Dirac — video compression using open technology

Tim Borer and Thomas Davies
BBC Research & Development, UK

The distribution, delivery and storage of video are core activities for broadcasters. In the digital world, compression is used to exploit limited storage and transmission capacity as efficiently as possible. The BBC is developing a video compression technology, called Dirac, so that we can understand the technology and use it at reasonable cost and without restrictions.

Dirac is a hybrid motion-compensated codec that uses modern techniques such as wavelet transforms and arithmetic coding. It is an open technology which means that it is freely available and can be used without the payment of licence fees. Open technology is well suited to the business model of public service broadcasters as it allows open collaboration by those interested in its future development.

We are in the middle of a digital revolution in which the fields of broadcasting, telecoms and IT are merging together. ***Video compression*** is at the heart of this process. Compression is used almost everywhere that video is used in order to exploit limited storage capacity and data bandwidths as efficiently as possible – in programme production, storage, distribution, as well as in broadcasting the finished product.

At opposite ends of the quality scale, compression is vital to both HDTV / D-Cinema, and to Web streaming and mobile phones. Domestically, it is ubiquitous in consumer electronic equipment such as DVD players, DTV receivers, video cameras, on the Internet and on personal computers. Clearly this is a key technology that broadcasters must understand and use efficiently and cost effectively.

For many years the BBC has played a leading role in the development of digital broadcast technology. Not surprisingly for such a crucial technology, the BBC has been investigating state-of-the-art video compression. In March 2004, BBC Research & Development released experimental software for a new video codec called ***Dirac***. Dirac is a state-of-the-art video codec aimed at applications from HDTV to web streaming. It is intended to be an open technology that anyone or any organization may use, for any purpose, without licence fees.

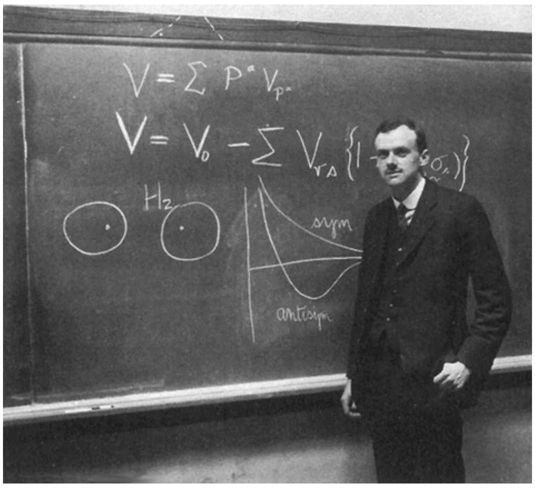

Figure 1
Paul Dirac

The Dirac codec is named after the British Physicist, Paul Dirac. Although little known, Dirac (1902 – 1984) was one of the most influential scientists of the 20th century [1][2]. In 1933 he shared the Nobel Prize for Physics with Erwin Schrödinger for his contributions to quantum mechanics. Speaking in 1995 at a ceremony to unveil a commemorative plaque to Dirac in Westminster Abbey, London, Stephen Hawking – the current Lucasian Professor of Mathematics at Cambridge (a post previously held by Dirac) – said *"Dirac has done more than anyone this century, with the exception of Einstein, to advance physics and change our picture of the universe"*.

This article is intended to give an overview of the Dirac video compression system. It discusses what we are trying to do and our rationale for doing so, it highlights the Dirac technologies and software and discusses some features of the performance of the Dirac codec and what the future might hold.

Dirac in context

Uncompressed standard definition video [1], straight from the camera, requires about 200 Mbit/s of data. This is far too much data to store or transmit. So, in the early 1990s, ***MPEG-2*** video compression [3] emerged and became the pre-eminent compression system for broadcast applications. It can reduce the raw data-rate from the camera by a factor of about 50. MPEG-2 was thus found to be suitable for the widespread adoption of SDTV and DVDs.

Around the turn of this century, with the explosion of digital technology, it became clear that the broadcasting landscape was changing. Many new technologies, such as HDTV, Internet streaming and IPTV (Broadband TV) were coming to the fore. The venerable MPEG-2 video compression system was beginning to show its age. A new system with better compression performance was plainly needed to make good use of the limited bandwidth available.

