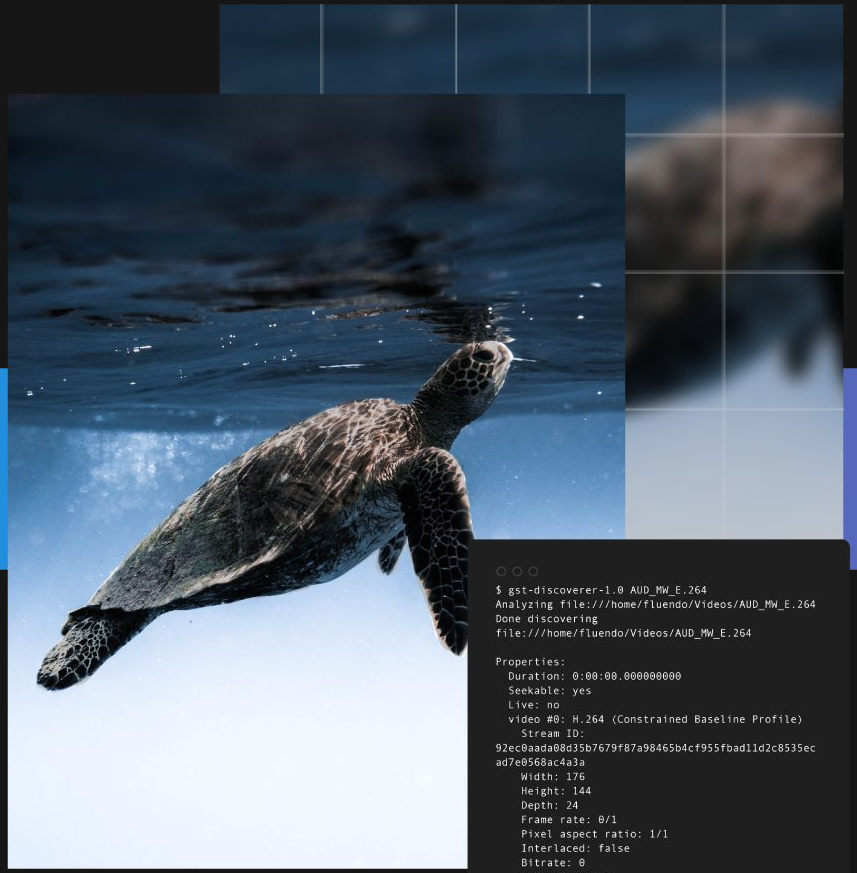
fluendo

VVC/H.266 Alpha Channel in GStreamer

Andoni Morales Alastruey

Diego Nieto

GStreamer Conference 2025

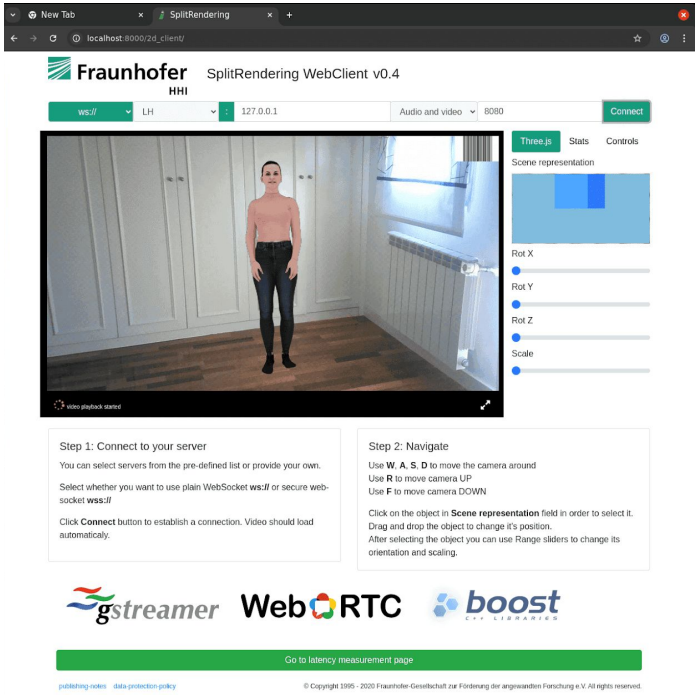


About Me

- Andoni Morales Alastruey @ylatuya
- Working for Fluendo for ~ 14 years
- GStreamer contributor since 2009
- Cerbero author
- Fluster co-author
- LongoMatch Founder

Context & Motivation: The STREAM Project

- **SPIRIT** – Scalable Platform for Real-time Immersive Telepresence:
  - Europe's first multisite and interconnected framework for telepresence applications.
  - Provides a toolkit for telepresence:
    - Capture volumetric data
    - Generate videos with 3D avatars
    - Low-latency video streaming
    - Client rendering in VR environment for an immersive experience
- **STREAM** – Scalable Telepresence with Real-time EnhAnced Multimedia:
  - SPIRIT Open Call 1
  - **Reduce bandwidth** preserving quality for 4k resolution
  - **GStreamer** + **Gst.WASM**
  - **H.266/VVC** + **Alpha Channel** support

