

REVIEWING PRODUCTION OPENEXR FILES ON THE WEB FOR ML

Max Grosse, Principal Software Engineer
DisneyResearch|Studios

PRODUCTION IMAGES?

- OpenEXR files
- High resolution, 1080p and up
- Extended / High Dynamic Range
- 16-bit or 32-bit floating point
- Not only final composed color data
 - Alpha masks
 - Feature buffers from the renderer
 - Error maps
 - ...

EXAMPLE USE CASE: DENOISER

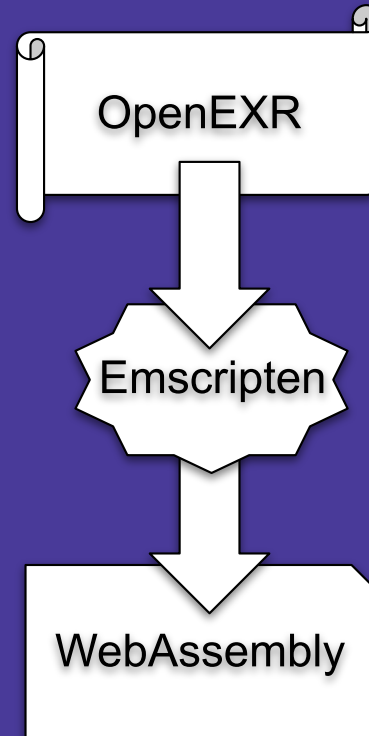
- *Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings*, Rousselle et al, ACM SIGGRAPH 2017
- Deep learning for denoising rendered images
- Suitable for production

INTRODUCING: JERI.IO

- The *JavaScript Extended-Range Image Viewer*
- HTTP server directly serves OpenEXR images with config
- Pure client-side app to decode EXR directly in the browser
- Thus, client receives original, unmodified pixels
- Like a very fancy, configurable, `` tag for EXRs

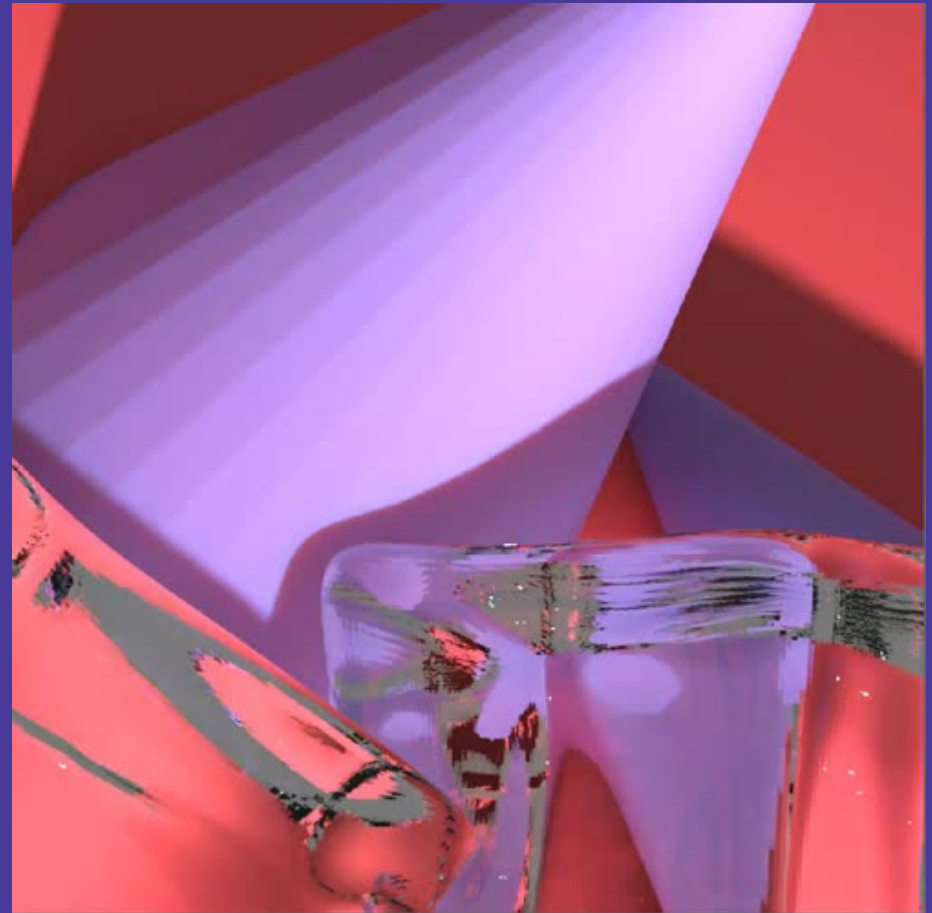
EXR ON THE WEB

- OpenEXR compiled to WebAssembly
- Emscripten did the job
- Speed is fine for our use cases, in particular when caching
- Now we have EXRs in the browser!



