Energy Preservation in PVQ-Based Video Coding

Jean-Marc Valin

March 17, 2012

1 Introduction

This mini-paper describes a proposal for adapting the CELT energy conservation principle to video coding based on a pyramid vector quantizer (PVQ). One potential advantage of conserving energy of the AC coefficients in video coding is preserving textures rather than low-passing them. Also, by introducing a fixed-resolution PVQ-type quantizer, we automatically gain a simple activity masking model.

The main challenge of adapting this scheme to video is that we have a good prediction (the reference frame), so we are essentially starting from a point that is already on the PVQ hyper-sphere, rather than at the origin like in CELT. Other challenges are the introduction of a quantization matrix and the fact that we want the reference (motion predicted) data to perfectly correspond to one of the entries in our codebook.

2 Encoder

Let vector $\mathbf{x}_d$ denote the (pre-normalization) DCT band to be coded in the current block and vector $\mathbf{r}_d$ denote the corresponding reference after motion compensation, the encoder computes and encodes the "band gain"

$$g = \sqrt{\mathbf{x}_d^T \mathbf{x}_d} \,. \tag{1}$$

Let $\mathbf{Q}$ be a diagonal matrix containing the quantization step size for each element of $\mathbf{x}_d$, the normalized band is computed as

$$\mathbf{x} = \frac{\mathbf{Q}^{-1}\mathbf{x}_d}{\left\|\mathbf{Q}^{-1}\mathbf{x}_d\right\|} \,, \tag{2}$$

with the normalized reference $\mathbf{r}$ similarly computed based on $\mathbf{r}_d$. The encoder then finds the position and sign of the maximum value in $\mathbf{r}$

$$m = \operatorname*{argmax}_i |r_i| \tag{3}$$

$$s = \operatorname{sgn}(r_m) \tag{4}$$

and computes the Householder reflection that reflects $\mathbf{r}$ to $-s\mathbf{e}_m$. The reflection vector is given by

$$\mathbf{v} = \mathbf{r} + s\mathbf{e}_m \ . \tag{5}$$

The encoder reflects the normalized band to find

$$\mathbf{z} = \mathbf{x} - \frac{2}{\mathbf{v}^T\mathbf{v}}\mathbf{v}\left(\mathbf{v}^T\mathbf{x}\right) \ . \tag{6}$$

The similarity between the current band and the reference band is represented by the angle (assuming no quantization)

$$\theta = \arccos\frac{-sz_m}{\|\mathbf{z}\|} \ .$$

Let N be the number of dimensions in $\mathbf{x}$ and K be the number of pulses in our codebooks, we search for the codebook entry

$$q = \underset{i}{\operatorname{argmax}}\frac{\mathbf{p}_i^T\left(\mathbf{z} + sz_m\mathbf{e}_m\right)}{\sqrt{\mathbf{p}_i^T\mathbf{p}_i}} \ , \tag{7}$$

where $\mathbf{p}_i$ is the i^{th} combination of magnitudes and signs that satisfies $\|\mathbf{p}_i\|_{L1} = K$. Let θ_{opt} be the post-quantization optimal angle, the mean square error becomes

$$E = \left(-s\cos\theta_{opt}\mathbf{e}_m + \sin\theta_{opt}\hat{\mathbf{p}} - \mathbf{z}\right)^2 \ ,$$

where $\hat{\mathbf{p}} = \mathbf{p}_q/\sqrt{\mathbf{p}_q^T\mathbf{p}_q}$Solving for $\frac{\partial E}{\partial\theta} = 0$ and knowing that $\hat{\mathbf{p}}^T\mathbf{e}_m = 0$, we have

$$\frac{\partial}{\partial\theta}\left(-s\cos\theta_{opt}\mathbf{e}_m + \sin\theta_{opt}\hat{\mathbf{p}}\right)\mathbf{z} = 0$$

$$\left(s\sin\theta_{opt}\mathbf{e}_m + \cos\theta_{opt}\hat{\mathbf{p}}\right)\mathbf{z} = 0$$

$$\sin\theta_{opt}\mathbf{e}_m\mathbf{z} = -s\cos\theta_{opt}\hat{\mathbf{p}}\mathbf{z}$$

$$\theta_{opt} = -s\arctan\frac{\hat{\mathbf{p}}\mathbf{z}}{\mathbf{e}_m\mathbf{z}} \ .$$

