PERFORMANCE ANALYSIS AND COMPARISON OF THE DIRAC VIDEO CODEC WITH H.264/MPEG-4 PART 10 AVC

ARUNA RAVI
Electrical Engineering Department
University of Texas at Arlington
Arlington, Texas 76019, USA
aruna.ravi@mavs.uta.edu

K. R. RAO
Electrical Engineering Department
University of Texas at Arlington, Box 19016
Arlington, Texas 76019, USA
rao@uta.edu

Received 4 January 2010
Revised 28 July 2010

Dirac is a hybrid motion-compensated state-of-the-art video codec that can be used without the payment of license fees. It can be easily adapted for new platforms and is aimed at applications ranging from HDTV to web streaming. The objective of this paper is to analyze Dirac video codec based on several input test sequences, and compare its performance with H.264/MPEG-4 Part 10 AVC. Both Dirac and H.264 are implemented using different video test sequences at various constant “target” bit rates ranging from 10 KBps to 200 KBps at image resolutions from QCIF to SD. The results have been recorded graphically and we arrive at a conclusion whether Dirac’s performance is comparable to H.264. We also research whether Dirac outperforms H.264/MPEG-4 Part 10 in terms of computational speed and efficiency.

Keywords: Dirac; H.264/MPEG-4 Part 10 AVC; performance comparison; wavelet transform.

AMS Subject Classification: 94A08, 68P30, 68U10

1. Introduction

Video compression is used to exploit limited storage and transmission capacity as efficiently as possible which is important for the internet and high definition media. Dirac\(^1\) is an open and royalty-free video codec developed by the BBC. It aims to provide high-quality video compression from web video up to HD,\(^4\) and as such competes with existing formats such as H.264\(^11\)–\(^14\) and SMPTE VC-1.\(^17\) Dirac can compress any size of picture from low-resolution QCIF (176 × 144 pixels) to HDTV
(1920 × 1080) and beyond, similar to common video codecs such as the ISO/IEC Moving Picture Experts Group (MPEG)’s MPEG-4 Part 2\textsuperscript{18–27} and Microsoft’s SMPTE VC-1.

Dirac employs wavelet compression, instead of the discrete cosine transforms used in most other codecs. The Dirac software is a prototype implementation that can freely be modified and deployed. Dirac’s decoder in particular is designed to be fast and more agile than other conventional decoders. The resulting specification is simple and straightforward to implement and is optimized for real-time performance.\textsuperscript{1}

2. Dirac Architecture

In the Dirac codec, image motion is tracked and the motion information is used to make a prediction of a later frame. A transform is applied to the prediction error between the current frame and the previous frame aided by motion compensation and the transform coefficients are quantized and entropy coded. Temporal and spatial redundancies are removed by motion estimation, motion compensation and discrete wavelet transform respectively. Dirac uses a flexible and efficient form of entropy coding called arithmetic coding which packs the bits efficiently into the bit stream.\textsuperscript{1}

2.1. Dirac encoder

In the Dirac encoder (Fig. 1) the entire compressed data is packaged in a simple byte stream. This has synchronization, permitting access to any frame quickly and efficiently — making editing simple. The structure is such that the entire byte stream can be packaged in many of the existing transport streams. This feature allows a wide range of coding options, as well as easy access to all the other data transport systems required for production or broadcast metadata.

Streaming video quality is partly dependent upon the video encoding process and the amount of bandwidth required for it to be viewed properly. While encoding a video, a high degree of compression is applied to both the video and audio tracks so that it will stream at this speed.\textsuperscript{1,21}

2.2. Dirac decoder

The Dirac decoder (Fig. 2) performs the inverse operations of the encoder. This decoder implementation is designed to provide fast decoding whilst remaining portable across various software platforms.

3. Stages of Encoding and Decoding in Dirac

3.1. Wavelet transform

The 2D discrete wavelet transform provides Dirac with the flexibility to operate at a range of resolutions. This is because wavelets operate on the entire picture at
once, rather than focusing on small areas at a time. In Dirac, the discrete wavelet transform plays the same role as the DCT in MPEG-2 in de-correlating data in a roughly frequency-sensitive way, whilst having the advantage of preserving fine details better than block-based transforms. Synthesis filters can undo the aliasing
introduced by critical sampling and perfectly reconstruct the input. The wavelet transform is constructed by repeated filtering of signals into low- and high-frequency parts. For two-dimensional signals, this filtering occurs both horizontally and vertically as shown in Fig. 3. At each stage, the low horizontal/low vertical frequency sub-band is split further, resulting in logarithmic frequency decomposition into sub-bands.

