Daala's advanced coding techniques

FFmpeg implementation and how they fit in AOMedia's codec

Rostislav Pehlivanov
atomnker@gmail.com

2016-01-30

Implementing a native Daala decoder in FFmpeg

Or how I learned to stop worrying and did it anyway

Some things happened...

- AOMedia's codec has begun development
  - `https://chromium.googlesource.com/webm/aom/`
  - `https://chromium-review.googlesource.com/#/q/project:webm/aom`

Some things happened...

- AOMedia's codec has begun development
  - `https://chromium.googlesource.com/webm/aom/`
  - `https://chromium-review.googlesource.com/#/q/project:webm/aom`
- Daala's development will slow down

Some things happened...

- AOMedia's codec has begun development
  - `https://chromium.googlesource.com/webm/aom/`
  - `https://chromium-review.googlesource.com/#/q/project:webm/aom`
- Daala's development will slow down
- VP9's codebase has been chosen as a starting point

Some things happened...

- AOMedia's codec has begun development
  - `https://chromium.googlesource.com/webm/aom/`
  - `https://chromium-review.googlesource.com/#/q/project:webm/aom`
- Daala's development will slow down
- VP9's codebase has been chosen as a starting point
- Xiph and Cisco's teams have started to implement some of their coding tools

Some things happened...

- AOMedia's codec has begun development
  - `https://chromium.googlesource.com/webm/aom/`
  - `https://chromium-review.googlesource.com/#/q/project:webm/aom`
- Daala's development will slow down
- VP9's codebase has been chosen as a starting point
- Xiph and Cisco's teams have started to implement some of their coding tools
- Daala might become an image-only codec
  - Hopefully with support for a lossy alpha channel

Why bother?

- Google succeeded in quickly pushing their VP9 codec though Chrome(ium)
- Other browsers were slow to follow (have to ship another library)
- libvpx had speed issues
- FFVP9 was not ready on time (Firefox just switched to using it)
- ...leading to fragmentation and user agent checks for webm support

The idea

- Have support in libavcodec for AOMedia/NetVC/Daala on bitstream freezing
- Keep maintaining it and improving it until the reference implementation is stable
- That way any browser wishing to have support would only need to wait until next stable release/cherry pick.

What a normal DCT based codec does

Encoder:

- Splits image into blocks
- Does a forward DCT transform on all the blocks
- Quantized the resulting coefficients (possibly using vector quantization)
- Transmits the quantized coefficients

Decoder:

- Receives and dequantized coefficients
- Applies an inverse DCT transform
- Applies filtering (e.g. deblocking)

What a normal DCT based codec does

Encoder:

- Splits image into blocks
- Does a forward DCT transform on all the blocks
- Quantized the resulting coefficients (possibly using vector quantization)
- Transmits the quantized coefficients

Decoder:

- Receives and dequantized coefficients
- Applies an inverse DCT transform
- Applies filtering (e.g. deblocking)

Daala does pretty much everything differently...

Daala's unique coding tools

- Entropy encoding
  - Range coding
  - Multi symbol
  - Adaptive
- Screen coding
  - Uses wavelet transforms for blocks
  - Sometimes uses Unary coding for DC coefficients
- Perceptual Vector Quantization
  - Activity masking
- Lapped transforms
- Deringing filter
- ~~Bilinear blur for I-frames~~

Daala's entropy encoder

- Unconventional - splits coding of uncompressable raw bits away
- Appends the raw bits buffer at the end of the stream
- Read/written sequentially from end to start
- Avoids the patent hell of arithmetic coding
- Codes multiple symbols

Daala's use of wavelets for blocks

- Uses a Haar wavelet transform to compress the coefficients
- Only used on fully lossless frames currently
- Possibility to be used in a mixed block transforms (since the overlap filter is invertible)
- Very simple (able to write a decoder in around 500 lines)

Perceptual Vector Quantization

- Splits coefficients into bands (similar to audio)
- 'Synthesizes' coefficients
  - Coefficients represented by a vector
  - Each coefficient is normalized e.g. [0.0f, 1.0f]
  - Multiplied by the vector gain (transmitted separately)
- Uses standard zigzag coding for the bands
- Can accept 'reference' coefficients to use as a base

