Discovering key real-world use cases for benchmark design

2024-10-16

Designing a good benchmark test set

- Relevant: The tests reflects real-world production usage
  - If a vendor's product excels in the benchmark test set, the product will likely introduce business benefit when deployed in production.
- Reproducible: Can be carried out by anyone from any organization.
  - Open source both data and code. Moderate size, can be copied around easily.
  - Does not require extraordinary compute resources and/or human expertise (targeting laptop grade computer).
- Scalable: The tests delivers meaningful results when run on a broad spectrum of infrastructures used in real-world production.
- Understandable and Acceptance:
  - The test results are reasonable to be understood by a practitioner in the art* but still meets robust scientific standards, ex. linear, orthogonal, monotonic etc.
  - Venders and studios are willing to use it (political good will)

Designing a good benchmark test set

- Relevant: The tests reflects real-world production usage
  - If a vendor's product excels in the benchmark test set, the product will likely introduce business benefit when deployed in production.
- Reproducible: Can be carried out by anyone from any organization.
  - Open source both data and code. Moderate size, can be copied around easily.
  - Does not require extraordinary compute resources and/or human expertise (targeting laptop grade computer).
- Scalable: The tests delivers meaningful results when run on a broad spectrum of infrastructures used in real-world production.
- Understandable and Acceptance:
  - The test results are reasonable to be understood by a practitioner in the art* but still meets robust scientific standards, ex. linear, orthogonal, monotonic etc.
  - Venders and studios are willing to use it (political good will)

Designing relevant EXR tests

- Restrict the scope - we can not cover all possible prod cases (in one semester?).
  - Half linear RGB only. No fp32/int32, no luminance separated (YUV/Lab etc), no Deep.
  - We only targeting two core metrics in this round: compression ratio on disk and reading/decoding/decompression throughput. We ignore writing/encoding throughput.
- What are the common patterns of EXR API calls in real-world prod?
  - Kimball has mentioned “compositing-based workflow do more scanline access while rendering-based workflow do more tile based access”.
  - This need to be much detailed from here – any further info/evidence?
- What are the approximate combination/importance of these patterns?
- Translate these use case patterns into code that carries out EXR API calls in specific sequences.

How to answer these questions?

- In this stage we would like to avoid vendor/studio questionnaires.
  - It could be done, and has been done successfully, but let's think about it in the next iteration.
- EXR has a special trait - API implementation *is the definition* of the file format (quote Florian Kainz).
  - The existing API defines the total possibility of business use cases. We can enumerate the combinations.
  - However, not all provided (possible) functionalities and combinations are used equally in prod context - importance varies in the real world.
- Can we infer EXR usage patterns by looking at how EXR's consumer software interacts with it?
  - Blender's source code is visible (Aras P wrote the blender EXR connectors…I might just ask)
  - For closed source software, some public info might also be available to ASWF projects. (Vendor neutral requirement – no insider knowledge should be used.)
  - Pending technical survey
- Any better ideas?

Karen Ruggles

https://www.desales.edu/academics/our-faculty/faculty-profiles/karen-ruggles

- One senior undergraduate students (Chris) w. One single semester capstone/seminar project on OpenEXR (Jan 2025 - April 2025); see slack.
- Chris to write a one pager research proposal by end of Nov.
  - Research Question
  - Prior Arts
  - Methodology
  - Goal/success criteria
- Open to ideas – if none I'll take him on the benchmarking designing project.