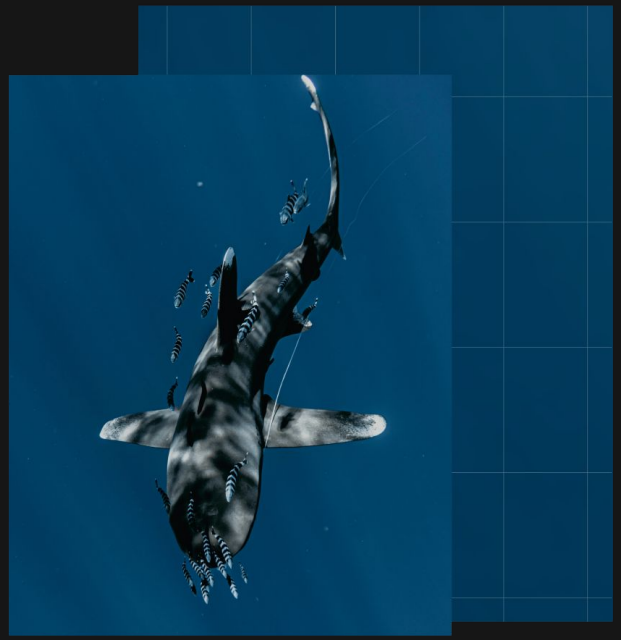


Alpha Channel Fundamentals

What is an Alpha Channel?

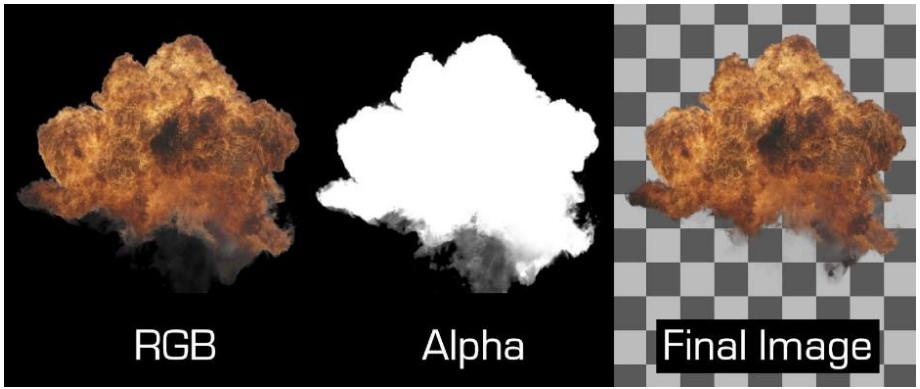
VP8/VP9

VVC / H.266



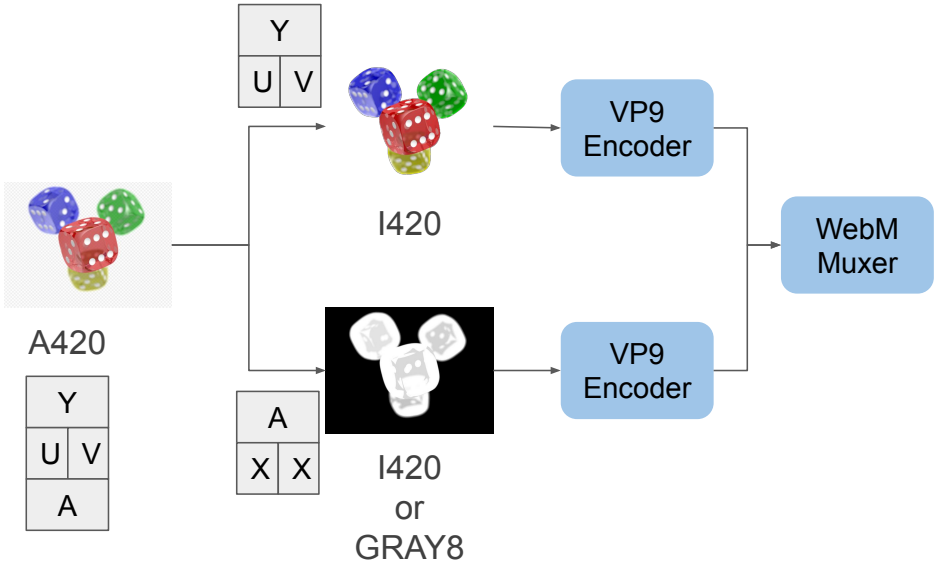
What is the Alpha Channel

- An extra grayscale channel that contains information about pixel transparency.
- Used to create an **alpha mask** to overlay videos without visible borders or backgrounds.
- RGB + A:
  a. 32 bits per pixel, 8R + 8G + 8B + 8A
  b. format: RGBA, ARGB, ...
- YUV + A:
  a. Alpha channel is usually not subsampled
  b. format: AYUV, A420, A422, A444, ...



Alpha Channel in VP8/VP9 with WebM

- Not defined in the codec specification.
- The A420 input stream is split in 2 streams:
  - Main: I420 with alpha stripped
  - Alpha: GREY 8 or I420 (A -> Y channel, U and V are ignored
- Streams are encoded independently
- The main track and the alpha track are muxed with WebM
- Supported in Firefox, Opera and Chrome

