Pyramid Vector Quantization for Video Coding
draft-valin-videocodec-pvq-00

Abstract

This proposes applying pyramid vector quantization (PVQ) to video
coding.

Status of this Memo

This Internet-Draft is submitted in full conformance with the
provisions of BCP 78 and BCP 79. This document may not be modified,
and derivative works of it may not be created, and it may not be
published except as an Internet-Draft.

Internet-Drafts are working documents of the Internet Engineering
Task Force (IETF). Note that other groups may also distribute
working documents as Internet-Drafts. The list of current Internet-
Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months
and may be updated, replaced, or obsoleted by other documents at any
time. It is inappropriate to use Internet-Drafts as reference
material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 18, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the
document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal
Provisions Relating to IETF Documents
(http://trustee.ietf.org/license-info) in effect on the date of
publication of this document. Please review these documents
carefully, as they describe your rights and restrictions with respect
to this document. Code Components extracted from this document must
include Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.
Table of Contents

1. Introduction ..................................................... 3
2. Terminology ....................................................... 3
3. Gain-Shape Coding and Activity Masking .............................. 3
4. Householder Reflection ........................................... 4
5. Angle-Based Encoding ............................................. 5
6. Pyramid-Based Encoding .......................................... 7
7. Bi-prediction .................................................... 7
8. Development Repository .......................................... 7
9. IANA Considerations .............................................. 8
10. Security Considerations ......................................... 8
11. Acknowledgements ............................................... 8
12. References ....................................................... 8
   12.1. Normative References ....................................... 8
   12.2. Informative References ..................................... 8
Author’s Address ..................................................... 8
1. Introduction

This draft describes a proposal for adapting the Opus RFC 6716 [RFC6716] energy conservation principle to video coding based on a pyramid vector quantizer (PVQ) [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. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Gain-Shape Coding and Activity Masking

The main idea behind the proposed video coding scheme is to code groups of DCT coefficient as a scalar gain and a unit-norm "shape" vector. A block’s AC coefficients may all be part of the same group, or may be divided by frequency (e.g. by octave) and/or by directionality (horizontal vs vertical).

It is desirable for a single quality parameter to control the resolution of both the gain and the shape. Ideally, that quality parameter should also take into account activity masking, that is, the fact that the eye is less sensitive to regions of an image that have more details. According to Jason Garrett-Glaser, the perceptual analysis in the x264 encoder uses a resolution proportional to the variance of the AC coefficients raised to the power $a$, with $a=0.173$. For gain-shape quantization, this is equivalent to using a resolution of $g^{(2a)}$, where $g$ is the gain. We can derive a scalar quantizer that follows this resolution:

$g = Q_g \gamma$

$1+2a$

Valin

Expires April 18, 2013

[Page 3]
where gamma is the gain quantization index and Q_G is the gain resolution and main quality parameter.

An important aspect of the current proposal is the use of prediction. In the case of the gain, there is usually a significant correlation with the gain of neighboring blocks. One way to predict the gain of a block is to compute the gain of the coefficients obtained through intra or inter prediction. Another way is to use the encoded gain of the neighboring blocks to explicitly predict the gain of the current block.

4. Householder Reflection

Let vector \( x_d \) denote the (pre-normalization) DCT band to be coded in the current block and vector \( r_d \) denote the corresponding reference (based on intra prediction or motion compensation), the encoder computes and encodes the "band gain" \( g = \sqrt{x_d^T x_d} \). The normalized band is computed as

\[
\frac{x_d}{\| x_d \|}
\]

with the normalized reference \( r \) similarly computed based on \( r_d \). The encoder then finds the position and sign of the maximum value in \( r \):

\[
m = \text{argmax}_i | r_i |
\]

\[
s = \text{sign}(r_m)
\]

and computes the Householder reflection that reflects \( r \) to \(-s e_m\). The reflection vector is given by

\[
v = r + s e_m .
\]

The encoder reflects the normalized band to find the unit-norm vector

\[
\frac{v^T x}{v^T v} = x - 2 \frac{v}{v^T v} .
\]

The closer the current band is from the reference band, the closer \( z \) is from \(-s e_m\). This can be represented either as an angle, or as a coordinate on a projected pyramid.
5. Angle-Based Encoding

Assuming no quantization, the similarity can be represented by the angle

\[ \theta = \arccos(-s \ z_m) \ . \]

If \( \theta \) is quantized and transmitted to the decoder, then \( z \) can be reconstructed as

\[ z = -s \ \cos(\theta) \ e_m + \sin(\theta) \ z_r \ , \]

where \( z_r \) is a unit vector based on \( z \) that excludes dimension \( m \).

