On the Performance of Promising Dirac Video Codec

André Filgueiras de Araujo

Departamento de Comunicações, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, Brazil. aaraujo@decom.fee.unicamp.br

Yuzo Iano

Departamento de Comunicações, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, Brazil. yuzo@decom.fee.unicamp.br

Abstract—Video source coding has become a theme of major importance in the communications domain, due to the growing use of digital video and to its potential for greatly reducing the impact on bandwidth needs. This paper investigates the recently released promising state-of-the-art video codec Dirac, based on wavelet transform. It is considered comparatively with respect to the well-known H.264/AVC scheme, one of the best methods for encoding video. They are compared in terms of visual quality, latency and bit rate.

Keywords—codec, Dirac, H.264, source coding, video compression, wavelets.

I. INTRODUCTION

Recently, the use of digital video has been increasing tremendously with the explosion in the number of tools for making, editing, coding and watching it. This provides a great challenge for today’s communication networks, because it is known that video content is their greatest bandwidth consumer. The utilization of streaming video serves to illustrate this point, as it has become an application that almost every web portal provides. To take another example, video on demand services are already available all around the globe, based also on digital television rapid expansion.

Hence, it is certainly of great importance to study state-of-the-art schemes for compressing digital video. As the flow of this kind of content grows unrestrictedly in every network around the globe, improvements must continue to be done to support this recent way of communicating. It is essential to enhance performance in terms of reduction in bitrates and latency without loss of visual quality (or even improving it).

One of the most efficient video codecs is H.264/MPEG-4 Part 10 [1] (also known as Advanced Video Coding - AVC), developed by a joint effort of ISO/IEC MPEG and ITU-T VCEG. This standard represents a number of advances in standard video coding technology, in terms of coding efficiency improvement, error/loss robustness enhancement, and flexibility for effective use over a broad variety of network types and application domains [2].

H.264 has been widely adopted all around the world as the compression method for a variety of applications, including the Brazilian standard for digital television - SBTVD [3]. Software and hardware implementations comprise among others Adobe Flash, QuickTime, Nero Digital, Sony Playstation, Microsoft Xbox 360. A major disadvantage of this codec, on the other hand, is that its use implies royalty payment to the patent holders.

With its 1.0.0 research version released in September 2008, Dirac video codec is a new open and royalty-free video compression technology developed by the BBC R&D. It aims at providing greater coding efficiency for applications ranging from internet streaming to HDTV and digital cinema with reduced complexity. It employs several improvements regarding state-of-the-art techniques for video coding, the most important being wavelet-based subband decomposition.

In this work, the Dirac video codec is investigated in detail, with the focus on its performance trade-offs comparatively to H.264. These represent two different and important ways of compressing video. The analysis is done in terms of visual quality (objectively measured by PSNR and MSSIM), latency and bit rate.

II. DIRAC VIDEO CODEC

Dirac is an open video codec developed by the BBC. It provides great compression efficiency with a very simple architecture, based on a small number of core tools. Its overall design is similar to a conventional hybrid motion-compensated codec, except that the function usually performed by a block transform in most standards (DCT, in most cases) is instead performed by the wavelet transform. Figure 1 presents the overall architecture of the codec.
The encoder consists in four major modules, supporting both frames and fields: Transform and Quantization, Motion Estimation, Motion Compensation and Entropy Coding. These are described on the following subsections.

A. Transform and Quantization

In Dirac Video Codec, the Transform operation is performed by wavelet filters. This key element allows for the support of any resolution without enlarging the codec toolset. A wide range of wavelet filters can be used. For compatibility with JPEG2000, Daubechies (9,7) filter is provided [5].

Both Intra and Inter pictures (a picture corresponding to a frame or a field) are wavelet transformed; Intra pictures are directly transformed, whilst Inter ones are motion compensated before entering this stage, resulting in a residual frame to be transformed.

The picture component data is coded in three stages. First, the data arrays are wavelet-transformed using separable wavelet filters and divided into subbands. Then, they are quantized (using RDO quantizes in the reference encoder). Finally, the quantized data is entropy coded [6]. Figure 2 illustrates this procedure.

```
Fig. 2: Coding of picture data [6]
```

The wavelet transform is performed on both columns and rows of each image, resulting in four subbands: Low-Low (LL, corresponding to low frequencies in both horizontal and vertical directions), Low-High (LH, low frequency in horizontal and high frequency in vertical direction), High-Low (HL, high frequency in horizontal and low frequency in vertical direction) and High-High (HH, corresponding to high frequencies in both horizontal and vertical directions). The LL band is then further processed, obtaining the decomposition as in Figure 3.