MPEG-4 Part 2 [4], standardized in 1999, improved on MPEG-2, but its 15 to 20% coding gains did not justify the expensive changes that would be necessary for its widespread adoption.

In the last few years, a number of other compression technologies – including ***H.264*** (also known as *AVC* and *MPEG-4 Part 10*) [5], ***Windows Media*** [6], ***Real Video*** [7] and ***On2*** [8] – have emerged which improve substantially on the performance of the older MPEG-2 standard, typically by a 50% reduction in bitrate for the same quality. This means that, in principle, we can now provide streaming media quality at a few hundred kbit/s, SDTV at about 2 Mbit/s and HDTV at about 10 Mbit/s.

Open technology

Given the surfeit of other codecs, why is there a need for Dirac and what does it provide to broadcasters that the others don't? The key feature of Dirac, in contrast to the other codecs, is that it is ***open technology***. Dirac has been designed to avoid patent infringement. This means that it may freely be used by anyone without the payment of royalties. This may seem like a minor issue but it could have a profound impact on the uptake of digital technology and, particularly, the way it is used by public service broadcasters.

It is probably worth clarifying the difference between "open standards" and "open technology", two concepts that are often confused. Broadcasters have always preferred open standards to vendor-specific technology, so that they can gain the benefits of interoperability and a competitive market place. Open standards are published and may be read by anyone, but implementers and users typically pay royalties to the owners of patents embodied in the standard. So open standards can be proprietary in the sense that the technology is owned and you may have to pay to use it. Open tech-

[1] "Standard Definition" here means the resolution of European PAL broadcasting, i.e. "625 line" television with a resolution of 720 pixels by 576 lines.

nology, on the other hand, takes openness a stage further and gives you the right to use the technology, for any purpose, without payment.

The increasing delivery of video via the Internet places new demands on public service broadcasters such as the BBC. Unlike traditional broadcasting, the cost of distribution via the Internet increases with the number of users. Whilst this is acceptable for a subscription service, it does not match the fixed-revenue model of public service broadcasting. One of these costs is simply that of the bandwidth needed to transmit the video, which can be minimized by using an efficient codec and by adopting distributed-delivery methods such as peer-to-peer and multicasting. Another significant cost is that of licence fees if proprietary technology is used. Such fees are payable even for open standards such as MPEG. Whilst these costs are manageable initially, they could become prohibitive if public service broadcasters try to scale up to millions of simultaneous users, or if new services are deployed which were not envisaged in the original licence agreements.

So it is difficult for conventional codec technologies (including open standards) to fit the model of Internet delivery used by public service broadcasters: the licence fees for these codecs are typically set with subscription business models in mind. “Open Technology” is needed that may be used freely without royalty payment, regardless of the number of users or the new types of services that are delivered.

The issue of open technology is, however, wider than simply trying to contain the costs for public service broadcasters. A fundamental principle of the BBC is universal access to all its services [9]. This means the BBC has an obligation to make its content freely available, independent of platform, without proprietary lock-in. Furthermore, the BBC is looking at new technologies such as facilities to allow download of almost all BBC programmes for a seven-day period after they are first broadcast. In the future, the BBC will examine ways to extend this access to its archive content. Ultimately, unlocking the full public-service value of BBC content will require its availability to the public where and when they want it and on a wide range of platforms such as broadband, peer-to-peer networks and mobile devices [9].

Achieving the free availability of content and the take-up of digital technology is facilitated by open technology and inhibited by proprietary technology. The platforms on which video is watched will become increasingly diverse. Proprietary codecs are unlikely to support all these platforms. Open-standard codecs, such as H.264, may be able to support multiple platforms but the licensing requirements are still an issue. In a digital world, where video is streamed around the home, shared on peer-to-peer networks, downloaded and viewed on demand, and perhaps edited and re-coded by consumers, there will be many codecs in each household. In this scenario, even modest licensing costs begin to add up, which inhibits the uptake of the new technology.