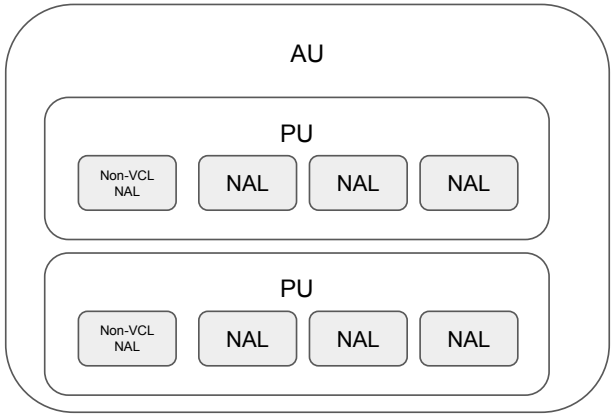


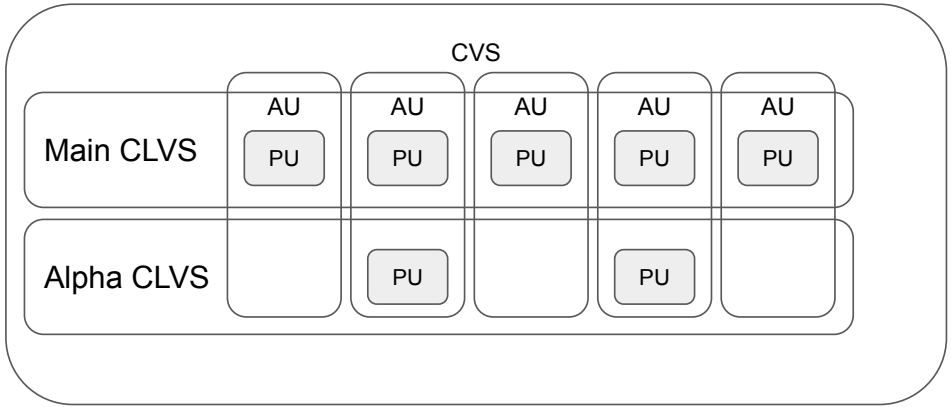
Alpha Channel in VVC/H.266

- Combination of **multi-layer** and **SEI signalling**
- Alpha channel is encoded as an **auxiliary layer**.
- Alpha channel is part of the **bitstream**.
- It's signalled through 2 SEI messages:
  - **Scalability Dimension Info** - SDI (8.19)
  - **Alpha Channel Info** - ACI (8.23)
- Signalling is defined in **ITU H.274**
- The H.274 specification was designed for VVC/H.266, with the goal of being **reusable** by **other codecs** and **applications**.

Definitions

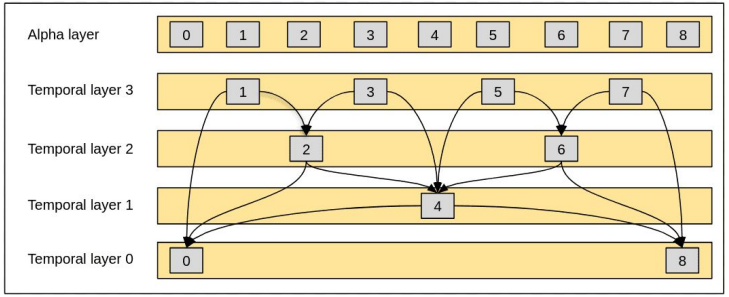
- **video coding layer (VCL) NAL unit:** coded slice NAL units
- **non-VCL NAL unit:** A NAL unit that is not a VCL NAL unit.
- **picture unit (PU)**: A set of NAL units that contain **all VCL NAL and non-VCL NAL** units of a coded picture
- **access unit (AU)**: **A set of PUs** that belong to different layers and contain **coded pictures** associated with the same **output time**.
- **coded layer video sequence** (**CLVS)**: A sequence of **PUs** of the **same layer.**
- **coded picture**: A coded representation of a picture containing all **CTUs of the picture**.
- **coded video sequence** (CVS): A sequence of **AUs**





Alpha Channel Auxiliary Layer

- Multi-layer encoding is commonly used for:
  - **Spatial** scalability: layers increases resolution
  - **Temporal** scalability: layers increase framerate
  - **SNR** scalability: layers increase quality
  - **Multi-view**: one layer per view
- HEVC: multi-layer supported through extensions:
  - **MV-HEVC** (Multiview)
  - **SHVC** (Scalable)
- VVC: multi-layer supported in the spec:
  - **Multilayer Main 10** profile
  - **Multilayer Main 10 4:4:4** profile
- The alpha channel is encoded as an auxiliary layer



Scalability Dimension Info - SDI (8.19)

This SEI message provides **information for the layers** in the current Coded Video Stream (CVS):

- **Multi-view**: the view ID of each layer
- **Alpha** and **Depth**: the auxiliary ID of each layer