Parent-child relationships between coefficients are also exploited when entropy coding wavelet data. This is based on the correlation between coefficients that represent the same area inside the picture, as shown in Figure 3.

Quantization follows by the use of a dead-zone quantizer (shown in Figure 4), which is characterized by the zero interval twice as wide as the uniform one. This presents the advantage of simple implementation and good denoising results. The selection of quantizers is then performed by the minimization of a Lagrangian combination of rate and distortion.

B. Motion Estimation

This task is specific to the encoder, and represents the heaviest part of the encoding process. Dirac uses a 3-stage approach, using luminance samples only. Firstly, motion vectors are found for every block at pixel accuracy using hierarchical motion estimation. In the second stage, these vectors are refined to sub-pixel accuracy (supporting until 1/8 pixel accuracy). In the third stage, mode decision is done, choosing which predictor to use and how to aggregate motion vectors by grouping blocks with similar motion together.

```
Fig. 3: Wavelet subband decomposition and parent-child relationship [6]
```

Dirac uses 3 types of picture. Intra pictures (I) are coded without reference to other pictures in the sequence. Level 1 pictures (L1) and Level 2 pictures (L2) are both inter pictures, i.e., they are coded with reference to other previously coded pictures. L1 pictures are forward-predicted only (they are also known as P-pictures) whereas L2 pictures are also known as B pictures (predicted from both earlier and later references).

```
Fig. 4: Quantizers:(a)Uniform quantizer.(b)Dead-zone quantizer[6]
```
C. Motion Compensation

Dirac uses Overlapped Block-based Motion Compensation (OBMC), which avoids block-edge artifacts that would be expensive to code using wavelets. Any block sizes can be used, with any degree of overlap selected. This information must be sent to the decoder. However, there must be an exact number of macroblocks (4x4 set of blocks) horizontally and vertically, which is achieved by padding the data.

The OBMC method lies on a separable linear ramp mask, which acts as a weighed function on the predicting block. A pixel may fall within only one block or in up to four if it lies at the corner of a block, as shown in Figure 5.

D. Entropy Coding

Wavelet coefficient entropy coding in Dirac is based on three stages: binarization, context modeling and adaptive arithmetic coding. Figure 6 illustrates this process.

The first step is the binarization of the coefficients: transforming them to bits. This is important because, otherwise, the values of the coefficients would happen very rarely, which would reduce coding efficiency. Dirac uses an interleaved exp-Golomb binarization scheme.

The context modeling stage follows, taking into account the dependencies that still exist between coefficients (especially zero coefficients, which tend to be together). Thus, the context is selected and fed into the arithmetic coder.

Fig. 5: OBMC scheme showing overlapping blocks (darker-shaded areas) [6].

The arithmetic coder, which is the third step of the entropy coding function, produces variable-length codes for entire sequences of symbols based on their probabilities, achieving an average length codeword very close to the limit imposed by Information Theory.

A similar scheme is used for entropy coding of motion vectors, except that a prediction stage is employed before binarization and that different context models are used.

Fig. 6: Entropy coding block diagram for wavelet coefficients [6]

III. SIMULATION

Dirac and H.264 were used for compressing 8 videos comprising different degrees and kinds of motion with various sizes: canal352 (352x288 – CIF resolution , 68 frames), snow352 (352x288 – CIF resolution , 75 frames), squirrel352 (352x288 – CIF resolution , 75 frames), canal720 (720x576 – SDTV 576p resolution , 50 frames), snow720 (720x576 – SDTV 576p resolution , 50 frames), squirrel720 (720x576 – SDTV 576p resolution , 50 frames), snow1280 (1280x720 – HDTV 720p resolution , 30 frames) and squirrel1280 (1280x720 – HDTV 720p resolution , 30 frames). These sample videos are available in raw RGB format at [7], and a frame rate of 30Hz is used. They were converted to YUV format, with chrominance sampling 4:2:2.

Each video was compressed many times with the same encoder for different target bit rates, with the objective of analyzing the performance constrained to the given bit rate.

The encoders used were the reference encoders for both codecs implemented in C++. For Dirac codec, the 1.0.2 version was employed [8] while the JM 15.1 was used for H.264 [9]. The simulations were done using a PC equipped with Intel Pentium M 740, 1.73GHz, 768 MB RAM.

For providing visual quality comparison, two distortion metrics were used: the Peak Signal-to-Noise Ratio (PSNR) and the Mean Structural Similarity Index (MSSIM) [10]. They were computed for the luminance samples of each frame within each sample video and the mean values for the entire videos were taken into account. The PSNR, based on the MSE, is the simplest and most widely used measure of quality. However, images with the same PSNR may have very different type of errors, some of which are much more visible than others. The MSSIM, on the other hand, quantify better the visibility of errors because the Human Visual System (HVS) is in general highly adapted for extracting structural information, which is measured by this index. To calculate it, the algorithm based on [10] was used.