Perhaps more important than the costs are the difficulties associated with licensing itself. Many new platforms – such as set-top boxes, video cameras, hard disk recorders, PDAs and an increasing number of PCs – use the open source Linux operating system, which is an “open technology”. In part, they do this because of its freedom from licensing. But the development and use of video codecs on Linux has been held back by the need to pay codec royalties. The open source development model means that it must be possible to distribute it freely, and free distribution does not allow the payment of royalties. Key video playback technologies on Linux, such as MPlayer [10], have ambiguous legality, which inhibits the use of video technology on Linux. An open technology video codec is therefore essential.

Being “open” guarantees the continued availability of the technology. Broadcasters rely on the longevity of technology so that they can access their archives. Consumers expect continued support for technology so that they can access their own collections. Many proprietary codecs have already come and gone.

This is why we are releasing standard fully-portable software (written in ISO C++) that can be implemented on any platform. Dirac will continue to be available for many years and can easily be recreated for new platforms.

For proof of the effectiveness of open technology, you need look no further than the Internet itself. Most of the technology of the Internet is open technology. For example, 70% of web servers use the open technology Apache software. The explosive growth of the Internet over the last decade demonstrates the effectiveness of this approach. In essence, in developing the Dirac codec, we have simply been trying to emulate the successful methodology used for the development of the Internet.

The Dirac philosophy

Developing Dirac as an open technology requires a different approach to the conventional way in which standard codecs are designed. Previous video compression standards have been developed under the auspices of the ITU and MPEG. Typically, new standards are developed at conferences of interested parties from industry and academia. This process has great advantages in pooling expertise from a wide variety of sources. However, it can also suffer from a number of drawbacks:

- it can be slow and cumbersome;
- parties have a vested interest in inserting patented technology into the standard – even if it increases the complexity;
- the process results in a standardized bitstream and a reference codec which implements the standard, but which is not practically useable;
- there is little guidance on encoder design.

The Dirac development consists of two parts: a ***compression specification*** for the bitstream and decoder, and ***software*** for compression and decompression. Unlike MPEG standards development, the software is not intended simply to provide reference coding and decoding: it is a prototype implementation that can freely be modified, enhanced and deployed. The decoder implementation, in particular, is designed to provide fast decoding whilst remaining portable across software platforms.

With Dirac, the software has been developed first, and work is ongoing to converge the compression specification and the software. The traditional approach is first to develop a specification and later an implementation. The Dirac approach is quicker and more agile. By focusing on developing "real-world" software, we aim to ensure that the resulting specification is simple and straightforward to implement.

As an open technology, Dirac must be straightforward to understand and easy to use. The key philosophy behind Dirac is "keep it simple". Of course, a modern state-of-the-art codec is, inevitably, quite complex, but we have tried to minimize this in Dirac. Reduced complexity makes Dirac easier to understand, easier to implement and easier to optimize for real-time performance. To complement this philosophy, we also aim to provide copious documentation, some of which is already available on our web site [11].

Dirac technology

Dirac uses a small number of core tools, chosen for subjective performance. Whilst some of its methods are new to mainstream video compression, they are based on long-standing techniques, and we are not aware that Dirac infringes any third-party patents.

Architecture

Dirac is a conventional hybrid motion-compensated video codec – as are the MPEG standards *(see Fig. 2)*. Image motion is tracked and the motion information used to make a prediction of a later frame. A transform is applied to the predicted frame and the transform coefficients are quantized and entropy coded [12][13][14]. The term "hybrid" is used because both a transform and motion compensation are used. Motion compensation is used to remove *temporal redundancy* and the transform is used to remove *spatial redundancy*. In contrast to most codecs, Dirac uses a ***wavelet transform*** rather than a block transform, such as the DCT (discrete cosine transform). Entropy coding packs the bits efficiently into the bitstream. Dirac uses a more flexible and efficient form of entropy coding called ***arithmetic coding*** [15], rather than the usual Huffman variable length codes.

