A Compressed Domain Distortion Measure for Fast Video Transcoding

Yicheng Huang, Vu An Tran, and Ye Wang
School of Computing
National University of Singapore
3 Science Drive 2, Singapore 117543
{huangyic, tranvuan, wangye}@comp.nus.edu.sg

ABSTRACT

Video applications on different mobile devices are becoming increasingly popular. It is an attractive alternative to transcode a high quality non-scalable video bitstream to match constraints (such as bandwidth or processing power) of different platforms with a similar functionality as a scalable video format. In principle, such a transcoder can reduce either the bit per frame (bpf) or the frame per second (fps) of the original bitstream to meet a particular constraint. In the case that multiple candidates with different combinations of bpf and fps satisfy the constraint, an objective video quality measure is needed for the transcoder to choose the candidate with the overall best quality considering both the spatial quality (reflected by bpf) and the temporal quality (reflected by fps). Conventional measures, such as PSNR and MSE operate in the pixel-domain, require full decoding of both the original and candidate video bitstreams and are computationally very expensive. This drawback renders them unsuitable for real-time transcoding applications. To solve this problem, we propose a Mean Compressed Domain Error (MCDE) to predict the quality of the transcoded video. Experimental results show that the proposed MCDE can predict video quality accurately with a negligible computational complexity in comparison with the conventional MSE/PSNR.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Video

General Terms

Algorithm, Performance, Design

Keywords

Video Transcoding, Mean Compressed Domain Error

1. INTRODUCTION

Video on mobile devices are quickly becoming an attractive application. In this paper we consider the following scenario: given a high quality non-scalable video bitstream such as MPEG-4, a transcoder converts it to multiple bitstreams matching the constraints of different mobile devices (see Figure 1). For applications such as video on demand (VoD), the transcoding needs to be fast. Furthermore, it should introduce a minimal distortion. To achieve this, an objective quality measure is needed and it should be calculated in the compressed domain directly to save computational workload. This is the problem we seek to address in this paper.

There are two popular methods to reduce the video bit rate and decoding workload, namely reducing the bit per frame (bpf) and frame per second (fps). Reducing bpf increases spatial distortion while reducing fps increases temporal distortion. For a given constraint, there could be multiple candidates with different combinations of spatial quality (bpf) and temporal quality (fps). Thus, an objective video quality measure which can predict the overall video quality considering both spatial and temporal distortions becomes a critical component.

Conventional measures such as peak signal to noise ratio (PSNR) and mean square error (MSE) operate in the pixel-domain, which require full decoding of both original and candidate video bitstreams and are computationally too expensive for real-time transcoding applications.

In this paper, we propose a new measure, Mean Compressed Domain Error (MCDE), which is used to objectively predict the quality degradation between the transcoded and the original bitstream. Compared to the conventional pixel-domain video quality measure, MCDE has a negligible computational complexity while it has the same or even better accuracy. Figure 1 shows a possible system architecture using the proposed MCDE.

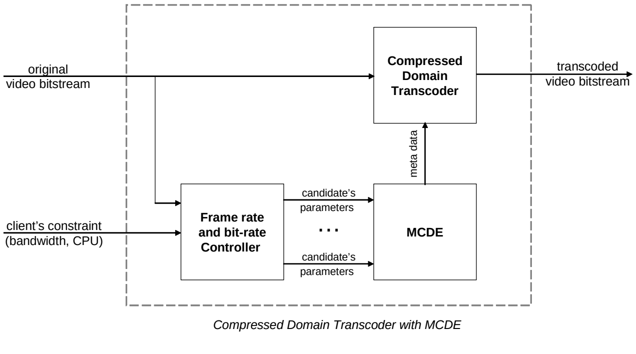

Figure 1: System architecture

The system takes the original video bitstream and client's constraint as input. Frame rate and bit-rate Controller generates a set of possible candidates with different combinations of *bpf* and *fps* that meet the client's constraint. MCDE evaluates all the

MM'07, September 23–28, 2007, Augsburg, Bavaria, Germany.
Copyright 2007 ACM 978-1-59593-701-8/07/0009...$5.00.

candidates and select the one with best quality. Then Compressed Domain Transcoder does the actually transcoding and generates the target video bitstream as the output. It should be noted that before MCDE selects the candidate with best quality, no actual transcoding is performed. MCDE is able to predict video quality based only on the compressed domain information such as Huffman codes and macroblock (MB).

