Abstract: 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 (encoder and decoder) \cite{1} based on several input test sequences, and compare its performance with H.264 / MPEG-4 Part 10 AVC \cite{11-14}. Analysis has been done on Dirac and H.264 using QCIF, CIF and SDTV video test sequences at various constant ‘target’ bit rates ranging from 10KBps to 200KBps. The test results have been recorded graphically and these indicate that Dirac's performance is comparable to H.264.
Enclosed is the manuscript "Performance analysis and comparison of the Dirac video codec with H.264 / MPEG-4 Part 10 AVC " for possible publication in Journal of Visual Communication and Image Representation.

Please address all your correspondence to me.

K. R. Rao,
Electrical Engineering Department,
The University of Texas at Arlington,
416 Yates Street, Box 19016, Arlington, Texas 76019, USA
Phone: +1 817-272-3478, Fax: +1 817-272-2253, Email: rao@uta.edu
Performance analysis and comparison of the Dirac video codec with H.264 / MPEG-4 Part 10 AVC

Aruna Ravi and K.R. Rao
Electrical Engineering Department, The University of Texas at Arlington, 416 Yates Street, Box 19016, Arlington, Texas 76019, USA.

Abstract
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 (encoder and decoder) based on several input test sequences, and compare its performance with H.264 / MPEG-4 Part 10 AVC. Analysis has been done on Dirac and H.264 using QCIF, CIF and SDTV video test sequences at various constant ‘target’ bit rates ranging from 10KBps to 200KBps. The test results have been recorded graphically and these indicate that Dirac’s performance is comparable to H.264.

Key words: Dirac; H.264 / MPEG-4 Part 10 AVC; Performance comparison

PACS:

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 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, and as such competes with existing formats such as H.264 and WMV 9. Dirac can compress any size of picture from low-resolution QCIF (176x144 pixels) to HDTV (1920x1080) and beyond, similar to common video codecs such as the ISO/IEC Moving Picture Experts Group (MPEG)’s MPEG-4 Part 2 and Microsoft’s WMV 9.

1 Email address: aruna.ravi@mavs.uta.edu
2 Email address: rao@uta.edu, Phone: +1-817-272-3478, Fax: +1-817-272-2253,
URL: http://www-ee.uta.edu/dip

Preprint submitted to Elsevier Science August 2009
However, it promises significant savings in data rate and improvements in quality over these codecs. Some claims have been made that it is even superior to those promised by the latest generation of codecs such as H.264/MPEG-4 AVC or SMPTE's VC-1. \textsuperscript{[20]}

Dirac employs wavelet compression, instead of the discrete cosine transforms used in most other codecs. The Dirac software \textsuperscript{[4][19]} is not intended simply to provide reference coding and decoding. It 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 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. \textsuperscript{[1]} Temporal and spatial redundancies are removed by motion estimation, motion compensation and discrete wavelet transform respectively. Dirac uses a more 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 (Fig. 1) \textsuperscript{[1][2]}
In the Dirac encoder, \textsuperscript{[1][2]} 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.

2.2 Dirac decoder (Fig. 2)
The Dirac decoder \textsuperscript{[1][2]} performs the inverse operations of the encoder
Fig. 1. Dirac encoder architecture

Fig. 2. Dirac decoder architecture
3 Encoding and Decoding in Dirac
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.

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. In one dimension, it consists of the iterated application of a complementary pair of half-band filters followed by sub-sampling by a factor 2 as shown in Fig. 3.

![Fig. 3: Perfect reconstruction analysis and synthesis filter pairs](image)

The synthesis filters can undo the aliasing introduced by critical sampling and perfectly reconstruct the input. The filters split the signal into a LH (low-high), high-frequency (HF) part and the wavelet transform then iteratively decomposes the low-frequency (LF) component to produce an octave-band decomposition of the signal. 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. At each stage, the low horizontal / low vertical frequency sub-band is split further, resulting in logarithmic frequency decomposition into sub-bands.
Wavelet transforms have been proven to provide a more efficient technique than block transforms with still images. Within the Dirac wavelet filters, the data is encoded in 3 stages as shown in Fig. 4.

