APPLYING PERCEPTUAL VECTOR QUANTIZATION OUTSIDE OF DAALA

Yushin Cho[A,B], David M. Barr[B], Jean-Marc Valin[A,B], and Timothy B. Terriberry[A,B]

A. Mozilla Corporation B. The Xiph.Org Foundation

ABSTRACT

Perceptual Vector Quantization (PVQ) was developed for the Daala video codec, which was intentionally designed to be different from traditional video codecs. Since Daala contains many unusual design elements, it can be difficult to separate out the contributions of each, leaving uncertainty as to how beneficial each one is on its own. We integrate PVQ into the Alliance for Open Media's AV1 codec, which follows a much more traditional design. This confirms that it delivers large perceptual improvements in such a codec, and uncovers implementation issues we will have to resolve before it can be adopted in the final codec design.

***Index Terms*—** AOM, AV1, Daala, compression, codec, video, perceptual

1. INTRODUCTION

The Daala video coding project includes many less-conventional techniques aimed at improving the perceptual quality of video [1]. Some are easily integrated into a traditional video codec, and some are not. Here, we give the results of integrating one of those techniques, Perceptual Vector Quantization, into the Alliance for Open Media's work-in-progress codec, AV1 [2]. PVQ is a form of gain-shape vector quantization designed to improve the perceptual quality of transform coefficient quantization over that of scalar quantization [3].

Section 2 gives a brief overview of PVQ, how it enables techniques such as signaling-free activity masking, and describes the challenges faced integrating PVQ into AV1. For complete details of PVQ in Daala, we refer readers to [3]. Section 3 describes a new metric used by Daala to weigh coding alternatives while taking account of human perception. We need this metric to avoid biasing top-level decisions against the very perceptual properties we hoped to preserve with PVQ. Finally, we present the results of integrating both that metric and PVQ into AV1 in Section 4.

The AV1 git repository contains the source code for all experiments [2]. We take all test results from our Are We Compressed Yet (AWCY) website, using the `objective-1-fast` sequence set [4] and the methodology currently proposed in the testing draft of the Internet Engineering Task Force (IETF)'s NETVC working group [5]. The source code for this is also available [6, 7]. It uses five objective metrics, whose implementations are available in the Daala git repository [8]. These are PSNR, SSIM [9], PSNR-HVS-M [10], multiscale SSIM [11], and CIEDE 2000 [12]. The first four are luma-only metrics, while the last is the only one to take into account color. Unlike PSNR, SSIM, PSNR-HVS-M, and MS SSIM all attempt to model human vision. CIEDE 2000 gives behavior very similar to PSNR.

2. PERCEPTUAL VECTOR QUANTIZATION

PVQ operates by extracting and explicitly coding perceptually meaningful parameters from transform coefficients. It partitions transform coefficients into *bands* following an octave-orientation structure, and computes, quantizes, and codes the overall energy in each band (the *gain*, g). It then normalizes the coefficients and codes the resulting unit vector (the *shape*) using a scheme based on Fischer's Pyramid Vector Quantization [13] with $N-1$ degrees of freedom (where N is the number of coefficients). This ensures the total number of degrees of freedom, N, remains unchanged, so there is no redundancy in the encoding.

When a prediction is available, PVQ first uses it to predict the gain. Then, after normalization, it quantizes and codes the angle, θ, between the predictor and the input viewed as N-dimensional vectors. A special *noref* flag (coded per-band) tells the decoder to ignore the prediction when θ would be $\frac{\pi}{2}$ or larger (i.e., when the correlation of the input and the prediction is zero or less). If PVQ does use the prediction, and θ is non-zero, it codes a point on the $N-1$ dimensional hypersphere of radius $g \sin\theta$ orthogonal to that prediction vector. This uses $N-2$ degrees of freedom with the same vector quantization scheme as in the unpredicted case.

The total number of degrees of freedom remains N, but instead of a collection of undiffierentiated AC coefficients, we now have two perceptually meaningful parameters. The gain indicates the overall contrast (energy) in each band, and θ signals how closely the input matches the prediction. This has a number of benefits. For example, it becomes cheaper to signal changes in the overall contrast, since they are collected in a single parameter, rather than requiring small updates to potentially dozens of non-zero coefficients. Additionally, the contrast is exactly the value the codec needs to know to use *activity masking* to control the quantization resolution [14].