The resolution of the gain and angle, as well as the number of pulses should all be derived from a single quality parameter. The encoder transmits the gain g, the quantized angle $\hat{\theta}$, and the $\mathbf{p}_q$ vector. Neither s nor m need to be transmitted since they can be obtained from the decoder. Encoding $\mathbf{p}_q$ should make use of the fact that K is known and is left as an exercise to the implementer.

3 Decoder

The decoder starts by decoding the codebook entry $\mathbf{p}_q$ and uses it to reconstruct the unit-norm reflected band as

$$\hat{\mathbf{z}} = -s\cos\hat{\theta}\mathbf{e}_m + \sin\hat{\theta}\frac{\mathbf{p}_q}{\sqrt{\mathbf{p}_q^T\mathbf{p}_q}} \ . \tag{8}$$

Because the decoder has access to exactly the same reference as the encoder, it is able to apply (3)-(5) to obtain the same $\mathbf{v}$ as used in the encoder. The decoded normalized band is

$$\hat{\mathbf{x}} = \hat{\mathbf{z}} - \frac{2}{\mathbf{v}^T\mathbf{v}}\mathbf{v}\left(\mathbf{v}^T\hat{\mathbf{x}}\right) . \tag{9}$$

The renormalized band is computed by taking into account the quantization resolution:

$$\hat{\mathbf{x}}_d = \hat{g}\frac{\mathbf{Q}\hat{\mathbf{x}}}{\|\mathbf{Q}\hat{\mathbf{x}}\|} . \tag{10}$$

4 Coding Resolution

It is desirable for a single quality parameter to control K and the resolution of gain and angle. That quality parameter should also take into account activity masking to some extent. According to Jason Garrett-Glaser, x264's activity masking uses a resolution proportional to the $g^{2\alpha}$, with $\alpha = 0.173$. We can derive a scalar quantizer that follows this resolution:

$$\hat{g} = Q_g \gamma^{1+2\alpha} ,$$

where γ is the gain quantization index and Q_g is the gain resolution and "master" quality parameter. If we assume that MSE is a good criterion, then the angle quantization resolution should be (roughly)

$$Q_\theta = Q_g \left(1 + 2\alpha\right) \gamma^{2\alpha} .$$

To derive the optimal K we need to consider the cosine distance between adjacent codevectors $\mathbf{p}_1$ and $\mathbf{p}_2$ for two cases: $K < N$ and $K > N$. For $K < N$, the worst resolution occurs when no value in $\mathbf{p}_q$ is larger than one. In that case, the two closest codevectors have a cosine distance

$$\begin{aligned}\cos\tau &= \frac{\mathbf{p}_1^T\mathbf{p}_2}{\sqrt{\mathbf{p}_2^T\mathbf{p}_2}\sqrt{\mathbf{p}_2^T\mathbf{p}_2}} \\ &= \frac{K-1}{K} \\ &= 1 - \frac{1}{K}\end{aligned}$$

By approximating the cosine, we then get

$$\begin{aligned}1 - \frac{1}{K} = \cos\tau &\approx 1 - \frac{\tau^2}{2} \\ K &\approx \frac{2}{\tau^2}\end{aligned} \tag{11}$$

For $K > N$ the worst resolution happens when all values are equal to K/N in $\mathbf{p}_1$ and $\mathbf{p}_2$ differs by one pulse. In that case

$$\begin{aligned}\cos\tau &= \frac{K^2/N}{\sqrt{\frac{K^2}{N}}\sqrt{\frac{K^2}{N}+2}} \\ &= \frac{1}{\sqrt{1+\frac{2N}{K^2}}} \\ &\approx 1-\frac{N}{K^2}\end{aligned}$$

