# AOM Analyzer Update
# AOM Testing SG

2021-Aug-13

Nasir Hemed <hemednas28@gmail.com>
Nathan Egge <negge@videolan.org>

# Who am I?

- University of Toronto
  - Comp Sci Major and Math minor

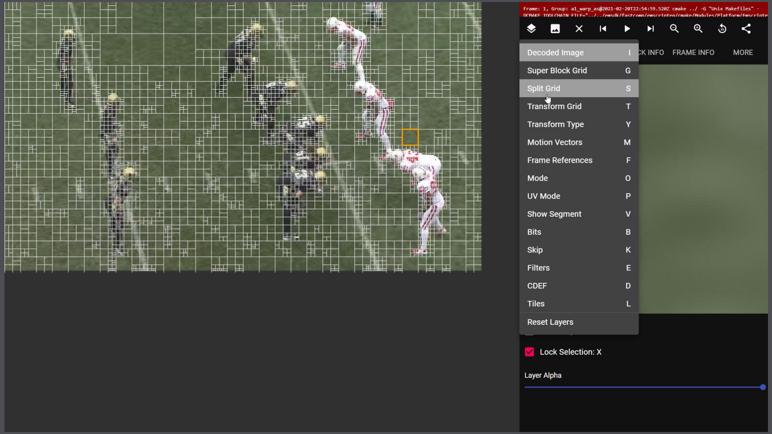
- Google - SWE Intern
  - Video Compression
  - Analyzed AV1 Tools

- VideoLAN GSoC Contributor
  - Project title: “Metadata extraction using AV1 decoder” [1]
  - Display new metadata in the AOMAnalyzer

[1] https://summerofcode.withgoogle.com/projects/#4771554950381568

# AOMAnalyzer: An Open Source AV1 Analyzer

- Browser-based Bitstream Analyzer
  - Frontend React
  - libaom decoder (wasm)
- Free to use (Open Source)
  - No license required
- Integrated with reference codebase
  - Allows you to extract almost anything the decoder can access
- Supports most of the core tools
  - MVs, Modes, Split + Tx Partitions, Transform Types, ..., etc.



```
Frame: 1, Group: a1_warp_as@2021-02-20T22:54:59.520Z cmake ../ -G "Unix Makefiles" -
DCMAKE_TOOLCHAIN_FILE="../../emsdk/fastcomp/emscripten/cmake/Modules/Platform/Emscripte
```

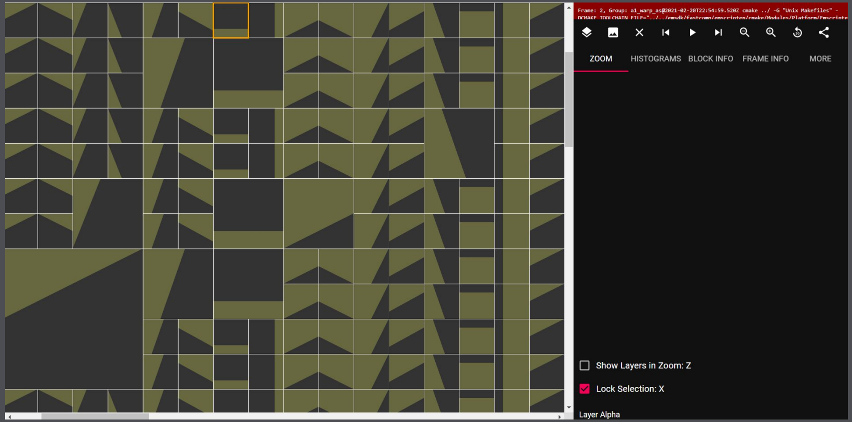
☑ Lock Selection: X

# Why a web based analyzer?

- Easy to Access
  - Anyone with a browser can analyze AV1 stream
  - Deep links into the video stream
  - Always has the latest version
- Continuous Integration (CI)
  - Bespoke analyzer per commit (even non-standard bitstreams)
  - Integrated with AWCY
  - Regression tests could produce a bitstream link on error
- No server infrastructure needed
  - Frontend + decoder run locally on your browser

