AVIF and OpenUSD

Why AVIF?

The **AV1 Image Format** is a royalty-free open-source picture format, with modern compression, principled color specification, high dynamic range values, depth images and alpha channels, and support for layered and sequential images.

Tooling is commonly available, and the format is supported by all major browsers. Support for hardware decoding has started to appear.

Emissive surfaces are a key driver for the use of EDR/HDR in USDZ. Emissive surfaces include common applications such as sky domes for image based lighting, and scene elements such as self illuminated control panels and hand held device screens. USDZ already supports OpenEXR for such use cases, but AVIF offers significant compression over what OpenEXR provides.

Alpha is a key feature of AVIF for use in USDZ. This feature exists in OpenEXR and PNG, but neither offers both adequate compression levels and EDR/HDR/WCG. Typical applications include cut out effects such as leaves on a card, or matting effects, to isolate glowing controls on a self-illuminated control panel.

Key Goals

- Read AVIF files, supporting features compatible with Hio
- As small and light an inlinable/embedded implementation as possible,
- Integration can be easily, regularly updated by us from fully-supported sources.
- Support EDR/HDR/WGC
- Support for Monochrome image layers (alpha, depth)
- Buildable with OpenUSD's currently required tool chain with no new tools required
- No new licensing requirements
- Does not introduce secondary library requirements (such as zlib and the like).

Stretch Goals

- write AVIF files
- support image sequences
- introduce SMPTE camdkit metadata OBU structure
- introduce cube map metadata to the scalability structure

Technical Requirements

- Embeddable within USD as part of the core build and distribution, following the pattern of HioOpenEXR to if possible. This means:

  - Optional ~ a unity build. libavif and libaom are light and could be adapted to this format. Have not investigated a unity build for other libraries
  - Symbol isolation through namespacing, so that the interred library does not leak symbols and therefore collide with a user supplied and linked copy of the library
  - Wrapping through Hio interfaces; this involves some conformance to an image spec, and thus might involve pixel format conversion or resizing
  - No third party external dependencies that are not also interred. In the case of OpenEXR, a minimal subset of libdeflate was symbol-isolated and interred.

- Hio has a mechanism to restrict writability, we can leverage it to reject unsupported write requests. This is similar to how Apple's ImageIO works, see Appendix B.

- AVIF encodes color gamut in a linear, fully general format as is done in OpenEXR. This is fully compatible with our plans for named, and mathematically specified color management.

- Library for codec is still TBD, and alternatives should be evaluated. Preliminary discussion held on 10/2/2023.
  - AOM ~ this is the reference codec provided by the Alliance for Open Media
  - SVT ~ a high performance production encoder, by AOM
  - Rav1e ~ a highly rated independent implementation written in Rust
  - dav1d ~ a high performance independent implementation
  - gav1 ~ an independent implementation by Google, not as highly developed as others, apparently
  - avm ~ a fork of aom, by a gplv3 community

Appendix A ~ AVIF Reference

The AVIF specification is here https://github.com/AOMediaCodec/av1-avif and it goes into great depth. Most of the information below is from this part of the spec: https://aomediacodec.github.io/av1-spec/av1-spec.pdf

Compression: AOMedia Video 1, AV1 is an open, royalty-free coding format for video and image transmission. The AV1 reference encoder achieved 34%, 46.2% and 50.3% higher data compression than libvpx-vp9, x264 High profile, and x264 Main profile respectively.

- "AV1 beats x264 and libvpx-vp9 in practical use case" https://web.archive.org/web/20191105053533/https://engineering.fb.com/video-engineering/av1-beats-x264-and-libvpx-vp9-in-practical-use-case/ . Facebook Engineering. 10 April 2018.

AVIF showed better compression efficiency than JPEG as well as better detail preservation, fewer blocking artifacts and less color bleeding around hard edges in composites of natural images, text, and graphics.

- Mavlankar, Aditya; De Cock, Jan; Concolato, Cyril; Swanson, Kyle; Moorthy, Anush; Aaron, Anne (2020-02-13). "AVIF for Next-Generation Image Coding". The Netflix Tech Blog. Archived

Compression artifacts are reduced through a de-ringing filter.

- https://arxiv.org/pdf/1602.05975.pdf

Multiple Images and Data: AV1 Image File Format, AVIF can store multiple images via AV1, including both single images and image sequences (such as bursts or animations), in a single file. GIF style flip book images are supported via multi-layer images. AVIF files are designed to be HEIF-compliant for both image items and sequences.

Alpha and depth data: AVIF calls these “monochrome” images, and a multi-component image may include an alpha channel for opacity information.

Layers: AVIF supports spatial layers which may either be composed with a prior layer to create a new image variant via an “Operating Point”, or the layer may be independent to provide a different resolution or quality independent of other layers. MIP maps are not explicitly supported, nor are cube maps.

