

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 10, 2015

N. Egge
T. Terriberry
Mozilla Corporation
March 9, 2015

Time Domain Lapped Transforms for Video Coding
draft-egge-videocodec-tdlt-01

Abstract

This proposes the use of Time Domain Lapped Transforms (TDLT) as the transform step for video coding.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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 September 10, 2015.

Copyright Notice

Copyright (c) 2015 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.

This document may not be modified, and derivative works of it may not be created, and it may not be published except as an Internet-Draft.

Table of Contents

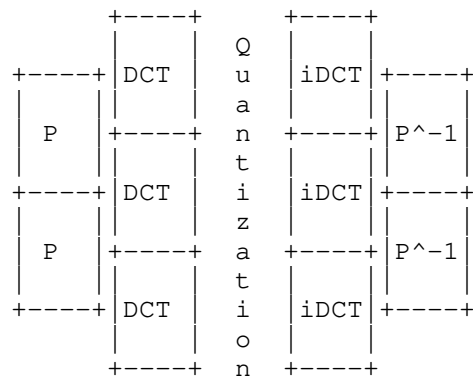
| | |
|---|----|
| 1. Introduction | 2 |
| 2. TDLT Defined | 2 |
| 3. Lapped-Transform Selection | 4 |
| 3.1. Coding Gain | 5 |
| 3.2. Transform Width | 6 |
| 4. Optimal Transform Coefficients | 6 |
| 4.1. Exhaustive Search | 6 |
| 4.2. Stochastic Search | 7 |
| 4.3. Ramp Constraint | 8 |
| 5. Intra Prediction | 10 |
| 6. Motion Compensation | 11 |
| 7. Multiple Block Sizes | 11 |
| 7.1. Variable Sized Lapping | 12 |
| 7.2. Fixed Sized Lapping | 15 |
| 8. IANA Considerations | 16 |
| 9. Security Considerations | 16 |
| 10. Acknowledgments | 16 |
| 11. Informative References | 16 |
| Authors' Addresses | 17 |

1. Introduction

This draft outlines a proposal to adapt the Time-Domain Lapped Transforms (TDLT) for use in video coding. Lapped transforms were proposed for video coding at least as far back as 1989 [Malv89]. Like the loop filters more commonly found in recent video coding standards, TDLTs use a post-processing filter that runs between block edges to reduce or eliminate blocking artifacts. Unlike a loop filter, the TDLT filter is invertible, allowing the encoder to run the inverse filter on the input video. This decorrelates blocks before they are passed through a normal block transform and quantization step, improving coding gain (which helps in both smooth and highly textured areas), in addition to reducing blocking artifacts.

2. TDLT Defined

The Time-Domain Lapped Transform can be viewed as a set of pre and post filters to an existing block-based DCT transform. The idea is to place an invertible filter along the block boundaries outside an existing block-based DCT encoder.



The pre-filter P operates in the time domain, processing block boundaries and removing inter-block correlation. The blocks are then transformed by the DCT into the frequency domain, where the resulting coefficients are quantized and encoded. When decoding, the inverse operator P^{-1} is applied as a post-filter to the output of the inverse DCT. This has two benefits:

1. Quantization errors are spread over adjacent blocks via the post-filter P^{-1} , reducing blocking artifacts. This eliminates the need for a separate deblocking filter.
2. The increased support region of the transform allows it to take advantage of inter-block correlation to achieve a higher coding gain than a non-overlapped DCT. This allows it to more effectively code both smooth and textured regions.

The pre-filter P is defined in [Tran01] as follows:

$$P = \frac{1}{2} \begin{bmatrix} I & J \\ J & -I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} I & J \\ J & -I \end{bmatrix}$$

Here I is the identity matrix and J is the "reversal matrix", obtained by simply re-ordering the rows of the identity matrix in reverse order. The V matrix is a free parameter, and as long as V is invertible, this filter structure guarantees perfect reconstruction, linear phase, and biorthogonality. If V is orthogonal, then the overall transform is also orthogonal instead of just biorthogonal.

For the case of the 4x8 TDLT, we use the following invertible matrix for V:

$$V = \begin{bmatrix} 1 & q_0 \\ & \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ & \\ p_0 & 1 \end{bmatrix} \begin{bmatrix} s_0 & 0 \\ & \\ 0 & s_1 \end{bmatrix}$$

Thus for the 4x8 case, the pre-filter and post-filter are completely described by the four parameters q_0 , p_0 , s_0 , and s_1 . In general, any invertible V matrix may be used. However, factoring V into a series of lifting steps ensures that it can be implemented efficiently, and can reduce the number of parameters required by the optimization process, since the full flexibility of an arbitrary invertible matrix is not required to achieve good coding gain.

[Tran01] proposes two reduced-parameter factorizations, dubbed Type III and Type IV. These are identical in the 4x8 case, but for larger transforms the differ in the order that the p_i and q_i steps are applied: interleaved for Type III and ascending and then descending order for Type IV. While Type III appears to give slightly higher coding gain when unconstrained, when coupled with the ramp constraint discussed below and the constraint that all coefficients be dyadic rationals, the number of feasible solutions is much smaller than with Type IV. The increased number of feasible solutions allows Type IV transforms to achieve higher coding gains than Type III when these constraints are imposed. This definition easily extends to the 8x16 and 16x32 TDLT case with similar parameterizations. In general, we use the Type IV factorization from [Tran01]. For a V matrix of size M , this has $(M-1)$ p_i and $(M-1)$ q_i parameters, and M s_i parameters. For a transform of size $N \times 2N$, this gives a total of $1.5N-2$ parameters. This is also the number of lifting steps that must be performed to implement the V portion of the pre- and post-filters.

3. Lapped-Transform Selection

We would like to find good candidate transform coefficients that perform well within a video coding framework. There are several metrics we can use for evaluating pre-filter parameters. Including

1. Coding Gain - how well energy is compacted into only a few coefficients
2. Side Band Attenuation - how much energy from frequencies outside the passband leaks into each basis function
3. Transform Width - how wide are the basis functions and how much ringing they will cause
4. Orthogonality - how linearly independent the basis functions are

Of these, the most important by far is coding gain as it allows us to directly measure the improvement in bits between different candidate transforms. At high bit rates using an efficient quantizer, every 6.02 dB improvement in coding gain saves a bit of entropy per coefficient.