![Fig. 4: Dirac’s wavelet transform architecture](image)

Daubechies wavelet filters such as the (9, 7) low pass wavelet filter and the (5, 3) high pass filter are used to transform and divide the data in sub-bands which then are quantized with the corresponding RDO (rate distortion optimization) parameters and then variable length encoded. These three stages are then reversed at the decoder. The choice of wavelet filters has an impact on compression performance, as well as encoding / decoding speed in software. Filters are required to have compact impulse response in order to reduce ringing artifacts and other effects so as to represent smooth areas compactly. There are numerous filters supported by Dirac to allow a tradeoff between complexity and performance that are configurable in the reference software.

The discrete wavelet transform packs most of the information into only a few sub-bands (at low frequency) which allows compression to be achieved. Most of the energy is concentrated in the LL sub-band. All the other sub-bands can be coarsely quantized. This process can be repeated to achieve higher levels of wavelet transform. In case of two-dimensional images, wavelet filters are normally applied in both vertical and horizontal directions to each image component to produce four so-called sub-bands termed Low-Low (LL), Low-High (LH), High-Low (HL) and High-High (HH).
In the case of two dimensions, only the LL band is iteratively decomposed to obtain the decomposition of the two-dimensional spectrum as shown in Fig. 5. A Dirac-coded picture is free from block artifacts and is clearly superior to pictures coded by block-based transforms in the case of moving images.

![Wavelet transform frequency decomposition](image)

**Fig. 5: Wavelet transform frequency decomposition**

### 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 a dead-zone quantization as shown in Fig. 6 which differs from orthodox quantization by making the first set of quantization steps twice as wide. This allows Dirac to perform coarser quantization on smaller values compared to other codecs such as the MPEG-4.

![Dead-zone quantizer with quality factor (QF)](image)

**Fig. 6: 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. 7. 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.

![Fig. 7: Dirac’s entropy coding architecture](image)

3.4 Motion estimation

Motion estimation exploits temporal redundancy in video streams by looking for similarities between adjacent frames. Dirac implements hierarchical motion estimation (Fig. 8) in three distinct stages. In the first stage, pixel accurate motion vectors are determined for each block and each reference frame by hierarchical block matching. In the second stage, these pixel-accurate vectors are refined by searching sub-pixel values in the immediate neighborhood. In the final stage, mode decisions are made for each macro-block, determining the macro-block splitting level and the prediction mode used for each prediction unit. This last stage involves further block matching since block motion vectors are used as candidates for higher-level prediction units.

![Fig. 8: Hierarchical motion estimation](image)
During hierarchical motion estimation, Dirac first down converts the size of the current and reference of all types of inter frames (both P and B) using the 12 taps down conversion filter. Down conversion filters are lowpass filters that pass only the desired signal and also perform anti-alias filtering prior to decimation. Any suitable lowpass filter can be used including FIR, IIR and CIC filters. The number of down conversion levels depends upon the frame format.

Dirac also defines three types of frames. Intra (I) frames are coded without reference to other frames in the sequence. Level 1 (L1) frames and Level 2 (L2) frames are both inter frames, that is, they are coded with reference to other previously coded frames. The difference between L1 and L2 frames is that L1 frames are also used as temporal references for other frames, whereas L2 frames are not. A prediction structure for frame coding using a standard group of pictures (GOP) structure is shown in Fig. 9.
Each frame in Dirac may be predicted from up to two reference frames. Prediction modes can be varied by prediction unit, and there are four possibilities: Intra, Reference 1 only, Reference 2 only, and Reference 1 and 2 (bi-directional prediction).

![Fig. 9: Prediction of L1 and L2 frames in Dirac](image)

3.5 Motion compensation
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. OBMC is performed with basic blocks arranged into macro-blocks consisting of a 4x4 array of blocks. Dirac's OBMC scheme is based on a separable linear ramp mask. This acts as a weight function on the predicting block. Given a pixel \( p = p(x,y,t) \) in frame \( t \), \( p \) may fall within only one block or in up to four blocks if it lies at the corner of a block as shown in Fig. 10 where the darker-shade areas show overlapping areas.