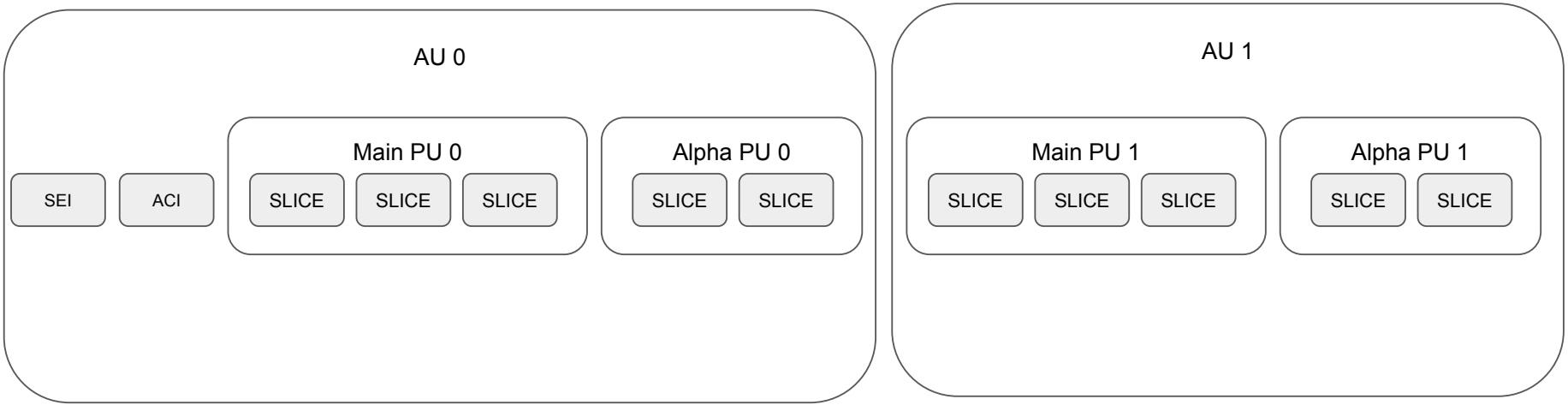
- `sdi_max_layers_minus1`: the maximum number of layers in the CVS
- `sdi_auxiliary_info_flag`: indicates whether auxiliary layers are present in the bitstream or not.
- For each layer:
  - `sdi_layer_id[i]:` ID of the layer
  - `sdi_aux_id`: AUX_ALPHA, AUX_DEPTH
  - `sdi_num_associated_primary_layers_minus1[i]`
  - For each associated primary layer:
    - `sdi_associated_primary_layer_idx`

Alpha Channel Info - ACI (8.23)

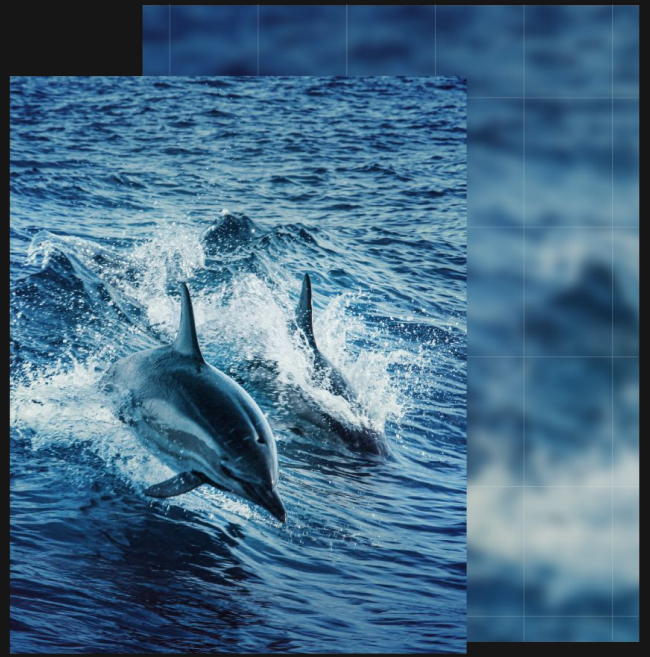
This SEI message provides information about the alpha channel **sample values** and **post-processing**. The alpha channel samples values of a picture **persists** until the next picture or end of CLVS.

- `alpha_channel_cancel_flag:` cancels the persistence of other ACI
- `alpha_channel_bit_depth_minus8`: the bit depth (eg: 8)
- `alpha_transparent_value`: transparent value (eg: 0)
- `alpha_opaque_value`: opaque value (eg: 255)

VVC/H.266 bitstream with alpha channel



Alpha Channel in GStreamer

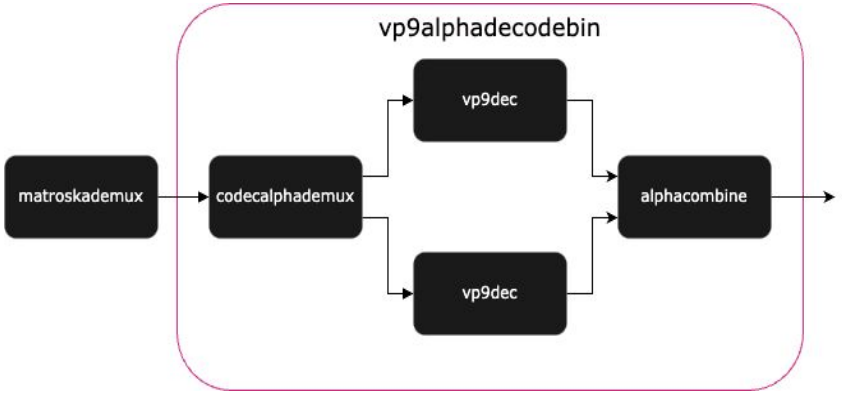


Alpha decoding support in GStreamer

- GStreamer **supports alpha** decoding for VP8/VP9 was introduced in **1.20**
- Streams with an encoded alpha layer are signaled with the ***codec-alpha*** caps.
- Decoders with alpha channel support will set ***codec-alpha=true*** in caps and an output pad with an alpha-capable format (eg: **A420)**
- GstAlphaDecodebin provides a way to support alpha channel decoding with decoders that **don’t support it**.