3.1. Coding Gain

Coding gain is a useful metric for comparing different candidate transforms. Roughly speaking, it is the measure of how well energy is compacted into only a few coefficients. The formula for coding gain of the lapped transform can be found in [Terr12]. Using an AR(1) model with $r=0.95$, we have

$$C_g = 10 * \text{Log}_{10} \left| \frac{1}{\prod_i ((G * \text{AR}(1) * G^T)[i,i] * (H^T * H)[i,i])} \right|$$

where G is the analysis filter of the lapped transform:

$$G = \begin{bmatrix} & & & \\ & & & \\ 0 & \text{DCT} & 0 & \\ & & & \end{bmatrix} \begin{bmatrix} P & 0 \\ & \\ & \\ 0 & P \end{bmatrix}$$

and H is the synthesis filter of the lapped transform:

$$H = \begin{bmatrix} & & & \\ & & & \\ & & & \\ 0 & P^{-1} & & \end{bmatrix} \begin{bmatrix} 0 \\ \text{idCT} \\ & \\ 0 \end{bmatrix}$$

In [Terr12] the coding gain of the non-lapped DCT is compared with the optimal non-lapped Karhunen-Loeve transform for the same AR(1) model with $r=0.95$.

| | 4 point | 8 point | 16 point |
|-----|-----------|-----------|-----------|
| DCT | 7.5701 dB | 8.8259 dB | 9.4555 dB |
| KLT | 7.5825 dB | 8.8462 dB | 9.4781 dB |

Similarly, in [Tran01] the coding gain of the TDLT using fast factorizations with real coefficients produced by unconstrained optimization are

| | 4x8 | 8x16 | 16x32 |
|---------------|-----------|-----------|-----------|
| Type III TDLT | 8.6349 dB | 9.6115 dB | 9.9496 dB |
| Type IV TDLT | 8.6349 dB | 9.6005 dB | 9.9057 dB |

3.2. Transform Width

In general, the wider the transform, the higher the coding gain: a 16-point DCT will always have a higher coding gain than a 4-point DCT. In the case of lapped transform, the width of the transform is more than just counting the number of points, it involves the shape of the basis functions. At equal coding gain, a narrower transform is better because it causes a smaller amount of ringing around edges. We define the width of the transform as

$$w = C * \left| \frac{\sum_{ij} (H[i,j]^2 * (j-N+1/2)^4)}{\sum_{ij} (H[i,j]^2)} \right|^{1/4},$$

where $C=2.991$ is a constant calibrated such that the width of the 1024-point non-overlapped DCT is equal to 1024.

4. Optimal Transform Coefficients

Of the four metrics described in Section 3 we chose to optimize our transform parameters for the highest coding gain.

To avoid the use of floating point operations, we use dyadic rationals to represent the parameters of our TDLT. These are the p 's, q 's and s 's that describe the V matrix in the pre-filter. We chose a base of 2^6 because it offered enough resolution to find good approximations of the optimal values for the p 's, q 's, and s 's and still allowed us to fit the results of multiplications in a 16 bit word. Increasing the base to 2^8 improves the achievable coding gain of the 4x8 transform by less than 0.002 dB. On the other hand, dropping it even one bit to 2^5 lowers the coding gain by 0.037 dB.

4.1. Exhaustive Search

For the smaller lapped transforms, it is possible to simply do an exhaustive search and check all possible transform candidates to find the one with the best coding gain. The limitation that the p 's, q 's, and s 's all be dyadic rationals allows us to simply enumerate all reasonable values. Additional constraints allowed us to further reduce the search space. Because the p 's and q 's are liftings steps that represent rotations in the plane their values are between -1.0

and 1.0. Likewise the limitation that the pre- and post-filter steps be reversible requires that the scale factors be greater than or equal 1.0, otherwise information would be lost during the transform. Finally, all things equal we prefer smaller scale factors as it makes quantizing and encoding the coefficients cheaper. We thus cap the scale factors at 2.0. Based on some limited experimentation, scale factors larger than this do not appear to produce useful transforms according to our metrics, anyway.

With a dyadic rational base of 2^6 , the number of possible candidates to consider is

$$\begin{aligned} |C| &= (2*(2^6)+1)^{(|p|+|q|)} * (2^6+1)^{|s|} \\ &= (2*(2^6)+1)^{(2*(N/2-1))} * (2^6+1)^{(N/2)} \end{aligned}$$

Thus for the transform sizes we are interested in, the number of candidates is tractable only for the 4x8 case:

| | N | C |
|------------|----|----------------------|
| 4x8 TDLT | 4 | 68161536 |
| 8x16 TDLT | 8 | $7.731400 * 10^{19}$ |
| 16x32 TDLT | 16 | $9.947082 * 10^{43}$ |

An exhaustive search for parameters that give the optimal coding gain for the 4x8 TDLT are below:

| | | | | | |
|-------|--------|-------|-------|-------|-------|
| p_0 | -11/64 | q_0 | 36/64 | s_0 | 91/64 |
| | | | | s_1 | 85/64 |

4.2. Stochastic Search

For the larger lapped transforms, doing an exhaustive search is not possible. Instead we formulate the optimization problem as an integer programming problem and use a robust industrial solver to find optimal integer values for the p's, q's, and s's.

For the 8x16 TDLT, the parameters are below:

| | | | | | |
|-------|--------|-------|-------|-------|-------|
| p_0 | -23/64 | q_0 | 48/64 | s_0 | 90/64 |
| p_1 | -18/64 | q_1 | 34/64 | s_1 | 73/64 |
| p_2 | -6/64 | q_2 | 20/64 | s_2 | 72/64 |
| | | | | s_3 | 75/64 |

For the 16x32 TDLT, the parameters are below:

| | | | | | |
|-----|--------|-----|-------|-----|-------|
| p_0 | -24/64 | q_0 | 50/64 | s_0 | 90/64 |
| p_1 | -23/64 | q_1 | 40/64 | s_1 | 74/64 |
| p_2 | -17/64 | q_2 | 31/64 | s_2 | 73/64 |
| p_3 | -12/64 | q_3 | 22/64 | s_3 | 71/64 |
| p_4 | -14/64 | q_4 | 18/64 | s_4 | 67/64 |
| p_5 | -13/64 | q_5 | 16/64 | s_5 | 67/64 |
| p_6 | -7/64 | q_6 | 11/64 | s_6 | 67/64 |
| | | | | s_7 | 72/64 |

In order to confirm that the integer approximations found are in fact optimal, we can compare them with the optimal real valued coding gains for the three lapped-transforms we are proposing. In [Tran01] a numeric solver was used to find optimal values for a Type IV lapped transform.

| | 4x8 | 8x16 | 16x32 |
|-------------|------------|------------|------------|
| Real Valued | 8.6349 dB | 9.6005 dB | 9.9057 dB |
| Approximate | 8.63473 dB | 9.60021 dB | 9.89338 dB |
| Loss | 0.00017 dB | 0.00029 dB | 0.01232 dB |

4.3. Ramp Constraint

It is also possible to constrain the lapped transform so that it is (1,2)-regular [DT03], i.e., that it has one vanishing moment in the analysis filter and two vanishing moments in the synthesis filter. This allows the synthesis filter to reconstruct any piecewise linear function solely from the DC coefficients. This causes the shape of the DC basis function to be a symmetric linear ramp. This can be particularly useful when it matches the shape of other windowing functions used in the codec. For example, a linear window is commonly used with Overlapped Block Motion Compensation (OBMC), which is one possible approach for avoiding blocking artifacts in the motion-compensation stage of the codec. More vanishing moments are possible, allowing reconstruction of piecewise quadratic or even higher-order functions, but these require additional overlap stages.