Non-Destructive Editing: The Operating Point concept allows the storage of multiple versions of an image, including original and edited versions, without the need to duplicate the entire image data. Edits can be applied as separate "edits" without altering the original data. Our use

of AVIF applies all edits to resolve an image for use as a texture, so this feature is only of marginal interest.

HDR and Wide Gamut Imaging: SDR & HDR are supported. Bit depths are 8, 10, or 12, see table below. Color gamuts formally mentioned by AOM on the promotional website include sRGB, BT.709, BT.2020, BT.2100, and Pro Photo RGB. ICC Profiles are supported, as is “color space signaling by CICP (ITU-TH.273, and ISO/IEC 23091-2)”. Note that color gamut support is fully general, and identical in form to OpenEXR’s (as noted in metadata below).

Metadata: AVIF has specialized metadata containers. Metadata applies per Open Bitstream Unit within the container ~ an image tile is an OBU. See section 5.8 ~ https://aomediacodec.github.io/av1-spec/av1-spec.pdf Note that OBUs appear sequentially in a container, and apply sequentially to following data, much as a typical streaming format employs via “chunks”.

There is no general metadata attribute scheme in AVIF, such as is found in OpenEXR. It would be reasonable to propose the “camdkit” metadata as a formal container for the specification. https://github.com/SMPTE/ris-osvp-metadata-camdkit

- metadata_itut_t35 - country code
- metadata_hdr_cll - content light levels
- Metadata_hdr_mdcv
    - Ciexy rgb chromaticities, 0.16 fp
    - Ciexy whitepoint 0.16 fp
    - luminance_max is a 24.8 fixed-point maximum luminance, represented in candelas per square meter (nits)
    - luminance_min is a 18.14 fixed-point minimum luminance, represented in candelas per square meter (nits)
- metadata_scalability
    - Spatial layers ~ width, height
    - Temporal group ~ flipbook
- metadata_timecode
    - Time is either a frame count with a rate specified in an unsigned 32 bit integer ~ no NTSC rates, or
    - An absolute time in h:m:s + frame * rate. (slight simplification)

Audio Integration: I can find mentions that it is possible to embed audio in an AVIF file, but not on the official website, and no documentation, or instruction where you’d expect it, for example: https://trac.ffmpeg.org/wiki/Encode/AV1

Supported encodings: We don't have to choose or restrict encoding, as the decoder will turn these into RGB in the specified gamut, and then we'll apply our own gamut conversion.

seq_profile	Bit depth	mono/alpha/depth	Chroma subsampling
0	8 or 10	yes	YUV 4:2:0
1	8 or 10	no	YUV 4:4:4
2	8 or 10	yes	YUV 4:2:2
2 (not a typo)	12	yes	YUV 4:2:0, 4:2:2, 4:4:4

Appendix B ~ ImageIO read/writability

> sips --formats

```
Supported Formats:
------------------------------------------
com.adobe.pdf                pdf   Writable
com.adobe.photoshop-image    psd   Writable
com.adobe.raw-image          dng
com.apple.atx                --    Writable
com.apple.icns               icns  Writable
com.apple.pict               pict
com.canon.cr2-raw-image      cr2
com.canon.cr3-raw-image      cr3
com.canon.crw-raw-image      crw
com.canon.tif-raw-image      tif
com.compuserve.gif           gif   Writable
com.dxo.raw-image            dxo
com.epson.raw-image          erf
com.fuji.raw-image           raf
com.hasselblad.3fr-raw-image 3fr
com.hasselblad.fff-raw-image fff
com.ilm.openexr-image        exr   Writable
com.kodak.raw-image          dcr
com.konicaminolta.raw-image  mrw
com.leafamerica.raw-image    mos
com.leica.raw-image          raw
com.leica.rwl-raw-image      rwl
com.microsoft.bmp            bmp   Writable
com.microsoft.cur            --
com.microsoft.dds            dds   Writable
com.microsoft.ico            ico   Writable
com.nikon.nrw-raw-image      nrw
com.nikon.raw-image          nef
com.olympus.or-raw-image     orf
com.olympus.raw-image        orf
com.olympus.sr-raw-image     orf
com.panasonic.raw-image      raw
com.panasonic.rw2-raw-image  rw2
com.pentax.raw-image         pef
```

Appendix C ~ references in no particular order

https://github.com/SMPTE/ris-osvp-metadata-camdkit
https://github.com/AOMediaCodec/av1-avif
https://aomediacodec.github.io/av1-spec/av1-spec.pdf
https://avif.io/blog/articles/avif-faq/#doesavifsupportlayers
https://www.smashingmagazine.com/2021/09/modern-image-formats-avif-webp/#avif
https://squoosh.app/
https://www.dpreview.com/forums/post/62532734
https://en.wikipedia.org/wiki/AVIF#cite_note-:0-2