[4]
Dirac also provides sub-pixel motion compensation with motion vectors and thereby improves the prediction rate up to 1/8th pixel accuracy. It supports the use of global motion estimates in a few bytes. Techniques such as predicting a frame using only motion information (without transmitting any wavelet coefficients) and predicting a frame to be identical to a previous frame at low bit rates are also supported. It involves using the motion vectors to predict the current frame in such a way as to minimize the cost of encoding residual data.

3.6 Decoder

The Dirac’s decoder implementation is designed to provide fast decoding whilst remaining portable across various software platforms. The decoding process is carried out in three stages as shown in Fig. 11. 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 transform is applied on the data to produce the decoded, uncompressed video output. A trade off is made between video quality and motion vector bit rate. Such techniques can provide substantial bit rate reductions when only a modest quality is required.
4 Implementation
Dirac can be 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]

5 Results
Objective test methods attempt to quantify the error between a reference and an encoded bitstream. [5] To ensure the accuracy of the tests, each codec must be encoded using the same bit rate. [5] Since the latest version of Dirac does include a constant bit rate (CBR) mode, the comparison between Dirac and H.264’s [11-14] 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.

Objective tests are divided into three sections, namely (i) Compression, (ii) Structural similarity index (SSIM), and (iii) Peak signal to noise ratio (PSNR). The test sequences “Miss-America” QCIF (176x144) [23], “Stefan” CIF (352x288) [23] and “Susie” standard-definition (SD) (720x480) [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 12, 13 and 14 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 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 and HD media.

![Compression ratio vs Bitrate at CBR (QCIF)](image)

**Fig. 12:** Compression ratio comparison of Dirac and H.264 for “Miss-America” QCIF sequence

![Compression ratio vs Bitrate at CBR (CIF)](image)

**Fig. 13:** Compression ratio comparison of Dirac and H.264 for “Stefan” CIF sequence
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 Figures 15, 16 and 17.
5.3 PSNR test

H.264 achieves higher PSNR than Dirac as seen in Figures 18, 19 and 20. Since Dirac has not been designed to maximize PSNR, measurements of PSNR performance against other codecs are not meaningful. [1]
Fig. 18: PSNR comparison of Dirac and H.264 for “Miss-America” QCIF sequence

Fig. 19: PSNR comparison of Dirac and H.264 for “Stefan” CIF sequence

Fig. 20: PSNR comparison of Dirac and H.264 for “Susie” SDTV sequence
Table 1: Performance comparison of Dirac with H.264 at CBR for QCIF sequence

<table>
<thead>
<tr>
<th></th>
<th>Dirac</th>
<th></th>
<th></th>
<th>H.264</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CBR(KBps)</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>
<td>MSE(Y)</td>
<td>SSIM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>69</td>
<td>95</td>
<td>38.913</td>
<td>9.353</td>
<td>0.996</td>
<td>83</td>
<td>90</td>
<td>44.167</td>
<td>2.404</td>
<td>0.988</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></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>45.726</td>
<td>1.739</td>
<td>0.987</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>247</td>
<td>23</td>
<td>44.848</td>
<td>2.230</td>
<td>0.986</td>
<td>243</td>
<td>23</td>
<td>47.257</td>
<td>1.223</td>
<td>0.988</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>477</td>
<td>12</td>
<td>48.180</td>
<td>1.587</td>
<td>0.988</td>
<td>481</td>
<td>12</td>
<td>49.054</td>
<td>0.808</td>
<td>0.986</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>684</td>
<td>9</td>
<td>48.840</td>
<td>1.410</td>
<td>0.999</td>
<td>801</td>
<td>9</td>
<td>49.826</td>
<td>0.677</td>
<td>0.995</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></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>
<td>0.403</td>
<td>0.996</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>1185</td>
<td>5</td>
<td>48.420</td>
<td>0.936</td>
<td>0.982</td>
<td>912</td>
<td>6</td>
<td>52.077</td>
<td>0.403</td>
<td>0.995</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*indicates encoded file size including all 150 frames after compression.