This regularity can be enforced solely by enforcing a series of constraints on the scale factors, s_i .

$$s_0 = N \cdot (1 - q_0)$$

$$s_i = \frac{N}{2^i + 1} * \left\lfloor \frac{(q_{i-1} - 1) * p_{i-1} - q_i}{N} \right\rfloor, \text{ for } i > 0$$

Since $2^i + 1$ is odd, but we want s_i to be a dyadic rational value, the remainder of the expression must be evenly divisible by $(2^i + 1)$. A similar set of constraints can be derived for Type III, but they involve more of the p 's and q 's per s_i value, and thus have far fewer admissible solutions when coupled with the dyadic rational constraint.

The additional restrictions described above greatly reduce the number of combinations to consider, both because there are fewer parameters (the s_i 's can no longer be chosen independently) and because there are fewer combinations of parameter values which produce dyadic rational coefficients. With these constraints, the number of combinations is small enough that an exhaustive search is now tractable for the 8x16 TDLT.

| | N | C |
|-----------|---|-----------|
| 4x8 TDLT | 4 | 442 |
| 8x16 TDLT | 8 | 331677320 |

An exhaustive search for parameters that give the optimal coding gain under the ramp and dyadic rational constraints for the 4x8 and 8x16 TDLT are below:

| | | | | | | | | | | | | | | | | | | | | | | |
|--|--------|--------|--|--------|-------|---|--|-------|-------|-------|-------|-----|-------|---|-----|-------|-----|-------|-----|-------|-----|-------|
| <table><tr><td>p_0</td><td>-16/64</td></tr></table> | p_0 | -16/64 | <table><tr><td>q_0</td><td>41/64</td></tr></table> | q_0 | 41/64 | <table><tr><td>s_0</td><td>92/64</td></tr><tr><td>s_1</td><td>93/64</td></tr></table> | s_0 | 92/64 | s_1 | 93/64 | | | | | | | | | | | | |
| p_0 | -16/64 | | | | | | | | | | | | | | | | | | | | | |
| q_0 | 41/64 | | | | | | | | | | | | | | | | | | | | | |
| s_0 | 92/64 | | | | | | | | | | | | | | | | | | | | | |
| s_1 | 93/64 | | | | | | | | | | | | | | | | | | | | | |
| <table><tr><td>p_0</td><td>-24/64</td></tr><tr><td>p_1</td><td>-20/64</td></tr><tr><td>p_2</td><td>-4/64</td></tr></table> | p_0 | -24/64 | p_1 | -20/64 | p_2 | -4/64 | <table><tr><td>q_0</td><td>53/64</td></tr><tr><td>q_1</td><td>40/64</td></tr><tr><td>q_2</td><td>24/64</td></tr></table> | q_0 | 53/64 | q_1 | 40/64 | q_2 | 24/64 | <table><tr><td>s_0</td><td>88/64</td></tr><tr><td>s_1</td><td>75/64</td></tr><tr><td>s_2</td><td>76/64</td></tr><tr><td>s_3</td><td>76/64</td></tr></table> | s_0 | 88/64 | s_1 | 75/64 | s_2 | 76/64 | s_3 | 76/64 |
| p_0 | -24/64 | | | | | | | | | | | | | | | | | | | | | |
| p_1 | -20/64 | | | | | | | | | | | | | | | | | | | | | |
| p_2 | -4/64 | | | | | | | | | | | | | | | | | | | | | |
| q_0 | 53/64 | | | | | | | | | | | | | | | | | | | | | |
| q_1 | 40/64 | | | | | | | | | | | | | | | | | | | | | |
| q_2 | 24/64 | | | | | | | | | | | | | | | | | | | | | |
| s_0 | 88/64 | | | | | | | | | | | | | | | | | | | | | |
| s_1 | 75/64 | | | | | | | | | | | | | | | | | | | | | |
| s_2 | 76/64 | | | | | | | | | | | | | | | | | | | | | |
| s_3 | 76/64 | | | | | | | | | | | | | | | | | | | | | |

Unfortunately, in the 16x32 TDLT case the number of combinations is still not tractable, even with these additional constraints. Again, we use an integer programming model to solve for the integer parameters that optimize coding gain in this context.

| | | | | | |
|-----|--------|-----|-------|-----|-------|
| p_0 | -32/64 | q_0 | 59/64 | s_0 | 80/64 |
| p_1 | -28/64 | q_1 | 53/64 | s_1 | 72/64 |
| p_2 | -24/64 | q_2 | 46/64 | s_2 | 73/64 |
| p_3 | -32/64 | q_3 | 41/64 | s_3 | 68/64 |
| p_4 | -24/64 | q_4 | 35/64 | s_4 | 72/64 |
| p_5 | -13/64 | q_5 | 24/64 | s_5 | 74/64 |
| p_6 | -2/64 | q_6 | 12/64 | s_6 | 74/64 |
| | | | | s_7 | 70/64 |

| | 4x8 | 8x16 | 16x32 |
|---------------|------------|------------|------------|
| Dyadic | 8.63473 dB | 9.60021 dB | 9.89338 dB |
| Ramp + Dyadic | 8.59886 dB | 9.56161 dB | 9.78294 dB |
| Loss | 0.03587 dB | 0.0386 dB | 0.11044 dB |

5. Intra Prediction

Since the final pixel values of a block are not available until after the post-filter runs, they cannot be used to predict neighboring blocks. There are a number of possible solutions to this. For example, one could simply use pixels from outside the overlap region. However, as these pixels are farther away, they are poorer predictors, and the extra distance reduces the range of prediction directions which have enough neighbors available to form an adequate extrapolation [OP11].