3.2. Scaling and quantization
Scaling involves taking frame data after application of wavelet transform and scaling the coefficients to perform quantization. Quantization employs a rate distortion optimization algorithm to strip information from the frame data that results in as little visual distortion as possible.

Dirac uses dead-zone quantization technique (Fig. 4) which differs from uniform quantization by making the first set of quantization steps twice as wide. This method is simple, efficient and allows Dirac to perform coarser quantization on smaller values.

---

Fig. 3. Wavelet transform frequency decomposition.

Fig. 4. Dead-zone quantizer with quality factor (QF).
3.3. Entropy coding

Entropy coding is applied after wavelet transform to minimize the number of bits used. It consists of three stages: binarization, context modeling and arithmetic coding as shown in Fig. 5. The purpose of the first stage is to provide a bit stream with easily analyzable statistics that can be encoded using arithmetic coding, which can adapt to those statistics, reflecting any local statistical features. The context modeling in Dirac is based on the principle that whether a coefficient is small or not is well-predicted by its neighbors and its parents. Arithmetic coding performs lossless compression and is both flexible and efficient.

3.4. Motion estimation and motion compensation

Motion estimation exploits temporal redundancy in video streams by looking for similarities between adjacent frames. An example of motion estimation technique used in the Dirac reference software is shown in Fig. 6.

Motion compensation is used to predict the present frame. Dirac uses overlapped block-based motion compensation (OBMC) to achieve good compression and avoid block-edge artifacts which would be expensive to code using wavelets. OBMC allows interaction of neighboring blocks (Fig. 7). Dirac also provides sub-pixel motion compensation with motion vectors and thereby improves the prediction rate up to 1/8th pixel accuracy. Techniques such as predicting a frame using only motion information and predicting a frame to be nearly identical to a previous frame at low bit rates are also supported.

3.5. Decoder

The decoding process is carried out in three stages as shown in Fig. 8. At the first stage, the input encoded bit stream is decoded by the entropy decoding technique. Next, scaling and inverse quantization is performed. In the final stage, inverse wavelet transform is applied on the data to produce the decoded, uncompressed video output. A tradeoff is made between video quality and motion vector bit rate.
Fig. 6. Hierarchical motion estimation.\textsuperscript{10}

Fig. 7. Overlapping blocks in OBMC.\textsuperscript{4}

Fig. 8. Stages of decoding in Dirac.
4. Implementation

Dirac reference software is fully implemented in C++ programming language which allows object-oriented development on all common operating systems. The C++ code compiles to produce libraries for common functions, motion estimation, encoding and decoding, which have an interface that allows them to be called from C. An application programmer’s interface can be written in C so that it can be kept simple and integrated with various media players, video processing tools and streaming software.\(^1\)

4.1. Code structure overview

The Dirac codec has an object-orientated code structure. The encoder consists of objects which take care of the compression of particular “objects” within a picture sequence. In other words, the compression of a sequence, a frame and a picture component are defined in individual classes as shown in Fig. 9.

The SequenceCompressor compress each frame of the sequence. SequenceCompressor::CompressNextFrame() calls FrameCompressor::Compress() up to three times in an iterative process to ensure frames are encoded with quality consistent throughout the video sequence.\(^4\) If there is a large difference between the desired quality (specified as a command-line option) and actual quality, coding parameters are altered and the frame is encoded again.

MotionEstimator::DoME() manages the motion estimation process which utilizes a hierarchical block-match method. The motion estimation process is split into three stages: pixel accurate block-matching, sub-pixel accurate block-matching and mode decision as shown in Fig. 10.\(^4\)

The motion estimation process has three stages:

(i) Pixel accurate block-matching: Initial pixel accurate block-match is carried out by PixelMatch::DoSearch() to produce the initial ME vectors.

(ii) Sub-pixel accurate block-matching: A SubpelRefine object is instantiated and its DoSubpel() method called. This method loop over all the ME vectors and refining them iteratively using the 1/2 pixel accuracy, then 1/4th pixel and finally 1/8th pixel, each time using the previous result as a guide.

(iii) Mode decision: The goal of mode decision is to evaluate the number of motion vectors assigned to a macroblock and which reference is used for block motion prediction.

4.2. Simplicity and relative speed of encoding

Due to the relative simplicity of the Dirac reference software, its encoding speed is found to be much faster compared to the H.264 JM 17.1 reference software. The decoding speeds of both the codecs are found to be comparable.