The latency of each encoder is calculated by the encoding time per frame. Although the implementations here are not optimized, they can provide a good idea of the computational complexity of each encoder.

A. Results

Examples of frame reconstruction are presented in Figure 7. The original frame is provided and can be visually
compared to the reconstructed ones.

Results concerning visual quality comparison are provided in Figures 8 and 9, presenting PSNR and MSSIM results respectively, for a wide range of bit rates in each case.

Regarding latency, a substantial difference was noticed for the results obtained. H.264 takes for a video in some cases more than 200 times the encoding time Dirac needs to encode the same video. In average, the former takes 145 times the encoding time of the latter for 352x288 format videos (of the order of 36-40s for H.264 and 0.2-0.3s for Dirac per frame), while this value achieves 138 for 720x576 format videos (of the order of 145-155s for H.264 and 0.6-1.5s for Dirac per frame) and 153 for 1280x720 format videos (of the order of 360-410s for H.264 and 1.8-3.2s for Dirac per frame).

**B. Discussion**

It can be immediately inferred by Figures 8 and 9 that H.264 performs better than Dirac. For each video analyzed, H.264’s PSNR is at least 2dB greater than Dirac’s for a wide range of

Fig. 7: Frame reconstruction. (a), (b), (c): canal352, frame #10. (a) original, (b) Dirac with bit rate = 761.001kbps, PSNR = 32.946dB, MSSIM = 0.913318, (c) H.264 with bit rate = 784.540kbps, PSNR = 37.2378dB, MSSIM = 0.956804. (c), (d), (e): snow352, frame #10. (d) original, (e) Dirac with bit rate = 983.619kbps, PSNR = 39.8511dB, MSSIM = 0.973384, (f) H.264 with bit rate = 774.650kbps, PSNR = 42.6733dB, MSSIM = 0.978814. (g), (h), (i): squirrel352, frame #10. (g) original, (h) Dirac with bit rate = 870.124kbps, PSNR = 36.9801dB, MSSIM = 0.968211, (i) H.264 with bit rate = 771.090kbps, PSNR = 36.5225dB, MSSIM = 0.963992.
bit rates.

MSSIM results also show superior performance of H.264 codec. Nevertheless, as this index tends to the unity for both codecs, the difference between them decreases in most cases along the bit rate axis, while PSNR analysis gives us the opposite: the difference increases along the bit rate axis.

This can be explained by observing that PSNR measures take into account only the differences of luminance values for each pixel, while the MSSIM index is calculated based on structural properties of the frames. Beyond a certain bit rate level, image structure changes slower because the main elements of the frame are already constructed, whilst PSNR is still calculating only differences of luminance values.

Encoding time results must be considered very carefully: they were done using reference implementations for both codecs. Optimized versions were not used. However, these results provide a good idea of the codec complexity. They have to be analyzed comparatively, for they were done using the same hardware under the same conditions.

It is then possible to remark that H.264 presents much greater complexity when compared to Dirac. As it was said above, Dirac consists on a very simple architecture, based on a

Fig. 8: Objective visual quality comparison, in terms of PSNR for each analyzed video. (a) canal352, (b) snow352, (c) squirrel352, (d) canal720, (e) snow720, (f) squirrel720, (g) snow1280, (h) squirrel128
small number of core tools. This leads to the conclusion that Dirac might be well-suited for real-time applications, such as videoconferencing.

IV. CONCLUSION

This paper investigates the performance of the Dirac Video Codec, a state-of-the-art scheme for video compression. It is compared with the well-established H.264/AVC codec in terms of visual quality (objectively measured by PSNR and MSSIM) and latency for various target bit rates. Eight short videos with different degrees and kinds of motion are used for simulations.

Video compression is critical for video transmission and storage, and the main contribution of this work is to analyze comparatively recent developments in the domain.

Results suggest that the simplicity of Dirac’s design is well-suited for applications requiring very low latency, such as videoconferencing. On the other hand, in cases where video is entirely coded before usage delay restrictions do not apply and then H.264 would be a better choice.

It must be noticed that Dirac is a codec that has just had its first version released; improvements will certainly come, and the results achieved so far make it a promising video codec.

The fact that it is an open codec is another of its advantages; as video use is increasing rapidly, reduction of costs will certainly become a key issue.
ACKNOWLEDGMENT

The authors would like to thank CAPES, FAEPEX/Unicamp and RH-TVD/CAPES for the financial support.

REFERENCES