An alternate approach is to perform the prediction in the frequency domain. Initial experiments suggest that this is just as effective as prediction in the time domain, and has similar computational requirements [Egge13]. However, because the frequency domain coefficients of a neighboring block are impacted both by what size DCT was used, and the lapping across all four of its edges, directional predictors can only be so good. At low rates, this meant more bits were spent correcting an incorrect predictor than were saved by coding only a directional mode.

A signal free technique was developed for doing limited intra prediction in the frequency domain when using lapped transforms. Note that when the spatial prediction mode is exactly horizontal or vertical, applying the filters described in this draft along the orthogonal direction is the identity. Thus it is possible to look at the horizontal coefficients of the neighboring block to the left, and the vertical energy of the neighboring block above and simply use the coefficients where the energy is larger. When this technique is

coupled with a quantization and coefficient coder that makes signaling no predictor cheap [Vali15], this becomes an effective frequency domain intra predictor.

Finally, a technique was developed for intra predicting chroma frequency domain coefficients from decoded coincident luma coefficients [Egge15]. While this technique does not strictly require the use of lapped transforms, because the block size extent (and thus the lapping region) for both the chroma and luma planes is the same, the use of a lapped transform does not change the effectiveness of this technique.

6. Motion Compensation

There have been several lapped transform proposals that perform block-by-block motion compensation by simply expanding the size of the prediction region for each block [TT01], [OPT11]. However, in addition to increasing the amount of motion-compensated prediction pixels that must be computed by a factor of four, this also increases the number of applications of the pre- and post-filter by a factor of four, since this must now be done separately for each block, using the motion-compensated frame difference for that block.

An alternate approach is simply perform motion compensation of the frame in a completely separate step, prior to any transform, using any method desired [Terr15]. The lapping can then be applied to this motion-compensated prediction, producing per-block predictors. This still allows the prediction mode (inter, intra, bi-prediction, etc.) to be chosen on a block-by-block basis. It also interacts well with other techniques designed to operate in the frequency domain, such as the Pyramid Vector Quantization (PVQ) proposed elsewhere.

The downside is that motion estimation in the encoder needs to be performed for regions slightly beyond the current block. However, this is already required by blocking-artifact-free motion compensation techniques, such as Overlapped Block Motion Compensation (OBMC). Experience with OBMC has shown that an encoder can mostly ignore look-ahead and still get acceptable results, unlike other techniques, such as control-grid interpolation (CGI).

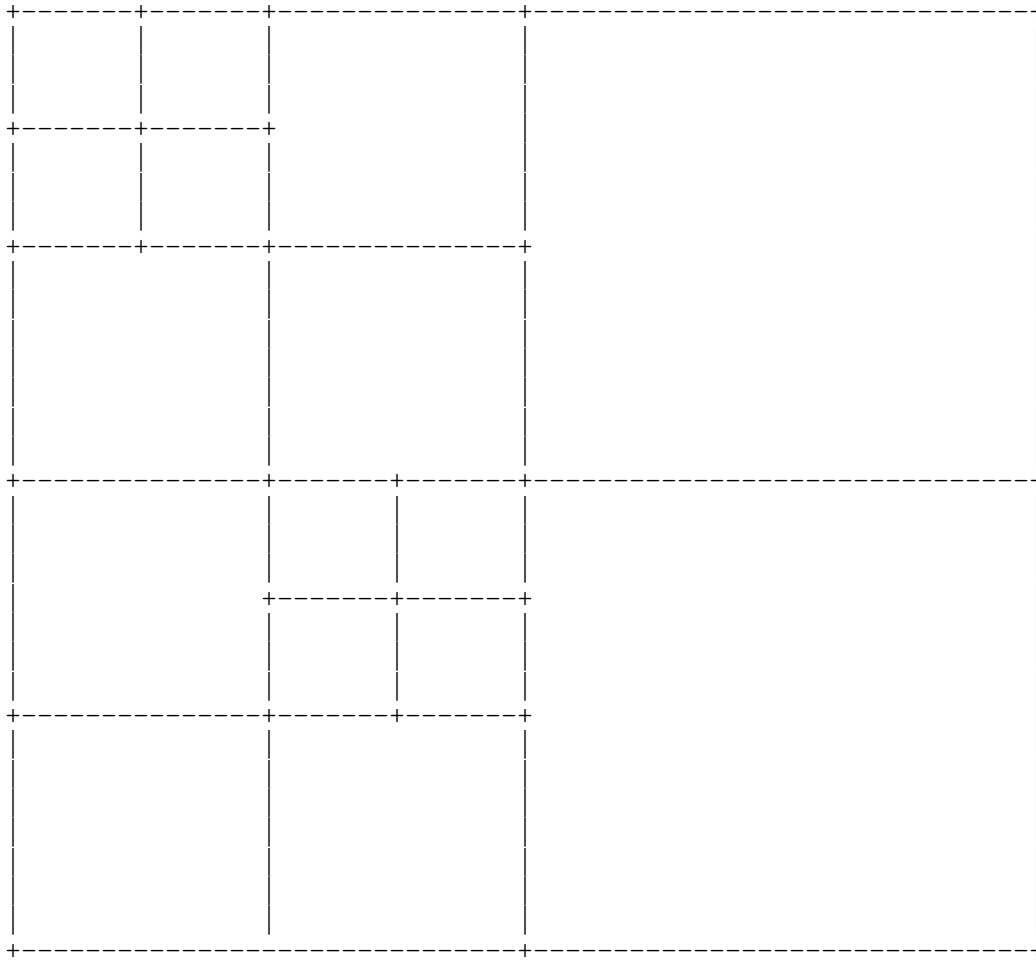
7. Multiple Block Sizes

Multiple block size support is important for lapped transforms, since the larger support region increases their susceptibility to ringing artifacts compared to a non-overlapped transform with the same number of coefficients (though it is greatly reduced compared to a non-overlapped transform with a support region of the same size).