GstAlphaDecodebin

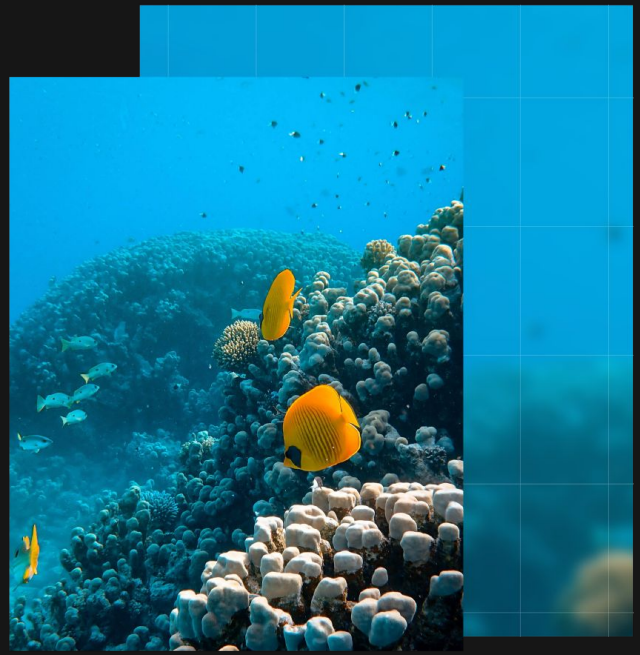
- Requires an input stream where buffers have a `GstVideoCodecAlphaMeta.`
- `codecalphademux`: splits the incoming buffers with the `GstVideoCodecAlphaMeta` into 2 output streams:
  a. Original buffer from the main stream.
  b. Alpha Channel buffers.
- `vp8dec` or `vp9dec`: 2 decoder instances that decode the main video track and the alpha one
- `alphacombine`: combines back the 2 decoded raw buffers into a single.



Encoder & Decoder Implementation

VVenc encoder

GStreamer decoder



Implementation choices

Encoder

- **VVenc:** Open source VVC/H.266 encoder by Fraunhofer.
- Initial support for multi-layer.

Decoder

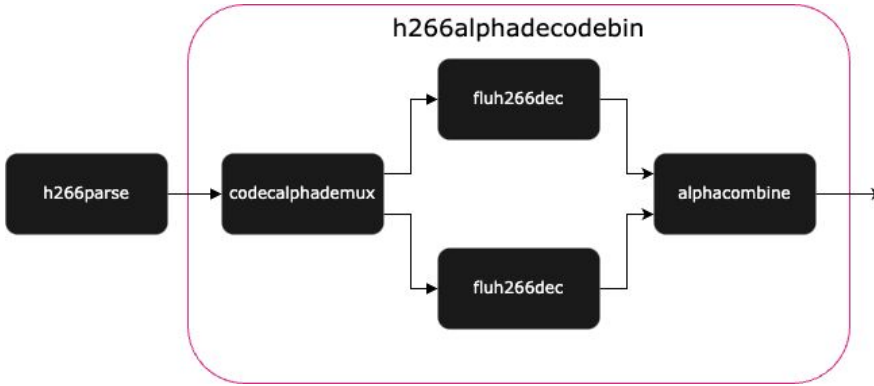
- We analyzed 2 options:
  - Implementing support in **VVdec** (Open source decoder by Fraunhofer)
  - Add support in GstAlphaDecodebin
- We choose GstAlphaDecodebin because it provides the flexibility of being able to switch between different decoder implementations, including hardware decoders.

VVC Encoder with Alpha Support

1. **Multi-layer Support:**
   - Rebase and integrate multi-layer work from the NHK's VVenc fork.
2. **Alpha Layer Encoding:**
   - Reuse existing multi-layer infrastructure; encoding the alpha channel as an independent layer.
   - Add new configurations for alpha layers.
   - *Note/Limitation:* Due to implementation constraints, the 8-bit alpha channel was temporarily encoded as an **I420** (YUV) stream, storing alpha in the Luma channel.
3. **SEI Message Implementation:** Add logic to inject the required **SDI** and **ACI** SEI messages into the bitstream, following ITU H.274.
4. Branch: https://github.com/fluendo/vvenc/tree/alpha-sei

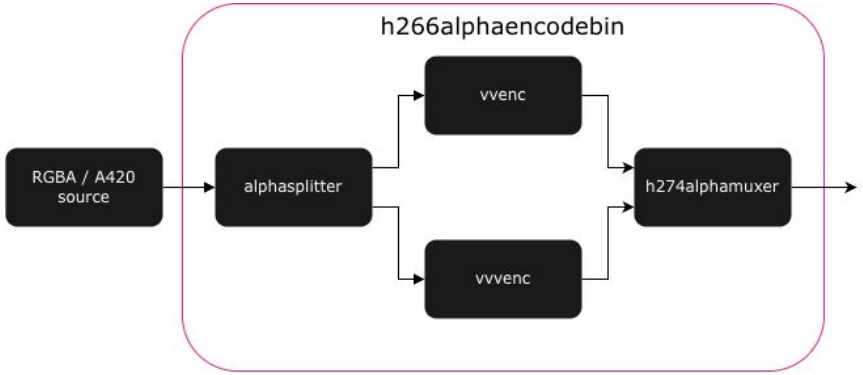
GStreamer Decoder

- `h266parse`:
  1. Add support to parse and process **ACI** and **SDI** SEI messages.
  2. Attach Alpha NALs as `GstVideoCodecAlphaMeta` to the main stream buffer.
- `h266alphadecodebin`: A that:
  1. New element (based on `GstAlphaDecodeBin`)
  2. Splits the stream based on the metadata.
  3. Uses two VVC decoders (main + alpha).
  4. Combines the two decoded raw buffers into a single **AYUV** stream.
- MR: https://gitlab.freedesktop.org/gstreamer/gstreamer/-/merge_requests/8675