We derive that resolution inside every block on a band-

by-band level with no additional signaling. The gain itself uses a non-linear companded quantizer. Then, once the gain is known, we use it to derive the quantization resolution of θ, as well as the size of the vector quantization codebook, which is related to the radius of the hypersphere: g in the non-predicted case, and $g \sin \theta$ in the predicted case. We disable this activity masking for chroma, and, since activity masking is less pronounced on edges, for 4×4 blocks.

To prevent packet loss from desynchronizing the bitstream, in the predicted case we use the approximation $\sin \theta \approx \theta$ to replace the dependency on $\sin \theta$ with the quantization index of θ. This also avoids a dependency on g, since the quantization resolution of θ depends on g in the same way (the terms cancel). Additionally, this allows a decoder to parse the bitstream without doing any of the signal processing necessary to reconstruct the final coefficient values.

2.1. The Integration of PVQ into AV1

In Daala, which uses lapped transforms, all prediction operates in the frequency domain to avoid causality issues caused by the lapping. AV1, however, uses traditional spatial intra prediction and block-based motion compensation. As a result, we must apply an extra forward transform to construct the frequency-domain predictions required on a per-band basis for PVQ, in both the encoder and decoder. In addition, transform coders commonly optimize the case of applying an inverse transform to only a small number of non-zero coefficients. However, with PVQ, the transform is not applied to a sparse residual. The output of dequantization is a complete vector in the neighborhood of the predictor, not a difference from the predictor. Thus it often has many non-zero values, making this optimization less effective. These issues contribute to some amount of decoder complexity increase, but transforms typically account for less than 10% of the computational budget of decoders for current-era codecs.

Still, the vector codebook search is slower than scalar quantization, and our implementation currently lacks SIMD. Compounding this, for each choice of prediction mode, transform type, block size, etc., the current encoder uses a full transform and quantization to evaluate the rate and distortion for Rate Distortion Optimization (RDO). In comparison, Daala has only one transform type and its encoder completely decouples prediction from transform coding, leaving only the transform block size decision at the transform stage. As a result, at its slowest settings, AV1 invokes the PVQ quantization routines up to 165 times more than in Daala [15]. That makes AV1 with PVQ enabled many times slower than with scalar quantization. We expect that much of this can be solved by using rate and distortion modeling to avoid invoking the quantizer as often [16]. Even for scalar, this would make searching the large space of potential coding options in AV1 significantly faster, and is expected to be necessary for practical encoders. Work to implement this is still underway.

PVQ must also account for the coefficient scanning order. AV1, like VP9 on which it is based, uses both a DCT as well as asymmetric DSTs as transforms, with multiple scanning orders depending on the transform type. PVQ, however, divides the coefficients into bands, and scans each band independently. To handle this, we keep the same band structure (for all transform types), and scan the coefficients within each band in the same order they would have been scanned in the whole block with scalar quantization. This works reasonably well as a first approximation, though we will likely have to revise the band structure when AV1 adds additional transform types, including, for example, the option of using the identity (no transform) in either the horizontal or vertical directions (or both). Similarly, we use the same quantization matrices used by Daala (for all transform types) when activity masking is enabled, and flat quantization when it is disabled.

3. THE DAALA DISTORTION METRIC

Making perceptual trade-offs during quantization cannot by itself ensure good perceptual quality. Quantization sits at the bottom of the Rate-Distortion Optimization (RDO) process, for a particular prediction block size, prediction mode, set of prediction parameters, transform size, and transform type. In practice, RDO selects from many possibilities for all of these things. In doing so, it must measure the distortion introduced by quantization in a way that is comparable for different choices of transform size and transform type.

If RDO uses a simple distortion metric, such as Mean Squared Error (MSE), it runs the risk of destroying the perceptual trade-offs made by the underlying quantization. For example, suppose a region contains areas of both high and low contrast. When coded with multiple transform blocks, PVQ will use less quantization in the regions of low contrast, and more quantization in the regions of high contrast. When coded with a single transform block, PVQ will quantize the whole region somewhere in between. MSE-based RDO will then tend to choose the latter over the former, since spending the extra bits in the low-contrast regions will appear to be inefficient and since it cannot tell that distortions hidden in the high-contrast region will be less visible. With the large block sizes of modern codecs, making macro-scale mistakes in the choice of prediction mode, transform size, or transform type can be devastating to visual quality. Another issue is that PVQ does not use activity masking for 4×4 blocks, as these typically contain edges. Using MSE as a top-level distortion metric creates a strong bias for choosing 4×4 blocks.