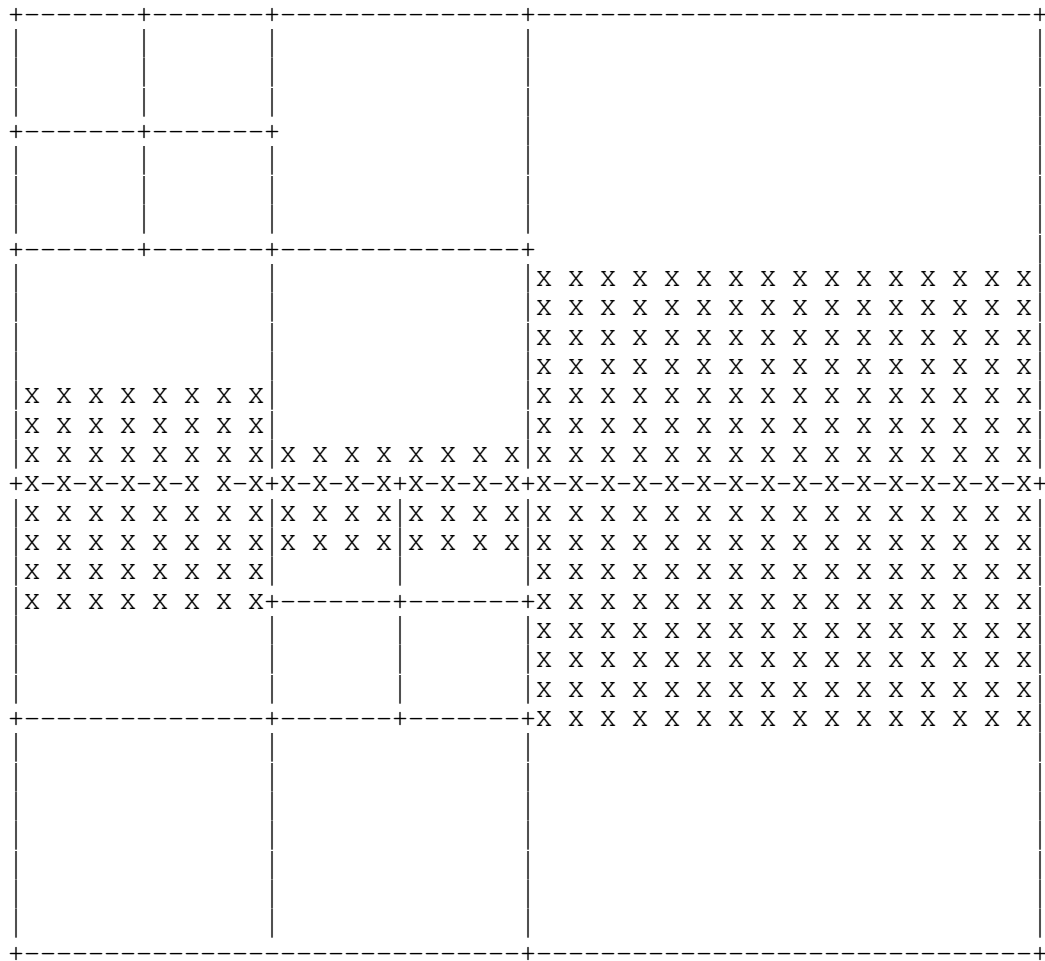
7.1. Variable Sized Lapping

The most obvious approach is to require that the size of the overlap filter be constrained by the smallest block adjacent to a given edge. This requires some amount of look-ahead in the encoder, but has the benefit of using the largest lapping possible in regions where all blocks are the same size while not introducing discontinuities where blocks of different sizes meet. Note that this has an effect on the coding syntax, as the block size decision for the block below the one being coded must be made and communicated to the decoder prior to coding. Using this convention no additional information need to be communicated other than the block size decision to completely describe how the variable sized lapping should be applied.

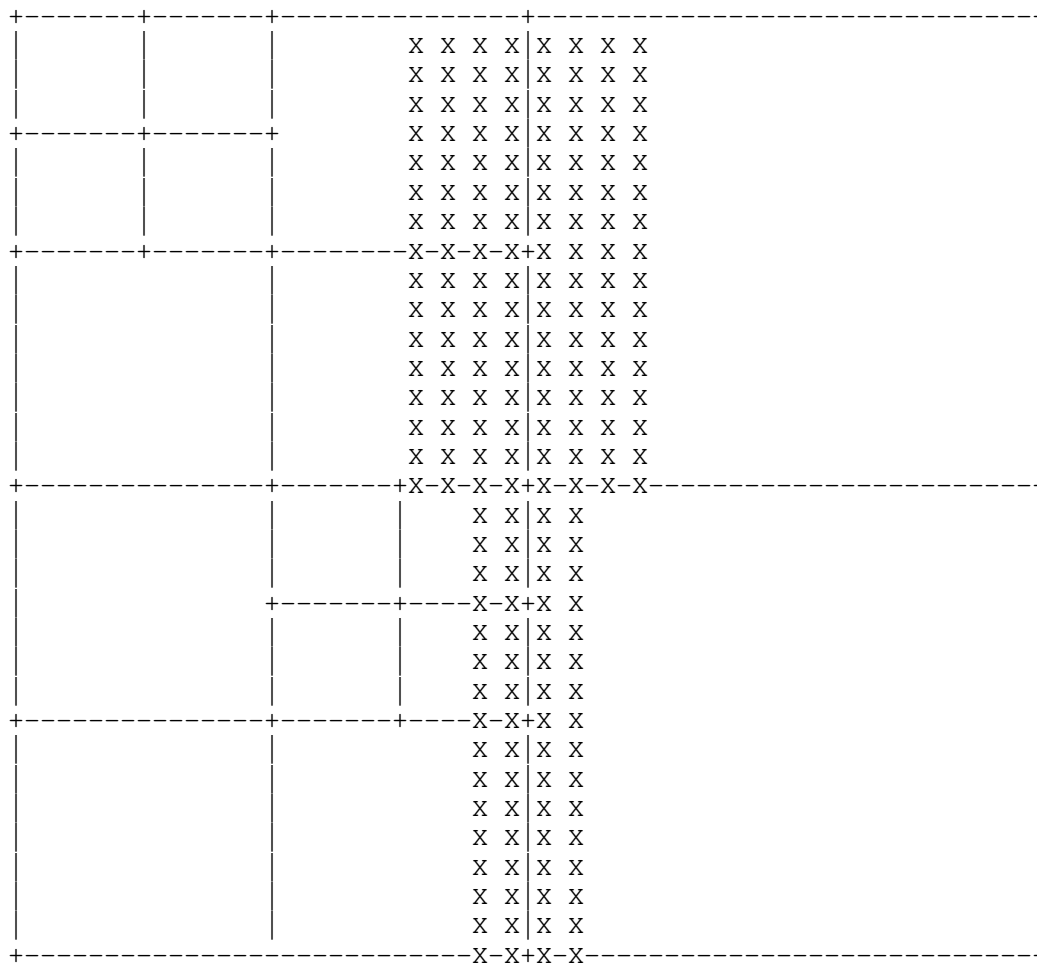
Consider an example image that is 32x32 with the following block size decisions. We apply the lapping recursively to blocks of 32x32 at a time until we reach a block that is not subdivided into smaller blocks. At each step in the recursions, we apply a filter vertically across the block edges that run left to right splitting the block in half. We then apply a horizontal filter across the block edges that split the block in half top to bottom.



