The Dirac Video Codec

Peter Meerwald

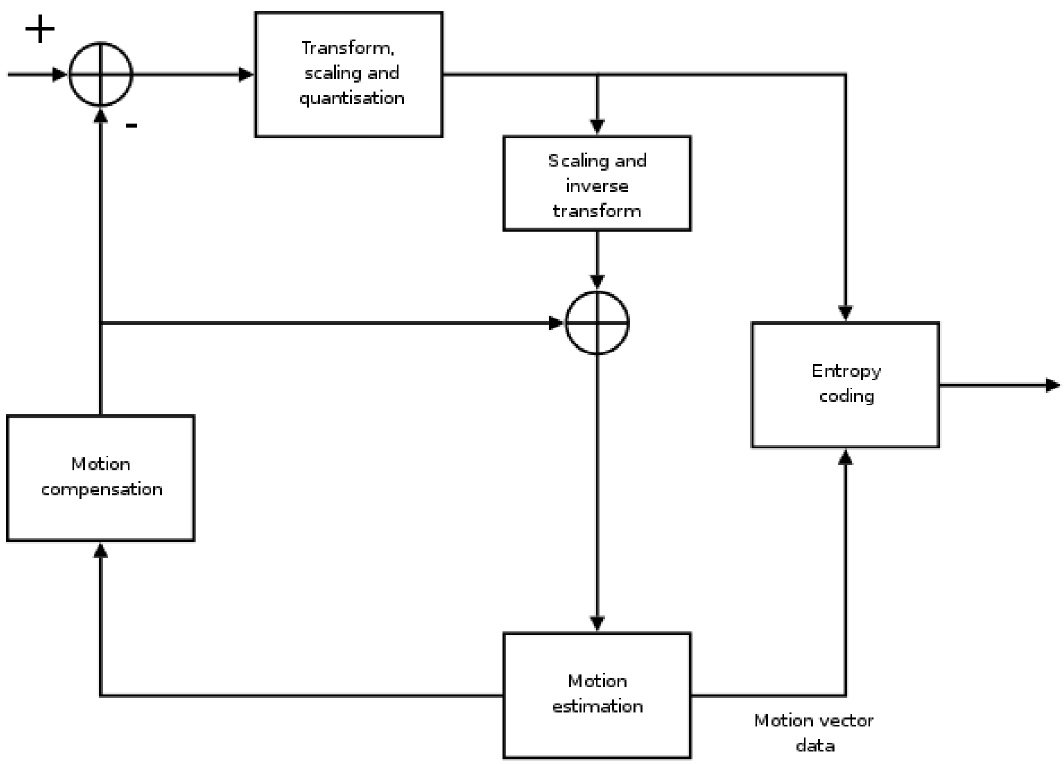
April 18, 2007

Dept. of Computer Science, University of Salzburg pmeerw@cosy.sbg.ac.at

Overview

- Developed by BBC R&D, C++
- Mozilla Public License (MPL) 1.1, patented
- Motion-compensated, hybrid video codec
- `http://dirac.sourceforge.net`
- alternative implementation: Schrodinger, `http://schrodinger.sourceforge.net`

Encoder Architecture



Wavelet Transform

4 level decomposition; edge extension for intra-, zero extension for inter frames