Perceptual Vector Quantization - Ref path

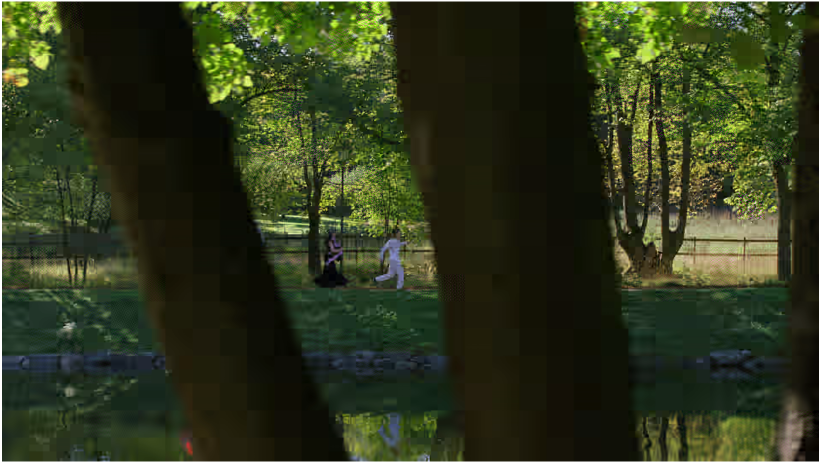
- Reduces coefficient delta by using the reference provided
- Uses the householder reflection to align the ref to an axis (flips sign)
- Encoder codes the difference between the current vector and the reference

- Used for Chroma from Luma
- Used for Intraprediction
- Used for Interprediction
  - Does a forward transform on the reference frame during decoding
- Can potentially be used for any other kind of prediction (e.g. alpha from luma)

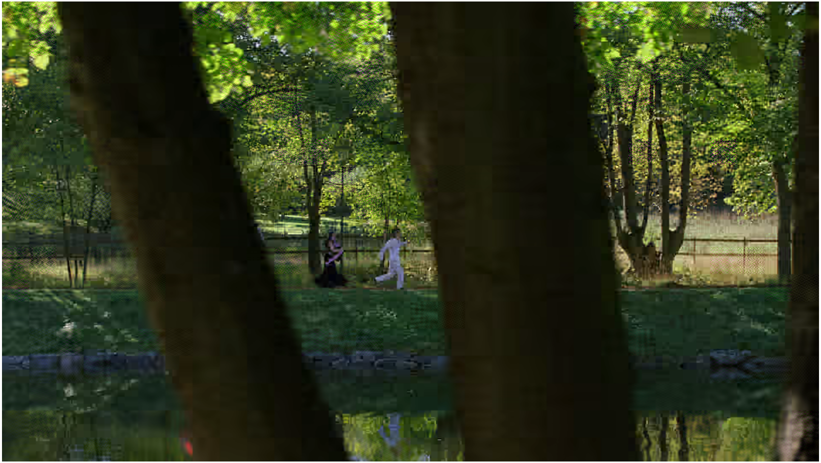
Perceptual Vector Quantization - Activity Masking

- Not signalled - only a single global flag to enable
- Acts on larger blocks (4x4 have too limited quantization)
- Increases quantization on blocks with contrast (impercievably)
- Gives more bits to blocks with low contrast