Block size decision for a 32x32 frame.



Apply the filter vertically across the horizontal internal edge.



Apply the filter horizontally across the vertical internal edge.

The filters are then applied recursively in this manner to the four quadrants of the block. By applying the filters recursively this way, we have prevented any discontinuities from appearing where block is split but its neighbor is not.

7.2. Fixed Sized Lapping

One of the challenges using variable sized lapping is that changing the block size decision (either splitting a block into four blocks a quarter as big, or merging four blocks into one four times the size) can have an impact on the coding performance outside the block considered. This makes computing the optimal block size decision for

a frame computationally difficult as traditional rate-distortion optimization (RDO) algorithms exploit this locality to iteratively improve an initial decision.

One way to simplify the problem is to assume a fixed sized lapping across the entire image. If only the 4-point filter is used across block boundaries, then it is possible to compare the distortion of an 8x8 block with that of four 4x4 blocks by simply computing the mean squared error (MSE) of the 64 spatial domain coefficients after applying the inverse lapped transform. Because changing the block size decision, and thus the interior lapping has no impact on the lap decision on the border of the 8x8 block, then just looking at the rate and distortion of the interior coefficients is sufficient.

This approach has does not leverage the additional coding gain and deblocking achieved by using larger lapping filters but may make up for this by allowing computationally cheap block size decision heuristics in real-time encoding environments.

8. IANA Considerations

This document has no actions for IANA.

9. Security Considerations

This draft has no security considerations.

10. Acknowledgments

Thanks to Greg Maxwell and Jean-Marc Valin for their assistance in the experimentation and other valuable contributions to this document.

11. Informative References

- [DT03] Dai, W. and T. Tran, "Regularity-Constrained Pre- and Post-Filtering for Block DCT Based Systems", IEEE Transactions on Signal Processing 51(10):2568--2581, October 2003.
- [Egge13] Egge, N., "Intra-Prediction in Daala", October 2013, <<http://people.xiph.org/~unlord/Daala-Intra.pdf>>.
- [Egge15] Egge, N. and J. Valin, "Predicting Chroma from Luma with Frequency Domain Intra Prediction", February 2015, <http://people.xiph.org/~unlord/spie_cfl.pdf>.

- [Malv89] Malvar, H. and D. Staelin, "The LOT: Transform Coding Without Blocking Effects", IEEE Transactions on Acoustics, Speech, and Signal Processing , April 1989, <<http://research.microsoft.com/apps/pubs/default.aspx?id=102073>>.
- [OP11] de Oliveria, R. and B. Pesquet-Popescu, "Intra-Frame Prediction with Lapped Transforms for Image Coding", Proc. of the 36th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'11) pp. 805--808, May 2011.
- [OPT11] de Oliveria, R., Pesquet-Popescu, B., and M. Trocan, "Inter Prediction Using Lapped Transforms for Advanced Video Coding", Proc. of the 18th IEEE International Conference on Image Processing (ICIP'11) pp. 3705--3708, September 2011.
- [Tran01] Tran, T., "Lapped Transform via Time-Domain Pre- and Post-Processing", IEEE Transactions on Signal Processing , October 2001.
- [TT01] Tran, T. and C. Tu, "Lapped Transform Based Video Coding", Proc. of the 24th SPIE Conference on Applications of Digital Image Processing vol. 4472, pp. 319--333, July 2001.
- [Terr12] Terriberry, T., "Introduction to Video Coding Part 1: Transform Coding", February 2012.
- [Terr15] Terriberry, T., "Adaptive Motion Compensation Without Blocking Artifacts", February 2015, <<https://people.xiph.org/~tterribe/daala/vbsobmc.pdf>>.
- [Vali15] Valin, J., "Perceptual Vector Quantization for Video Coding", February 2015, <http://jmvalin.ca/video/spie_pvq.pdf>.

Authors' Addresses

Nathan E. Egge
Mozilla Corporation
650 Castro Street
Mountain View, CA 94041
USA

Email: negge@dgql.org

Timothy B. Terriberry
Mozilla Corporation
650 Castro Street
Mountain View, CA 94041
USA

Phone: +1 650 903-0800
Email: tterribe@xiph.org

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 18, 2013

GM. Maxwell
Xiph.Org
K. Walsh
October 15, 2012

Requirements for an Internet Video Codec
draft-maxwell-videocodec-requirements-00

Abstract

This document provides specific requirements for an Internet video codec. These requirements address quality, bit-rate, loss robustness, application suitability, as well as other desirable properties.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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. Definitions | 4 |
| 3. Applications | 5 |
| 3.1. Point to point video calls | 5 |
| 3.2. Video Conferencing | 5 |
| 3.3. Telepresence | 5 |
| 3.4. Teleoperation and Remote Software Services | 5 |
| 3.5. Other applications | 6 |
| 4. Constraints Imposed by the Internet on the Codec | 7 |
| 5. Detailed Basic Requirements | 9 |
| 5.1. Quality and bit-rate | 9 |
| 5.2. Computational resources | 9 |
| 6. Additional considerations | 10 |
| 7. Encoder side potential for improvement | 11 |
| 8. Bit error robustness | 12 |
| 9. Legacy compatibility | 13 |
| 10. Security Considerations | 14 |
| 11. IANA Considerations | 15 |
| 12. Informative References | 16 |
| Authors' Addresses | 17 |

1. Introduction

This document provides requirements for a video codec designed specifically for use over the Internet. The requirements attempt to address the needs of the most common Internet video transmission applications and to ensure good quality when operating in conditions that are typical for the Internet. These requirements address quality, bit-rate, and loss robustness. Other desirable codec properties are considered as well.

2. Definitions

Codec bit-rates in bits per second (b/s) will be considered without counting any overhead (IP/UDP/RTP headers, padding, ...).

3. Applications

The following applications should be considered for Internet video codecs, along with their requirements:

- o Live video streaming
- o Video on demand
- o Point to point video calls
- o Video Conferencing
- o Telepresence
- o Teleoperation
- o Remote software services
- o Other applications

3.1. Point to point video calls

Point to point calls are calls from two "standard" (fixed or mobile) phones, desktop, portable computers, or tablets, and implemented in hardware or software.

3.2. Video Conferencing

Video Conferencing applications (which support multi-party calls) have additional requirements on top of the requirements for point-to-point calls. Conferencing systems often have greater network bandwidth available. The ability to vary the bit-rate (VBR) is a desirable feature for the codec. This not only saves bandwidth "on average", but it can also help conference servers make more efficient use of the available bandwidth by using more bandwidth for important video streams and less bandwidth for less important ones.

