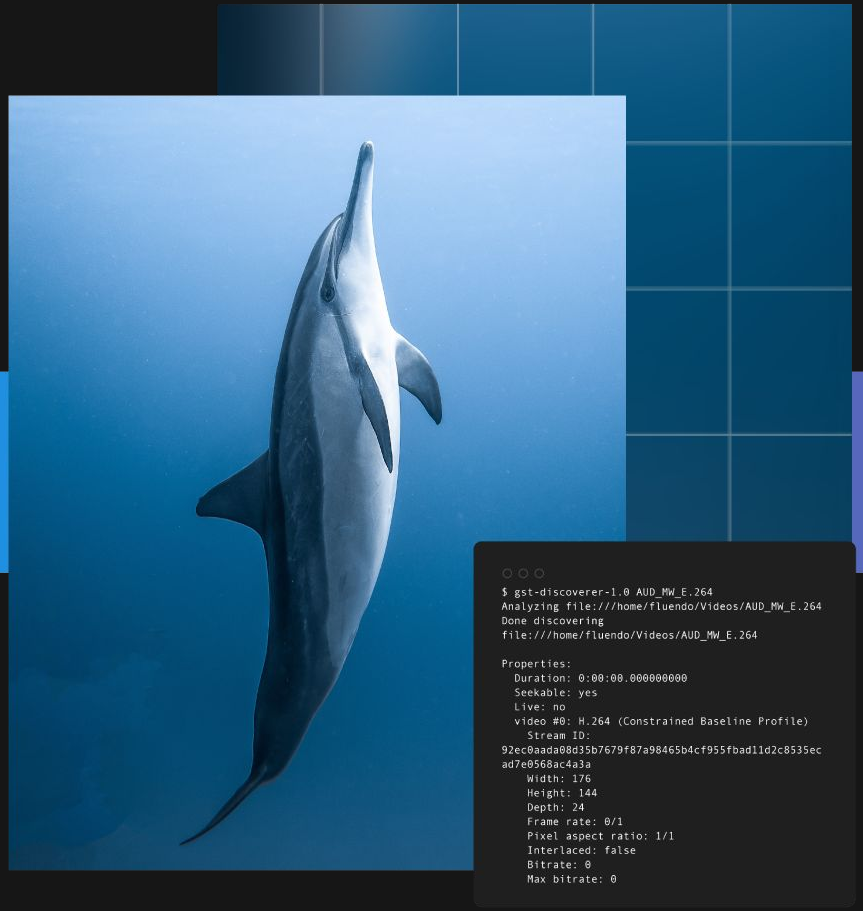
fluendo

Fluster

A Coruña, September 2023

Ruben Gonzalez

Michalis Dimopoulos



Index

Intro

Conformance: what and why | Tool stats | Project stats

What is fluster?

- Conformance test tool for multimedia decoders
- Open-source
- Multi-platform
- Modular code architecture
  - Object-oriented design
- Easy-to-digest and lightweight
  - Python (3.6+)
  - No external dependencies

Why fluster?

- Why conformance testing?
  - Decoder behaviour defined by corresponding standard
  - Reference test suites provided by owning bodies (ITU, Fraunhofer)
- Identified need: Conformance test integration in CI workflow
  - Establish conformance level of our decoders
  - Detect regressions fast
- Identified gap: lack of offer of similar tools in open source community
  - … so the journey began back in 2020

Tool statistics

Codec	# of decoder implementations	# of test suites	# of test vectors
AAC	2	2	73
AV1	9	3	275
H.264	24	2	204
H.265	**27**	**4**	220
H.266	1	1	282
VP8	14	1	61
VP9	15	2	**311**
Total	**92**	**15**	**1426**

Linux community adoption

- Linux kernel for ARM: H.265 GStreamer decoder conformance
  - https://www.spinics.net/lists/arm-kernel/msg876595.html
- Linux kernel: VP9 decoder conformance
  - https://lkml.iu.edu/hypermail/linux/kernel/2109.1/07229.html
- Debian package (testing stage)
  - https://tracker.debian.org/pkg/fluster