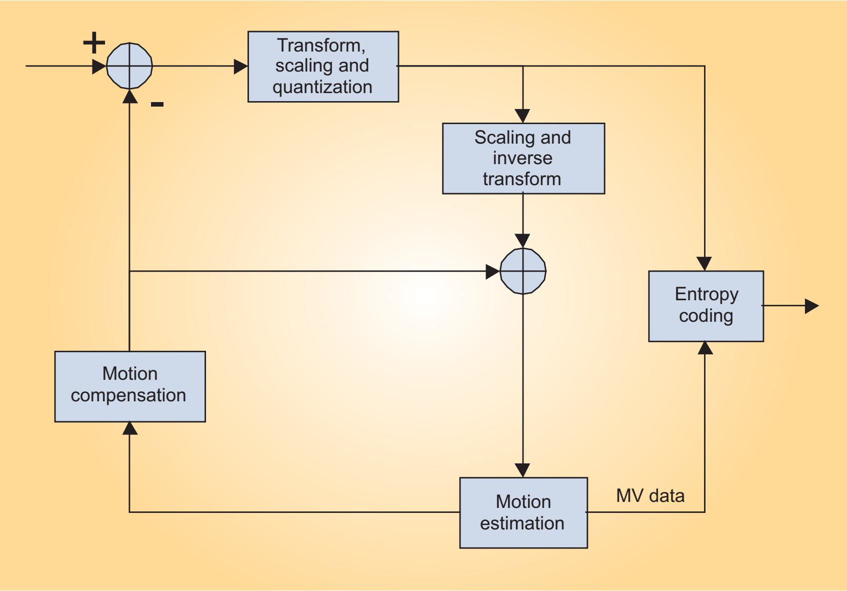
+
-
Transform, scaling and quantization
Scaling and inverse transform
Entropy coding
Motion compensation
Motion estimation
MV data

Figure 2
Encoder architecture (the decoder performs the inverse operations)

Wavelets

Wavelets have been proved more effective than block transforms for still image compression and are used in the JPEG2000 still image compression standard [16]. The artefacts induced at restricted bitrates tend to be more benign than those due to block transforms. Wavelets operate on the whole of the picture at once, rather than focusing on small areas at a time as block transforms do. This provides Dirac with the flexibility to operate at a range of resolutions from Internet streaming resolution up to HDTV quality.

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.

Fig. 3 shows an original image on the left, a first level wavelet transform in the middle and a second level transform on the right. Considering the first level transform, mid grey represents zero. So, although the transform is the same size as the original, most of the information is now concentrated in the low frequency top left corner. If you look carefully you may be able to see detail in the other parts of the transform but it is, obviously, at a much lower level. The transform packs most of the information into only a proportion of the transform (the low frequency top left), which allows us to achieve compression.