To combat these issues, we use a more sophisticated distortion metric for higher-level RDO decisions. It is intended as a crude approximation to PSNR-HVS-M [10] for use inside the coding loop, and contains many features that have long been used to weight the perceptual visibility of errors [17]:

1. A frequency-dependent weighting that approximates the Contrast Sensitivity Function (CSF).

2. An activity measure to account for spatial masking.
3. An energy preservation term that penalizes mismatches in the amount of activity in a region.

3.1. The Definition of the Metric

The metric operates over $N \times M$ regions in the luma plane, where N and M are multiples of 8 (for smaller regions or regions in the chroma planes, we simply use MSE). To begin, we apply a simple spatial filter whose frequency response is meant to approximate that of the human CSF. PSNR-HVS-M uses the JPEG quantization matrix for this purpose, and an earlier version of our metric applied a DCT and weighted the errors in the frequency domain in the same way, but this is very computationally expensive. Instead, we apply a 3-tap filter of the form $\begin{bmatrix}1 & K & 1\end{bmatrix}/(K+2)$ to the coding error in both the horizontal and vertical directions, using mirroring at the boundaries of the $N \times M$ block (mirroring tested slightly better than simple edge extension). We find that $K = 5$ provides a close fit for the desired frequency weighting, as Fig. 1 shows. The filter falls off a bit more steeply at high frequencies and does not apply a smaller weight to DC than the lowest AC frequency, but we saw improvements in PSNR-HVS-M compared to using a DCT and weighting errors in the frequency domain (see Table 1), possibly due to reduced aliasing and round-off errors in the calculations.

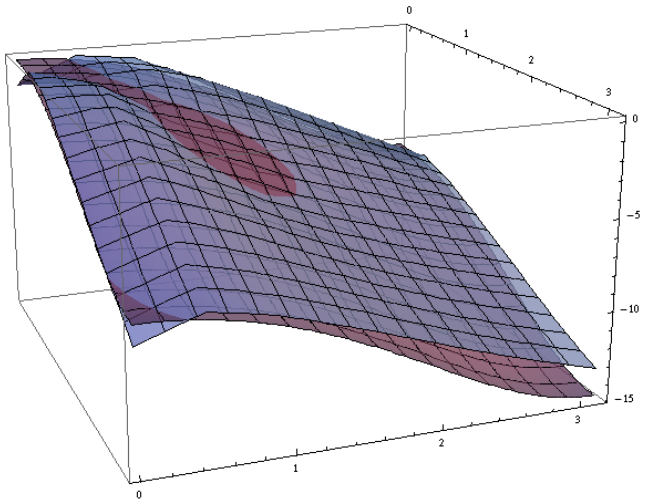

Fig. 1: 2-D frequency response in dB (vertical axis) of our spatial filter (purple, bottom) vs. an 8×8 DCT matrix (blue, top). The two horizontal axes are spatial frequency in radians.

Then we partition the region into 8×8 blocks. In each block, we compute the variance of overlapping 4×4 sub-blocks of the input image at every even offset (nine variance calculations total), and take the harmonic mean:

$$\sigma^2_{\text{avg}} = \frac{9}{\sum_{i=0}^{2}\sum_{j=0}^{2}\frac{1}{C_{\text{harm}}+\sigma^2_{2i,2j}}}, \tag{1}$$

PSNR	PSNR-HVS	SSIM	MS-SSIM	CIEDE 2000
−0.10%	−1.17%	0.53%	0.14%	−0.18%

Table 1: BD-rate changes when moving from an 8×8 DCT and frequency-dependent weighting to a spatial filter that approximates that weighting. All numbers are percent changes in bitrate at equivalent quality (more negative is better).

where $\sigma^2_{2i,2j}$ is the variance of the 4×4 sub-block at position $(2i, 2j)$ in the parent 8×8 block. We add the noise floor C_{harm}, currently set to 1 in AV1, to prevent division by zero. The harmonic mean gives more weight to less active (low contrast) regions, as this is where errors are more visible. In practice simpler measures such as the minimum variance also work, but the harmonic mean provided slightly better results when activity masking was enabled for PVQ in Daala.