Integer (approximation) lifting filters, lossless

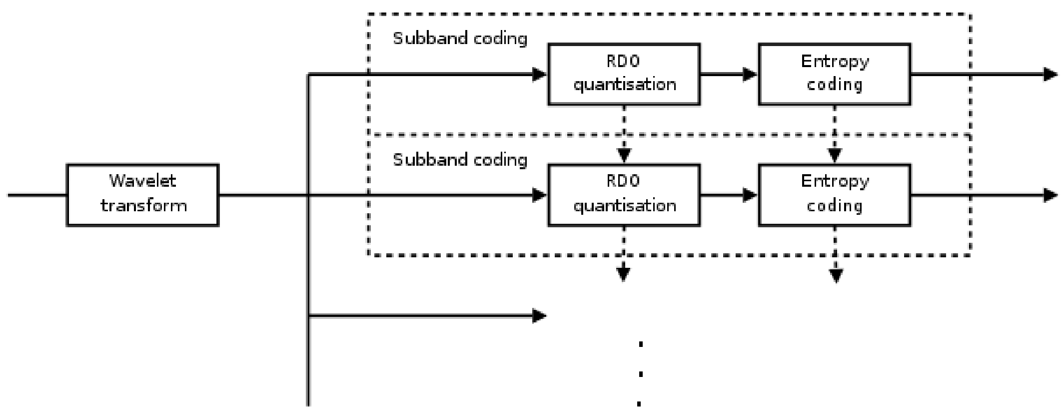
- Deslauriers-Debuc (9,3) [default intra-frame filter]
- LeGall (5,3) [default inter-frame filter]
- Deslauriers-Debuc (13,5)
- Haar (no shift, single shift, double shift)
- Fidelity (for down-conversion)
- Daubechies (9,7)

Subband coding

Intra-frame coding uses prediction of DC component coefficients

Arithmetic coding (AC) is expensive, no pass-through [2, 3]

AC context depends on parent subband, zero neighbourhood & sign prediction (12 contexts: {COEFF_DATA, [ZPZN_F1|ZPNN_F1|NPZN_F1|NPNN, ...,ZP_F6+|NP_FP6+], SIGN_ZERO|_POS|_NEG})



Observations

Not using bit-plan coding

Why not multi-symbol arithmetic coder? Most symbols occur rarely, sparse statistics

Little parent-child coding gain [4, 1]

Transform coefficients roughly Laplacian distributed, probability of 0 or 1 occuring in any unary bin (VLC) is constant

Dirac initializes contexts with equal probability, H.264/AVC / JPEG2000 uses precomputed model; separate contexts for Y, U, V and HL, LH, HH improve performance [3]

Dirac halves the model's frequency counts at regular intervals

Entropy Coding

3 stages:

1. Binarisation, interleaved Exp.-Golomb code

2. Context modelling: model sign, data bits and follow bits independently
   eg. context for motion vector, first reference, x coordinate: {REF1x_DATA, [REF1x_F1, . . . , REF1x_F4, REF1x_F5+], REF1x_SIGN}

3. Adaptive arithmetic coding

H.264/AVC uses Truncated Unary (TU) / k-th order Exp.-Golomb codes [5] (in CABAC); eg. TU(14) EG(0) for coding transform coefficients and TU(9) EG(3) for motion vectors

Exp.-Golomb Code (1)

Interleaved Bit Sequence	Value	Bit Sequence
1	0	0
001	1	100
011	2	101
00001	3	11000
00011	4	11001
01001	5	11010
01011	6	11011

Table 1: Exp.-Golomb Bit Sequence for Dirac and H.264/AVC.

Exp.-Golomb Code (2)

To code number N, EG(0):