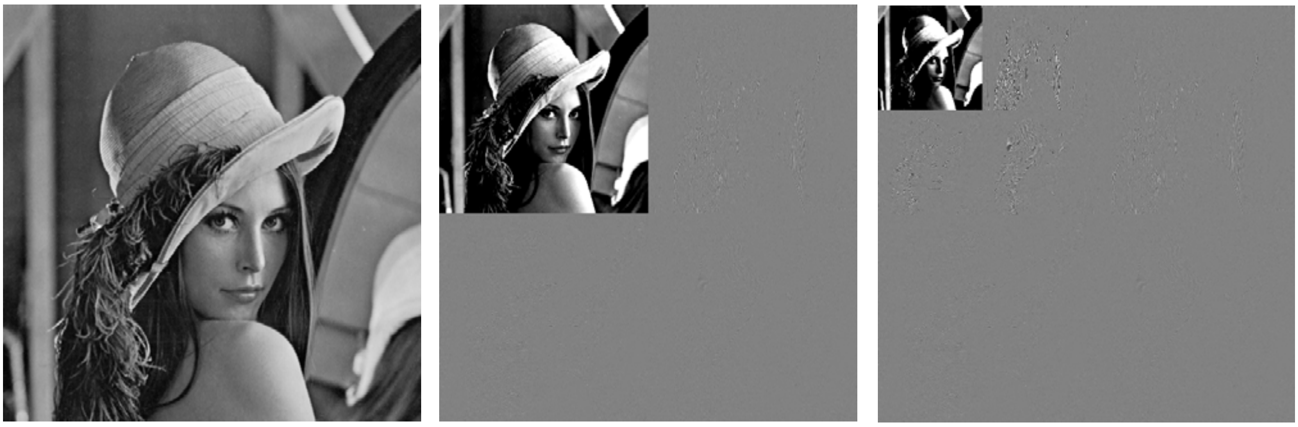

Figure 3
Stages of a wavelet transform

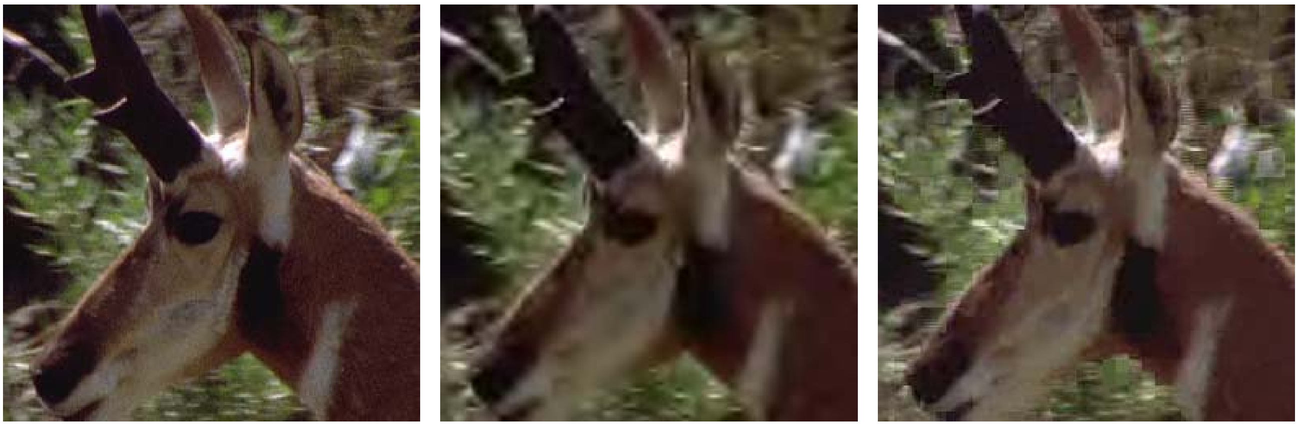

Figure 4
Original image *(left)*, Dirac compression *(middle)*, MPEG-2 compression *(right)*

A second level of transform coding can be used to pack the information even more tightly. The right-hand image of *Fig. 3* performs a second wavelet transform, but this time only on the top left portion of the first level transform. Now, although information is more tightly packed, you may be able to see some detail emerging outside the very top left. We can repeat the process to achieve higher levels of wavelet transform. The benefits of higher levels diminish with each level, so Dirac usually only uses a four-level wavelet transform.

The difference between wavelets and block transforms can be seen in *Fig. 4*. The pictures are parts of frames taken from an original image, and after compression using Dirac and using MPEG-2. The original bitrate was 160 Mbit/s, which was then compressed to 1 Mbit/s with the two compression systems. So the data-rate has been reduced by a factor of 160! The MPEG-coded picture clearly shows block artefacts while these are absent from the Dirac-coded picture. Superficially, the MPEG picture may look sharper but this is an illusion caused by looking at a still image rather than moving video. In fact the block artefacts remain in a fixed position while the image moves, which is a most disturbing artefact. In the moving image, the Dirac-compressed image is clearly superior.

Entropy coding

After applying the wavelet transform, entropy coding is applied to minimize the number of bits used. The key to entropy coding is to use a variable number of bits for the different values. Normally we use eight bits to define the brightness of a pixel. In the wavelet transform, we can see from *Fig. 3* that many of the values are approximately zero. If we only use a single bit to indicate a value of zero, we immediately need far fewer bits to store the transform. Of course this only works after wavelet transforming because few of the values in the original picture are zero. See reference [12] for more details.

Dirac uses an advanced technique for entropy coding called “arithmetic coding” [12], which is both flexible and efficient. The non-zero values in the higher frequency sub-bands of the wavelet transform (and they are there if you look carefully) are often in the same part of the picture as they are in lower frequency sub-bands. That is, there are statistical correlations across sub-bands. Dirac creates statistical models of these correlations and arithmetic coding allows us to exploit these correlations to achieve better compression [17].