We then define the activity, A, as

$$A \triangleq C^2_{\text{act}}\left(\frac{1}{4}+\sigma^2_{\text{avg}}\right)^{-\frac{1}{3}}, \tag{2}$$

where C_{act} is a calibration constant chosen roughly to keep equal bitrate with and without the activity term. We currently use $C_{\text{act}} = 1.95$. Again, the $\frac{1}{4}$ term is added as a noise floor. We chose the exponent, $-\frac{1}{3}$, to match the activity masking exponent used by PVQ itself [3].

Now, let $\hat{\sigma}^2_{2i,2j}$ be the variance of the 4×4 sub-block of the coded image and let $e_{i,j}$ be the filtered error signal. Then the total distortion in the 8×8 block is

$$D^{8\times8}_{\text{hvs}} \triangleq A\left(\sum_{i=0}^{7}\sum_{j=0}^{7} e^2_{i,j} + \sum_{i=0}^{2}\sum_{j=0}^{2}\left(\sigma_{2i,2j}-\hat{\sigma}_{2i,2j}\right)^2\right). \tag{3}$$

The first term measures the (frequency-weighted) error pixel-by-pixel, and the second measures mismatches in the amount of energy present (e.g., the injection of noise in flat regions or the low-passing of textured regions). We originally introduced the second term in Daala to avoid a bias towards skipping in regions with a small amount of texture. It also appears to provide a small gain on perceptual metrics in AV1 (see Table 2), but could be dropped for simplicity. The total is weighted by the overall activity of the region.

The total distortion for the $N \times M$ region is then just the sum of $D^{8\times8}_{\text{hvs}}$ for each block, scaled by another calibration constant, $C_{\text{hvs}}(QP)$. We use $C_{\text{hvs}}(QP)$ to scale the average distortion to match MSE when using the given Quantization Parameter (QP). This allows us to use the same lambda during RDO when using this distortion function, and allows us to easily combine distortion measured in the luma and chroma planes. To tune $C_{\text{hvs}}(QP)$ we fixed it at constant 1, collected coded sequences for 8 values spanning the QP range and for

PSNR	PSNR-HVS	SSIM	MS-SSIM	CIEDE 2000
0.17%	−0.01%	−0.32%	−0.19%	0.65%

Table 2: BD-rate changes by adding the energy preservation term in AV1. Measured on the first five frames of `objective-1-fast` only. All numbers are percent changes in bitrate at equivalent quality (more negative is better).

	PSNR	PSNR-HVS	SSIM	MS-SSIM	CIEDE 2000
PVQ	3.39%	5.06%	3.40%	4.06%	2.87%
DD	11.47%	−3.64%	−2.41%	−8.51%	7.58%
PVQ+AM	30.64%	−5.88%	1.29%	−11.62%	23.78%

Table 3: BD-rate changes from MSE-optimized scalar quantization using MSE-optimized PVQ, the Daala distortion metric (DD), and PVQ with activity masking and the Daala distortion metric (PVQ+AM). All numbers are percent changes in bitrate at equivalent quality (more negative is better).

each QP value calculated a simple linear regression of MSE against $D_{\text{hvs}}^{8\times8}$. The trends for keyframes and other frames differ significantly, but keyframes tend to use lower QP values, so we fit a piecewise quadratic interpolation that eased between the trends across the range.

The resulting distortion function is slower than plain MSE, as each 8×8 block requires applying two 3-tap filters, and one multiply per pixel for both the input and the target to compute the variances, plus $O(1)$ additional work. However, it is not too slow to be impractical. Some computation could be saved by re-using the variances and activity calculations for the input image when comparing multiple alternatives.

4. RESULTS

We test the codec[1] on 15 frames of `objective-1-fast`[2] in four configurations: scalar quantization, MSE-optimized PVQ (with flat quantization and no activity masking), scalar quantization with the Daala distortion metric, and PVQ with non-flat quantization, activity masking, and the Daala distortion metric. Table 3 summarizes the results.

In Daala, MSE-optimized PVQ was significantly better than scalar [3]. Here it is somewhat behind. This is likely due to relatively poor tuning of Daala's scalar implementation. Conversely, in preliminary results presented last year, MSE-optimized PVQ in AV1 gave performance very close to that of scalar: within 1% BD-rate on all metrics, and a slight advantage on PSNR and CIEDE 2000 [15]. Improvements in the tuning of the scalar quantizer in AV1 since then and regressions in PVQ due to other changes have widened the gap. However, these results demonstrate that quality of implementation matters more than the choice of scalar vs. vector.