# Project Contributions

- Continuous Integration + Linting
- Motion mode layer
- Compound Modes
    - Wedge shapes and sign
- Film Grain



```
Frame: 2, Group: a1_warp_as@2021-02-20T22:54:59.520Z cmake ../ -G "Unix Makefiles" -
DCMAKE_TOOLCHAIN_FILE="../../emsdk/fastcomp/emscripten/cmake/Modules/Platform/Emscripte
```

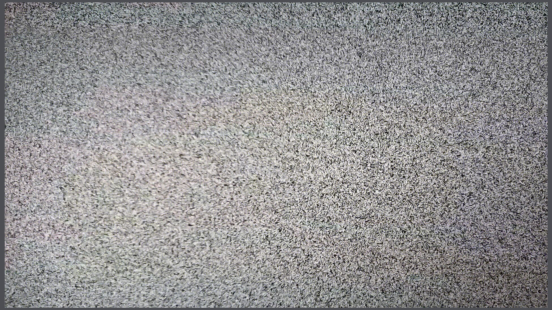
☐ Show Layers in Zoom: Z

☑ Lock Selection: X

- [https://github.com/nasirhemed/aomanalyzer/tree/add-visualization](https://github.com/nasirhemed/aomanalyzer/tree/add-visualization)
- [https://github.com/nasirhemed/aom/tree/gsoc-2021](https://github.com/nasirhemed/aom/tree/gsoc-2021)

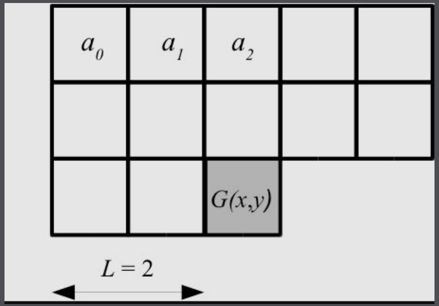
# AV1 Film Grain

- Grain in a noisy image
    - Sometimes used in movies, and film
- But… encoding noise is expensive
    - Randomness makes it difficult to predict
    - TX and quant naturally removes noise



- AV1 Approach: Grain Synthesis
    - Denoise input video before encoding
    - Estimate Film Grain Parameters from original
    - Generate and apply grain after decoding the frame

# Estimating Film Grain Parameters

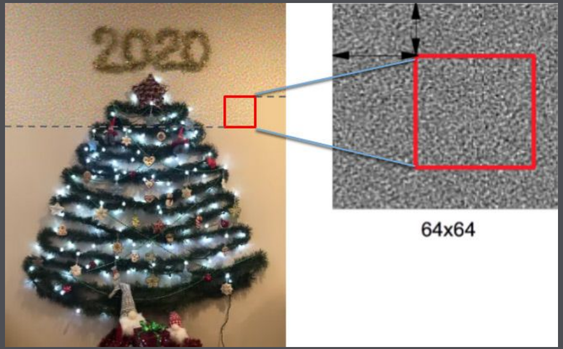


- Modeling Grain Pattern
  - 64x64 Block of grain for luma
  - Autoregressive Model
  - $G(x, y) = a_0 G(x-2, y-2) + a_1 G(x-1, y-2) + \cdots + z$

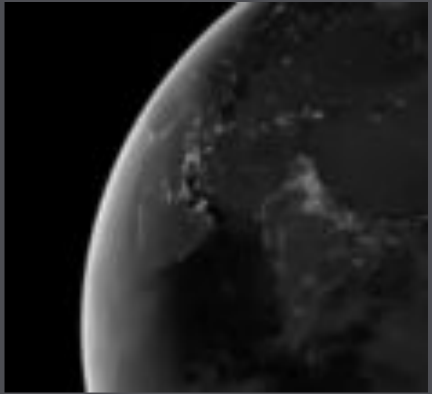
- Modeling Grain Intensity
  - Grain scaled based on the intensity of the signal
  - Linear scaling function applied on grain
  - $f(Y)G(x, y)$

