DAALA: A PERCEPTUALLY-DRIVEN STILL PICTURE CODEC

Jean-Marc Valin, Nathan E. Egge, Thomas Daede, Timothy B. Terriberry, Christopher Montgomery

Mozilla, Mountain View, CA, USA
Xiph.Org Foundation

ABSTRACT

Daala is a new royalty-free video codec based on perceptually-driven coding techniques. We explore using its keyframe format for still picture coding and show how it has improved over the past year. We believe the technology used in Daala could be the basis of an excellent, royalty-free image format.

1. INTRODUCTION

Daala is a royalty-free video codec designed to avoid traditional patent-encumbered techniques used in most current video codecs. In this paper, we propose to use Daala's keyframe format for still picture coding. In June 2015, Daala was compared to other still picture codecs at the 2015 Picture Coding Symposium (PCS) [1, 2]. Since then many improvements were made to the bitstream to improve its quality. These include reduced overlap in the lapped transform, finer quantization of chroma, encoder search improvements, as well as a new deringing filter.

Rather than describe in detail all of the coding techniques used in Daala, this paper provides references to previously published documents describing them. We then present improvements since the version presented at PCS 2015. Finally, we present the results obtained and compare them to previously reported results [1].

2. FUNDAMENTAL DAALA TECHNIQUES

One of the goals of the Daala codec is to explore techniques that are very different from those typically used in most codecs. Most of these techniques have been fundamental to Daala since the initial stages of the project. They are described below.

Lapping

Daala uses lapped transforms [3, 4] rather than a regular DCT followed by a deblocking filter. This reduces blocking artifacts but prevents the use of standard pixel-based intra-prediction techniques [5]. Instead, we use a simple frequency-domain intra predictor that only handles horizontal and vertical directions [1]. Also, DC coefficients are combined recursively using a Haar transform, up to the level of 64x64 superblocks.

Multi-Symbol Entropy Coder

Most recent video codecs encode information using binary arithmetic coding, meaning that each symbol can only take two values. The Daala range coder supports up to 16 values per symbol, making it possible to encode fewer symbols [6]. This is equivalent to coding up to four binary values in parallel and reduces serial dependencies.

Perceptual Vector Quantization

Rather than use scalar quantization like the vast majority of picture and video codecs, Daala is based on perceptual vector quantization (PVQ) [7]. PVQ makes it possible to take into account masking effects with no extra signaling.

Chroma from Luma (CfL) Prediction

Because of its structure, PVQ makes it especially easy to predict chroma planes from the luma plane. Daala's chroma from luma (CfL) [8] prediction uses the luma transform coefficients to predict the chroma transform coefficients.

3. RECENT IMPROVEMENTS

Since the last evaluation at PCS, Daala has improved significantly. In addition to general encoder tuning, there have been more visible changes, each described below.

3.1. Deringing Filter

The previous version of Daala evaluated in [1] included a deringing filter based on a directional *paint* algorithm. While it provided an improvement in quality, the paint deringing filter proved very hard to vectorize and efficiently implement. This was mostly due to the complicated per-pixel weights used in block interiors, which blended up to four points from the block boundary, selected in ways that were not regular over an entire block.

Send correspondence to Jean-Marc Valin <jmvalin@jmvalin.ca>.

In the current version of Daala, the paint deringing filter is replaced by a new directional deringing filter based on a conditional replacement filter [9, 5]. Let $x(n)$ denote a 1-dimensional signal and w_k denote filter tap weights, a linear finite impulse response (FIR) filter with unit DC response is defined as

$$y(n) = \frac{1}{\sum_k w_k} \sum_k w_k x(n+k) \ , \quad (1)$$

which can alternatively be written as

$$y(n) = x(n) + \frac{1}{\sum_k w_k} \sum_{k,k\neq 0} w_k \left[x(n+k) - x(n)\right] \ . \quad (2)$$

The main advantage of expressing a filter in the form of Eq. (2) is that the normalization term $\frac{1}{\sum_k w_k}$ can be approximated relatively coarsely without affecting the unit gain for DC. This makes it easy to use small integers for the weights w_k.