There is also a high-speed ANSI C implementation of Dirac called Schrödinger\(^4\) which is a cross-platform implementation of the Dirac video compression
Fig. 9. The Dirac compression engine's structure.\textsuperscript{48}
Fig. 10. Motion estimation classes.
specification as a C library. Many media frameworks such as GStreamer\textsuperscript{52} and ffmpeg\textsuperscript{53} and applications such as VLC\textsuperscript{54} use Schrödinger to encode and decode Dirac video. Schrödinger outperforms Dirac-research in most encoding situations, both in terms of encoding speed and visual quality.\textsuperscript{19}

There are quite a few research papers\textsuperscript{3,46,47} suggesting techniques to optimize Dirac’s entropy coder. According to one paper,\textsuperscript{46} a much faster video codec can be achieved by replacing the original arithmetic coder of the Dirac algorithm with an accurately configured M-coder. The new arithmetic coder is three times faster for high bit rates and even outperforms the original compression performance.

Another paper\textsuperscript{47} suggests a rate control algorithm based on optimization of quality factor for Dirac codec. This method exploits the existing constant-quality control, which is governed by a parameter called quality factor (QF) to give a constant bit rate. A mathematical model called the rate-quality factor (R-QF) is derived to generate optimum QF for the current coding frame using the bit rate resulting from the encoding of the previous frame in order to meet the target bit rate.

In another research project\textsuperscript{48} different approaches to encoder optimization such as multi-threading, Streaming SIMD (Single Instruction Multiple Data) Extensions (SSE)\textsuperscript{49} and compilation with Intel’s C/C++ compiler\textsuperscript{50} using the Visual Studio add-in\textsuperscript{51} have been extensively discussed.

5. Results

Objective test methods attempt to quantify the error between a reference and an encoded bit stream.\textsuperscript{5} To ensure the accuracy of the tests, each codec must be encoded using the same bit rate.\textsuperscript{5,47}

Since the latest version of Dirac includes a constant bit rate (CBR) mode, the comparison between Dirac and H.264’s performance was produced by encoding several test sequences at different bit rates. By utilizing the CBR mode within H.264, we can ensure that H.264 is being encoded at the same bit rate as that of Dirac.\textsuperscript{47}

Objective tests are divided into three sections, namely (i) compression, (ii) structural similarity index (SSIM),\textsuperscript{16} and (iii) peak-to-peak signal-to-noise ratio (PSNR). The test sequences “Miss-America” QCIF (176 × 144),\textsuperscript{23} “Stefan” CIF (352 × 288)\textsuperscript{23} and “Susie” standard-definition (SD) (720 × 480)\textsuperscript{24} are used for evaluation. The two methods are very close and comparable in compression, PSNR and SSIM. Also, a significant improvement in encoding time is achieved by Dirac, compared to H.264 for all the test sequences.

5.1. Compression ratio test

By evaluating the magnitude of the *.drc and *.264 files, compression ratio results in comparison to the file size of the original sequence are produced from Dirac and H.264 respectively. Figures 11–13 show a comparison of how Dirac and H.264 perform in compression for QCIF, CIF and SDTV sequences respectively. Dirac achieves slightly higher compression ratios for lower bit rates than H.264 in case of
Fig. 11. Compression ratio comparison of Dirac and H.264 for “Miss-America” QCIF sequence.

Fig. 12. Compression ratio comparison of Dirac and H.264 for “Stefan” CIF sequence.

Fig. 13. Compression ratio comparison of Dirac and H.264 for “Susie” SDTV sequence.
QCIF sequences. At higher QCIF bit rates both Dirac and H.264 achieve similar compression. H.264 provides better compression at higher bit rates especially for CIF, SD media.

5.2. **SSIM test**

Structural similarity (SSIM) operates by way of comparing local patterns of pixel intensities that have been normalized for luminance and contrast. This basically means that SSIM is computed based on the combination of luminance similarity, contrast similarity and structural similarity encompassed into one value. The maximum possible value for SSIM is 1, which indicates the encoded sequence is an exact replica of the reference sequence. SSIM is an alternative method of objectively evaluating video quality. H.264 achieves slightly better SSIM than Dirac as seen in Figs. 14–16.

![SSIM vs Bitrate for QCIF](image)

**Fig. 14.** SSIM comparison of Dirac and H.264 for “Miss-America” QCIF sequence.

![SSIM vs Bitrate for CIF](image)

**Fig. 15.** SSIM comparison of Dirac and H.264 for “Stefan” CIF sequence.
5.3. **PSNR test**

H.264 achieves slightly higher PSNR than Dirac (about 3–4 dB) as seen in Figs. 17–19.