N+1 $= 1\, b_{k-1}\, b_{k-2}\, \ldots\, b_0$

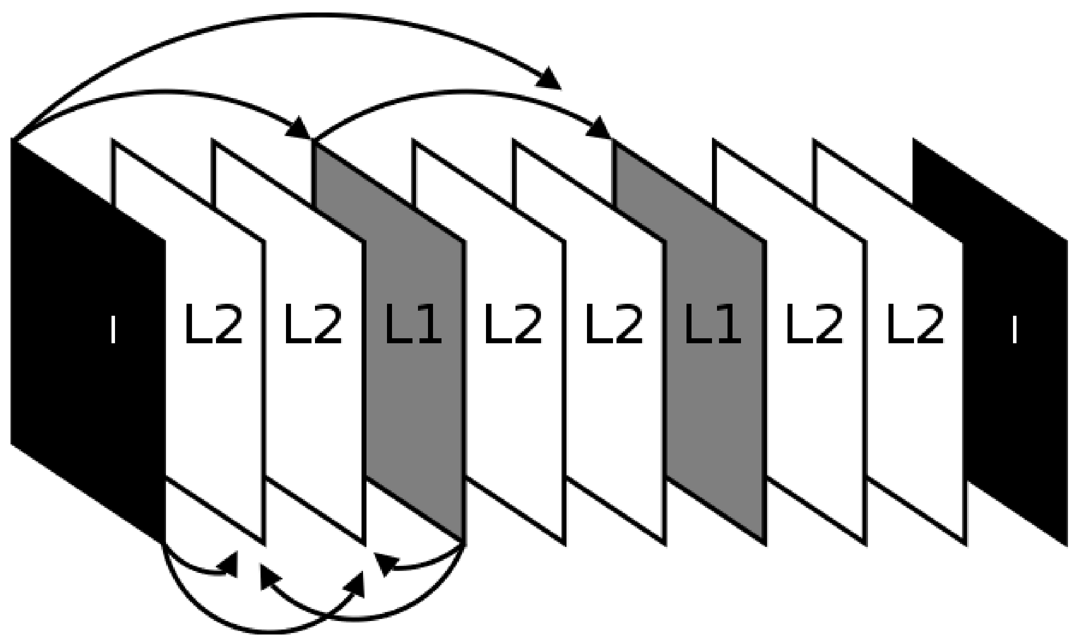
For Dirac: Interleave b with 0, terminate with 1

For H.264/AVC: Write k 1, write 0, write b

Temporal Prediction

L1 and L2 are predicted; L1 is itself used for prediction

Arbitrary prediction possible, but using standard GOP structure



Motion Estimation (1)

3 stages:

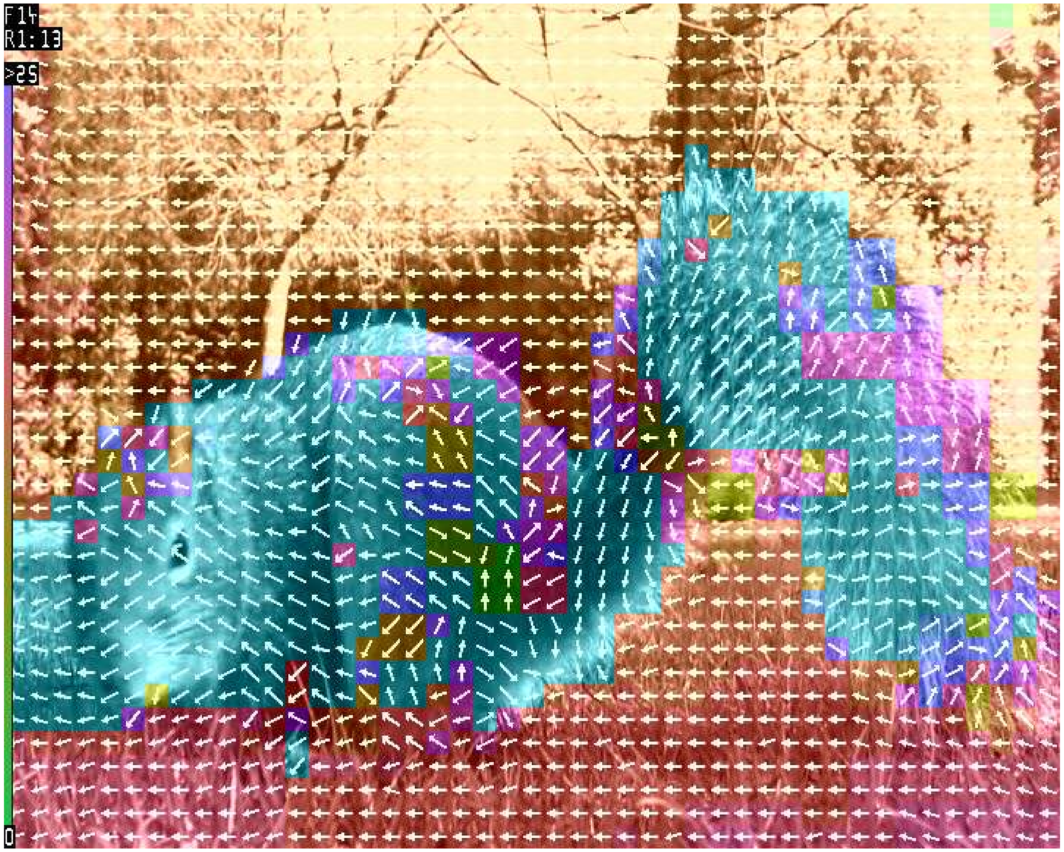
1. Hierarchical per-block motion estimation with pixel accuracy in Y component