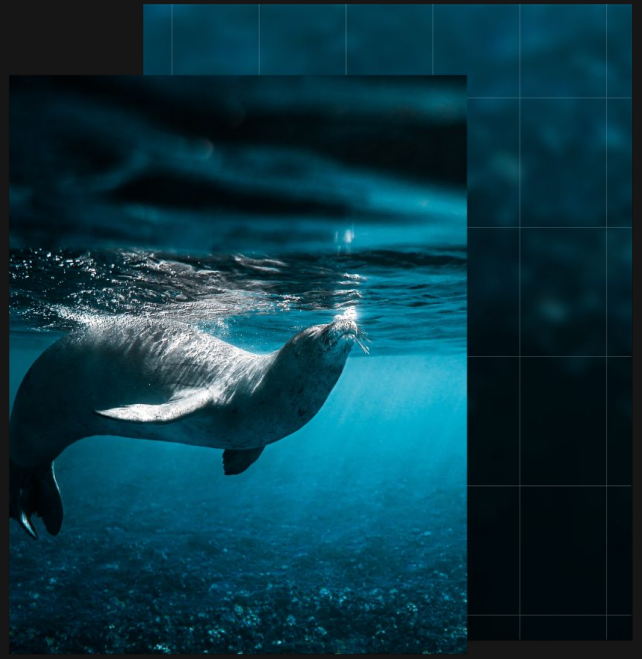
Github project info

License	Contributors	Commits	Active forks
LGPL-3.0	21	379	24

From:

Demo

list | download | run

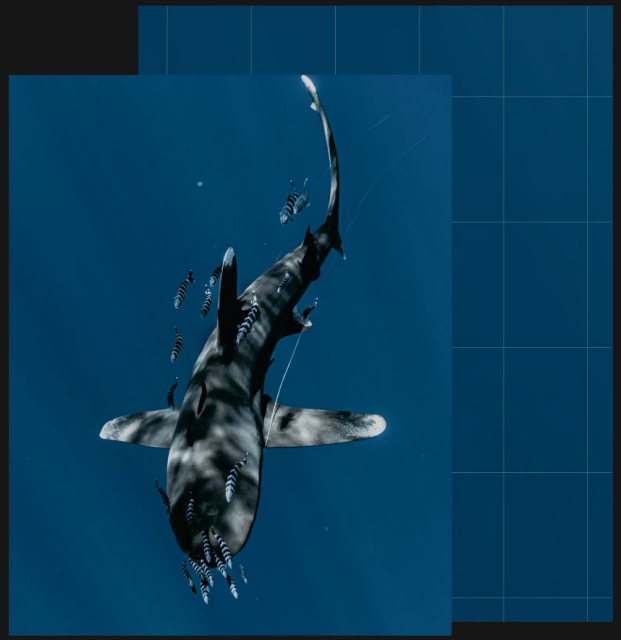


Demo

https://github.com/fluendo/fluster/blob/gstconf2023/gst-conf.demo.out

Tool extension

Decoder | Test suite



Decoder class abstraction level 1

```
class Decoder(ABC):
    """Base class for decoders"""
    name = ""
    codec = Codec.NONE
    hw_acceleration = False
    description = ""
    binary = ""

...

@abstractmethod
def decode(...)

...
def register_decoder(cls: Type[Decoder]) -> Type[Decoder]:
    """Register a new decoder implementation"""
```

GStreamer-based decoder creation

```
@register_decoder
class GStreamerD3d12H265Gst10Decoder(GStreamer10Video):
    '''GStreamer H.265 D3D12 decoder implementation for GStreamer 1.0'''
    codec = Codec.H265
    decoder_bin = ' h265parse ! d3d12h265dec '
    api = 'D3D12'
    hw_acceleration = True
```

FFmpeg-based decoder creation

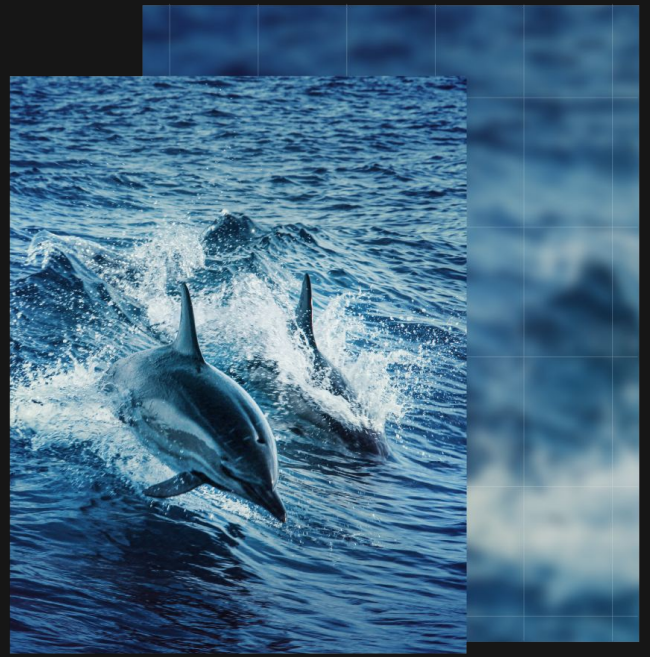
```
@register_decoder
class FFmpegVP8VaapiDecoder(FFmpegVaapiDecoder):
    '''FFmpeg VAAPI decoder for VP8'''
    codec = Codec.VP8

...

class FFmpegVaapiDecoder(FFmpegDecoder):
    '''Generic class for FFmpeg VAAPI decoder'''
    hw_acceleration = True
    api = 'VAAPI'
```