Tables 1–3 and Figs. 20–22 show the performance comparison of Dirac with H.264/MPEG-4 at constant bit rates (CBR) ranging from 10–200 KBps for QCIF, CIF and SD sequences, respectively.

6. **Conclusion**

Overall Dirac codec is very promising. According to BBC Research, Dirac was developed with a view to optimize its performance with compression ratio and perceptual quality at the forefront. Its simplistic nature provides robustness and fast compression which is very beneficial, therefore to a large extent Dirac has succeeded in its aim.\(^5\)
Despite its early reference code, it is creditable that Dirac produces good results relative to state-of-the-art codecs such as H.264. SSIM indicates that H.264 has slightly greater improvement in terms of quality. The choice of the codec will depend on the end user’s application which will decide if the enormous cost in royalty fees justifies the additional increase in quality (as in the case of H.264/MPEG-4). Both Dirac and H.264 maintain a near constant quality at low bit rates, which is beneficial for applications such as video streaming.

In conclusion, Dirac is an extremely simple yet robust codec, and has the potential to achieve compression results very close to H.264, at reduced complexity and without royalty payments. SMPTE VC-2 is based on the intra-frame version of Dirac called DiracPro. It is being standardized by SMPTE (SMPTE S2042). It is aimed at professional applications and not for end user distribution.
Table 1. Performance comparison of Dirac with H.264 at CBR for QCIF sequence.

<table>
<thead>
<tr>
<th>CBR (KB/s)</th>
<th>Dirac</th>
<th></th>
<th></th>
<th></th>
<th>H.264</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Size (KB)</td>
<td>Compression</td>
<td>PSNR-Y (dB)</td>
<td>MSE (Y)</td>
<td>SSIM</td>
<td>Size (KB)</td>
<td>Compression</td>
<td>PSNR-Y (dB)</td>
</tr>
<tr>
<td>10</td>
<td>59</td>
<td>95</td>
<td>38.913</td>
<td>8.353</td>
<td>0.966</td>
<td>63</td>
<td>90</td>
<td>44.462</td>
</tr>
<tr>
<td>20</td>
<td>120</td>
<td>46</td>
<td>42.911</td>
<td>3.326</td>
<td>0.981</td>
<td>123</td>
<td>45</td>
<td>45.729</td>
</tr>
<tr>
<td>40</td>
<td>247</td>
<td>23</td>
<td>44.648</td>
<td>2.230</td>
<td>0.986</td>
<td>243</td>
<td>23</td>
<td>47.257</td>
</tr>
<tr>
<td>80</td>
<td>477</td>
<td>12</td>
<td>46.180</td>
<td>1.587</td>
<td>0.988</td>
<td>481</td>
<td>12</td>
<td>49.054</td>
</tr>
<tr>
<td>100</td>
<td>594</td>
<td>9</td>
<td>46.640</td>
<td>1.410</td>
<td>0.989</td>
<td>601</td>
<td>9</td>
<td>49.826</td>
</tr>
<tr>
<td>160</td>
<td>949</td>
<td>6</td>
<td>47.717</td>
<td>1.100</td>
<td>0.991</td>
<td>911</td>
<td>6</td>
<td>52.073</td>
</tr>
<tr>
<td>200</td>
<td>1,186</td>
<td>5</td>
<td>48.420</td>
<td>0.936</td>
<td>0.992</td>
<td>912</td>
<td>6</td>
<td>52.077</td>
</tr>
</tbody>
</table>

Note: *Indicates encoded file size including all 150 frames after compression.

Table 2. Performance comparison of Dirac with H.264 at CBR for CIF sequence.

