R&D White Paper

WHP 124

September 2005

Dirac video compression

T. Borer, T. Davies *and* A. Suraparaju

Research & Development
BRITISH BROADCASTING CORPORATION

BBC Research & Development
White Paper WHP 124

Dirac video compression

T. Borer, T. Davies and A. Suraparaju

Abstract

Dirac is a new general purpose video compression system, suitable for resolutions from internet streaming to HDTV, that has been developed by the BBC. Dirac achieves state of the art compression performance and is expected to improve as it develops. It is a hybrid motion compensated codec that uses modern techniques such as wavelet transforms and arithmetic coding. The design of Dirac is intended to minimise complexity, thereby facilitating understanding and reducing the resources needed for implementation. Dirac is open technology, which means 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. The open model of development adopted for Dirac supports open collaboration by anyone interested in its future development.

This document was originally published in the Proceedings of the International Broadcasting Convention, September 2005.

White Papers are distributed freely on request.

Authorisation of the Chief Scientist is required for publication.

DIRAC VIDEO COMPRESSION

T. Borer, T. Davies & A. Suraparaju

BBC Research & Development, UK

ABSTRACT

Dirac is a new general purpose video compression system, suitable for resolutions from internet streaming to HDTV, that has been developed by the BBC. Dirac achieves state of the art compression performance and is expected to improve as it develops. It is a hybrid motion compensated codec that uses modern techniques such as wavelet transforms and arithmetic coding. The design of Dirac is intended to minimise complexity, thereby facilitating understanding and reducing the resources needed for implementation. Dirac is open technology, which means 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. The open model of development adopted for Dirac supports open collaboration by anyone interested in its future development.

INTRODUCTION

Video compression technology is ubiquitous in broadcasting. In the digital world compression is used to exploit limited storage and transmission bandwidth as efficiently as possible. It is used in production, newsgathering and distribution between broadcast centres. Video is compressed for digital delivery by terrestrial, cable or satellite broadcasting and for web streaming. And compression is essential to provide the new generation of HDTV services. Clearly it is vital for broadcasters to understand and use the most effective technologies available.

In March 2004 BBC Research and Development released experimental software for a new video codec called Dirac. Dirac is a general purpose video codec aimed at applications from HDTV to web streaming. It is intended to be licence free so that anyone can use it for any purpose free of charge. The quality of pictures compressed with Dirac is comparable to other state of the art codecs such as H264 (a.k.a MPEG4 AVC) and it has the potential for further improvements in performance.

This paper is intended to present an overview of Dirac. For those who may have heard of Dirac and are curious, it discusses what we are trying to do and our rationale for doing so. The paper highlights the technology used in Dirac and how this has been embodied in software. Finally we discuss some features of the performance of Dirac and what the future might hold for this new video compression system.

THE GENESIS OF DIRAC

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 IP TV were coming to the fore. The venerable MPEG2 video compression system, used for digital video broadcasting and on DVDs, 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. We started a project to develop a video codec to meet this need. By 2003 we had a promising prototype and by 2004 we had released software for the Dirac codec.

The increasing delivery of video by the Internet places new demands on public service broadcasters like 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 cost is simply that of the bandwidth needed to transmit the video, which can be minimised by using an efficient codec. 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 become prohibitive if we try to scale up by orders of magnitude to millions of simultaneous users. Furthermore a public service broadcaster has an obligation to make its content freely available, independent of platform, without proprietary lock in.

Conventional open standards, such as MPEG, do not fit the model of Internet delivery by public service broadcasters. This is because their licence fees are typically set with subscription business models in mind. “Open Technology” is needed that may freely be used without royalty payment. Open technology also supports the take up of digital technology by reducing the cost to consumers. Dirac aims to be an open technology: it is released “open source”, in a similar way to the well-known Linux operating system, so that it can both be used freely and developed collaboratively. Anyone can download the software from our web site[1], use it a modify it for their own purposes.

THE TAO OF DIRAC

As an open technology Dirac must be straightforward to understand and easy to use. The key philosophy behind Dirac is “keep it simple”. This contrasts with other systems, which achieve improved compression by adding ever more complexity. Of course, a modern state of the art codec is, inevitably, quite complex, but we have tried to minimise this in Dirac. Reduced complexity makes Dirac easier to understand, easier to implement and easier to optimise for real time performance. To complement this philosophy we also aim to provide copious documentation, some of which is already on our web site [1].

Dirac uses a small number of core tools, which are chosen to achieve good subjective performance. In developing Dirac as an Open Technology we were free to use the best available techniques without the need to generate new patentable technologies. Whilst some techniques we have used 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.

