	21.4	24.4	25.0	16.9	18.2	15.0
	Mobile	1.1	2.4	6.7	15.7	31.3	49.3	17.7	15.7	20.3
	Mother&dgh1	5.7	17.9	27.0	34.2	37.5	39.1	26.9	26.5	20.7
	Mother&dgh2	3.3	14.0	25.7	35.9	42.4	45.1	27.7	26.7	23.1
	News	0.8	6.3	12.2	17.7	23.5	26.9	14.6	14.4	12.8
	Paris	-0.4	3.7	8.9	15.7	24.1	31.0	13.9	13.1	13.7
Salesman	2.5	9.8	18.2	26.2	33.1	36.1	21.0	19.7	13.3	
Silent	4.5	13.2	20.2	27.3	31.9	35.2	22.0	20.9	16.0	
Stefan	2.0	3.9	7.3	13.6	25.3	42.3	15.7	14.4	16.7	
Students	1.0	9.5	17.4	25.7	31.9	35.3	20.1	18.7	14.2	
Suzie	6.5	20.8	31.8	39.7	44.4	45.5	31.4	31.7	24.6	
Tennis	3.7	15.7	28.0	39.9	48.5	52.7	31.4	29.1	23.4	
Trevor	3.4	10.8	18.4	26.5	32.6	36.0	21.3	20.3	18.0	
Walk	7.7	13.9	21.6	30.7	38.6	43.1	25.9	25.1	24.8	
<i>Average</i>	<i>3.1</i>	<i>11.8</i>	<i>20.1</i>	<i>28.8</i>	<i>36.2</i>	<i>41.3</i>	<i>23.5</i>	<i>22.9</i>	<i>21.0</i>	
CIF	Akiyo	4.8	15.2	23.5	30.3	34.0	35.2	23.8	24.7	18.8
	Bus	6.8	15.5	24.1	35.0	44.9	52.0	29.7	28.7	28.8
	City	6.0	17.7	30.2	42.8	51.2	55.4	33.9	33.0	30.9
	Coastguard	11.3	19.7	31.8	48.2	61.6	68.8	40.2	39.8	42.2
	Container	0.4	22.1	37.8	51.2	59.7	63.5	39.1	39.5	35.9
	Crew	11.5	20.5	29.0	38.2	45.0	48.0	32.1	31.2	27.2
	Crew	9.8	20.4	29.5	37.6	42.7	43.7	30.6	30.6	24.5
	Foreman	5.3	21.4	31.6	38.7	42.3	42.9	30.4	31.2	26.7
	Hall monitor	4.0	35.2	47.5	54.3	57.5	57.5	42.6	46.9	40.6
	Harbour15fps	10.3	16.8	26.2	40.3	54.1	63.3	35.2	34.0	38.2

Table 3 Coding efficiency of PropME over ConvME for D1MD0 configuration in video transcoding applicaiton (Continued)