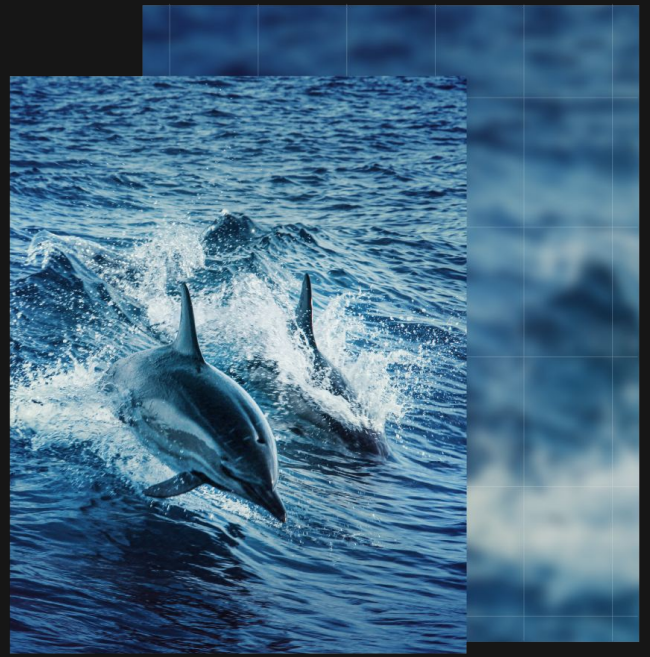


GStreamer VVC/H.266 Alpha Encoder?

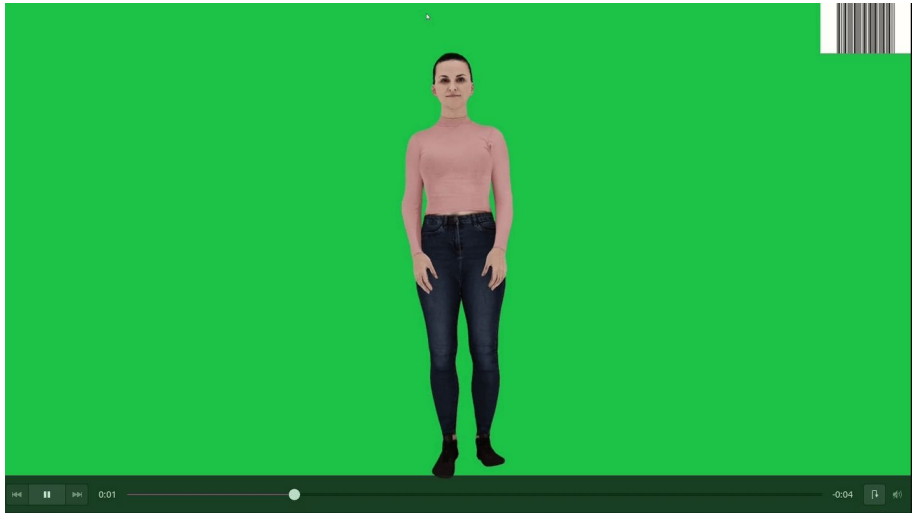
- A similar architecture could be used to support Alpha for encoders.
- It would allow supporting alpha encoder for encoders without multi-layer/alpha support
- Alpha and main are encoded independently
- `h266alphamuxer`:
  1. Combines the 2 bitstreams into a single bitstream, merging CLVS from main and alpha into a single bitstream.
  2. Injects SDI and ACI SEI signalling.



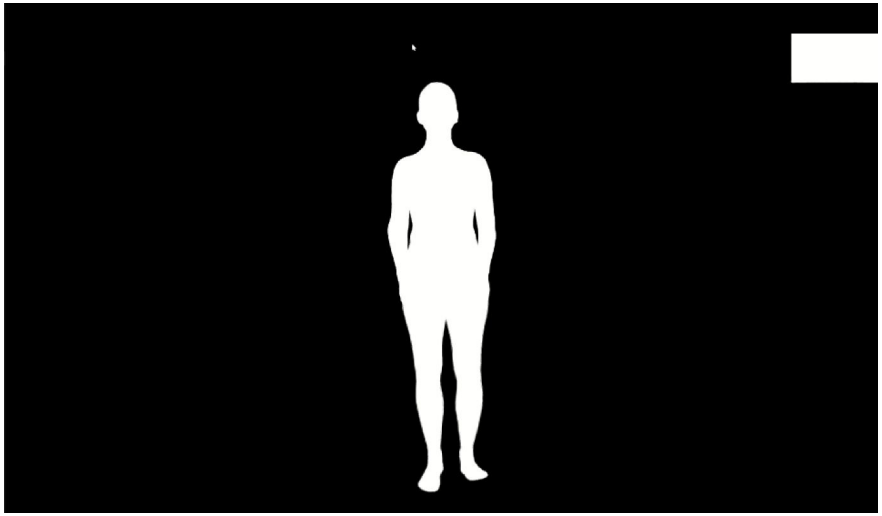
VVC / H266 Alpha Demo



Input Streams



I420 Main



I420 Alpha

VVenc encoding

```
#======== Layers ===============
MaxLayers                     : 2
MaxSublayers                  : 7
AllIndependentLayersFlag      : 1
#======== Alpha ================
SEIAlphaChannel               : 1
SEIScalabityDimension         : 1
#======== OLSs =================
EachLayerIsAnOlsFlag          : 1
NumOutputLayerSets            : 2
NumPTLsInVPS                  : 1
#======== Layer-0 ==============
LayerId0                      : 0
#======== Layer-1 ==============
LayerId1                      : 1
#======== OLS-0 ================
OlsPTLIdx0                    : 0
#======== OLS-1 ================
OlsPTLIdx1                    : 1
```