We have tried to make Dirac easy to use. For example the bit stream has been designed to be easy to parse. In MPEG2 there are no unique frame numbers. In long GOP MPEG2 the only way to locate frames is to parse the bit stream looking for start codes. This makes navigating the video sequence, for example jumping ahead or “scrubbing the video”, very difficult. In Dirac, by contrast, each frame has a unique frame number and contains the offset to the next frame. In software parlance the frames constitute a doubly linked list. This makes navigating a stream both easier and quicker.

DIRAC TECHNOLOGY

The Dirac development consists of two parts: a compression specification for 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, but also as 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. Real-time decoding of modern compression systems is difficult without extensively exploiting hardware support (in coprocessors and video cards) or assembly-language code, but these features can easily be added to Dirac’s modular codebase.

With Dirac the software has been developed first, and work is ongoing to converge the 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.

Architecture

Dirac is a conventional hybrid motion compensated video codec (figure 1), except that the block transforms used in other codecs, such as the MPEG standards, are replaced by a single wavelet transform applied to the whole frame. Wavelets have commonly been used for still image compression, and for the core of the JPEG2000 standard [2]. This hybrid architecture means that wavelet transforms are applied to both Intra pictures and to motion-compensated residuals.

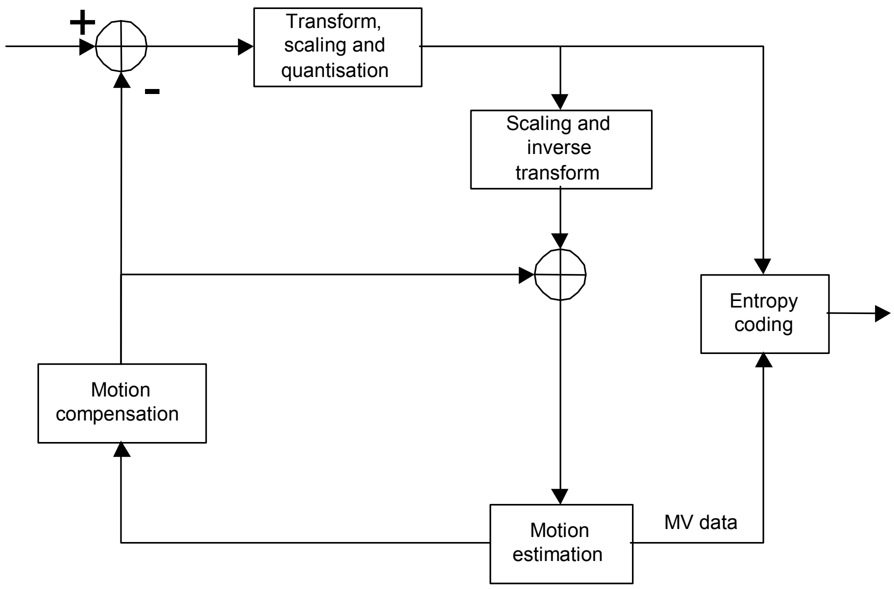

Figure 1: Encoder Architecture (The decoder performs the inverse operations)

The wavelet transform is constructed by repeated filtering of signals into low- and high-frequency parts. For two-dimensional signals this filtering occurs horizontally and vertically. At each stage the ‘low-low’ subband is further split, resulting in logarithmic frequency decomposition into subbands (figure 2 below).

Motion compensation

Motion compensation in Dirac uses variable-sized overlapped blocks, which are grouped into 4x4 sets to form macroblocks. Overlapping greatly mitigates the impact of block edges in subsequent wavelet transform coefficient coding; it also generally results in a smoother motion field, yielding improved compression.

There are three macroblock splitting modes to enhance performance over a range of bit-rates and types of content. These allow single motion vectors per reference to be used for the macroblock, or for four ‘sub-macroblocks’ of 2x2 blocks, or per block. Motion vector accuracy is variable, up to (and beyond) 1/8th pixel meaning that coarser motion vectors can be used at higher resolutions and for simpler decoding at all resolutions.

Within the Dirac software, block sizes are completely flexible because they are decoupled from a particular block transform size. This enhances scalability and compression performance, since, for example, block sizes can be chosen which ensure a constant number of blocks whatever the source resolution. For hardware implementations, such extreme flexibility is undesirable, and as Dirac matures levels and profiles will be developed that constrain block sizes for ranges of picture sizes.

Figure 2: Original Image (left) and 3 level wavelet transform (right)

Global motion compensation tools are also under development. These will provide a parameterised global motion field that can be used to derive block motion vectors, which can then be used in block motion compensation in the usual way.