	Harbour30fps	9.0	15.5	25.2	39.1	52.2	60.3	33.5	32.5	35.1
	Ice	2.1	8.7	14.4	20.7	24.7	28.4	16.5	17.5	15.7
	Ice	1.1	8.0	14.4	19.9	24.0	25.9	15.5	16.8	14.2
	Mobile	1.6	6.3	12.9	23.7	36.9	47.0	21.4	20.8	26.7
	Mother	7.9	22.0	30.4	36.0	38.2	39.0	28.9	29.5	21.8
	News	2.8	9.9	15.7	21.4	26.5	29.4	17.6	18.0	15.7
	Paris	1.6	8.2	14.7	23.7	32.2	37.8	19.7	19.3	19.9
	Sign Irene	6.3	15.4	24.3	33.4	40.3	42.7	27.1	27.6	24.9
	Silent	5.8	18.7	27.8	35.8	41.0	42.8	28.6	27.4	21.4
	Stefan	5.2	10.3	17.0	27.3	39.9	49.7	24.9	24.1	28.9
	Student	4.1	16.3	26.2	33.9	39.1	41.8	26.9	26.5	22.4
	Tempete	4.2	10.5	19.0	31.2	43.1	50.2	26.4	25.6	24.5
	Walk	11.0	19.1	27.6	36.2	43.1	47.1	30.6	30.0	27.6
	<i>Average</i>	<i>5.8</i>	<i>16.2</i>	<i>25.3</i>	<i>34.7</i>	<i>42.3</i>	<i>46.6</i>	<i>28.5</i>	<i>28.5</i>	<i>26.6</i>
4CIF	City	10.1	33.5	51.3	64.2	71.5	74.4	50.8	53.0	48.8
	Crew	12.8	33.2	44.7	52.6	56.8	56.6	42.8	44.3	36.9
	Harbor	14.4	27.9	40.5	52.6	60.9	66.3	43.8	43.6	40.9
	Ice	4.6	17.4	25.3	31.0	34.3	35.1	24.6	26.0	19.8
	Soccer	10.1	29.1	43.4	54.9	62.2	64.3	44.0	43.6	40.3
	<i>Average</i>	<i>10.4</i>	<i>28.2</i>	<i>41.1</i>	<i>51.1</i>	<i>57.1</i>	<i>59.3</i>	<i>41.2</i>	<i>42.1</i>	<i>37.4</i>
HD	Mobcal	5.3	28.1	47.1	63.7	73.2	76.9	49.0	51.2	48.9
	Parkrun	6.3	14.1	24.8	39.0	54.2	67.0	34.2	32.2	35.1
	Shields	7.2	44.3	63.3	73.5	78.3	78.9	57.6	62.0	59.5
	Stockholm	7.6	45.5	69.9	78.2	80.8	80.7	60.5	64.2	59.0
	<i>Average</i>	<i>6.6</i>	<i>33.0</i>	<i>51.3</i>	<i>63.6</i>	<i>71.6</i>	<i>75.9</i>	<i>50.3</i>	<i>52.4</i>	<i>50.6</i>

The coding efficiency results have been presented in Table 2 for different video sizes, different configurations, and different number of reference frames. The averaging is performed among different QP values and all test sequences.

6.2 Quality performance for video transcoding

The R-D performance of PropME in the transcoding system has been evaluated when the high quality bitstream is generated at the encoder using both ConvME and PropME, and then it is transcoded by the CPDT method with mode and MV reuse [1]. In our test scenario, as shown in Figure 1, each video sequence is coded by a constant QP_1 equal to 20 (or by the corresponding target bit rate and initial QP_1 of 20 when RC is enabled), that generates the high quality bitstream. Then, CPDT transcoding is performed on this bitstream by re-quantizing the residuals with some constant QP_2 values of 24, 28, 32, 36, and 40 (o by the corresponding target bit rates and initial QP_2 values when RC is enabled). These QPs generate

bitstreams with approximately 50%, 32%, 22%, 16%, and 14% of the bit rate of the high quality stream, respectively.

Figure 11 shows the quality performance of different transcoding methods for football test sequence with QCIF resolution. The performance of the video when originally coded by QP_2 ('Encode' curve) is also presented in the figure as an upper bound of quality performance. It is shown that the cascaded transcoding method ('Cascaded' curve) has the best performance. However, it suffers from the high computational complexity at the transcoding node. The CPDT transcoding method on the sequence that is generated by ConvME ('CPDT(ConvME)' curve) has the worst R-D performance, while transcoding the bitstream that was generated by PropME ('CPDT(PropME)' curve) results in lower bit rate at the same PSNR quality.

The results for all test sequences and the configuration of D1MD0 are presented in Table 3, which shows the coding efficiency when the high quality bitstream is transcoded by the CPDT method ('Transcoding QP_2 ' columns in Table 3). The bit rate reduction is presented

Table 4 Coding efficiency (-BDR%) of PropME over ConvME for different configurations in video transcoding application