3.3. Telepresence

Most telepresence applications can be considered to be essentially very high quality video conferencing environments, so all of the conferencing requirements also apply to telepresence.

3.4. Teleoperation and Remote Software Services

Teleoperation applications are similar to telepresence, with the exception that they involve remote physical interactions. For

example, the user may be controlling a robot while receiving a real-time video feed from that robot. The other requirements of telepresence apply to teleoperation as well.

The requirements for remote software services are similar to those of teleoperation. These applications include remote desktop applications, remote virtualization, and interactive media applications being rendered remotely (e.g. video games rendered on central servers).

3.5. Other applications

The above list is by no means a complete list of all applications involving interactive video transmission on the Internet. However, it is believed that meeting the needs of all these different applications should be sufficient to ensure that most applications not listed will also be met.

4. Constraints Imposed by the Internet on the Codec

The bandwidth requirements of video are a significant obstacle for Internet deployment. A substantial portion of the hosts on the Internet have connectivity sufficient to carry perceptually lossless audio, even in inefficient uncompressed form. However, a much smaller portion of hosts have connectivity sufficient for the 200+ megabits per second required for uncompressed 30fps standard definition video, and expected resolutions for Internet video are increasing. Even the highest resolutions widely used off the Internet, where wide-area bandwidth is not a constraint, have not yet reached perceptual losslessness. In addition to increases in resolution, operating models which are less broadcast-oriented (including video on demand and video conferencing) limit the traffic mitigation effectiveness of CDNs and multicast, though support for these technologies remains essential. Because there are a few applications where increases in bandwidth efficiency are not important and many where improved efficiency is essential--such as delivering HD video to bandwidth-constrained network edges--it is important that the codec deliver competitive quality per bitrate and support a wide range of bandwidths.

Packet losses are inevitable on the Internet and dealing with them is one of requirements for an Internet video codec. Efficient video compression typically uses very high gain backward prediction, which can result in infinite error propagation in the worst case if measures are not taken to mitigate it. Error propagation is usually mitigated in traditional file- and broadcast-oriented codecs through key-frames, periodic intra refresh, and constrained back-reference structure. While these techniques are important, and also enable random access, a codec designed for the Internet should also be able to take advantage of bidirectional communication to reduce the impact of loss when possible.

In many high-latency and non-realtime applications, however, the relevant transport is lossless. While random access still is important, general error tolerance is not, and the codec may support modes which have very low error tolerance--including ones which prevent packet decode in the presence of loss, if it results in efficiency gains for non-realtime applications.

For interactive applications latency is an important codec performance metric and many common input and output devices add frames of latency. To avoid adding further delays the codec must support operating in a mode that adds no more delay than that from processing a single frame at a time. Modes which permit sub-frame encoding may be useful but are hampered by the lack of subframe support in existing input and output devices.

Another important property of the Internet is that it is mostly a best-effort network, with no guaranteed bandwidth. This means that the codec has to be able to vary its output bit-rate dynamically (in real-time), without requiring an out-of-band signaling mechanism, and without causing artifacts at the bit-rate change boundaries. Because the complete range of useful bit-rates may not be achievable at a single resolution the codec may need to support changing resolutions on the fly. Additional desirable features are:

- o Having the possibility to use smooth bit-rate changes with high bit-rate resolution;
- o Making it possible for a codec to adapt its bit-rate based on the source signal being encoded (source-controlled VBR) to maximize the quality for a certain `_average_` bit-rate.

Because the Internet transmits data in bytes, a codec should produce compressed data in integer numbers of bytes. In general, the codec design should take into consideration explicit congestion notification (ECN) and multicast and may include features that would improve the quality of an ECN or multicast enabled deployment.

The IETF has defined a set of application-layer protocols to be used for transmitting real-time transport of multimedia data, including video. It is thus important for the resulting codec to be easy to use with these protocols. For example, it must be possible to create an [RTP] payload format that conforms to BCP 36 [PAYLOADS]. If any codec parameters need to be negotiated between end-points, the negotiation should be as easy as possible to carry over SIP [RFC3261]/SDP [RFC4566] or alternatively over XMPP [RFC6120]/Jingle [XEP-0167].

5. Detailed Basic Requirements

This section summarizes all the constraints imposed by the target applications and by the Internet into a set of actual requirements for codec development.

5.1. Quality and bit-rate

The quality of a codec is directly linked to the bit-rate, so these two must be considered jointly. When comparing the bit-rate of codecs, the overhead of IP/UDP/RTP headers should not be considered, but any additional bits required in the RTP payload format after the header (e.g. required signaling) should be considered. In terms of quality vs bit-rate, the codec to be developed must be better than the following codecs, that are generally considered as royalty-free:

- o VP8
- o Theora

It is desirable for the codecs to support source-controlled variable bit-rate (VBR) to take advantage from the fact that different inputs require a different bitrate to achieve the same quality.

5.2. Computational resources

The resulting codec should be implementable on a wide range of devices, and should not have a design which gratuitously complicates low power ASIC implementations. While the codec must not depend on special hardware features or instructions, the codec design should allow implementations to take full advantage of hardware accelerators and vector instructions where available. Complexity should generally scale with resolution, and it is also desirable to support multiple encoder and decoder complexity levels via mechanisms other than resolution, in order to achieve the best possible bitrate/quality trade-off available across many kinds of devices without unduly constraining resolution. The codec should also be able to take advantages of advances in computer speed and the deployment of hardware accelerators which would allow the use of higher complexity modes in a broader set of applications.

In addition to computational complexity, dynamic memory for reference storage is a significant resource constraint for video codecs. It is desirable that the codec support different memory usage tradeoffs to fit on more devices, and that the codec not require implementations to utilize more memory without reasonable efficiency gains.

6. Additional considerations

There are additional features or characteristics that may be desirable under some circumstances, but should not be part of the strict requirements. The benefit of meeting these considerations should be weighted against the associated cost.

7. Encoder side potential for improvement

In most video codecs, it is possible to improve the quality by improving the encoder without breaking compatibility (i.e. without changing the decoder). Potential for improvement varies from one codec to another. All things being equal, being able to improve a codec after the bit-stream is a desirable property. However, this should not be done at the expense of quality in a straight-forward encoder.