Fig. 21: Comparison of Dirac and H.264 at CBR = 10KBps, QCIF

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

|        | Dirac |         |         | H.264 |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|--------|-------|---------|---------|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| CBR(KBps) | Size(KB) | Compression | PSNR-Y(dB) | MSE(Y) | SSIM | Size(KB) | Compression | PSNR-Y(dB) | MSE(Y) | SSIM |
| 10     | 146   | 82      | 27.408   | 118.470| 0.996 | 142     | 84      | 31.617   | 44.816 | 0.955 |
| 20     | 295   | 47      | 31.813   | 44.851 | 0.951 | 282     | 48      | 34.650   | 22.288 | 0.874 |
| 40     | 569   | 24      | 35.208   | 18.207 | 0.976 | 558     | 24      | 36.066   | 10.178 | 0.894 |
| 80     | 1114  | 12      | 38.012   | 8.184  | 0.986 | 1112    | 12      | 42.103   | 4.007  | 0.991 |
| 100    | 1386  | 10      | 40.343   | 8.008  | 0.988 | 1386    | 10      | 43.134   | 3.160  | 0.992 |
| 160    | 2216  | 6       | 43.773   | 3.061  | 0.992 | 2199    | 6       | 48.840   | 1.346  | 0.995 |
| 200    | 2757  | 6       | 44.864   | 2.211  | 0.994 | 2731    | 6       | 48.726   | 0.871  | 0.997 |

*indicates encoded file size including all 50 frames after compression.

Fig. 22: Comparison of Dirac and H.264 at CBR = 100KBps, CIF
Table 3: Performance comparison of Dirac with H.264 at CBR for SD sequence

<table>
<thead>
<tr>
<th>CBR(Kbps)</th>
<th>Dirac</th>
<th>H.264</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size(KB)</strong></td>
<td>Compression</td>
<td>PSNR-Y(dB)</td>
</tr>
<tr>
<td>10</td>
<td>180</td>
<td>94</td>
</tr>
<tr>
<td>20</td>
<td>396</td>
<td>44</td>
</tr>
<tr>
<td>40</td>
<td>751</td>
<td>22</td>
</tr>
<tr>
<td>80</td>
<td>1470</td>
<td>11</td>
</tr>
<tr>
<td>100</td>
<td>1822</td>
<td>9</td>
</tr>
<tr>
<td>160</td>
<td>2849</td>
<td>6</td>
</tr>
<tr>
<td>200</td>
<td>3539</td>
<td>5</td>
</tr>
</tbody>
</table>

*Indicates encoded file size including all 25 frames after compression.

![Dirac output sequence, CBR = 100KBps](image1)
![H.264 output sequence, CBR = 100KBps](image2)

Fig. 23: Comparison of Dirac and H.264 at CBR = 100KBps, SDTV

### 6 Conclusions

Overall Dirac is very promising. According to BBC R&D, Dirac was developed with a view to optimize its performance be it compression ratio and perceptual quality at the forefront. Its simplistic nature provides robustness which is a very beneficial feature, therefore to a large extent Dirac has succeeded in its aim. \(^{[5]}\)

Expert viewing suggests that, despite its simple toolset, Dirac is very comparable to other state-of-the-art codecs such as H.264. \(^{[5]}\) SSIM indicates that H.264 has slightly greater improvement. However the question remains whether the enormous cost in royalty fees justifies the additional increase in quality. \(^{[5]}\) 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, nearly equaling H.264 in performance sans the cost of royalty fees. 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.

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 or 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


[10] /CMPT 365 Course Slides/, School of Computing Science, Simon Fraser University,fig3:


[21] Dirac video codec - A programmer's guide:
http://dirac.sourceforge.net/documentation/code/programmers_guide/toc.htm
[24] Video test sequences ITU601:
http://www.cipr.rpi.edu/resource/sequences/itu601.html
[30] Daubechies wavelet filter design: http://cnx.org/content/m11159/latest/
[34] H.264 encoder and decoder: http://www.adalta.it/Pages/407/266881_266881.jpg
[40] FLAC - Free Lossless Audio Codec: http://flac.sourceforge.net/
[41] Vorbis: http://www.vorbis.com/