The disadvantage of linear filters for removing ringing artifacts is that they tend to also cause blurring. To reduce the amount of blurring, the conditional replacement filter used in Daala excludes the signal taps $x(n+k)$ that would cause blurring and replaces them with $x(n)$ instead. This is determined by whether $x(n+k)$ differs from $x(n)$ by more than a threshold T. The FIR filter in Eq. (2) then becomes a conditional replacement filter expressed as

$$y(n) = x(n) + \frac{1}{\sum_k w_k} \sum_{k,k\neq 0} w_k R\left(x(n+k) - x(n), T\right) \ , \quad (3)$$

where

$$R(x,T) = \begin{cases} x & , |x| < T \\ 0 & , \text{otherwise} \end{cases} \ . \quad (4)$$

To further reduce the risk of blurring the decoded image, the conditional replacement filter is applied along the main direction of the edges in each 8x8 block. The algorithm for finding directions is described in [9] and is the same as the one previously used in the paint deringing filter. For each 8x8 block, it determines which of eight different directions best represents the content of the block. It can be efficiently implemented in SIMD. A 7-tap conditional replacement filter is applied in Daala for a single pixel in a 8x8 block. The process is repeated for each pixel in each block being filtered.

To reduce ringing in very smooth regions of the image, the filter is applied a second time to combine multiple output values of the first filter. The second filter is applied either vertically or horizontally – in the direction most orthogonal to the one used in the first filter. For example, for a 45-degree direction, the second filter would be applied vertically. The combined effect of the two filters is a separable deringing filter that covers a total of 35 pixel taps.

Fig. 1 shows the effect of the deringing filter on edges at low bitrate. For the impact on objective metrics, see Table 2 in Section 4. An interactive demonstration of the deringing filter is also available [10].

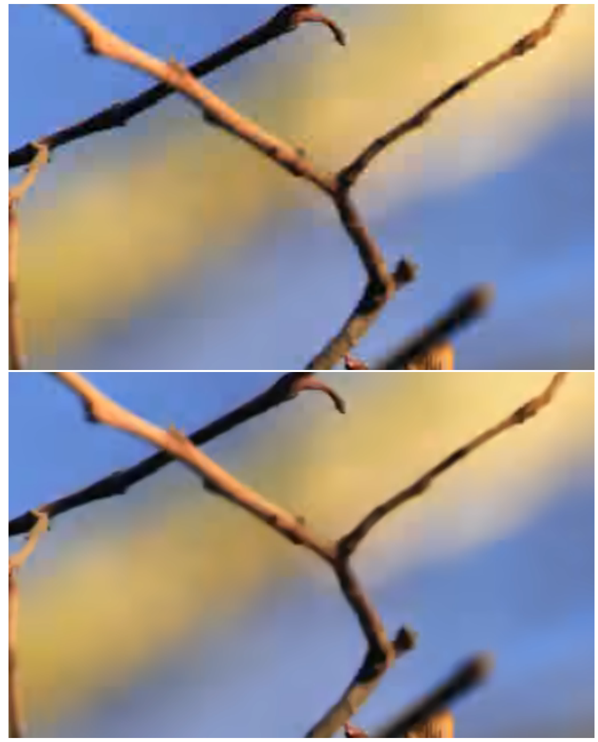

Fig. 1. Effect of the deringing filter at low bitrate. Top: without deringing. Bottom: with deringing.

3.2. 64x64 DCT

Larger transform sizes generally provide better coding efficiency, particularly for HD and UHD content. Since the last subjective test [2], Daala has added support for block sizes up to 64x64. When a 64x64 block is used in a keyframe, only the lower quadrant of coefficients is quantized and coded. This provides additional performance as regions where 64x64 blocks are used typically lack high frequency detail. This also makes it possible for a Daala based picture codec to use a lower complexity 64-point DCT which does not compute the upper 32 frequencies.

3.3. Reduced Lapping

Lapped transforms have complex interactions with variable transform sizes. In a previous version of Daala that used maximum lapping width for any given transform size, making block size decisions based on rate-distortion optimization (RDO) proved computationally intractable. This is why the lapping as implemented in [1] was fixed at 8-point almost everywhere, with the exception that an 8x8 block being subdivided into four 4x4 blocks used 4-point lapping on the internal boundaries. This caused 4x4 transform blocks to have

4-point lapping on two sides and 8-point lapping on the other two sides. The current version of Daala uses 4-point lapping on all block boundaries. Although it makes textures slightly worse and causes slightly more blocking artifacts, the ringing on edges is significantly reduced by using 4-point lapping.