Ref	Configuration	Rate control = OFF					Rate control = ON				
		QCIF	CIF	4CIF	HD	Avr.	QCIF	CIF	4CIF	HD	Avr.
1	D1MD0	23.1	28.8	42.3	53.3	36.9	21.1	26.7	37.5	51.2	34.1
	D1MD4	17.8	23.5	36.8	41.7	29.9	16.9	22.8	33.2	38.4	27.8
	D1MD1	2.8	3.2	4.2	8.1	4.6	2.9	3.7	4.1	7.8	4.6
	D1MD2	3.0	3.3	4.6	8.8	4.9	2.9	3.9	4.6	8.6	5.0
	D0MD0	11.2	16.9	27.6	39.9	23.9	10.4	16.6	25.4	38.3	22.7
	D0MD4	9.6	13.9	21.7	27.3	18.1	9.1	13.9	21.2	25.8	17.5
	D0MD1	1.2	1.4	2.4	5.1	2.5	0.9	1.9	2.2	4.9	2.5
	D0MD2	1.2	1.6	2.6	5.5	2.7	1.0	1.8	2.4	5.4	2.6
5	D1MD0	19.2	27.1	42.0	47.0	33.8	17.8	24.6	37.1	45.0	31.1
	D1MD4	6.8	11.8	22.8	35.9	19.3	6.7	12.0	21.1	31.6	17.8
	D1MD1	2.9	4.0	5.9	7.4	5.1	3.1	4.5	5.7	6.7	5.0
	D1MD2	3.0	4.3	6.4	7.8	5.4	3.0	4.9	6.3	7.0	5.3
	D0MD0	6.2	11.8	25.1	27.1	17.6	6.2	12.1	23.1	26.0	16.8
	D0MD4	1.9	4.2	9.7	16.7	8.1	2.2	3.8	10.1	15.3	7.9
	D0MD1	0.9	1.5	1.8	3.9	2.0	0.8	1.7	1.7	3.8	2.0
	D0MD2	0.8	1.6	1.9	4.0	2.1	0.9	1.8	1.7	4.0	2.1

for the encoding of the high quality bitstream ('Enc. QP1' column in Table 3) which is coded with QP₁ of 20. The average of encoding and transcoding bit rate reduction and BDR are also presented in the last two columns of Table 3. The bit rate reduction is more noticeable when the transcoder aims to produce a bitstream with lower bit rates or quality (e.g., 'QP₂(40)' compared to 'QP₂(36)'). Similar to the encoding application, the bit rate reduction is more significant for the sequences with higher motion activity or higher resolution. The reasons for the above observations are the same as the ones discussed in Subsection 6.1 for video coding application. The results of average bit rate reduction are presented in Table 4 for different video sizes, different configurations, and different number of reference frames.

6.3 Complexity performance analysis

We experimentally study the computational complexity overhead of the PropME over ConvME using the

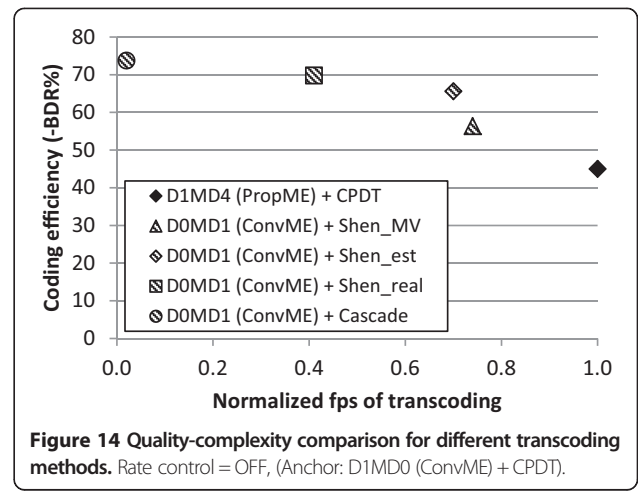
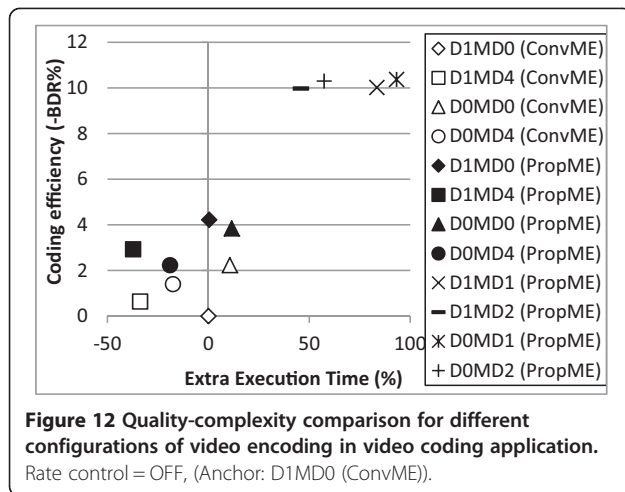
encoding time reported by JM16.2 software. In this study, we extracted the extra execution time (EET) of each method using (30).

$$EET = \frac{\text{Time}}{\text{Time}_{\text{Anchor}}} - 1 \quad (30)$$

The underlying platform is a computational grid with different personal computers and Windows operating system. It should be noted that different methods for the same video and QP that are used together to compute the EET were scheduled to be executed on the same computer in the computational grid. The average values of EET are reported in Table 5 for different video sizes and different QP values when the encoding configuration is D1MD0. EET with the other configurations shows the similar results. The results in Table 5 indicate that the computational overhead of the PropME over ConvME is 0.4% in average which is quite negligible.