USABLE EXR ON THE WEB

- Want to be able to change gamma, exposure
- Want to visualize error metrics and compare inputs
- Use WebGL for all image processing client-side
- Done in TypeScript with support for React.js



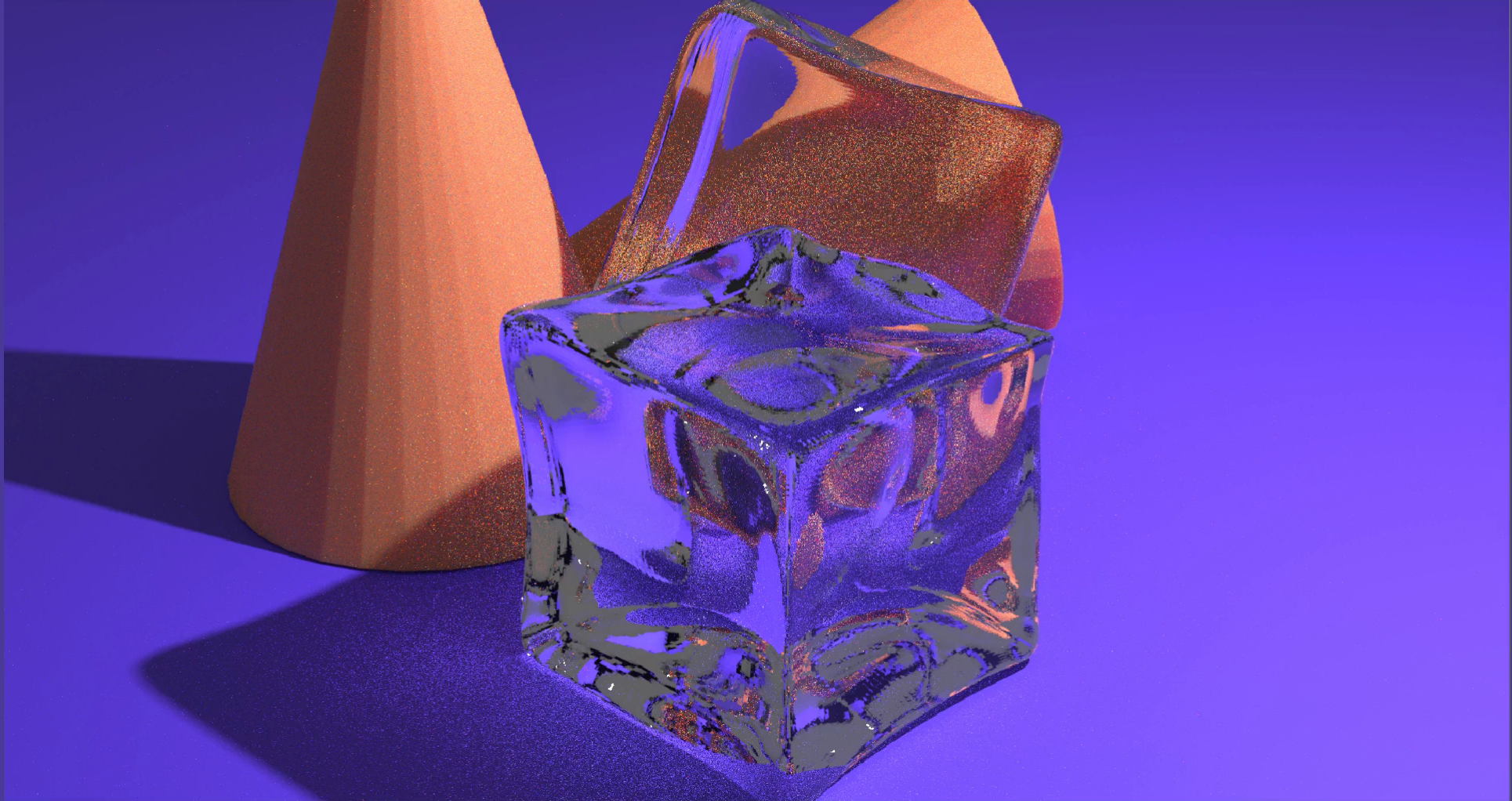
CONFIGURATION

- JSON to configure viewer
- Images to load and from where
- How to group these images
- Which metrics, images to compare

```
const config = {
  title: 'root',
  children: [
    { title: 'Improved', image: 'glass.exr' },
    { title: 'Reference', image: 'glass-ref.exr' },
    {
      title: 'SSIM',
      lossMap: {
        'function': 'SSIM',
        imageA: 'glass.exr',
        imageB: 'glass-ref.exr'
      }
    }
  ],
};
```

JERI DEMO

Ice Blue¹ Ice Red²
Input Denoised Reference SSIM



[HTTPS://JERI.IO](https://jeri.io)

RELEASED AS OPEN SOURCE

THANK YOU

max.grosse@disneyresearch.com