While using perceptually-tuned quantization with a simple top-level distortion metric will not work, using the Daala distortion metric with scalar quantization demonstrates that the reverse does work. Despite the degredation in PSNR, we show significant improvements in all perceptual metrics. Ad-hoc inspection confirms that they do look better, with sharper edges in low-contrast areas and better detail retention.

PVQ with all of its perceptual features enabled amplifies this trend even further, with the exception of SSIM. The perceptual gains are even more significant given that MSE-optimized PVQ was starting from behind.

(a) Scalar (b) PVQ+AM

Fig. 2: Detail section frame 20 of `dark70p` encoded at equal bitrate with (a) scalar quantization and (b) PVQ with activity masking. The shirt retains more texture, and banding, ringing and blockiness are reduced.

(a) Scalar (b) PVQ+AM

Fig. 3: Detail section of frame 20 of `drivingpov` encoded at equal bitrate with (a) scalar quantization and (b) PVQ with activity masking. Details in the road are better preserved.

5. CONCLUSION

Overall, PVQ delivers substantial visual quality improvements in a traditional codec design. Some issues remain to be solved, such as the increase in encoder complexity, and some integration issues with other current or planned AV1 tools. We are actively working to solve them in future work.

[1] Using commit `e93acb2d228c` in the AV1 git repository [2].

[2] 15 frame results were used only to meet publication deadlines. If accepted, the final paper will include results for the complete sequences.

6. REFERENCES

[1] Yushin Cho, Thomas J. Daede, Nathan E. Egge, Guillaume Martres, Tristan Matthews, Christopher Montgomery, Timothy B. Terriberry, and Jean-Marc Valin, "Perceptually-driven video coding with the Daala video codec," in *Proc. of the 39th SPIE Conference on Applications of Digital Image Processing*, San Jose, Aug. 2016, vol. 9971.

[2] "Alliance for open media git repository," https://aomedia.googlesource.com/aom.

[3] Jean-Marc Valin and Timothy B. Terriberry, "Perceptual vector quantization for video coding," in *Proceedings of Visual Information Processing and Communication VI*, Amir Said, Onur G. Guleryuz, and Robert L. Stevenson, Eds., San Francisco, CA, Mar. 2015, vol. 9410.

[4] "Are we compressed yet? website," https://arewecompressedyet.com/.

[5] Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik, "Multi-scale structural similarity for image quality assessment," in *Proceedings of the 37th IEEE Conference on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2003, vol. 2, pp. 1398–1402.

[6] "Are we compressed yet? git repository," https://github.com/tdaede/awcy.

[7] "rd_tool git repository," https://github.com/tdaede/rd_tool.

[8] "Daala git repository," https://git.xiph.org/daala.git.

[9] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[10] Nikolay Ponomarenko, Flavia Silvestri, Karen Egiazarian, Marco Carli, Jaakko Astola, and Vladimir Lukin, "On between-coefficient contrast masking of DCT basis functions," in *CD-ROM Proceedings of the 3rd International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM '07)*, Scottsdale, Arizona, Jan. 2007.

[11] 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.

[12] Gaurav Sharma, Wencheng Wu, and Edul N. Dalal, "The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations," *Color Research & Applications*, vol. 30, no. 1, pp. 21–30, Feb. 2005.

[13] Thomas R. Fischer, "A pyramid vector quantizer," *IEEE Transactions on Information Theory*, vol. 32, no. 4, pp. 568–583, July 1986.

[14] Wilfried Osberger, Anthony J. Maeder, and Neil Bergmann, "A perceptually based quantization technique for MPEG encoding," in *Proc. of the 3rd SPIE Conference on Human Vision and Electronic Imaging*, San Jose, Jan. 1998, vol. 3299, pp. 148–159.

[15] Yushin Cho, "Applying PVQ outside Daala," https://tools.ietf.org/html/draft-cho-netvc-applypvq, Nov. 2016.

[16] Hsueh-Ming Hang and Jiann-Jone Chen, "Source model for transform video coder and its application—part I: Fundamental theory," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 2, pp. 287–298, Apr. 1997.

[17] J. O. Limb, "Distortion criteria of the human viewer," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 12, pp. 778–793, Dec. 1979.