# Applying Film Grain

1. Grain Pattern sub-block taken
   a. 32X32 sub-block for luma
   b. chroma depends on subsampling
   c. Random x, y offset for sub-block location
2. Grain Pattern Scaled
   a. Linear scaling function used
3. Grain applied on pixel
   a. $\hat{Y} = Y + f(Y)G_l$



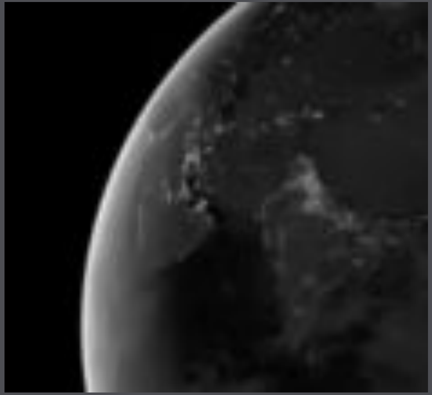
# Film Grain in AV1



$Y$

decoded luma

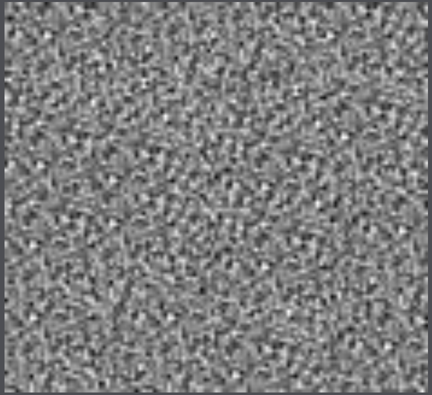
# Film Grain in AV1



$Y$

↑

decoded luma

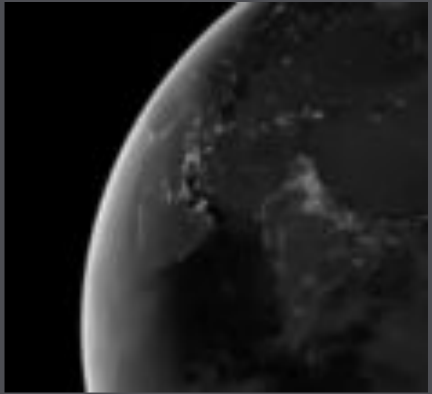


$G_l$

↑

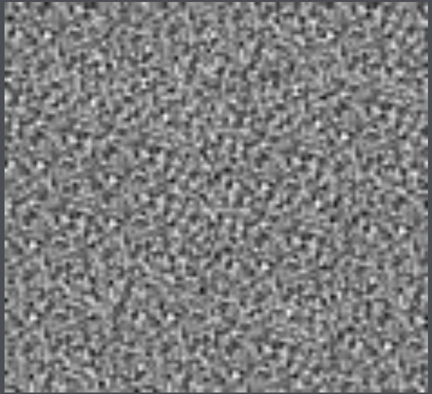
unscaled grain

# Film Grain in AV1



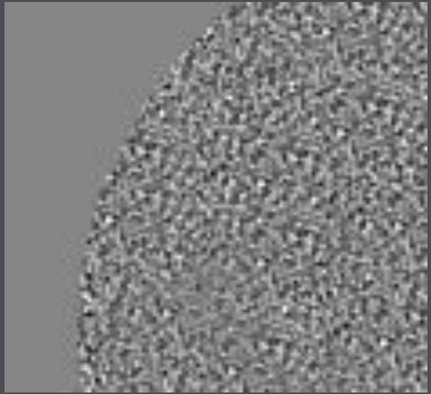