<table>
<thead>
<tr>
<th>CBR (KB/s)</th>
<th>Dirac</th>
<th></th>
<th></th>
<th></th>
<th>H.264</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Size (KB)</td>
<td>Compression</td>
<td>PSNR-Y (dB)</td>
<td>MSE (Y)</td>
<td>SSIM</td>
<td>Size (KB)</td>
<td>Compression</td>
<td>PSNR-Y (dB)</td>
</tr>
<tr>
<td>10</td>
<td>146</td>
<td>92</td>
<td>27.468</td>
<td>116.476</td>
<td>0.896</td>
<td>142</td>
<td>94</td>
<td>31.617</td>
</tr>
<tr>
<td>20</td>
<td>285</td>
<td>47</td>
<td>31.613</td>
<td>44.851</td>
<td>0.951</td>
<td>282</td>
<td>48</td>
<td>34.650</td>
</tr>
<tr>
<td>40</td>
<td>559</td>
<td>24</td>
<td>35.296</td>
<td>19.207</td>
<td>0.975</td>
<td>559</td>
<td>24</td>
<td>38.055</td>
</tr>
<tr>
<td>80</td>
<td>1,114</td>
<td>12</td>
<td>39.012</td>
<td>8.164</td>
<td>0.966</td>
<td>1,112</td>
<td>12</td>
<td>42.103</td>
</tr>
<tr>
<td>100</td>
<td>1,386</td>
<td>10</td>
<td>40.343</td>
<td>6.005</td>
<td>0.989</td>
<td>1,389</td>
<td>10</td>
<td>43.134</td>
</tr>
<tr>
<td>160</td>
<td>2,216</td>
<td>6</td>
<td>43.273</td>
<td>3.061</td>
<td>0.992</td>
<td>2,199</td>
<td>6</td>
<td>46.840</td>
</tr>
<tr>
<td>200</td>
<td>2,757</td>
<td>5</td>
<td>44.684</td>
<td>2.211</td>
<td>0.994</td>
<td>2,731</td>
<td>5</td>
<td>48.729</td>
</tr>
</tbody>
</table>

Note: *Indicates encoded file size including all 90 frames after compression.
<table>
<thead>
<tr>
<th>CBR (KB/s)</th>
<th>Dirac</th>
<th>H.264</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Size (KB)</td>
<td>Compression</td>
</tr>
<tr>
<td>10</td>
<td>180</td>
<td>94</td>
</tr>
<tr>
<td>20</td>
<td>388</td>
<td>44</td>
</tr>
<tr>
<td>40</td>
<td>751</td>
<td>22</td>
</tr>
<tr>
<td>80</td>
<td>1,470</td>
<td>11</td>
</tr>
<tr>
<td>100</td>
<td>1,822</td>
<td>9</td>
</tr>
<tr>
<td>160</td>
<td>2,849</td>
<td>6</td>
</tr>
<tr>
<td>200</td>
<td>3,539</td>
<td>5</td>
</tr>
</tbody>
</table>

Note: *Indicates encoded file size including all 25 frames after compression.
Performance Analysis and Comparison of Dirac Video Codec

Fig. 20. Comparison of Dirac and H.264 at CBR = 10 Kbps, QCIF.

Fig. 21. Comparison of Dirac and H.264 at CBR = 100 Kbps, CIF.

Fig. 22. Comparison of Dirac and H.264 at CBR = 100 Kbps, SDTV (progressive).
7. Future Research

This implementation of the Dirac codec is directed towards high-quality video compression from web video up to ultra HD. However, the standard just defines a video codec and has no mention of any audio compression. It is necessary to associate an audio stream along with the video in order to have meaningful delivery of the video to the end user.

The Dirac video codec can be further improved by integrating it with an audio codec such as MPEG Layer 2 (MP2) or the AAC. MP2 is royalty-free, applicable to high quality audio and has performance similar to MP3 at higher bit rates. The Dirac research group at BBC also suggests Vorbis audio codec and FLAC (free lossless audio codec) developed by Xiph.Org Foundation as high quality audio formats available under royalty-free terms that can be used with Dirac video codec.

Hence it is possible to transmit by multiplexing the video and audio coded bit streams to create a single bit stream for transmission and de-multiplexing the streams at the receiving end. This can be followed by synchronization of the audio and video during playback so that it can be suitable for various applications.

References

3. H. Eeckhaut et al., Speeding up Dirac’s entropy coder, in Proc. 5th WSEAS Int. Conf. on Multimedia, Internet and Video Technologies, Greece (2005), pp. 120–125.
10. CMPT 365 Course Slides, School of Computing Science, Simon Fraser University.
29. Daubechies wavelet, Wikipedia.
31. Digital down converter, Wikipedia.
32. H.264/MPEG-4 AVC, Wikipedia.
33. M. Fieldler, Implementation of basic H.264/AVC Decoder, seminar paper, Chemnitz University of Technology (June 2004).
34. H.264 encoder and decoder, www.adalta.it/Pages/407/266881_266881.jpg.
41. MPEG Layer II, Wikipedia.
42. MP3/MPEG Layer III, Wikipedia.
43. T. Borer, Dirac coding: Tutorial and implementation, EBU Networked Media Exchange seminar (June 2009).
54. VLC media player, www.videolan.org/vlc/.