By approximating the cosine, we get

$$\begin{aligned}1-\frac{N}{K^2} &= \cos\tau \approx 1-\frac{\tau^2}{2} \\ K &\approx \sqrt{\frac{2N}{\tau}}\ .\end{aligned} \tag{12}$$

By combining (11) with (12), we have

$$K \approx \min\left(\sqrt{\frac{2N}{\tau}}, \frac{2}{\tau}\right)$$

The last step is to set

$$\tau = \sin\hat{\theta}$$

to account for the fact that the more the image differs from the reference, the higher the resolution needs to be.

5 Theoretical Ramblings on SSIM

According to Wikipedia, the SSIM metric is defined as

$$\mathrm{SSIM}\left(x,y\right) = \left(\frac{\mu_x\mu_y+c_1}{\mu_x^2+\mu_y^2+c_1}\right)\cdot\left(\frac{\sigma_{xy}+c_2}{\sigma_x^2+\sigma_y^2+c_2}\right)\ .$$

Where μ_x and μ_y are the DC of images x and y and σ_x and σ_y are the RMS value of the AC coefficients of images x and y. From now on, we will consider x to be the reference image and y to be the coded image. Now, let's ignore the DC for now and define a Simplified SSIM metric as

$$\mathrm{SSSIM}\left(x,y\right) = \frac{2\sigma_{xy}+c_2}{\sigma_x^2+\sigma_y^2+c_2}\ .$$

This is the metric we'll try optimizing here. First, let $g = \sigma_y/\sigma_x$ be gain that the codec causes on the AC coefficients and $\hat{y} = y/g$. Solving for

$$\frac{d}{dg}\mathrm{SSSIM}\left(x,y\right) = \frac{d}{dg}\frac{2g\sigma_{x\hat{y}}+c_2}{\sigma_x^2\left(1+g^2\right)+c_2} = 0$$

we find that the optimal gain that maximizes SSSIM is $g_{max} \approx 1 - \frac{c_2}{2\sigma_x^2} \cdot \left(\frac{\sigma_x^2}{\sigma_{x\hat{y}}} - 1\right)$. This means that conserving energy ($g_{max} = 1$) is a good thing to do as long as the contrat is high enough ($\frac{c_2}{2\sigma_x^2}$ is small) or the bit-rate is high enough ($\frac{\sigma_x^2}{\sigma_{x\hat{y}}}$ close to 1).

Now, let's consider a spherical horse in simple harmonic motion... or to be more exact, let's consider that the PVQ codebook is perfectly uniform over the sphere and that $g_{max} = 1$. We get

$$\mathrm{SSSIM}\left(x, y\right) = \frac{\sigma_{xy} + c_2/2}{\sigma_x^2 + c_2/2} \,,$$

where $\sigma_{xy}/\sigma_x^2 = \cos\theta$ is the cosine distance between x and y. Assuming a uniform quantizer, we have

$$\theta \propto 2^{-b/(N-1)} \,,$$

where b is the number of bits allocated and N is the number of AC coefficients. Let $c' = c/(2\sigma_x^2)$...

<FIXME: This needs to be cleaned up>

$$\mathrm{SSSIM}\left(x, y\right) = \frac{\cos\theta + c'}{1 + c'} \approx \frac{1 - \theta^2 + c'}{1 + c'} \,,$$

Trying to make SSIM equal for two blocks:

$$\frac{1 + c_1' - 2^{-2b_1/(N-1)}}{1 + c_1'} = \frac{1 + c_2' - 2^{-2b_2/(N-1)}}{1 + c_2'}$$

The optimal bit offset is

$$b = -\frac{N-1}{2} \log_2 \left(1 + 2c_2/\sigma_x^2\right)$$

From this (theoretically) optimal offset, we can encode only the deviation from the optimal allocation. In practice, b would not be an exact bit allocation like for CELT, but only the "quantization step exponent".

6 Conclusion

While it seems like a good idea, we're still experimenting with the details.