$Y$

decoded luma



$G_l$

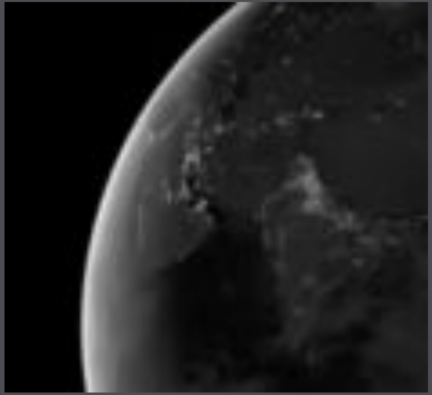
unscaled grain



$f(Y)G_l$

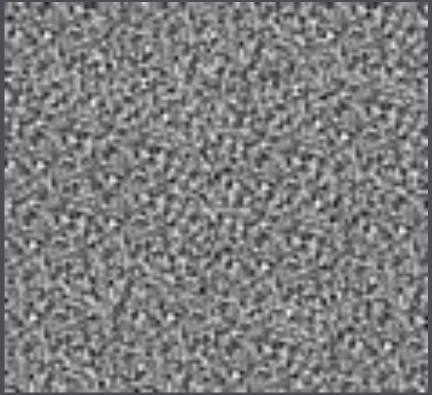
scaled grain

# Film Grain in AV1



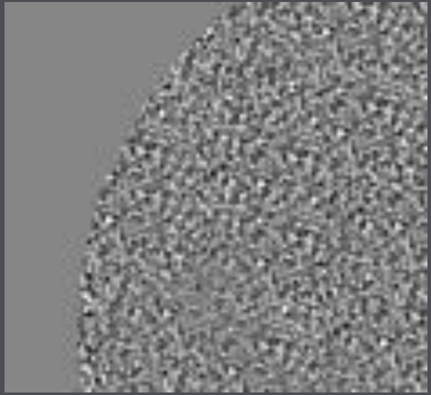
$Y$

decoded luma



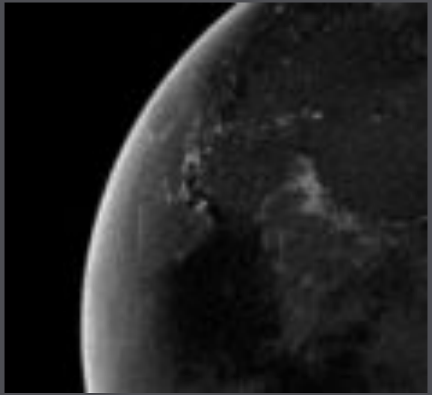
$G_l$

unscaled grain



$f(Y)G_l$

scaled grain

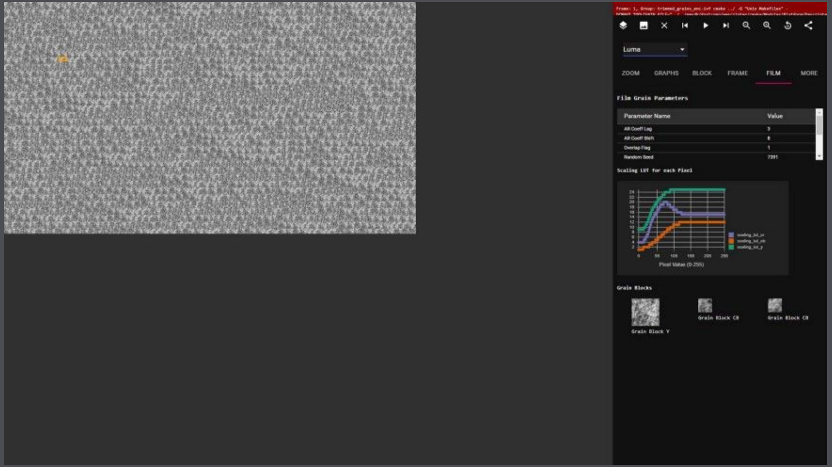


$Y + f(Y)G_l$

luma with grain added

# Displaying Grain Properties on Analyzer

- Metadata extracted
  - Film grain parameters
  - Scaling Lookup Table (Linear Scaling Function) $f(Y)$
  - Unscaled Grain $G_l$
  - Scaled Grain $f(Y)G_l$