The motion information estimated at the encoder also uses statistical modelling and arithmetic coding to compress it into the fewest number of bits. This compressed data is put into the bitstream, to be used by the decoder as part of the compressed video.

Motion compensation

Dirac uses motion compensation, as do many other systems, to achieve good compression [11]. To avoid the block artefacts common in other codecs, we estimate motion using overlapping blocks.

Dirac also supports the use of global motion estimates, which can specify camera motion, e.g. pans and zooms, in a few bytes and thereby reduce the bitrate. At low bitrates it may be useful to simply predict a frame using only the motion information without transmitting any wavelet coefficients at all. Even more extreme may be to simply predict a frame to be the same as a previous frame. Such techniques, which are supported by Dirac, can provide substantial bitrate reductions when only a modest quality is required, e.g. for Internet streaming.

Bitstream

The bitstream syntax for Dirac is quite different from the conventional MPEG syntax. With a new codec we had the opportunity to develop a new syntax, whilst learning from and incorporating some of the best features of the MPEG syntax, to providing a simpler, more coherent and easier to use structure. We have been able to implement new features not present in MPEG (e.g. unique frame numbers) and omit obsolete features (e.g. obsolete HDTV colorimetry parameters). For example, each frame in a Dirac bitstream indicates where the previous and next frame are in the bitstream. This is an unusual feature in compression systems and allows software that uses Dirac to navigate much more easily through the bitstream. Simple bitstream navigation supports editing using compressed video. Applications using older codecs that do not support this feature, such as MPEG-2, require more complex software and have reduced performance.

Implementation

The Dirac software is implemented in the (ISO) C++ programming language. We wanted to use a language that was well known and readily available to all. C++ is ideal for this and allows Dirac to be built on all common operating systems. Dirac has been tested under the Microsoft Windows, Linux and Apple operating systems and others.

Although the Dirac internals are written in C++, an application programmer’s interface (API) has been written in the C programming language. C language APIs are the lingua franca of software, which allow pieces of software to operate together. The API is as simple as possible and allows the straight-forward integration of Dirac into media players, video processing tools and streaming software.

Performance of Dirac

Dirac is designed to achieve good subjective compression performance by using a few tools guided by psychovisual criteria. The psychovisual modelling is performed solely in the encoder – there are no quantization matrices as there are in MPEG-2 or H.264 FRExt. This allows future implementations of Dirac more freedom and, therefore, more scope to improve performance.

Unlike most codecs, Dirac has not been designed to maximize PSNR (Peak Signal to Noise Ratio). It is well known that PSNR does not correlate well with the subjective quality of compressed pictures, particularly at low bitrates. Instead, Dirac tries to improve quality by concentrating more on large errors and de-emphasizing high frequency errors, e.g. at the edges of objects or in textured areas, which are less noticeable. Although the quality metric used by Dirac is quite simple, it is surprisingly effective.

Since Dirac has not been designed to maximize PSNR, measurements of PSNR performance against other codecs are not meaningful. Expert viewing suggests that, despite its simple toolset, Dirac is very comparable to other state-of-the-art codecs such as H.264.

The relative simplicity and clean architecture of the Dirac codec supports high performance. In principle, this should allow Dirac to achieve simpler and more efficient implementations than competing codecs. Efficient implementations would lend themselves to use on mobile and low performance platforms such as cell phones and PDAs. Such efficient implementations require considerable optimization effort, which the project is now starting to address.

Since early in 2005, Dirac has been able to play back up to 1 Mbit/s video, in real time, on ordinary PCs running Microsoft Windows or Linux. This provides quality that approaches that of standard definition broadcasts and is more than adequate for Internet streaming. As we optimize the software, the speed of playback will continue to improve. Dramatic improvements in decoding speed are possible if we exploit the processing power of graphics accelerators, as other codecs do.

The future

Although it has been demonstrated to provide high quality in real time on a variety of platforms, Dirac is still in development. Looking forward, we hope there will be significant improvements over the coming months. These will include maturing of the specification [11] and the convergence of the software with the specification. There is already experimental support for Dirac in a number of media players, including Windows Media Player, and this support will continue to strengthen. Looking further ahead, we hope that standardization through a formal standards body may be possible, although the specification and technology are already available to all.