In Dirac, frames have two essential properties. Firstly, they are either predicted from other frames (Inter) or not (Intra). Secondly, they can be used to predict other frames (Reference) or not (Non Reference). All combinations of these properties are possible, and any Inter frame may be predicted from up to two reference frames, which may be future or past frames. This means that Dirac supports conventional MPEG-style GOP structures, but also any other prediction structure that satisfies the constraints imposed by decoder buffering capabilities, as specified in a level/profile.

Access to the Dirac bitstream is through a separate Random Access Point (RAP) header, which precedes an I-frame. However not all I-frames need to be RAPs, which allows I-frames to be inserted at any point to achieve efficient compression, for example if there is a picture cut, without affecting the prediction structure. Dirac supports frame skipping to improve quality at very low bit rates. Software decoders may skip non-reference frames to speed up decoding.

Entropy coding

Once data has been transformed entropy coding is used to minimise the number of bits used. The entropy coding technique used by Dirac is arithmetic coding [3] because it is both flexible and efficient. Arithmetic coding separates statistical modelling from the compression process itself, and better compression is afforded if the inter-dependence of data is exploited by switching between models (or contexts, as they are known) based on previously-coded data [4]. Dirac codes the coefficients in each subband taking account of statistical correlations spatially across subbands and also between subbands. The subbands are coded from the lowest frequency band to the highest. The entropy coding scheme is straightforward, using a single pass of the coefficients and employing only 17 statistical contexts [5].

In coding each subband, the wavelet coefficients are quantised. Each subband is spatially partitioned into juxtaposed blocks. The encoder can elect to provide a separate quantiser for each block (supporting Region Of Interest coding, or more efficient coding of heterogeneous material) or a single quantiser per subband. Each block can be skipped or coded. This arrangement allows smooth areas, which transform to large contiguous areas of zero coefficients in a subband, to be coded efficiently. The use of skipping flags also greatly speeds up decoding performance.

Motion vectors are coded losslessly, using the same arithmetic coding engine. In this case there are many more data types, including macroblock and block modes as well as the motion vectors themselves, and hence more contexts. All motion data is coded predictively, using data from previously coded neighbours.

Encoding

A key part of the Dirac project is the development of a software encoder. Although the software has not been optimised as much as the decoder, it has been designed to provide a basis for realistic encoding implementations, rather than merely a reference codec implementing 'optimal' encoding. The encoder uses reduced-complexity rate-distortion optimisation (RDO) techniques to select quantisation parameters and motion vectors. Motion estimation uses hierarchical block matching to reduce search costs. Dirac is designed to allow for purely encoder-side psychovisual modelling: unlike, for example, MPEG2 or H264 FRExt there are no built-in psychovisual factors such as quantisation matrices. This allows future implementations an entirely free rein in encoder control methodologies. Wavelets are particularly flexible in allowing different perceptual frequency weightings to be applied, and spatial masking to be exploited, merely by adjusting quantisation parameters or directly modifying coefficients.

The encoder software integrates a number of features into the RDO control mechanism. Overall, the encoder supports Constant Quality (CQ) coding using a perceptual metric. This metric uses the fourth-power difference between low-pass filtered versions of the original and coded data, thus penalising large errors much more than a PSNR metric would, and also weighting high-frequency error much less. An internal control engine adapts RDO parameters to meet quality targets. The Dirac quality metric has proved very stable for control purposes, and very useful in limiting the worst coding artefacts.

Quantiser selection uses weights based on spatial frequency to trade-off bit-rate and quantisation error; this quantisation error is also based on fourth-power differences rather than Mean Square Error.

The resulting encoder has been designed to exploit the capabilities of Dirac for high perceptual quality rather than optimising PSNR-based objective metrics, as it is well-known that PSNR gains can significantly overestimate real improvements in quality. Other implementations of Dirac encoders will be free to use these approaches, and also to implement quite different control methodologies.

Extensions

Dirac uses a small range of relatively simple tools for compression, whose power derives from their flexibility. The bitstream itself is designed so that new tools can easily be supported in new profiles at a later date. For example, although Dirac already provides effective compression of interlaced pictures, we plan to improve this further by providing a specific interlaced compression mode. New tools can be added without changing the bitstream syntax thanks to the use of index tables of coding tools.

Whilst conventional block transform codecs are approaching the end of their useful development cycle Dirac, by contrast, is only at the beginning of its development. The bitstream syntax supports techniques, which we are developing, to further improve quality. Global motion can describe camera motion such as pans, zooms and changes of perspective, in only a few bytes per frame. Extensive support is available for “skipping” updates to all or parts of frames that have not changed much or have changed in an easily predictable way. We have only just started to exploit these techniques. Frame skipping, for example, is known to be particularly effective at low, internet, bandwidths. So by taking advantage of these techniques, which are already supported in the bitstream, significant improvements in performance can be expected.

DIRAC IMPLEMENTATION

The Dirac software is implemented in the (ISO) C++ programming language. Codec implementation can be very complex, so we wanted to use a language that encourages object oriented design, particularly function and data abstraction. We also wanted to use a language that was well known and readily available to all. This was made possible by using the GCC C++ compiler software [6] and the GNU Build system [7, 8]. Using GCC, or other tools such as Microsoft Visual C++, Dirac can be built on all common operating systems. Dirac has been tested under different distributions of GNU/Linux, BSD, Solaris, IRIX, Darwin and MS Windows 2000/XP.

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

The Dirac software is released Open Source under the Mozilla triple licence [9]. This licence allows the Dirac software to be used in both free and proprietary products. It also allows for relicensing under the well known General Public License (GPL [10]), used for much open source software or the Lesser General Public License (LGPL [11]). The reference implementation and related documentation is hosted on Sourceforge and can be downloaded via the Dirac Project page [1].

THE 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 MPEG, 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 colourimetry parameters).

The bit stream syntax, the latest version of which is available on the web site, is intended to make Dirac simple to use. The principle is that only the minimum amount of meta data is present in the stream, unspecified values are defined to be defaults. So, for example, if you have standard definition 625 line video you need only specify the Video Format to be “SD (625 Digital)”, which implicitly specifies every other parameter that is needed. However there is also the flexibility to use unusual video parameters if necessary. For video originated on film the default interlace parameter would have to be overridden by the correct progressive parameter. For each video format only the minimum number of parameters need be specified, the rest falling back to default values. In most cases only a few parameters are needed. This simplifies controlling the encoder, writing the bitstream and also minimises the data that needs to be transmitted in the bitstream.

PERFORMANCE

Dirac is designed to achieve good subjective compression performance by using a few tools guided by psychovisual criteria. Dirac derives much of its power from the flexibility of these, relatively simple, tools. The use of variable-size blocks and variable levels of wavelet decomposition means that Dirac can perform very well across a great range of picture sizes and bit rates.

Wavelets are a powerful compression tool because of their psychovisual properties. Wavelet compression yields quite different, and usually more acceptable, artefacts than those resulting from block transforms. In other compression systems a “deblocking filter” is sometimes used to reduce block artefacts. However deblocking filters can make pictures look “plastic” at low bit rates. In Dirac there is no intrinsic blockiness, and hence no need for a deblocking filter. The iterative nature of the wavelet transform effectively applies short filters for high-frequency data and long-filters for low-frequency data, which works well in coding edges. The transform allows data redundancy across wide areas of a picture to be readily exploited. This is effective not just in Intra pictures, but in Inter pictures too, which have large areas of zero coefficients.

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

The Dirac encoding software uses a single “quality” factor to make it simple for the user to specify the quality they require. The quality factor is specified as a number between 0 and 10. For a specific sequence an increase of 1 unit roughly doubles the bit rate, but the bit rate may vary significantly between sequences to achieve a roughly constant quality. This way bit rate is not wasted coding to a specific PSNR that is not required to deliver the desired quality.

Since Dirac has not been designed to maximise 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 H264.

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

Since early in 2005 Dirac has been able to playback up to 1 Mbit/s video, in real time, on ordinary PCs running Windows or Linux. This provides quality that approaches that of standard definition broadcasts and is more than adequate for Internet streaming. As we optimise the software the speed of playback will continue to improve.

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, which is already available on the web site, and the convergence of the software with the specification. Compression performance is expected to improve, as features already in Dirac start to be exploited, yielding improved pictures. Computing performance will continue to improve as optimisation proceeds and code is written to take advantage of graphics accelerators. 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 out we hope that standardisation through a formal standards body may be possible. However with open technology such standardisation is less pressing because the specification and technology are available to all.

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.

REFERENCES

1. The Dirac Web page: http://dirac.sourceforge.net
2. International Organization for Standardization, www.iso.org ISO/IEC 15444
3. J. Rissanen, 'Generalised Kraft inequality and Arithmetic Coding', IBM Journal of Research and Development Vol 20 No 3 pp 198-203, May 1976
4. G.G Langdon and J. Rissanen, 'A double-adaptive file compression algorithm', IEEE Trans. Comms. Vol COM-31, No 11, pp 1253-1255, 1983
5. http://dirac.sourceforge.net/documentation/algorithm/algorithm/algorithm.pdf
6. http://gcc.gnu.org
7. http://www.gnu.org/software/automake
8. http://www.gnu.org/software/autoconf
9. http://www.mozilla.org/MPL/MPL-1.1.html
10. http://www.gnu.org/copyleft/gpl.html
11. http://www.gnu.org/copyleft/lesser.html