Frame: 1, Group: trimmed_grains_enc.ivf cmake ../ -G "Unix Makefiles" -

Film Grain Parameters

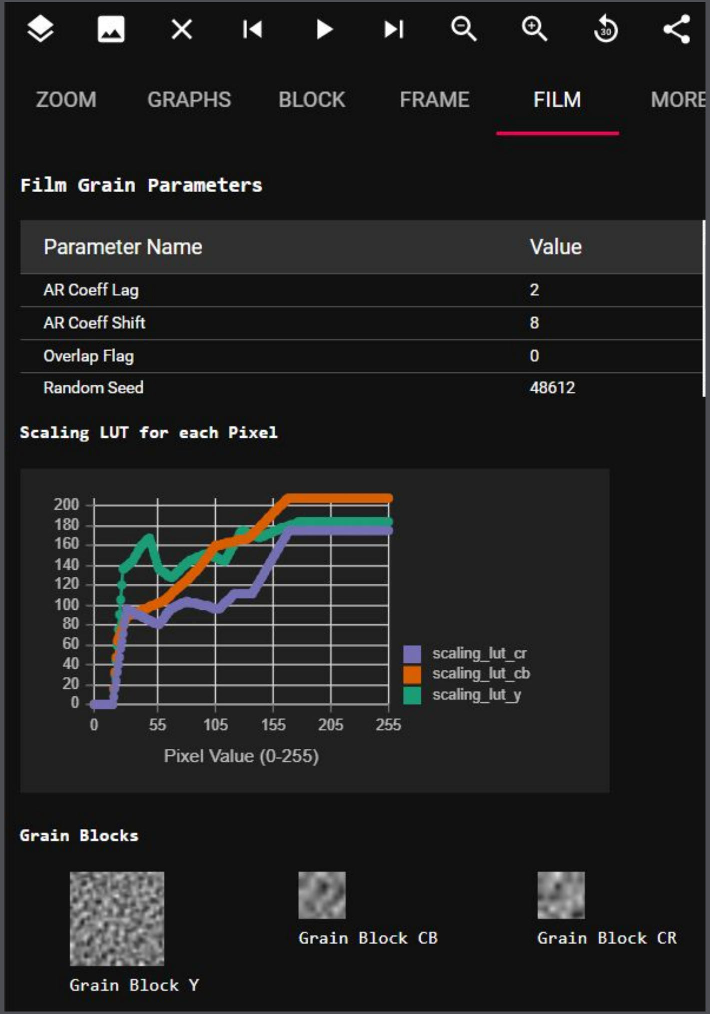
| Parameter Name | Value |
|---|---|
| AR Coeff Lag | 3 |
| AR Coeff Shift | 8 |
| Overlap Flag | 1 |
| Random Seed | 7391 |

Scaling LUT for each Pixel

Grain Blocks

# AOMAnalyzer: Film Grain Params

- Film Grain Parameters from bitstream
- Scaling Lookup Table
- Grain patterns for luma and chroma



Film Grain Parameters

| Parameter Name | Value |
|---|---|
| AR Coeff Lag | 2 |
| AR Coeff Shift | 8 |
| Overlap Flag | 0 |
| Random Seed | 48612 |

Scaling LUT for each Pixel

Grain Blocks

# Demo

[1] https://nasirhemed.github.io/aomanalyzer/?decoder=aomanalyzer/inspect.js&file=earth.ivf
[2] https://nasirhemed.github.io/aomanalyzer/?decoder=aomanalyzer/inspect.js&file=hammer.ivf
[3] https://nasirhemed.github.io/aomanalyzer/?decoder=aomanalyzer/inspect.js&file=man.ivf

# Future Work

- Film Grain
  - Flag to indicate whether parameters were updated
- Residual Data Layer
- Superresolution Layer
- Features for HBD bitstreams
- Support for SIMD [1]

[1] https://v8.dev/features/simd

# Questions?