Perceptual Vector Quantization - Without Activity Masking



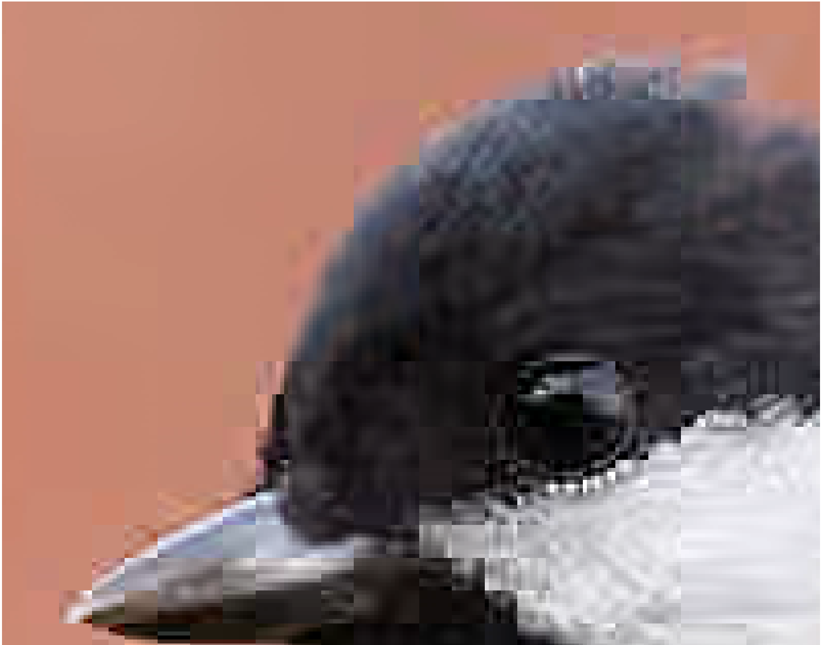
Perceptual Vector Quantization - With Activity Masking



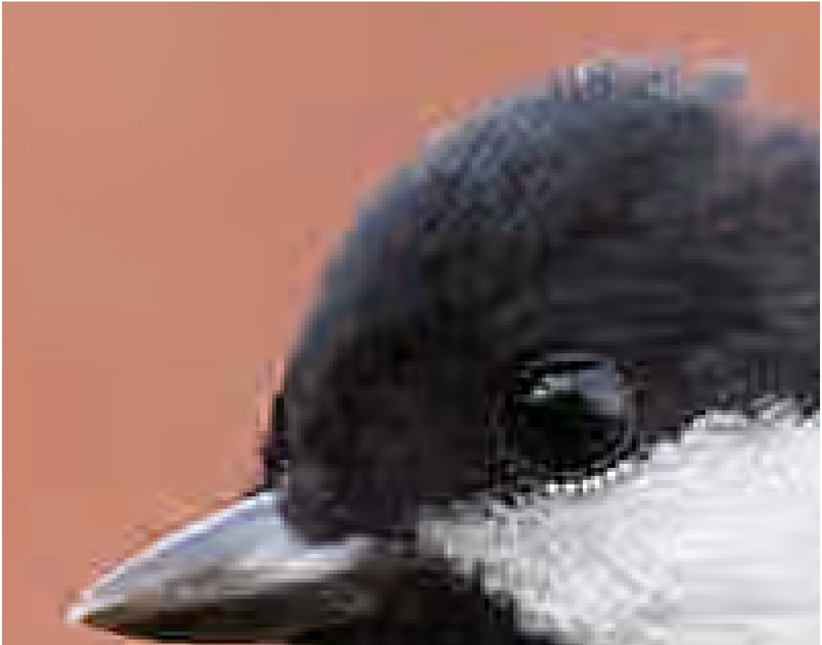
Lapped transforms

- Makes the image appear more blocky
- 'Resizes' the block + some outside zone inside the block