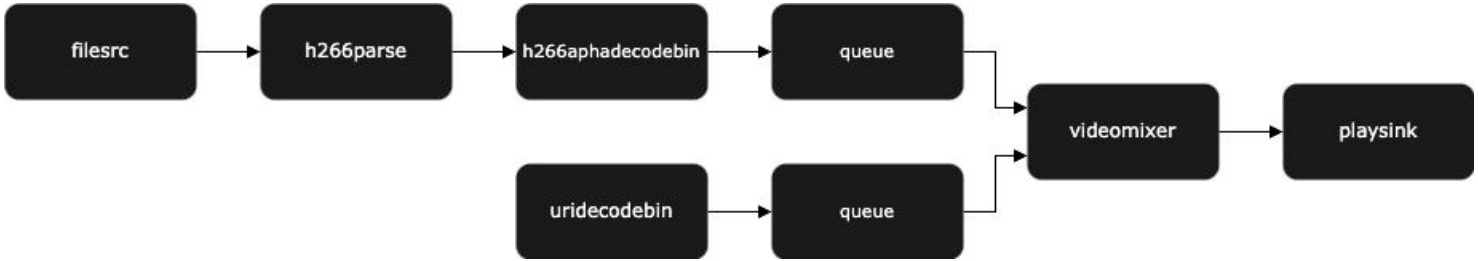
```
./vvencFFapp \

 -c ../../cfg/two_layers_alpha.cfg \

 -c ../../cfg/randomaccess_fast.cfg \

 -l0 -c sequence2K.cfg \

 -l1 -c sequenceAlpha.cfg \

 -l0 -q 22 \

 -l1 -q 22 \

    -l1 --Level=6.1 --VerCollocatedChroma=1 -v 6
```

VVenc encoding

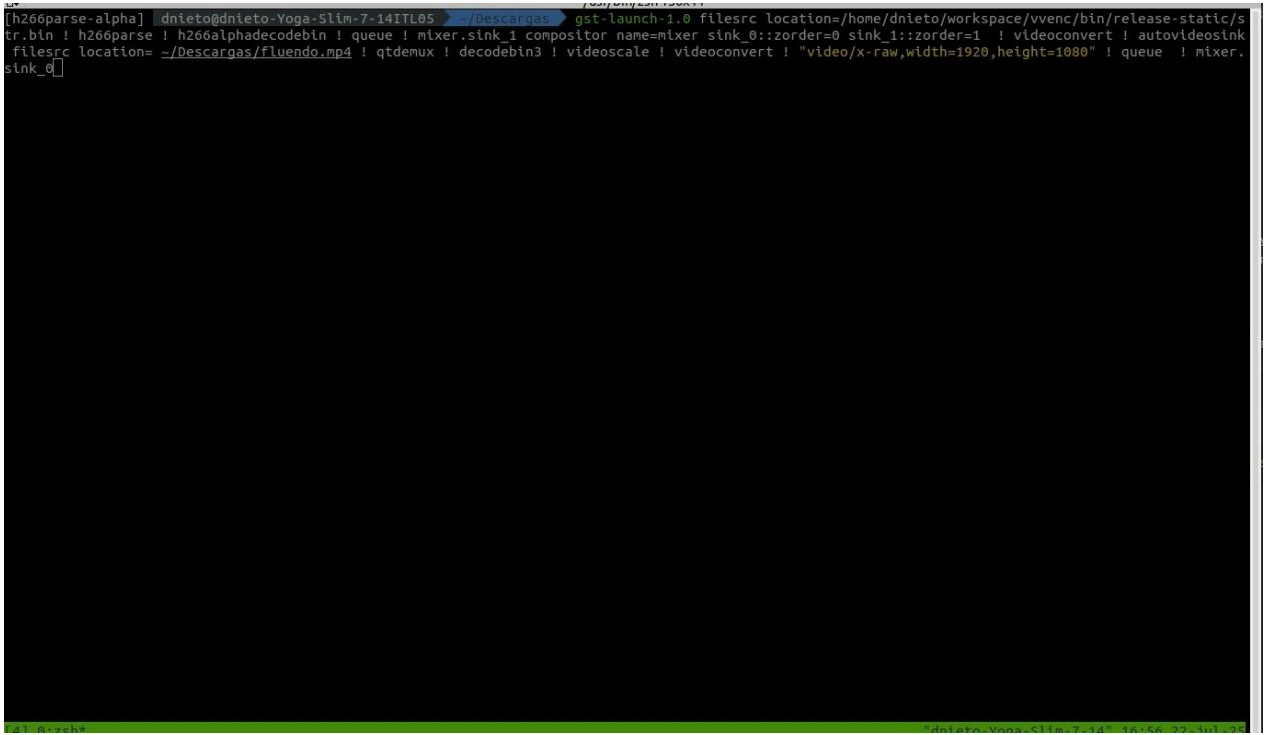
```
LayerId  0

vvenc [info]: SUMMARY ------------------------------------------------------
vvenc [info]:   Total Frames |   Bitrate     Y-PSNR    U-PSNR    V-PSNR    YUV-PSNR   Y-Lossless  U-Lossless  V-Lossless
vvenc [info]:         150    a0     80.5984   62.6694      -nan      -nan   64.0628          0         150         150
vvenc [info]: I Slices-----------------------------------------------------
vvenc [info]:   Total Frames |   Bitrate     Y-PSNR    U-PSNR    V-PSNR    YUV-PSNR   Y-Lossless  U-Lossless  V-Lossless
vvenc [info]:           4    i0    749.8200   70.1888      -nan      -nan   71.9405          0           4           4
vvenc [info]: P Slices-----------------------------------------------------
vvenc [info]:   Total Frames |   Bitrate     Y-PSNR    U-PSNR    V-PSNR    YUV-PSNR
vvenc [info]:           0    p0        -nan      -nan      -nan      -nan      -nan
vvenc [info]: B Slices-----------------------------------------------------
vvenc [info]:   Total Frames |   Bitrate     Y-PSNR    U-PSNR    V-PSNR    YUV-PSNR   Y-Lossless  U-Lossless  V-Lossless
vvenc [info]:         146    b0     62.2636   62.4634      -nan      -nan   63.9643          0         146         146
vvencFFapp [details]: Bytes written to file: 54402 (inf kbps)

LayerId  1

vvenc [info]: SUMMARY ------------------------------------------------------
vvenc [info]:   Total Frames |   Bitrate     Y-PSNR    U-PSNR    V-PSNR    YUV-PSNR   Y-Lossless  U-Lossless  V-Lossless
vvenc [info]:         150    a1     29.3376   71.6481      -nan      -nan   72.7819          0         150         150
vvenc [info]: I Slices-----------------------------------------------------
vvenc [info]:   Total Frames |   Bitrate     Y-PSNR    U-PSNR    V-PSNR    YUV-PSNR   Y-Lossless  U-Lossless  V-Lossless
vvenc [info]:           4    i1    203.7600   82.0603      -nan      -nan   83.8188          0           4           4
vvenc [info]: P Slices-----------------------------------------------------
vvenc [info]:   Total Frames |   Bitrate     Y-PSNR    U-PSNR    V-PSNR    YUV-PSNR
vvenc [info]:           0    p1        -nan      -nan      -nan      -nan      -nan
vvenc [info]: B Slices-----------------------------------------------------
vvenc [info]:   Total Frames |   Bitrate     Y-PSNR    U-PSNR    V-PSNR    YUV-PSNR   Y-Lossless  U-Lossless  V-Lossless
vvenc [info]:         146    b1     24.5589   71.3628      -nan      -nan   72.6736          0         146         146
vvencFFapp [details]: Bytes written to file: 22369 (inf kbps)
```