Table 5 Extra execution time (EET%) of the PropME over ConvME (D1MD0 configuration)

QP	Rate control = OFF							Rate control = ON						
	20	24	28	32	36	40	Avr.	20	24	28	32	36	40	Avr.
QCIF	0.8	-0.7	-0.9	-0.7	0.1	-0.1	-0.2	0.5	-0.3	-0.2	-0.1	0.5	0.0	0.1
CIF	1.5	-0.7	-1.8	0.2	-0.9	-1.1	-0.5	1.5	-0.4	0.6	0.4	-0.7	0.3	0.3
4CIF	4.3	3.0	1.4	0.2	0.9	1.4	1.9	3.1	1.4	-0.2	0.2	0.5	1.0	1.0
HD	4.0	-0.2	-1.3	-0.9	1.1	0.3	0.5	3.1	0.2	-1.3	-1.2	0.5	-0.6	0.1
Avr.	2.7	0.4	-0.6	-0.3	0.3	0.1	0.4	2.1	0.2	-0.3	-0.2	0.2	0.2	0.4



6.4 Complexity-quality performance analysis

We have studied the quality performance of the PropME for video coding and transcoding in Subsections 6.1 and 6.2, respectively, and the complexity performance in Subsection 6.3. In this subsection, we study the complexity-quality performance of different combinations of motion estimation algorithm and mode decision methods. For this purpose, we extracted BDR and EET parameters of all combinations with respect to D1MD0 with ConvME. Then, we have summarized and compared the quality-complexity performance of different methods for video coding application in Figure 12 and for video transcoding application in Figures 13 and 14.

For video coding application, quality-complexity comparison has been presented for different configurations and motion estimation methods in Figure 12. As noted in Subsection 6.1, PropME and ConvME have almost the same quality performance when mode decision is MD1 and MD2. Hence, only the results of PropME are

presented in the figure for MD1 and MD2. In this figure, BDR and EET have been extracted by averaging all video sizes. It can be observed from Figure 12 that in the case of simplified mode decision (i.e., MD0 and MD4) using D1 instead of D0 reduces the computational overhead of the encoder by about 15% to 35%. At the same time, the coding efficiency of the ConvME is also reduced, while the coding efficiency of the PropME is increased. This proves that PropME is properly working when D1 to reduce the complexity of the encoder. This configuration is useful when a very low complexity encoder is needed.

Results for video transcoding application have been presented in Figures 13 and 14, where BDR is calculated using the R-D points of video encoding with QP_1 of 20 and video transcoding with QP_2 of 24, 28, 32, 36, and 40. Figure 13 presents the quality-complexity comparison of different configurations and the joint multi-rate optimization method [7]. Since the PropME has only 3% better coding efficiency than the ConvME when using with MD1 and MD2, their results have only been reported with PropME in Figure 13. According to Figure 13, the PropME provides better performance in all configurations. It is worth noticing that PropME with MD0 or MD4 also provides better performance than MD1 and MD2. Joint multi-rate optimization method is the best in quality performance, but it is computationally expensive.