Whilst conventional block transform codecs are approaching the end of their useful development cycle, Dirac is only at the beginning of its development. The use of wavelets and variable levels of the transform, means that Dirac can perform very well across a great range of picture sizes and bitrates. The use of arithmetic coding allows efficient coding, taking into account the statistical properties of the signal. Features such as global motion parameters and frame skipping, support high quality images at low bitrates. We have only just started to exploit these techniques, which are already supported in the bitstream. Significant further improvements in performance can be expected.

Summary

Dirac is a state-of-the-art, general-purpose, video compression system. It achieves high performance through the use of modern techniques such as wavelet transforms and arithmetic coding. It is still young and has the potential for significant further improvements. Dirac is open technology, freely available to all, which suits the requirements of public service broadcasters and supports the use of digital technology by consumers.

Acknowledgements

The authors would like to acknowledge the enthusiasm and support both of the Dirac team at the BBC and the many others who have contributed to this open collaborative development. Without their support, Dirac would not have been possible.

Dr Tim Borer, a Principal R&D Engineer at BBC Research & Development, is the team leader for the Dirac project. He has worked in the broadcast industry for more than 20 years – for the BBC, Snell & Wilcox and Leitch. He gained a Ph.D. in video processing in 1992 and has worked on video processing, motion compensated processing, video standards conversion and compression, both in hardware and in software. He is the author of more than a dozen patents.

Dr Thomas Davies is a Senior R&D Engineer at BBC Research & Development, and the principal algorithm developer for Dirac. He joined the BBC in 2000 after obtaining a Ph.D. in Mathematics and spending some time working in communications and satellite networking. In addition to his work on video coding systems and compression algorithms, his work at the BBC has covered digital radio cameras, advanced OFDM modulation and error-control coding, as well as video quality assessment studies.

References

[1] Paul Dirac biography: http://www.iop.org/diracbio.html

[2] Paul Dirac biography: http://nobelprize.org/physics/laureates/1933/dirac-bio.html

[3] MPEG-2, ISO/IEC 13818-2
International Organization for Standardization, http://www.iso.ch

[4] MPEG-4 Part 2, ISO/IEC 14496-2
International Organization for Standardization, http://www.iso.ch

[5] Paola Sunna: **AVC/H.264 – an advanced video coding system for SD and HD broadcasting**
EBU Technical Review No. 302 (April 2005). http://www.ebu.ch/trev_302-sunna.pdf

[6] Microsoft Windows Media: http://www.microsoft.com/windows/windowsmedia

[7] RealVideo: http://www.realnetworks.com/products/codecs/realvideo.html

[8] On2 / vp7: http://www.on2.com/vp7.php

[9] Review of the BBC's Royal Charter, May 2005
http://www.bbc.co.uk/thefuture/pdfs/green_paper_response.pdf

[10] Mplayer: http://www.mplayerhq.hu

[11] The Dirac Web page: http://dirac.sourceforge.net

[12] Mohammed Ghanbari: **Standard Codecs – Image Compression to Advanced Video Coding**
Institution of Electrical Engineers, 2003, ISBN: 085296710
http://www.iee.org/Publish/Books/index.cfm

[13] Ian Richardson: **Video Codec Design – Developing Image and Video Compression Systems**
John Wiley and Sons Ltd, 2002, ISBN: 0471485535
http://www.wiley.com/WileyCDA/Section/index.html

[14] Ian Richardson: **H.264 and MPEG-4 Video Compression – Video Coding for Next-Generation Multimedia**
John Wiley and Sons Ltd, 2003, ISBN: 0470848375
http://www.wiley.com/WileyCDA/Section/index.html

[15] I.H. Witten, R.M. Neal and J.G. Cleary: **Arithmetic Coding for Data Compression**
Communications of the ACM, Vol. 30, No. 6, pp. 520 - 540, June 1987.

[16] JPEG2000: http://www.jpeg.org/jpeg2000/

[17] G.G. Langdon and J. Rissanen: **A double-adaptive file compression algorithm**
IEEE Trans., Comms. Vol. COM-31, No. 11, pp. 1253 - 1255, 1983
http://www.ieee.org/portal/site