Decoding and playback with GStreamer

```
gst-launch-1.0
 filesrc location=aplphastream.vvc !
 h266parse ! h266alphadecodebin ! queue ! mixer.sink_0
 uridecodebin uri=file:///Path/To/fluendo.mp4 !
 queue !  mixer.sink_1
 mixer.sink_1 compositor name=mixer sink_0::zorder=0 sink_1::zorder=1 !
 playsink
```



Decoding and playback with GStreamer



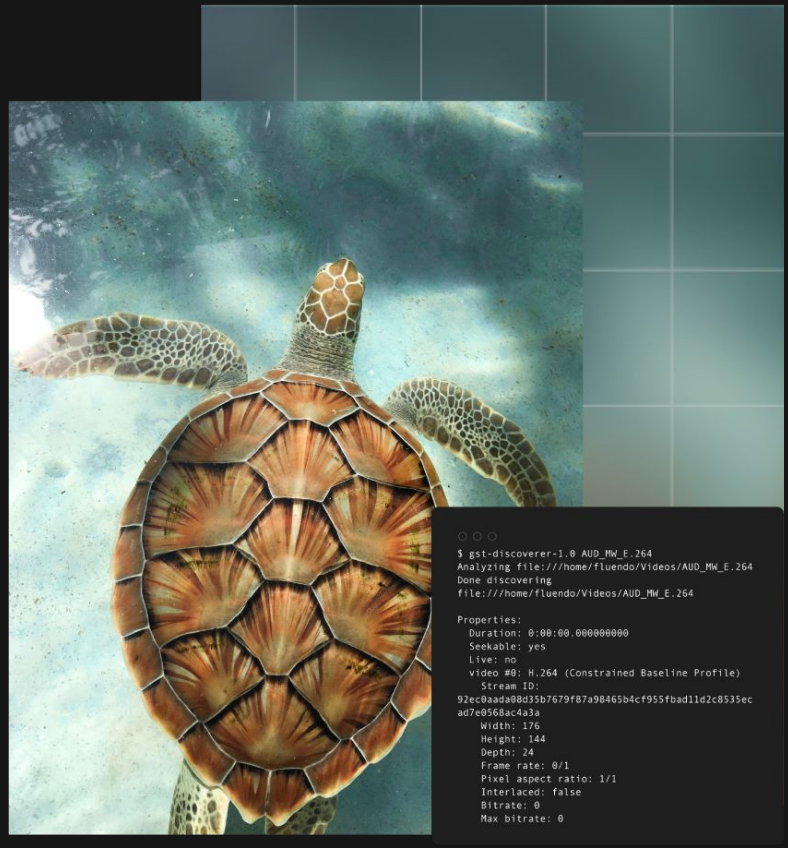
Q&A

i Don’t miss Carlos Bentzen’s talk “VVC/H.266 in GStreamer” at 16pm

fluendo

Stop taking risks.
Start finding solutions.

Thank you!



Funded by the European Union (SPIRIT, 101070672). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them. The SPIRIT project has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI).