Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2016

JM. Valin
Mozilla

October 19, 2015

Directional Deringing Filter
draft-valin-netvc-deringing-00

Abstract

This document describes a deringing filter that takes into account the direction of edges and patterns being filtered. The filter works by identifying the direction of each block and then adaptively filtering along the identified direction. In a second pass, the blocks are also filtered in a different direction, with more conservative thresholds to avoid blurring edges.

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 21, 2016.

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This document may not be modified, and derivative works of it may not be created, and it may not be published except as an Internet-Draft.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. Direction Search 2
3. Directional Filter 3
4. Second Stage Filter 4
5. Setting Thresholds 5
6. Superblock Filtering 6
7. Results 6
8. Conclusion 6
9. IANA Considerations 6
10. Security Considerations 7
11. Informative References 7
Author's Address 7

1. Introduction

This document describes a deringing filter that takes into account the direction of edges and patterns being filtered. The filter works by identifying the direction of each block and then adaptively filtering along the identified direction. In a second pass, the blocks are also filtered in a different direction, with more conservative thresholds to avoid blurring edges.

2. Direction Search

The first step is to divide the image into blocks of fixed or variable size. Variable-size blocks make it possible to use large blocks on long, continuous edges and small blocks where edges intersect or change direction. A fixed block size is easier to implement and does not require signaling the sizes on a block-by-block basis. For this work, we consider a fixed block size of 8x8.

Once the image is divided into blocks, we determine which direction best matches the pattern in each block. One way to determine the direction is to minimize mean squared difference (MSD) between the input block and a perfectly directional block. A perfectly directional block is a block for which each line along a certain direction has a constant value. For each direction, we assign a line number to each pixel, as shown below.

```
                    +---+---+---+---+---+---+---+---+
                    | 0 | 0 | 1 | 1 | 2 | 2 | 3 | 3 |
                    +---+---+---+---+---+---+---+---+
                    | 1 | 1 | 2 | 2 | 3 | 3 | 4 | 4 |
                    +---+---+---+---+---+---+---+---+
                    | 2 | 2 | 3 | 3 | 4 | 4 | 5 | 5 |
                    +---+---+---+---+---+---+---+---+
                    | 3 | 3 | 4 | 4 | 5 | 5 | 6 | 6 |
                    +---+---+---+---+---+---+---+---+
                    | 4 | 4 | 5 | 5 | 6 | 6 | 7 | 7 |
                    +---+---+---+---+---+---+---+---+
                    | 5 | 5 | 6 | 6 | 7 | 7 | 8 | 8 |
                    +---+---+---+---+---+---+---+---+
                    | 6 | 6 | 7 | 7 | 8 | 8 | 9 | 9 |
                    +---+---+---+---+---+---+---+---+
                    | 7 | 7 | 8 | 8 | 9 | 9 |10 |10 |
                    +---+---+---+---+---+---+---+---+
```

For each direction d, we compute the value s_d, which is equal to a direction-independent offset minus the MSD (see [Deringing-Note] for detauls) as:

```
                    __                        __           2
                    \          1      /     \           \
            s_d=    /_      ------- * |     /_      x_p | ,
                k in block  N_(d,k)   \ p in P_(d,k)    /
```

where x_p is the value of pixel p, P_(d,k) is the set of pixels in like k along direction d, and N_(d,k) is the cardinality of P_(d,k). From there, the direction is computed as the value of d that maximizes s_d.

3. Directional Filter

The directional filter for pixel (i,j) is defined as the 7-tap non-linear filter

```
                   3     _
               1  --    |  /                                          \
 y(i,j)=x(i,j)+---*\ w_k*| f| x(i,j)-x(i+floor(k*d_y),j+floor(k*d_x), T|
               W  /_    |_ \                                          /
                  k=1                                                _
                         /                                          \ |
                      + f| x(i,j)-x(i-floor(k*d_y),j-floor(k*d_x), T| |
                         \                                          /_|
```

where d_x and d_y define the direction, W is a constant normalizing factor, T is the filtering threshold for the block, and f(d,T) is defined as

$$f(d, T) = \begin{cases} d & , |d| < T \\ 0 & , \text{otherwise} \end{cases}$$

The direction parameters are shown in the table below. The weights w_k can be chosen so that W is a power of two. For example, Daala currently uses w=[3 2 2] with W=16. Since the direction is constant over 8x8 blocks, all operations in this filter are directly vectorizable over the blocks.

Direction	d_x	d_y
0	1	-1
1	1	-1/2
2	1	0
3	1	1/2
4	1	1
5	1/2	1
6	0	1
7	-1/2	1

Table 1

4. Second Stage Filter

The 7-tap directional filter is sometimes not enough to eliminate all ringing, so we use an additional filtering step that operates across the direction lines used in the first filter. Considering that the input of the second filter has considerably less ringing than the input of the second filter, and the fact that the second filter risks blurring edges, the position-dependent threshold T_2(i,j) for the second filter is set lower than that of the first filter T. The filter structure is the same as the one used for the directional filter. The direction parameters for the second stage filter are shown in the table below and the filter weights are w=[1 1] with W=16/3.

Direction	d_x	d_y
0	0	1
1	0	1
2	0	1
3	0	1
4	0	1
5	1	0
6	1	0
7	1	0

Table 2

5. Setting Thresholds

The thresholds T and T_2 must be set high enough to smooth out ringing artefacts, but low enough to avoid blurring important details in the image. Although the ringing is roughly proportional to the quantization step size Q, as the quantizer increases the error grows slightly less than linearly because the unquantized coefficients become very small compared to Q. As a starting point for determining the thresholds, we use a power model of the form T_0=alpha_1*Q^beta}, with beta=0.842 in Daala, and where alpha_1 depends on the input scaling.

Another factor that affects the optimal filtering threshold is the presence of strong directional edges/patterns. These can be estimated from the s_d parameters computed in the direction search as delta=s_(d_opt)-s_(d_ortho), where d_ortho=d_opt+4 (mod 8). We compute the direction filtering threshold for each block as

```
                /  1      /                            1/6  \ \
     T = T_0*max| ---, min| 3, alpha_2*(delta*delta_sb)     | |,
                \  2      \                                 / /
```

where delta_sb is the average of the delta values over the entire superblock and alpha_2 also depends on the input scaling. For the second filter, we use a more conservative threshold that depends on the amount of change caused by the directional filter.

```
                          /     T                    \
              T_2(i,j) = min| T, --- + |y(i,j)-x(i,j)| |.
                          \     3                    /
```

As a special case, when the pixels corresponding to the 8x8 block being filtered are all skipped, then T=T_2=0, so no deringing is performed.

6. Superblock Filtering

The filtering is applied one superblock at a time, conditional on a flag coded in the bit-stream. This binary flag is the only information coded in the bitstream by the deringing filter. The flag is only coded for superblocks that are not skipped and it is entropy-coded based on the neighbour values.

The deringing process sometimes reads pixels that lie outside of the superblock being processed. When these pixels belong to another superblock, the filtering always uses the unfiltered pixel values -- even for the second stage filter -- so that no dependency is added between the superblocks. This makes it possible -- in theory -- to filter all superblocks in parallel. When the pixels used for a filter lie outside of the viewable image, we set f(d,T)=0.

7. Results

The deringing filter described here has been implemented for the Daala [Daala-website] codec. It is available from the Daala Git repository [Daala-Git]. We tested the deringing filter on the Are We Compressed Yet [AWCY] ntt-short1 set over the 0.025 bit/pixel to 0.1 bit/pixel range, corresponding to a 1080p30 bitrate of 1.5 Mbit/s to 6 Mbit/s. The Bjontegaard-delta [I-D.daede-netvc-testing] rate reduction over that range was 6.5% for PSNR, 4.7% for PSNR-HVS, 5.6% for SSIM and -6.0% (regression) for FAST-SSIM. Visual inspection confirmed that the quality is indeed improved, despite the regression in the FAST-SSIM result.

8. Conclusion

We have demonstrated an effective algorithm to remove ringing artefacts from coded images and videos. The proposed filter takes into account the directionality of the patterns it is filtering to reduce the risk of blurring.

9. IANA Considerations

This document makes no request of IANA.

10. Security Considerations

This draft has no security considerations.

11. Informative References

[AWCY] "Are We Compressed Yet?", Xiph.Org Foundation , <https://arewecompressedyet.com/>.

[Daala-Git] "Daala Git repository", Xiph.Org Foundation , <http://git.xiph.org/?p=daala.git;a=summary>.

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

[Deringing-Note] Valin, JM., "Directional Deringing Filter", Xiph.Org Foundation , <http://jmvalin.ca/notes/dir_dering.pdf>.

[I-D.daede-netvc-testing] Daede, T. and J. Jack, "Video Codec Testing and Quality Measurement", draft-daede-netvc-testing-01 (work in progress), July 2015.

Author's Address

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

Email: jmvalin@jmvalin.ca