8. Bit error robustness

The vast majority of Internet-based applications do not need to be robust to bit errors because packets either arrive unaltered, or do not arrive at all. Considering that, the emphasis should be on packet loss robustness and packet loss concealment. That being said, it is often the case that extra robustness to bit errors can be achieved at no cost at all (i.e. no increase in size, complexity or bit-rate, no decrease in quality or packet loss robustness, ...). In those cases then it is useful to make a change that increases the robustness to bit errors. This can be useful for applications that use UDP Lite transmission (e.g. over a wireless LAN). Robustness to packet loss should **never** be sacrificed to achieve higher bit error robustness.

9. Legacy compatibility

In order to create the best possible codec for the Internet, there is no general requirement for compatibility with legacy Internet codecs. However, compatibility with commonly used video color formats is desirable.

10. Security Considerations

Although this document itself does not have security considerations, this section describes the security requirements for the codec.

Just like for any protocol to be used over the Internet, security is a very important aspect to consider. This goes beyond the obvious considerations of preventing buffer overflows and similar attacks that can lead to denial-of-service or remote code execution. One very important security aspect is to make sure that the decoders have a bounded and reasonable worst case complexity. This prevents an attacker from causing a DoS by sending packets that are specially crafted to take a very long (or infinite) time to decode.

11. IANA Considerations

This document has no actions for IANA.

12. Informative References

- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011.
- [XEP-0167] Ludwig, S., Saint-Andre, P., Egan, S., McQueen, R., and D. Cionoiu, "Jingle RTP Sessions", XSF XEP 0167, December 2009.
- [PAYLOADS] Handley, M. and C. Perkins, "Guidelines for Writers of RTP Payload Format Specifications", RFC 2736, BCP 36.
- [RTP] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for real-time applications", RFC 3550.

Authors' Addresses

Gregory Maxwell
Xiph.Org

Email: greg@xiph.org

Kat Walsh

Email: kat@mindspillage.org

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2015

JM. Valin
Mozilla
March 9, 2015

Pyramid Vector Quantization for Video Coding
draft-valin-videocodec-pvq-02

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.

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 September 10, 2015.

Copyright Notice

Copyright (c) 2015 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.

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.

Table of Contents

| | |
|---|---|
| 1. Introduction | 2 |
| 2. Gain-Shape Coding and Activity Masking | 2 |
| 3. Householder Reflection | 3 |
| 4. Angle-Based Encoding | 4 |
| 5. Bi-prediction | 5 |
| 6. Coefficient coding | 6 |
| 7. Development Repository | 6 |
| 8. IANA Considerations | 6 |
| 9. Security Considerations | 6 |
| 10. Acknowledgements | 6 |
| 11. References | 7 |
| 11.1. Informative References | 7 |
| 11.2. URIs | 7 |
| Author's Address | 7 |

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) [Pyramid-VQ]. 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. This proposal is described in greater details in [Perceptual-VQ], as well as in demo [PVQ-demo].

2. 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:

$$g = Q_g \gamma^{\frac{b}{2a}},$$

where γ is the gain quantization index, $b=1/(1-2a)$ 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.

3. 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

$$x = \frac{x_d}{\|x_d\|},$$

with the normalized reference vector r similarly computed based on r_d . The encoder then finds the position and sign of the largest component in vector r :

$$\begin{aligned} m &= \operatorname{argmax}_i |r_i| \\ s &= \operatorname{sign}(r_m) \end{aligned}$$

and computes the Householder reflection that reflects r to $-s e_m$, where e_m is a unit vector that points in the direction of dimension m . The reflection vector is given by

$$v = r + s e_m.$$

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

$$z = x - 2 \frac{v^T x}{v^T v} v .$$

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.

4. Angle-Based Encoding

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

$$\theta = \arccos(-s z_m) .$$

If θ 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)

$$Q_{\theta} = \frac{dg}{d(\gamma)} \frac{1}{g} = \frac{b}{\gamma} .$$

To derive the optimal K we need to consider the normalized distortion for a Laplace-distributed variable found experimentally to be approximately

$$D_p = \frac{(N-1)^2 + C(N-1)}{24K^2},$$

with $C \approx 4.2$. The distortion due to the gain is

$$D_g = \frac{b^2 Q_g^2 \gamma^{(2b-2)}}{12}.$$

Since PVQ codes $N-2$ degrees of freedom, its distortion should also be $(N-2)$ times the gain distortion, which eventually leads us to the optimal number of pulses

$$K = \frac{\gamma \sin(\theta)}{b} \sqrt{\frac{N + C - 2}{2}}.$$

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_θ 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 K depends only on the θ quantization index, with no dependency on the gain. Alternatively, instead of quantizing θ , 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 $\sin(\theta) \approx f(\theta)$. 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.

5. Bi-prediction

We can use this scheme for bi-prediction by introducing a second θ 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 θ_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 θ_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.

6. Coefficient coding

Encoding coefficients quantized with PVQ differs from encoding scalar-quantized coefficients from the fact that the sum of the coefficients magnitude is known (equal to K). It is possible to take advantage of the known K value either through modeling the distribution of coefficient magnitude or by modeling the zero runs. In the case of magnitude modeling, the expectation of the magnitude of coefficient n is modeled as

$$E(|y_n|) = \alpha * \frac{K_n}{N - n},$$

where K_n is the number of pulses left after encoding coefficients from 0 to n-1 and alpha depends on the distribution of the coefficients. For run-length modeling, the expectation of the position of the next non-zero coefficient is given by

$$E(|run|) = \beta * \frac{N - n}{K_n},$$

where beta also models the coefficient distribution.

7. 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 [1]. See [2] for more information.

8. IANA Considerations

This document makes no request of IANA.

9. Security Considerations

This draft has no security considerations.

10. Acknowledgements

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

11. References

11.1. Informative References

[PVQ-demo]

Valin, JM., "Daala: Perceptual Vector Quantization (PVQ)", November 2014, <https://people.xiph.org/~jm/daala/pvq_demo/>.

[Perceptual-VQ]

Valin, JM. and TB. Terriberry, "Perceptual Vector Quantization for Video Coding", Proceedings of SPIE Visual Information Processing and Communication, February 2015, <http://jmvalin.ca/papers/spie_pvq.pdf>.

[Pyramid-VQ]

Fischer, T., "A Pyramid Vector Quantizer", IEEE Trans. on Information Theory, Vol. 32 pp. 568-583, July 1986.

[RFC6716] Valin, JM., Vos, K., and T. Terriberry, "Definition of the Opus Audio Codec", RFC 6716, September 2012.

11.2. URIs

[1] <https://git.xiph.org/daala.git>

[2] <https://xiph.org/daala/>

Author's Address

Jean-Marc Valin
Mozilla
331 E. Evelyn Avenue
Mountain View, CA 94041
USA

Email: jmvalin@jmvalin.ca