3.4. Reduced Overhead Entropy Coder

The original entropy coder in Daala is based on a multiply-free algorithm described in [11]. We have replaced the piece-wise linear integer mapping from that paper with a new mapping that has less approximation error. Using the notation from [11], the new partition function is

$$e = \max(2R - 3t, 0) \tag{5}$$

$$f(x, t, R) = x + \min(x, e) + \min\left(\left\lfloor \frac{\max(x - e, 0)}{2} \right\rfloor, R - t\right) \tag{6}$$

This mapping is about three times as expensive to evaluate, but remains an order of magnitude less complex than a division and reduces the average overhead introduced by entropy coder by about 0.3%. While this is an acceptable trade-off for still picture coding, it may ultimately prove to be too expensive for real-time video to justify this reduction.

3.5. Finer Chroma Quantization

Among the most visible artifacts from the Daala version in [1] are chroma quantization artifacts caused by the chroma quantization being coarser than luma quantization at all bitrates. Since then, the chroma quantizers have changed to be coarser than luma at high bitrate, but finer at low bitrate. Although this results in a significant improvement in visual quality, it obviously regresses all the commonly-used luma-only metrics.

4. RESULTS

Because of the changes in chroma quantization (Section 3.5), it is hard to show the recent improvements in Daala using objective metrics. For this reason, we will show metrics for the current version *minus* the chroma quantization changes. The PSNR, PSNR-HVS [12], SSIM, and FAST-SSIM [13] metrics are obtained using the *Are We Compressed Yet?*[1] testing infrastructure. A test set *subset1*[2] composed of 50 still images is compressed and the metrics averaged together[3]. Table 1 shows the recent improvements at low (0.05 to 0.2 bit/pixel), medium (0.2 to 0.5 bit/pixel), and high (0.5 to 1.0 bit/pixel) bitrate. Among the improvements listed in Section 3, the one with the largest impact on metrics is the deringing filter, as shown in Table 2.

[1] https://arewecompressedyet.com/

[2] https://people.xiph.org/~tterribe/daala/subset1-y4m.tar.gz

[3] https://github.com/tdaede/rd_tool

Metric	Low (%)	Medium (%)	High (%)
PSNR	-6.3	-6.9	-7.9
PSNR-HVS	-6.7	-6.6	-6.3
SSIM	-4.8	-5.3	-6.5
FAST-SSIM	-2.4	-2.2	-0.6

Table 1. Bjøntegaard-delta [14] rate between the version of Daala presented in [1] and the current version on the *subset1* test set. Lower is better.

Metric	Low (%)	Medium (%)	High (%)
PSNR	-3.6	-3.0	-1.7
PSNR-HVS	-2.9	-2.2	-0.9
SSIM	-1.5	-1.6	-1.0
FAST-SSIM	+2.7	+3.4	+3.4

Table 2. Bjøntegaard-delta rate impact of the deringing filter on the *subset1* test set. Lower is better.

Fig. 2 shows the visual improvement on the picture that was the most problematic [5] for Daala in [2]. Both the reduction in deringing (due to deringing and reduced lapping) and the reduction in chroma artifacts (due to finer chroma quantization) are visible.

Fig. 3 shows the PSNR-HVS rate distortion performance of Daala compared to JPEG (libjpeg-turbo [15] and mozjpeg [16]), WebP [17], x264 [18], and BPG (HEVC) [19], using PSNR-HVS for *subset1*. The default compression parameters were used with all codecs with the exception of x264 which used the command line: `-preset placebo -crf=`. Subjective results from the Image Compression Grand Challenge at ICIP 2016 will provide more information on the perceptual quality of Daala as compared to other codecs.

5. FUTURE WORK

The Daala bit-stream has not been frozen yet, so it is still being improved. Among the recent features not demonstrated here are support for image bit depths up to 12 bits, as well as a special coding tool for non-photographic computer-generated images [20], which are generally better compressed with pixel prediction than block-based transform coding. However, some features such as metadata support and lossless RGB are still missing.

6. REFERENCES

[1] N. E. Egge, J.-M. Valin, T. B. Terriberry, T. Daede, and C. Montgomery, "Using Daala intra frames for still picture coding," in *Proceedings of Picture Coding Symposium*, 2015.

Fig. 2. Compressing the "bike" image at 0.25 bit/pixel. Left: Previous results presented in [2]. Right: current result.