2. Sub-pixel refinement, up to 1/8 pixel

3. Mode decision: Choose prediction and aggregate motion vectors

Trading Sum-of-absolute difference (SAD) minimum against local motion vector smoothness

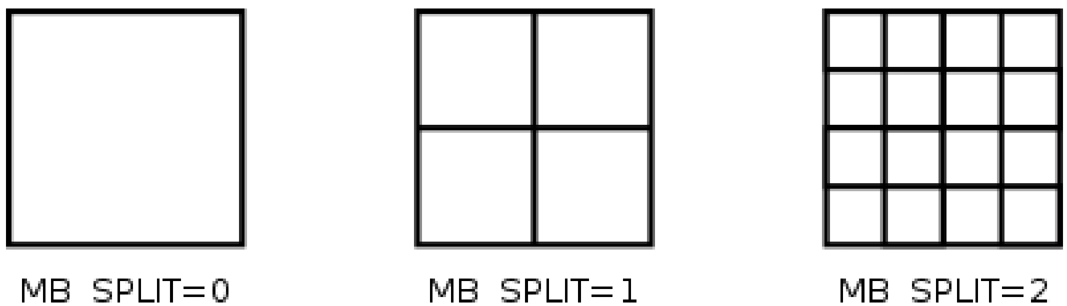
Motion Estimation (2)



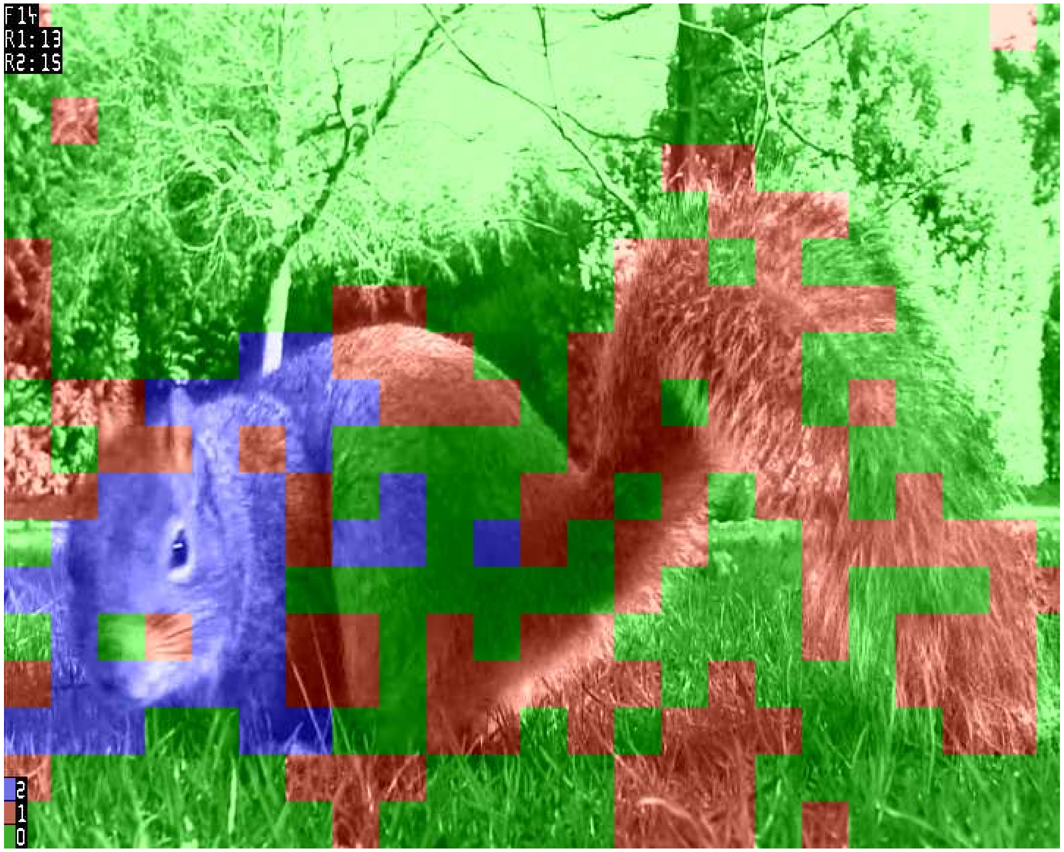
Macroblock Structure (1)

