Network Working Group   JM. Valin
Internet-Draft   Mozilla
Intended status: Standards Track   July 6, 2015
Expires: January 7, 2016

              Screencasting Considerations and L1-Tree Wavelet Coding
                          draft-valin-netvc-l1tw-01

Abstract

   This document proposes a screencasting encoding mode based on the
   Haar wavelet transform and L1-tree wavelet (L1TW) coding.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   Screensharing is an important application for an Internet video
   codec. Screensharing content differs from photographic images in
   many ways, including:

   o Text: screenshots often contain anti-aliased text on a perfectly
      flat background. This makes ringing artefacts highly perceptible.
      Also, typical photographic codecs based on the discrete cosine
      transform (DCT) cannot take advantage of the fact that the
      background often has a constant colour.

   o Lines and edges. Screenshots often contain perfectly straight
      horizontal and/or vertical lines. They appear in window frames,
      toolbars, widgets, spreadsheets, etc. DCT-based codecs can
      represent those lines and edges, but not as compactly as codecs
      like PNG.

   o Reduced number of colours: Screenshots are much less "noisy" than
      photographic images. It is common for a certain region of an
      image to only contain a handful of different colours, another
      property we would like to exploit in a video codec.

   o A very common motion pattern in screensharing content is the
      displacement of windows. This typically involves rectangular
      boundaries.

   The technique described in this document only deals with still images
   for now and focuses on the problem of efficiently coding anti-aliased
   text. While it is implemented for the Daala [Daala-website] codec,
   it should be applicable to most other video codecs.

2. The Haar Wavelet

   The Haar wavelet <https://en.wikipedia.org/wiki/Haar_wavelet> is the
   simplest of all orthogonal wavelets, and also the only one with
   linear phase. We use the Haar transform both because it is spatially
   compact and because it makes it easy to switch between a wavelet
   transform and the DCT.

   In 1-D, a single level of the Haar transform is expressed as:


```
                      ___
     [ y0 ]      / 1 [  1 1 ] [ x0 ]
     [    ] = / --- [       ] [    ]
     [ y1 ]   v   2 [ -1 1 ] [ x1 ]
```

   The 2-D Haar transform is implemented from a 2x2 lifting Haar kernel:

```
                  inputs: x0, x1, x2, x3
                  x0 <= x0 + x2
                  x3 <= x3 - x1
                  tmp <= (x0 - x3) >> 1
                  x1 <= tmp - x1
                  x2 <= tmp - x2
                  x0 <= x0 - x1
                  x3 <= x3 + x2
                  outputs: x0, x1, x2, x3
```

   This kernel has perfect reconstruction, making it also useful for
   lossless compression.

   The kernel above is applied on 5 levels for 32x32 superblocks. The
   resulting wavelet coefficients are quantized non-uniformly using the
   following quantization scales relative to the DC quantizer (from low
   frequency to high frequency):

```
          horizontal/vertical: [1.0, 1.0, 1.0, 1.5, 2.0]
          diagonal:            [1.0, 1.0, 1.5, 2.0, 3.0]
```

3. L1-Tree Coding

   Like other wavelet coding methods such as EZW and SPIHT, we code the
   wavelet coefficients using trees. The main difference however is
   that rather than being based on the maximum coefficient value in a
   tree, this technique is based on the sum of the absolute values of
   all coefficients in the tree. Let $x(i,j)$ denote the quantized
   wavelet coefficient at position (i,j), the children of $x(i,j)$ are
   $x(2*i,2*j)$, $x(2*i,2*j+1)$, $x(2*i+1,2*j)$, and $x(2*i+1,2*j+1)$. The
   absolute sum of the tree rooted in (i,j) is defined recursively as:

$$S(i,j) = |x(i,j)| + S(2*i,2*j) + S(2*i,2*j+1)$$
$$+ S(2*i+1,2*j) + S(2*i+1,2*j+1),$$

with $S(i,j)=0$ for i or $j >= N$. $C(i,j)$ is defined as $S(i,j) - |x(i,j)|$.

Coefficient coding starts at the root of each of the three "direction trees": $(1,0)$, $(0,1)$, and $(1,1)$. At each level we code the value of $|x(i,j)|$ using a cumulative density function adapted based on the value of $S(i,j)$. Coding $|x(i,j)|$ implies that the value of $C(i,j)$ is known to the decoder, so it does not need to be coded. Three symbols are then required to encode each of the new roots: $S(2*i,2*j)$, $S(2*i,2*j+1)$, $S(2*i+1,2*j)$, and $S(2*i+1,2*j+1)$.

At the top level, we have $S(0,0) = S(1,0) + S(0,1) + S(1,1)$, so that completely flat blocks can be coded with a single $S(0,0)=0$ symbol. The DC is coded separately.

4. Results

The coded images obtained with the Haar transform and L1TW have far better subjective visual quality than those obtained with the lapped DCT or with JPEG, and of comparable quality to those obtained with x264 <http://www.videolan.org/developers/x264.html> and x265 <http://x265.org/>. An example image at around 0.35 bit/pixel is provided at <http://jmvalin.ca/video/haar_example/>. The x264 image encoded with options "--preset placebo --crf=27" and the x265 image is encoded with "--preset slow --crf=29".

While the technique presented here works relatively well on the example above, there are still cases where it performs significantly worse than x265. These include gradients, such as those in toolbars and window titlebars, and long horizontal and vertical lines such as those found in spreadsheets. These cases should improve once we implement the ability to dynamically switch between the lapped DCT and the Haar transform. Other ways of improving performance on long lines and edges would be to extend to use a different 2D wavelet decomposition, or use an overcomplete basis.

5. Objective Evaluation

As a first step for evaluating screensharing quality, we have added a small collection of screenshot images to the "Are We Compressed Yet?" (AWCY) <https://arewecompressedyet.com/> website, under the "screenshots" set name. AWCY currently runs four quality metrics: PSNR, PSNR-HVS, SSIM, and FAST-SSIM [I-D.daede-netvc-testing]. It is not yet clear that and of these metrics is suitable for evaluating the quality of screensharing material.

6. 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 [Daala-website]
   for more information.

7. IANA Considerations

   This document makes no request of IANA.

8. Security Considerations

   This draft has no security considerations.

9. Acknowledgements

   Thanks to Timothy B. Terriberry for useful feedback and for
   designing the 2-D Haar lifting kernel.

10. Informative References

   [Daala-website]
           "Daala website", Xiph.Org Foundation , <https://xiph.org/
           daala/>.

   [I-D.daede-netvc-testing]
           Daede, T. and J. Jack, "Video Codec Testing and Quality
           Measurement", draft-daede-netvc-testing-00 (work in
           progress), March 2015.

Author's Address

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

   Email: jmvalin@jmvalin.ca