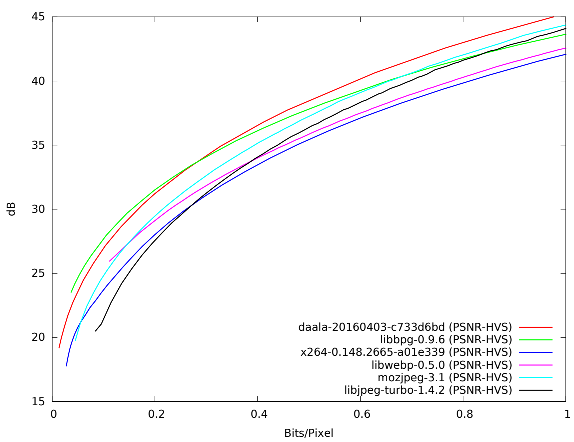

Fig. 3. Rate-distortion comparison between different codecs using PSNR-HVS on *subset1*.

[2] "Picture Coding Symposium 2015 feature event: Evaluation of current and future image compression technologies," `http://www.pcs2015.org/`.

[3] Henrique S. Malvar and David H. Staelin, "The LOT: Transform coding without blocking effects.," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 37, no. 4, pp. 553–559, 1989.

[4] T.D. Tran, Jie Liang, and Chengjie Tu, "Lapped transform via time-domain pre- and post-filtering," *Signal Processing, IEEE Transactions on*, vol. 51, no. 6, pp. 1557–1571, June 2003.

[5] T. J. Daede, N. E. Egge, J.-M. Valin, G. Martres, and T. B. Terriberry, "Daala: A perceptually-driven next generation video codec," `arXiv:1603.03129 [cs.MM]` `http://arxiv.org/abs/1603.03129`, 2016.

[6] T. B. Terriberry, "Coding tools for a next generation video codec," `https://tools.ietf.org/html/draft-terriberry-codingtools-02`, 2015.

[7] J.-M. Valin and T. B. Terriberry, "Perceptual vector quantization for video coding," in *Proceedings of SPIE Visual Information Processing and Communication*, 2015, vol. 9410, pp. 941009–941009–11, `arXiv:1602.05209 [cs.MM]` `http://arxiv.org/abs/1602.05209`.

[8] N. E. Egge. and J.-M. Valin, "Predicting chroma from luma with frequency domain intra prediction," in *Proceedings of SPIE Visual Information Processing and Communication*, 2015, vol. 9410, pp. 941008–941008–10, `arXiv:1603.03482 [cs.MM]` `http://arxiv.org/abs/1603.03482`.

[9] J.-M. Valin, "The Daala directional deringing filter," `arXiv:1602.05975 [cs.MM]` `http://arxiv.org/abs/1602.05975`, 2016.

[10] J.-M. Valin, "A deringing filter for daala... and beyond," `https://people.xiph.org/~jm/daala/deringing_demo/`, 2016.

[11] Lang Stuiver and Alistair Moffat, "Piecewise integer mapping for arithmetic coding," in *Data Compression Conference, 1998. DCC'98. Proceedings*. IEEE, 1998, pp. 3–12.

[12] N. Ponomarenko, F. Silvestri, K.Egiazarian, M. Carli, and V. Lukin, "On between-coefficient contrast mask-

ing of dct basis functions," in *Proceedings of Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics VPQM-07*, 2007.

[13] Ming-Jun Chen and Alan C. Bovik, "Fast structural similarity index algorithm," *Journal of Real-Time Image Processing*, vol. 6, no. 4, pp. 281–287, Dec. 2011.

[14] J. Moffitt T. Daede, "Video codec testing and quality measurement," `https://tools.ietf.org/html/draft-daede-netvc-testing`, 2015.

[15] "libjpeg-turbo website," `http://libjpeg-turbo.org/`.

[16] "mozjpeg website," `https://github.com/mozilla/mozjpeg`.

[17] "WebP website," `https://developers.google.com/speed/webp/`.

[18] "x264 website," `http://www.videolan.org/developers/x264.html`.

[19] "BPG website," `http://bellard.org/bpg/`.

[20] Jean-Marc Valin, "Screencasting considerations and L1-tree wavelet coding," `https://tools.ietf.org/html/draft-valin-netvc-l1tw-01`, 2015.