Macroblock . . . 4x4 blocks with 3 splitting modes (MB_SPLIT)

4 prediction modes: INTRA, REF1_ONLY, REF2_ONLY, REF1AND2



Macroblock Structure (2)

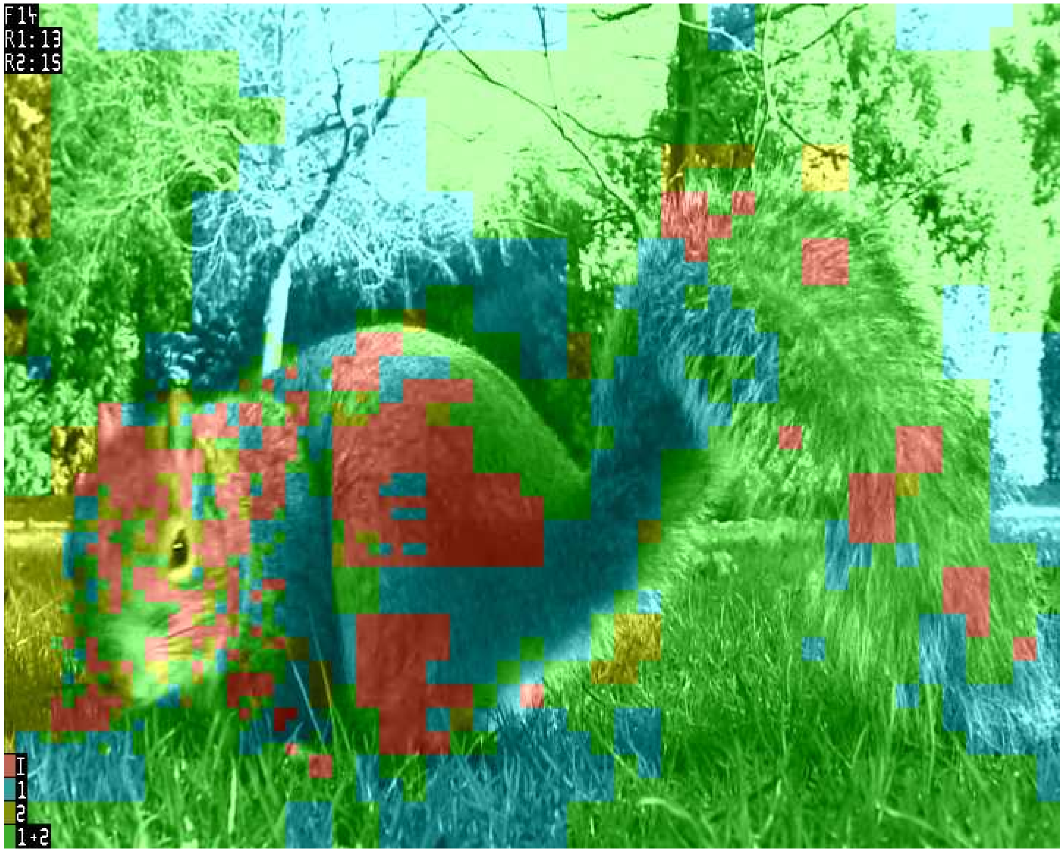


Motion Compensation (1)