Lapped transforms



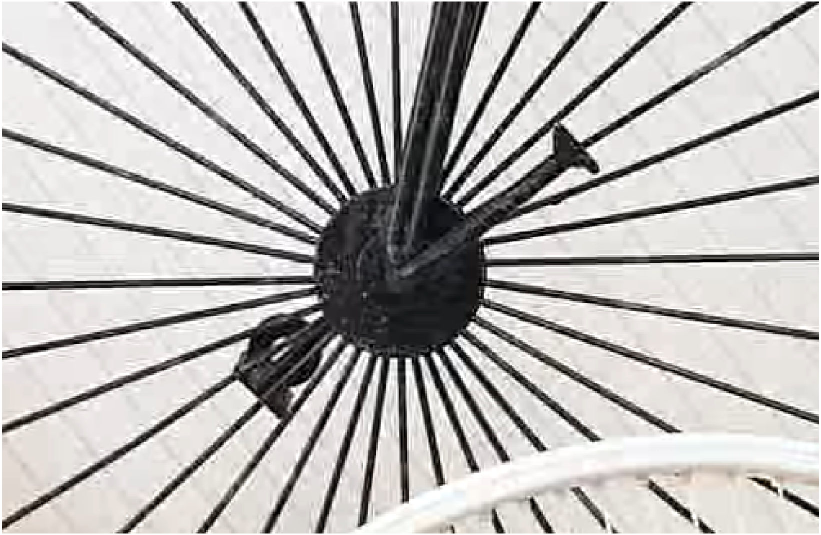
Lapped transforms



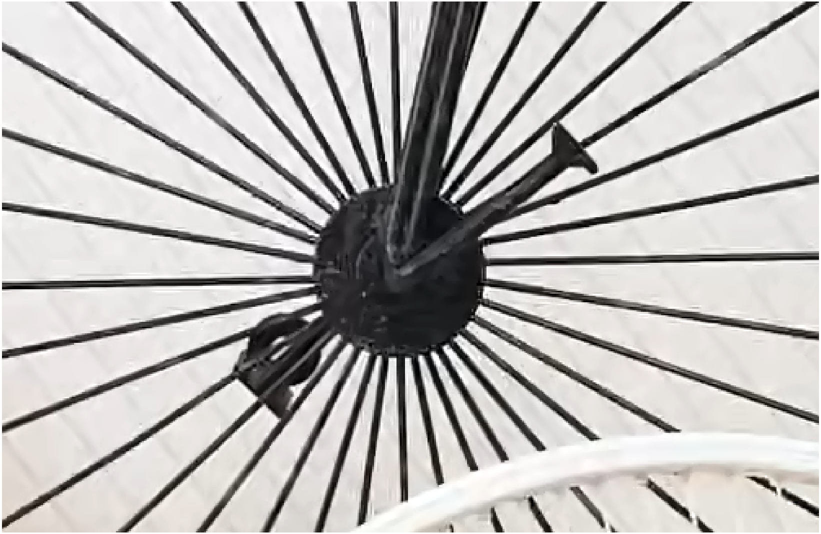
Deringing

- Conditional Replacement Filter
- Ringing will usually manifest itself as noise above the quantization step
- Picks a center pixel and scans every pixel around it
- If a pixel is deviating above the quantization step, replace it with the value of the center pixel.

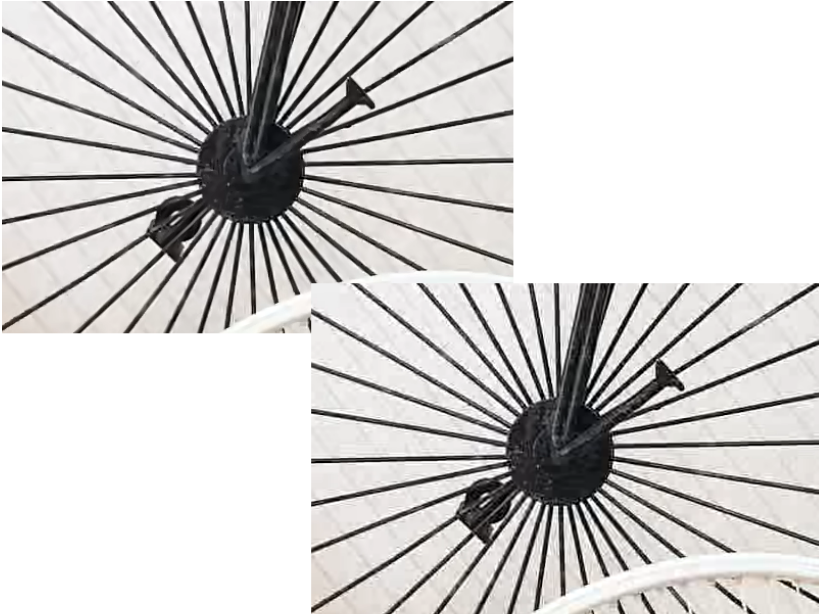
Deringing



Deringing



Deringing



FFmpeg Daala decoder

- Can decode Daala I-frames only
- Some code written from scratch, most is rewritten libdaala
- Still no support for the deringing filter
- Still some artifacts with 64x64 transforms
- Fully templated DSP
- But nearly bit identical

The End

Questions?