The vector \( z_r \) can be quantized using PVQ. Let \( y \) be a vector of integers that satisfies

\[ \sum_i(|y[i]|) = K \ , \]

with \( K \) determined in advance, then the PVQ search finds the vector \( y \) that maximizes \( y^T \ z_r / (y^T \ y) \) . The quantized version of \( z_r \) is

\[ z_{rq} = \frac{y}{|| y ||} \ . \]

If we assume that MSE is a good criterion for optimizing the resolution, then the angle quantization resolution should be (roughly)

\[ Q_{\theta} = \frac{d}{\gamma} \frac{1}{g} \frac{1+2a}{\gamma} \ . \]

To derive the optimal \( K \) we need to consider the cosine distance between adjacent codevectors \( y_1 \) and \( y_2 \) for two cases: \( K<N \) and \( K>N \). For \( K<N \), the worst resolution occurs when no value in \( y \) is larger than one. In that case, the two closest codevectors have a cosine distance

\[ \cos(\tau) = 1 - \frac{1}{K} \ . \]

(derivation left as an exercise for the reader)

By approximating \( \cos(\tau) \) as \( 1 - \tau^2 \), we get
\[
K = \frac{2}{\tau}.
\]

For \( K > N \) the worst resolution happens when all values are equal to \( K/N \) in \( y_1 \), and \( y_2 \) differs by one pulse. In that case

\[
\frac{N}{\cos(\tau)} = 1 - \frac{\tau}{K^2}.
\]

(also left as an exercise for the reader)

which gives the approximation

\[
K = \frac{\sqrt{2 N}}{\tau}.
\]

By combining the two cases, we have

\[
K = \min\left| \frac{\sqrt{2 N}}{\tau}, \frac{2}{\tau^2} \right|.
\]

To achieve uniform resolution in all dimensions,

\[
\frac{Q_{\theta}}{\tau} = \frac{\sin(\theta)}{\sin(\theta)}.
\]

The value of \( K \) does not need to be coded because all the variables it depends on are known to the decoder. However, because \( Q_{\theta} \) depends on the gain, this can lead to unacceptable loss propagation behavior in the case where inter prediction is used for the gain. This problem can be worked around by making the approximation \( \sin(\theta) \approx \theta \). With this approximation, then \( \tau \) is equal to the inverse of the \( \theta \) quantization index, with no dependency on the gain. Alternatively, instead of quantizing \( \theta \), we can quantize \( \sin(\theta) \) which also removes the dependency on the gain. In the general case, we quantize \( f(\theta) \) and then assume that \( \sin(\theta) = f(\theta) \). A possible choice of \( f(\theta) \) is a quadratic function of the form:

\[
f(\theta) = a_1 \theta - a_2 \theta^2.
\]
where $a_1$ and $a_2$ are two constants satisfying the constraint that $f(\pi/2)=\pi/2$. The value of $f(\theta)$ can also be predicted, but in case where we care about error propagation, it should only be predicted from information coded in the current frame.

6. Pyramid-Based Encoding

Instead of explicitly encoding an angle, it is also possible to apply PVQ directly on $z$. In that case, the angle is replaced by $\nu = K + s y[m]$, with $0 \leq \nu \leq 2K$, with smaller values more likely (assuming the predictor is good). Based on calculations similar to those for the angle-based encoding, the value of $K$ is set to

$$K = \min \left( \frac{c_1 \gamma}{N'}, \frac{c_2 \gamma^2}{N'} \right),$$

where $c_1$ and $c_2$ are empirical constants.

As is the case for angle-based encoding, $K$ does not need to be coded. However, if the gain parameter $\gamma$ is predicted from a different frame, then this would lead to unacceptable error propagation behavior. To reduce the error propagation, instead of coding $\nu$ we can code $\nu' = K - |y[m]|$, along with the sign of $s*y[m]$. In this way, any error in the gain will lead to the wrong value of $K$, but will not cause a desynchronization of the range coder as would happen when decoding the wrong number of symbols.

7. Bi-prediction

We can use this scheme for bi-prediction by introducing a second $\theta$ parameter. For the case of two (normalized) reference frames $r_1$ and $r_2$, we introduce $s_1 = (r_1 + r_2)/2$ and $s_2 = (r_1 - r_2)/2$. We start by using $s_1$ as a reference, apply the Householder reflection to both $x$ and $s_2$, and evaluate $\theta_1$. From there, we derive a second Householder reflection from the reflected version of $s_2$ and apply it to $z$. The result is that the $\theta_2$ parameter controls how the current image compares to the two reference images. It should even be possible to use this in the case of fades, using two references that are before the frame being encoded.

8. Development Repository

The algorithms in this proposal are being developed as part of Xiph.Org’s Daala project. The code is available in the Daala git repository at <https://git.xiph.org/daala.git>. See
Internet-Draft                  Video PVQ                   October 2012

<https://xiph.org/daala/> for more information.

9.  IANA Considerations

This document makes no request of IANA.

10. Security Considerations

This draft has no security considerations.

11. Acknowledgements

Thanks to Jason Garrett-Glaser, Timothy Terriberry, Greg Maxwell, and Nathan Egge for their contribution to this document.

12. References

12.1.  Normative References


12.2.  Informative References


Author’s Address

Jean-Marc Valin
Mozilla
650 Castro Street
Mountain View, CA  94041
USA

Email: jmvalin@jmvalin.ca