Overlapped-block motion compensation (OBMC) to reduce rough edges

Each pixel may be predicted by up to 4 blocks -> weighing

Motion Compensation (2)



Rate-Distortion Optimization

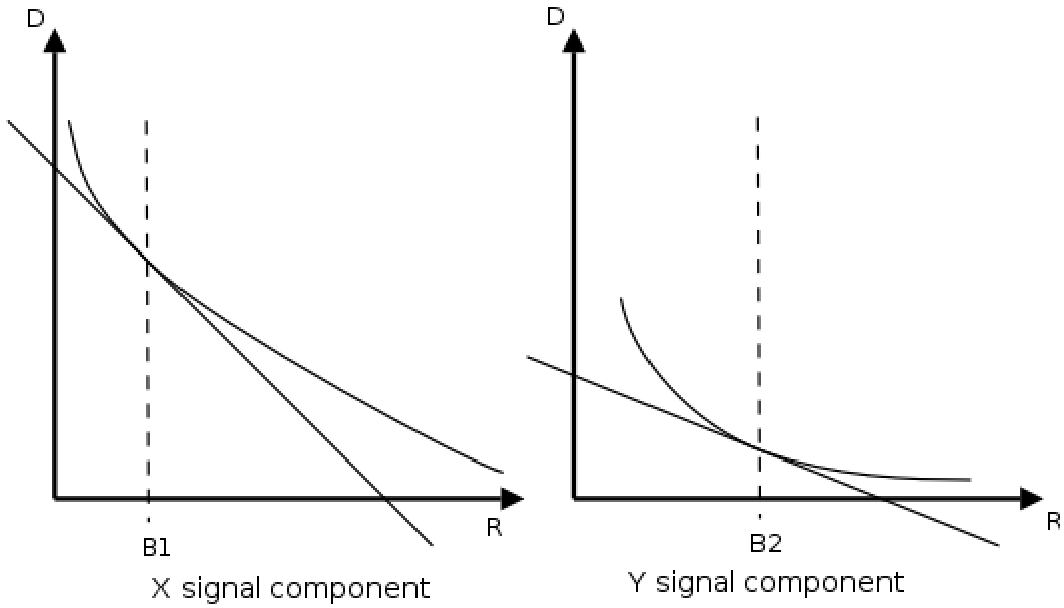
No common distortion measure, perceptual weighting with Contrast Sensitivity Function (CSF)

Rate not directly measureable, using statistical model based on coefficient entropy

Bitstream components (motion vectors vs. residual coefficients) are interdependent

$$\lambda \cdot C \cdot Ent(q) + \frac{\sqrt[2]{\sum |p_{i,j} - q(i,j)|^4}}{w_s}$$

Principle of Equal Slope



Scalability

No scalability options in Dirac, but

- Resolution – intrinsic
- Temporal – by suitable GOP L1/L2 structure
- SNR – ?
- Spatial – ?

Sequences and Codecs Used

Dirac 0.60 (CVS)

H.264/AVC . . . x264 (SVN) [7]

MPEG2 & MPEG4 . . . mencoder / libavcodec (SVN)

Carphone . . . 383 frames, QCIF, 176x144, 4:2:0

Foreman . . . 300 frames, CIF, 352x288, 4:2:0

Squirrel . . . 50 frames, PAL, 720x576, 4:2:0

Snowboard . . . 30 frames, HD720, 1280x720, 4:2:0

Compression Results (1): QCIF & CIF

Carphone (QCIF)	1024	768	512	384	256	128
Dirac	43.79	42.76	41.14	39.79	37.87	33.97
H.264/AVC	48.43	46.76	44.45	42.73	40.40	36.70
MPEG4	42.44	42.28	40.67	38.80	36.14	31.31
MPEG2	43.67	42.35	39.66	37.33	34.31	30.94

Foreman (CIF)	1024	768	512	384	256	128
Dirac	38.06	36.93	34.95	-	-	-
H.264/AVC	40.13	38.78	36.93	35.67	33.71	30.34
MPEG4	37.32	35.65	33.34	31.25	30.93	-
MPEG2	35.78	33.62	31.68	31.33	31.00	-

Table 2: PSNR (Y) results in dB for the first 30 frames encoded at 1024, . . . , 128 Kbit/s.