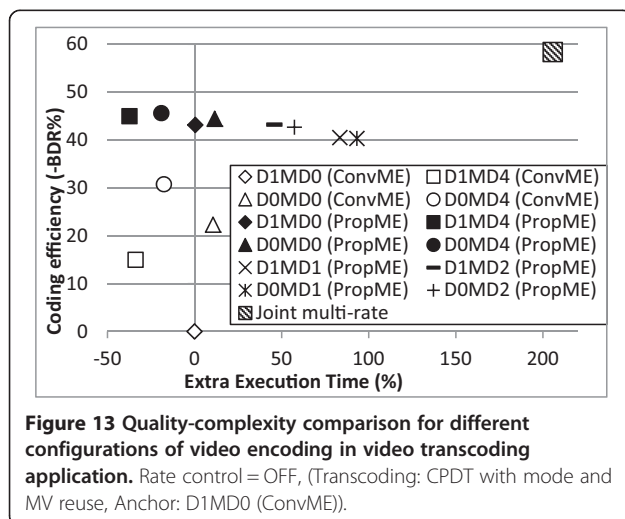


Table 6 Different combinations of encoder and transcoder

Configuration	Encoder	ME	Transcoder
D1MD4	PropME	CPDT	
D0MD1	ConvME	Shen_MV	[5]
D0MD1	ConvME	Shen_est	[5]
D0MD1	ConvME	Shen_real	[5]
D0MD1	ConvME	Cascade	[1]

In Figure 14, we compare the performance of the combination of PropME and CPDT transcoding method with the different fast transcoding methods proposed in [5], namely Shen_est and Shen_real and Shen_MV. Different combinations are presented in Table 6. As the computational complexity of transcoding methods are a lot different, we presented the normalized frame rate per second (NFPS) at which transrating can be performed. NFPS is the number of frames that can be transrated by a given method divided by that of the CPDT method. In Figure 14, it can be observed that the PropME with CPDT transcoding method results in the fastest transcoding with acceptable quality performance. Different variation of the methods, presented in [5], not only have lower transcoding frame rate, but also need higher computational complexity at the encoder side as they are using DOMD1 configurations.

7 Conclusions

In this paper, we proposed a new optimization condition for MV selection. The motion vector selected by the proposed motion estimation is less sensitive to bit rate or quantization parameter changes, making it suitable for both high bit rate and low bit rate compression. As a result, the MV in the high quality bitstream may be used even after the sequence has been transcoded. This enables using fast transcoding where the bit rate is reduced by re-quantization of residuals, and modes and motion vectors are reused from the high bit rate bitstream to eliminate the motion re-estimation cost in the adaptation node. Compared to the conventional motion estimation algorithm, the proposed motion estimation scheme not only improves the R-D performance of the coded bitstream but also results in significantly improved transcoding efficiency. The above improvements have been achieved with a quite negligible computational overhead. This makes the propose motion estimation method suitable for video encoding in ubiquitous multimedia system where different users have different network bandwidth and/or decoding bit rate capabilities.

Competing interests

The authors declare that they have no competing interests.

Author details

¹School of Electrical and Computer Engineering, University of Tehran, North Kargar Street, Tehran, P.O. Box: 14395-515, Iran. ²Distributed Collaborative Virtual Environments Research Laboratory, School of Electrical Engineering and Computer Science, University of Ottawa, 75 Laurier Avenue East, Ottawa, ON K1N 6N5, Canada. ³Digital System Level Design Laboratory, School of Electrical and Computer Engineering, Nano-Electronic Center of Excellence, University of Tehran, North Kargar Street, Tehran, P.O. Box: 14395-515, Iran.

Received: 28 May 2014 Accepted: 6 November 2014

Published: 23 November 2014

References

1. J Xin, C-W Lin, M-T Sun, Digital video transcoding. *Proc IEEE* **93**(1), 84–97 (2005)

2. DG Morrison, ME Nilson, M Ghanbari, *Reduction of the Bit-Rate of Compressed Video While in its Coded Form* (Proc. 6th International Workshop Packet Video, Portland, Oregon, USA, 1994), pp. D17.1–D17.4
3. Y-J Wang, C-C Cheng, T-S Chang, A fast algorithm and its VLSI architecture for fractional motion estimation for H.264/MPEG-4 AVC video coding. *IEEE Trans Circuits Syst Video Technol* **17**(5), 578–583 (2007)
4. WC Siu, YH Kam, WL Hui, WH Wong, L Wang, CW Kok, On the architecture of H.264 to H.264 homogeneous transcoding platform, in *Paper presented at the 2008 International Conference on Neural Networks and Signal Processing ICNNSP, IEEE*, pp. 654–659. Nanjing, 7–11 June 2008
5. H Shen, X Sun, F Wu, Fast H.264/MPEG-4 AVC transcoding using power-spectrum based rate-distortion optimization. *IEEE Trans. Circuits Syst. Video Technol.* **18**(6), 746–755 (2008)
6. GJ Sullivan, T Wiegand, Rate-distortion optimization for video compression. *IEEE Signal Process. Mag* **15**, 74–90 (1998)
7. M Semsarzadeh, MR Hashemi, *A Joint Multi Rate Optimization Framework for Video Adaptation in H (264/AVC)*, (IEEE International Symposium on Consumer Electronics (ISCE), Braunschweig, Germany, 2010), pp. 1–4
8. Y Shoham, A Gersho, Efficient bit allocation for an arbitrary set of quantizers, *IEEE Trans. Acoustics, Speech. Signal Process.* **36**, 1445–1453 (1988)
9. WC Chung, F Kossentini, MJT Smith, *An Efficient Motion Estimation Technique Based on a Rate-Distortion Criterion*, vol. 4 (IEEE International Conference on Acoustics, Speech and Signal Processing, Atlanta, USA, 1996), pp. 1926–1929
10. *Advanced Video Coding for Generic Audiovisual Services, ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)* (ITU-T and ISO/IEC JTC 1, Version 13, 2010)
11. T Wiegand, GJ Sullivan, G Bjøntegaard, A Luthra, Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* **13**(7), 560–576 (2003)
12. JR Jain, AK Jain, Displacement measurement and its application in interframe image coding. *IEEE Trans. Commun* **29**, 1799–1808 (1981)
13. GJ Sullivan, RL Baker, Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks, in *Paper presented at IEEE Global Telecommunications Conference (GLOBECOM), Volume 1* (Phoenix USA), pp. 85–90. 2–5 Dec 1991
14. T Chiang, YQ Zhang, A new rate control scheme using quadratic rate distortion model. *IEEE Trans. Circuits Syst. Video Technol.* **7**(1), 246–250 (1997)
15. J Ribas-Corbera, S Lei, Rate control in DCT video coding for low-delay communications. *IEEE Trans. Circuits Syst. Video Technol.* **9**(1), 172–185 (1999)
16. Z He, YK Kim, SK Mitra, Low-delay rate control for DCT video coding via p-domain source modeling. *IEEE Trans. Circuits Syst. Video Technol.* **11**(8), 928–940 (2001)
17. Z He, SK Mitra, A unified rate-distortion analysis framework for transform coding. *IEEE Trans. Circuits Syst. Video Technol.* **11**(12), 1221–1236 (2001)
18. Z He, SK Mitra, Optimum bit allocation and accurate rate control for video coding via p-domain source modeling. *IEEE Trans. Circuits Syst. Video Technol.* **12**(10), 840–849 (2002)
19. Z He, SK Mitra, A linear source model and a unified rate control algorithm for DCT video coding. *IEEE Trans. Circuits Syst. Video Technol.* **12**(11), 970–982 (2002)
20. K Takagi, Y Takishima, Y Nakajima, A study on rate distortion optimization scheme for JVT coder, in *visual communication and image processing (VCIP)*. *Proc. SPIE* **5150**, 914–923 (2003)
21. S Ma, W Gao, Y Lu, Rate-distortion analysis for H.264/AVC video coding and its application to rate control. *IEEE Transactions on Circuits and Systems for Video. Technology.* **15**(12), 1533–1544 (2005)
22. Z Zhang, G Liu, H Li, Y Li, A novel PDE-based rate-distortion model for rate control. *IEEE Trans. Circuits Syst. Video Technol.* **15**(11), 1354–1364 (2005)
23. N Kamaci, Y Altunbasak, RM Mersereau, Frame bit allocation for the H.264/AVC video coder via cauchy-density-based rate and distortion models. *IEEE Trans. Circuits Syst. Video Technol.* **15**(8), 994–1006 (2005)
24. DK Kwon, MY Shen, CCJ Kuo, Rate control for H.264 video with enhanced rate and distortion models. *IEEE Trans. Circuits Syst. Video Technol.* **17**(5), 517–529 (2007)
25. A Malewar, A Bahadarpurkar, V Gadre, *A Linear Rate Control Model for Better Target Buffer Level Tracking in H.264* (Signal Image and Video Processing, 2011). DOI 10.1007/s11760-011-0235-7
26. YK Tu, JF Yang, MT Sun, Efficient rate-distortion estimation for H.264/AVC coders. *IEEE Trans. Circuits Syst. Video Technol.* **16**(5), 600–611 (2006)
27. YK Tu, JF Yang, MT Sun, Rate-distortion modeling for efficient H.264/AVC encoding. *IEEE Trans. Circuits Syst. Video Technol.* **17**(5), 530–543 (2007)
28. KY Liao, JF Yang, MT Sun, Rate-distortion cost estimation for H.264/AVC. *IEEE Trans. Circuits Syst. Video Technol.* **20**(1), 38–49 (2010)

29. MG Sarwer, LM Po, Fast bit rate estimation for mode decision of H.264/AVC. *IEEE Trans. Circuits Syst. Video Technol.* **17**(10), 1402–1407 (2007)
30. W Zhenyu, NN King, A fast rate-distortion optimization algorithm for H.264/AVC. *IEEE Int. Conf. Acoustics, Speech and Signal Process. Proc* **1**, 1157–1160 (2007)
31. S Sun, S Chen, Efficient bit rate estimation for mode decision of H.264/AVC. *Lect. Notes in Comput. Sci.*, 204–212 (2008)
32. L Liu, X Zhuang, CABAC based bit estimation for fast H.264 RD optimization decision. Paper presented at the 6th IEEE Consumer Communications and Networking Conference CCNC (2009). IEEE, Nevada, 10–13 Jan 2009
33. J Hahm, C-M Kyung, Efficient CABAC rate estimation for H.264/AVC mode decision. *IEEE Trans. Circuits Syst. Video Technol.* **20**(2), 310–316 (2010)
34. X Zhao, J Sun, S Ma, W Gao, Novel statistical modeling, analysis and implementation of rate-distortion estimation for h.264/AVC coders. *IEEE Trans. Circuits Syst. Video Technol.* **20**(5), 647–660 (2010)
35. JM Moon, JH Kim, JM Moon, JH Kim, A new low-complexity integer distortion estimation method for H.264/AVC encoder. *IEEE Trans. Circuits Syst. Video Technol* **20**(2), 207–212 (2010)
36. J Sun, W Gao, D Zhao, Q Huang, Statistical model, analysis and approximation of rate-distortion function in MPEG-4 FGS videos. *IEEE Trans. Circuits Syst. Video Technol.* **16**(4), 535–539 (2006)
37. G Van Der Auwera, M Reisslein, LJ Karam, Video texture and motion based modeling of rate Variability-Distortion (VD) curves. *IEEE Trans. Circuits Syst. Video Technol.* **53**(3), 637–647 (2007)
38. CY Chang, CF Chou, DY Chan, T Lin, MH Chen, A q-domain characteristic-based bit-rate model for video transmission. *IEEE Trans. Circuits Syst. Video Technol.* **18**(9), 1307–1311 (2008)
39. J Xie, LT Chia, Study on the distribution of DCT residues and its application to R-D analysis of video coding. *J. Vis. Commun. Image Representation* **19**(7), 411–425 (2008)
40. L Guo, OC Au, M Ma, Z Liang, PHW Wong, A novel analytic quantization-distortion model for hybrid video coding. *IEEE Trans. Circuits Syst. Video Technol.* **19**(5), 627–641 (2009)
41. H Schwarz, D Marpe, T Wiegand, Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Trans. Circuits Syst. Video Technol.* **17**(9), 1103–1120 (2007)
42. M Cover Thomas, *A Thomas Joy, Elements of Information Theory*, 2nd edn. (John Wiley and Sons, Hoboken, New Jersey, 2006)
43. T Wiegand, B Girod, *Lagrange multiplier selection in hybrid video coder control* (Paper presented at the 2001 International Conference on Image Processing, IEEE, Thessaloniki), pp. 542–545. 7–10 Oct 2001
44. *H.264 Reference Software JM 18.4*. <http://iphome.hhi.de/suehring/tml/download>
45. G Bjøntegaard, *Calculation of Average PSNR Differences Between RD-Curves*, (in the 13th VCEG-M33 Meeting (TX, Austin, 2001)

doi:10.1186/1687-6180-2014-166

Cite this article as: Aminlou et al.: An R-D optimized transcoding resilient motion vector selection. *EURASIP Journal on Advances in Signal Processing* 2014 **2014**:166.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