Test suite creation


```
{
  "name": "JCT-VC-HEVC_V1",
  "codec": "H.265",
  "description": "JCT-VC HEVC version 1",
  "test_vectors": [
    {
      "name": "AMP_A_Samsung_7",
      "source":
"https://www.itu.int/wftp3/av-arch/jctvc-site/bitstream_exchange/draft_conformance/HEVC_v1/AMP_A_Samsung_7.zip",
      "source_checksum": "0c3edda6a77b4dad7ea70fd24b0b6270",
      "input_file": "AMP_A_Samsung_7.bin",
      "output_format": "yuv420p",
      "result": "93a5875d58072db5539c04e1e943ed9d"
    },
```

Future steps

Pypi package | External test suites and decoders | WebCodecs | GStreamer CI | Performance improvement



Pypi package

- Github release v 0.1.0 (.zip)
- Pypi package creation
  - Encapsulate reference decoder build process (makefile-based) to native python code
  - Refactor path handling to accommodate for user-space installation without administrator rights

External test suites and decoders

- Be able to define external, closed-source test suites for patented codecs
- Be able to define external decoders

WebCodecs

- Private prototype has been implemented to check the conformance of chromium browser full H.264 decoding stack. Fluster for WebCodecs would be a good solution.
- Chromium Hardware H.264 decoder has a very bad result for the conformance test

chromium / media / gpu / **h264_decoder.cc**

Code | Blame | 1769 lines (1519 loc) · 60.7 KB

```
1123  bool H264Decoder::ProcessSPS(int sps_id, bool* need_new_buffers) {
1130    *need_new_buffers = false;
1131
1132    if (sps->frame_mbs_only_flag == 0) {
1133      DVLOG(1) << "frame_mbs_only_flag != 1 not supported";
1134      return false;
1135    }
1136
```

GStreamer CI integration

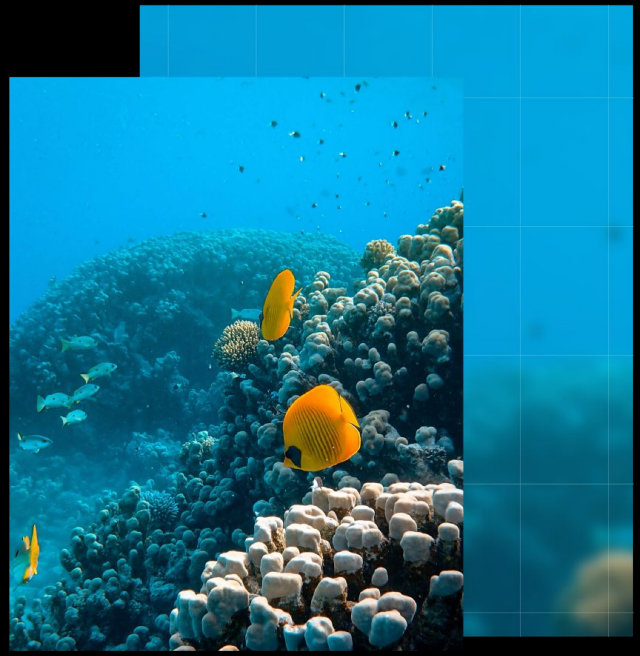
- Early-stage idea
  - Effort estimation
  - Reach consensus with community

Performance improvement

- MD5 checksum calculation
  - Currently: writing raw output streams to disk, calculating and discarding
  - Improvement: on-the-fly calculation on output of decoder element
  - For GStreamer-based decoder using element `videocodectestsink`
  - For FFmpeg-based decoder using argument `-f md5`

Question time

Thank you!



References

- Github repository: https://github.com/fluendo/fluster