Compression Results (2)

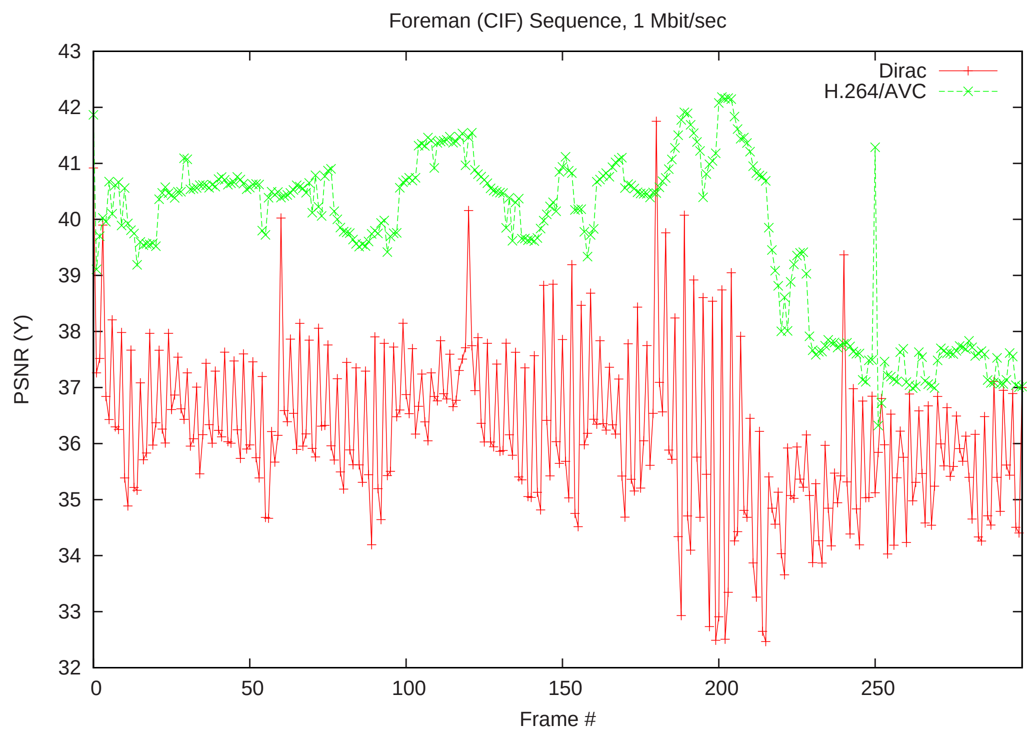


Figure 1: PSNR (Y) in dB of Foreman sequence, for Dirac and H.264/AVC.

Compression Results (3): PAL

more results by Onthriar [6]

Squirrel (PAL)	4096	3072	2048	1536	1024
Dirac	35.52	34.19	32.32	31.04	-
H.264/AVC	36.80	35.07	32.90	31.50	29.76
MPEG4	33.03	31.25	28.15	27.72	27.15
MPEG2	32.50	29.25	28.46	28.20	-

Table 3: PSNR (Y) in dB for the first 30 frames encoded at 4096, . . . , 1024 Kbit/s.

Compression Results (4)

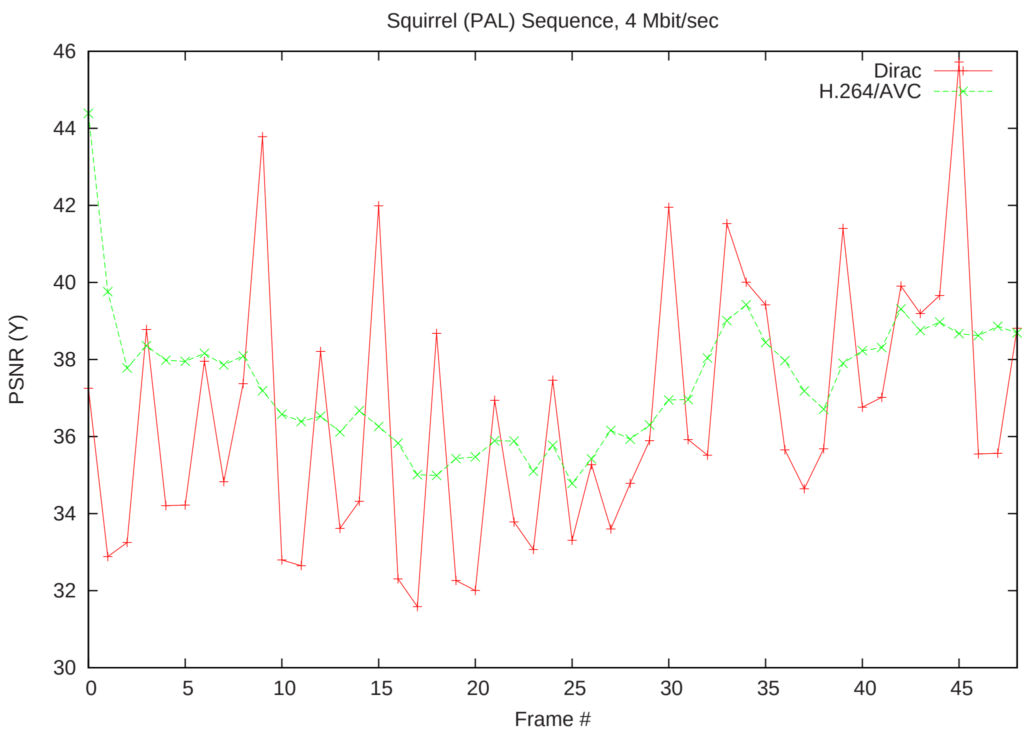


Figure 2: PSNR (Y) in dB of Squirrel sequence, for Dirac and H.264/AVC.

Runtime Results

Sequence	Dirac	x264
carphone (QCIF / 383 frames)	0.05	0.01
foreman (CIF / 300 frames)	0.22	0.03
squirrel (PAL / 50 frames)	0.87	0.08
snowboard (HD / 30 frames)	1.61	0.20

Table 4: Encoding time per frame in seconds on AMD Athlon MP 1250 MHz.

References

[1] A. Bilgin and Michael W. Marcellin. On parent-child coding gain in zero-tree based coders. page 485, Snowbird, UT, USA, March 2001. IEEE.

[2] Hendrik Eeckhaut, Benjamin Schrauwen, Mark Christiaens, and Jan van Campenhout. Speeding up dirac's entropy coder. In *Proceedings of the 5th WSEAS International Conference on Multimedia, Internet and Video Technologies*, pages 120–125, Corfu, Greece, August 2005.

[3] Hendrik Eeckhaut, Benjamin Schrauwen, Mark Christiaens, and Jan van Campenhout. Tuning the M-coder to improve dirac's entropy coding. *WSEAS Transactions on Information Science and Applications*, 2:1563–1571, 2005.

[4] Michael W. Marcellin and Ali Bilgin. Quantifying the parent-child coding gain in zero-tree based coders. *IEEE Signal Processing Letters*, 8(3):67–69, March 2001.

[5] Detlev Marpe, Heiko Schwarz, and Thomas Wiegand. Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):620–636, July 2003.

[6] K. Onthriar, K. K. Loo, and Z. Xue. Performance comparison of emerging dirac video codec with H.264/AVC. In *Proceedings of the International Conference on Digital Telecommunications, ICDT '06*, 2006.

[7] Jörn Ostermann, Jan Bormans, Peter List, Detlev Marpe, Matthias Narroschke, Fernando Pereira, Thomas Stockerhammer, and Thomas Wedi. Video coding with H.264/AVC: Tools, performance, and complexity. *IEEE Circuits and Systems Magazine*, 4(1):7–28, 2004.