The rest of this paper is organized as the follows: we present the details of MCDE in Section 2 and compare it with MSE and subjective result in Section 3. We conclude the paper in Section 4.

2. MEAN COMPRESSED DOMAIN ERROR (MCDE)

A transcoder operating in the compressed domain can reduce the bit rate and decoding workload by dropping Huffman codes or frames. Dropping Huffman codes causes spatial distortion while dropping frames causes temporal distortion. Dropping frames also results in different frame numbers between the original and transcoded video clips, thus making the calculation of the distortion measure harder. Fortunately, the work in [1, 2] provides a solution. They replace the dropped frames by copying the previous frames in the display order. The rationale is that a player can maintain the current frame on the screen before displaying the next frame. In the proposed MCDE, we use a similar approach. However, the distortion between two frames is calculated in the compressed domain. Then the MCDE is calculated as the average distortion between the original and transcoded frames.

It is noted that the distortion of the remaining frames (after frame dropping) can be regarded as the spatial distortion and the distortion of the replaced frames can be regarded as the temporal distortion. To simplify the problem, we analyze the two types of distortion separately and then combine them to produce the overall distortion. Before we go to the details of the algorithm, we first introduce some notations:

- D(F_A, F_B) is the estimated distortion between frames F_A and F_B.
- D_S(F_A, F_B) is the estimated spatial distortion between frames F_A and F_B.
- D_T(F_A, F_B) is the estimated temporal distortion between frames F_A and F_B.
- H(F) is the number of non-zero DCT coefficients of the frame F.

In the rest of this section, we present the measure for MPEG frame structure. Briefly, there are three types of frames in an MPEG stream: I-, P-, and B-frame. I-frames contain all data necessary for decoding and do not depend on any other frames. P-frames depend on the previous I- or P-frames. B-frames depend on both the previous I- or P-frame in display order as well as the subsequent I- or P-frame in display order. If an I-frame is dropped, all frames until the next I-frame cannot be decoded correctly. Because the dependency of P-/B-frames forms a chain, the effects of losing a P-frame depend on the length of the dependency chain and the frame's position within the chain. Similarly, if a P-frame is dropped, all the frames in its dependency chain cannot be decoded correctly. This form of structure exists in many video formats, such as H.261 and H.263. In this paper, we assume the transcoder drops B-frame first; then P-frame; and I-frame at last. Thus, all the remaining frames can be decoded correctly.

4.2.1 Spatial Distortion

Spatial distortion happens when Huffman codes are dropped during transcoding. Therefore spatial distortion is related to the number of Huffman codes dropped. For I-frames, the number of Huffman codes can be used directly to measure the spatial distortion. However, for P- and B-frames, error propagation has to be considered as well. It is because the frames that P- and B-frame depend on could also be distorted. In our measure, the spatial distortions caused by dropping Huffman codes for different types of frames are estimated by the following equations:

For I-frame

$$D_S(I, I') = H(I) - H(I') \tag{1}$$

where I and I′ are the original and transcoded frames.

For P-frame

$$D_S(P, P') = \alpha D_S(F, F') + (H(P) - H(P')) \tag{2}$$

where P and P' are the original and transcoded frames; F and F' are the frames P and P' depend on, respectively; α is a parameter for presenting the effect of error propagation.

For B-frame

$$\begin{aligned} D_S(B, B') = {} & \alpha(D_S(F_1, F_1') + D_S(F_2, F_2'))/2 \\ & +(H(B) - H(B')) \end{aligned} \tag{3}$$

where B and B' are the original and transcoded frames; F_1, F_2 and F_1', F'$_2$ are the frames B and B' depend on, respectively; α is the same parameter as in Equation 2.

4.2.2 Temporal Distortion

In addition to dropping Huffman codes, frames are also dropped during transcoding, resulting in temporal distortion. As mentioned before, the temporal distortion is estimated by replacing the dropped frame by its previous un-dropped frame. We calculate the distortion for every individual frame and sum the result up as the distortion of the whole video. We present how to estimate temporal distortion for different types of frames in the following paragraph. To simplify the problem, we assume the transcoder does not drop any Huffman coefficient.

For P-frame

Assume P_1 and P_2 are two P-frames in the original video and P_2 depends on P_1. After transcoding, P_1 is transcoded into P_1'. P_2 is dropped and is replaced by P_1'. Now we want to estimate the distortion between P_2 and P_1'. By our assumption, since the transcoder does not drop any Huffman coefficient from P_1, P_1 and P_1' are identical. The distortion between P_1' and P_2 should be equal to the difference between P_1 and P_2. Since P_2 depends on P_1, the difference between P_1 and P_2 can be estimated by the residual error after motion compensation. The residual error again can be estimated by the number of Huffman codes of P_2:

$$D_T(P_1, P_2) = H(P_2) \tag{4}$$

It is noted that a dropped P-frame may not be replaced by the frame it depends on. But it must be replaced by a frame in its dependency chain. So a more generic equation for estimating the distortion between a dropped P-frame and the replacing frame is:

$$\begin{aligned} D_T(P_0, P) &= \alpha D_T(P_0, P_1) + D_T(P_1, P) \\ &= \alpha D_T(P_0, P_1) + H(P) \end{aligned} \tag{5}$$

where P is the dropped P-frame, P_0 is the frame replacing P and P_1 is the frame P depends on. It is noted that P_0 and P_1 can be the same frame and they can be either P- or I-frame. α (the same parameter in Equation 2) is the parameter representing the effect for error propagation.

For B-frame

Estimating the distortion for a B-frame is more complex because B-frame depends on two frames and a dropped B-frame can be replaced by a frame that is not in its dependency chain. If a dropped B-frame is replaced by a frame that is in its dependency chain, we estimate the distortion by:

$$D_T(B,P_0) = \alpha(D_T(P_0,P_1) + D_T(P_0,P_2))/2 + H(B) \qquad (6)$$

where B is the dropped B-frame, P_1 and P_2 are the frames B depends on. P_0 is the frame to replace B; and P_0, P_1 and P_2 can be the same frame and they can be either P- or I-frame. α (the same parameter in Equation 2) is the parameter representing the attenuation effect for error propagation. If a dropped B-frame is replaced by a frame that is not in its dependency chain, the frame replacing it must be another B-frame having the same dependent frames as the dropped B-frame. We estimate the distortion by:

$$D_T(B,B_0) = H(B_0) + H(B) \qquad (7)$$

where B is the dropped B-frame and B_0 is the frame replacing B.

For I-frame

In our scheme, we drop I-frame only after all the P- and B-frames are dropped. So the dropped I-frame must be replaced by another I-frame. We estimate the distortion by:

$$D_T(I,I_0) = H(I_0) + H(I) \qquad (8)$$

where I is the dropped I-frame and I_0 is the frame replacing I.

4.2.3 Total Distortion

Now we combine spatial distortion and temporal distortion together. Assume F is the original frame. It is dropped during the transcoding. F_0' is the frame replacing F and F_0 is the original frame of F_0'. We estimate the distortion between F and F_0' by:

$$D(F,F_0^{'}) = wD_S(F_0,F_0^{'}) + (1-w)D_T(F\ ,F_0) \qquad (9)$$

where *w* is the weight between spatial distortion and temporal distortion. The average of the distortion of all the original and their transcoded frames is calculated as the final MCDE.

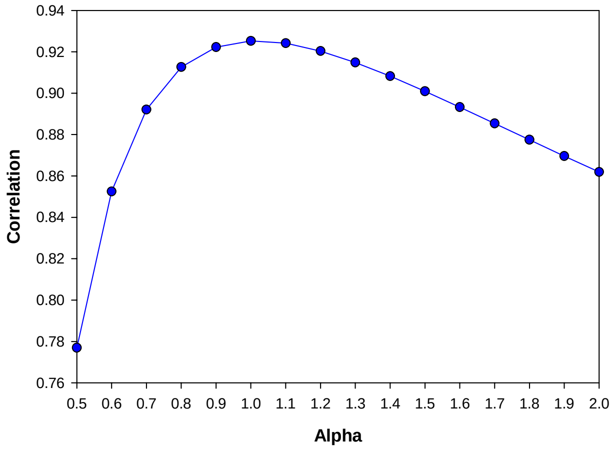

Figure 2: The correlation between MCDE and subjective result with different α values

There are two parameters in MCDE, *w* and α. It is difficult to select an optimal value for *w*, because the optimal value can be different for different video content. For example, when the movement of the video is low, the spatial distortion is more important, thus *w* should be small, and vice versa. In our scheme, considering the balance for all the cases, *w* is set to 0.5.

To choose the value for α in Equation 2, 3, 5 and 6, we conduct the experiments varying α from 0.1 to 2.0 (with *w* is fixed as 0.5). For each value of α, we compare the MCDE and the subjective results. The comparison is shown in Figure 2. And we can see that when α is set to 1.0, the correlation between MCDE and subjective result is the largest.

3. EVALUATION

3.1 Comparison among MCDE, MSE and DSCQS

In our experiments, we have three original CIF-size MPEG-4 video clips, which are shown in Table 1:

Name	Bit rate	Descriptions
Hall_768	768 KBps	Still background and two objects with moderate movements
Highway_1024	1024 KBps	Moving background
Walk_512	512 KBps	Both background and two foreground objects are with very fast movements

Table 1

Each of them is transcoded using different configurations. First, we fix the target frame rate as 8fps and 15fps and vary the number of Huffman coefficients as one of 10%, 20%, 40%, 60%, 80% and 100% of that of the original video clip. Then we fix the number of Huffman coefficient as 30% and 50% of the original video clip and vary the target frame rate as one of 5fps, 8fps, 12fps, 15fps, 20fps and 25fps. Thus, totally we have $3\times2\times6+3\times2\times6=72$ transcoded video clips for testing.

For each transcoded video clip, we calculate its MCDE and MSE. We also evaluate them using subjective testing. The 72 video clips are divided into three groups, and the video clips in each group have the same content. Thirty normal-eyesight viewers are invited to our test. Each of them evaluates one group of video clips. We select double stimulus continuous quality scale (DSCQS) as our subjective video quality methodology [3]. In DSCQS, the viewers are shown pairs of video clips (the original clip and the transcoded clip) in a randomized order. Each pair is displayed twice. After the second display, viewers are asked to rate the quality of each clip in the pair. The difference between these two scores is then used to qualify changes in quality. [3]

Figure 3 shows the comparison among MCDE, MSE and DSCQS for different percentages of the number of Huffman codes with the same target frame rate (15fps). The y-axis represents the quality distortion after normalization. The x-axis represents the percentages Huffman codes of the original video clips. It is observed that all MCDE, MSE and DSCQS decrease as the number of Huffman codes increases. The three curves follow the same trend.

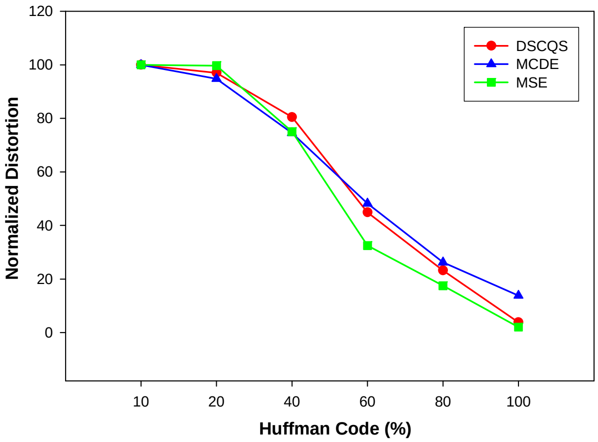

Figure 3: comparison among MCDE, MSE and DSCQS for Hall_768 with 15fps

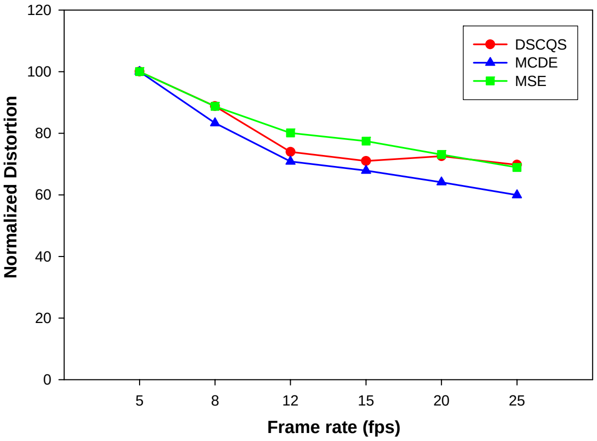

Figure 4: comparison among MCDE, MSE and DSCQS for Highway_1024 with 50% Huffman codes

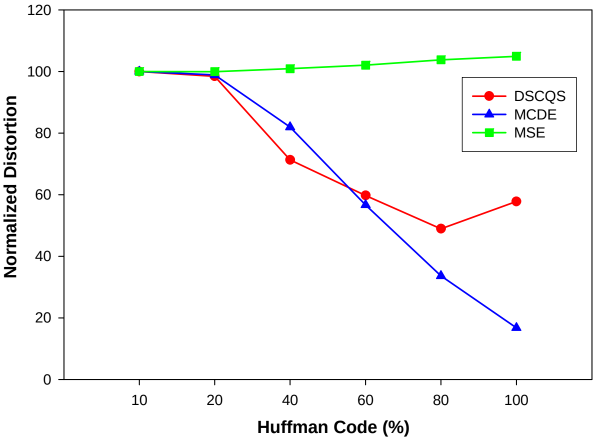

Figure 5: Comparison among MCDE, MSE and DSCQS for Walk_512 with 8fps

Figure 4 shows the comparison among MCDE, MSE and DSCQS for different frame rates with the same number of Huffman codes (50% of original). The y-axis represents the quality distortion after normalization. The x-axis represents the frame rate. It is observed that the curves of MCDE and MSE follow the same trends. Both of them decrease as the frame rate increases. However DSCQS increases as the frame rate increases from 15fps to 20fps. The similar results are also found in other subjective testing groups. It might be because that it is hard for people for to distinguish the temporal difference when the frame rate is larger than 15fps. This exactly matches the result in [4]

Figure 5 shows the comparison among MCDE, MSE and DSCQS for different number of Huffman codes with the same target frame rate (8fps). In this figure, only MCDE decreases as the number of Huffman codes increases. DSCQS almost has the same trend with MCDE except it increases as the number of Huffman codes increases from 80% to 100%. It is probably because for this video clip, the spatial distortion between 80% and 100% are very close. It is hard for people to distinguish them. MSE increases as the Huffman codes increases. It may be because 'Walk_512' has very fast motion, when the frame rate is low, MSE cannot measure the distortion correctly.

3.2 Computational Cost between MCDE and MSE

We also measure the computation complexity for both MSE and MCDE. Given the information of how to drop frames and Huffman codes, to calculate MSE we need to 1) actually transcode the video clip, 2) decode both original and transcoded video clips, and 3) calculate MSE. On average, that costs about 5 seconds for a 10-second video clip, on a Pentium 4, 3GHz, 1G RAM PC. On the other hand, the calculation of MCDE only takes around 0.5 seconds on the same PC. It is noted that we implement MSE using C++ and MCDE using Python. Although Python is much slower than C++, the calculation of MCDE is still 10 times faster than MSE.

4. CONCLUSIONS

In this paper, we propose a new objective video quality measure, MCDE, for the transcoding applications. Our experiments show that MCDE can be used to accurately predict the subjective quality of the transcoded video with negligible computational complexity in comparison with the conventional MSE.

MCDE is accurate and fast, mainly because it makes use of the compressed domain information of the original video clip. However, this also makes it unable to apply to transcoding applications involving resizing the video resolution. As a future work, we will extend MCDE for those applications.

5. REFERENCES

[1] M. Bonuccelli, F. Lonetti, F. Martelli, "Temporal Transcoding for Mobile Video Communication", the second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2005.

[2] K. Ngan, T. Meier, Z. Cheng, "Improved Single-video Object Rate Control for MPEG-4", IEEE CSVT, May 2003.

[3] M. Pinson, S. Wolf, "Comparing Subject Video Quality Testing Methodologies", Proceedings of SPIE, 2003.

[4] A.H. Anderson, L. Smallwood, R. MacDonald, J. Mullin, A. Fleming, "Video Data and Video Links in Mediated Communication: What do users value", International Journal of Human Computer Studies, 2000.