Developing a High Level Preservation Strategy for Virtual Reality Artworks

Tom Ensom & Jack McConchie

No Time To Wait 3, BFI, London
23 October 2018

Supported by Carla Rapoport, Lumen Art Projects Ltd.
Introduction

- Moving from exploratory research to strategy for acquisition and preservation
- First stages of project have explored:
  - The core components of VR systems (software and hardware) and their relationships
  - Production and playback of 360 video VR works
  - Production and playback of real-time 3D VR works
  - Initial exploration of preservation strategies for both the above
VR System Overview: **Hardware**

- **Computer System**
  - Graphics Processing Unit (GPU)

- **Headset**
  - Head Tracking System
  - Displays & Lenses

- **User**

- **Controller**

- **Positional Tracking System**
VR System Overview:

Software

Computer System

Operating System

VR Application

VR Runtime

Graphics Processing Unit (GPU)

Controller

Headset / HMD

Positional Tracking System
VR Platform Fragmentation

Image credit: Khronos Group
https://www.khronos.org/openxr/
360 Video: Production

Capture: Monoscopic 360 with dual fisheye lenses

Image credit: http://theta360.guide/plugin-guide/fisheye/
360 Video: Production

Capture: Stereoscopic 360 with multiple lenses

Image credit: https://www.mysterybox.us/blog/2017/1/31/shooting-360-vr-with-gopro-odyssey-and-google-jump-vr
360 Video: Production

Capture: Stereoscopic 360 with multiple lenses
360 Video: **Production**

Common file characteristics

<table>
<thead>
<tr>
<th>Containers</th>
<th>MP4, MKV</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Aspect Ratio</strong></td>
<td>Equirectangular 2:1</td>
</tr>
<tr>
<td><strong>Frame rates</strong></td>
<td>Commonly 60 fps</td>
</tr>
<tr>
<td><strong>Resolution</strong></td>
<td>4096 x 2048- equivalent to 1024 FOV</td>
</tr>
<tr>
<td><strong>Compression</strong></td>
<td>H264, H265</td>
</tr>
</tbody>
</table>
360 Video: **Playback**

Equirectangular
360 Video: **Playback**

*Cube Map*
360 Video: **Playback**

Pyramid

Image credit: https://code.fb.com/virtual-reality/next-generation-video-encoding-techniques-for-360-video-and-vr/
360 Video: **Playback**

Top bottom stereoscopic 360 equirectangular projection

Identifying string: “3dv” or “_tb”

examplemedia_3dv_.mp4

Left right stereoscopic 360 equirectangular projection

Identifying string: “3dh” or “_lr”

examplemedia_3dh_.mp4
360 Audio: **Production**

“A” Format
Raw audio

“B” Format
Amplitude and spatial information
# 360 Audio: Production

<table>
<thead>
<tr>
<th>Audio encoding</th>
<th>WAV, AIFF, AAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambisonic “order” or number of channels</td>
<td>4, 9, 16</td>
</tr>
<tr>
<td>Audio channel ordering</td>
<td>Furse-Malham, ACN</td>
</tr>
<tr>
<td>Audio channel spherical normalisation</td>
<td>SN3D, maxN, N3D.</td>
</tr>
</tbody>
</table>
360 Audio: **Playback**

- Direct sound
- Filtered sound through head delayed
- Diffracted sound around head delayed
Real-Time 3D: Production

- **Assets** created using various proprietary and open-source tools and brought together in an engine

- Assets typically portable with open file format potential in some cases:
  - **3D models**: meshes and materials
  - **Textures**: texture maps (raster images)
  - **Sounds**: Ogg Vorbis and WAV

- **glTF** as open standard for archiving?
3D Modelling: Meshes and UVs

Mesh

UV Map

blender

ZBrush

MAYA
3D Modelling: Materials and Textures

Diffuse map: color

Normal map: surface features

Metallic map: metallic (white) non metallic (black)

Roughness map: roughness (white) smoothness (black)
Real-Time 3D: Production

- **Engines** typically non-portable and proprietary/licenced
- Engines integrate assets and engine features in scenes/levels and include:
  - Lighting
  - Material shaders
  - Physics
  - Visual or typed scripting
  - Sound engine
  - Plugins and extensions
Real-Time 3D: **Playback**

- Built as executable software package
- Usually consists of executable file(s) and packaged assets
- Can be executed in a specific technical environment which might include:
  - VR Runtimes e.g. SteamVR, Oculus Runtime, OpenXR, WebVR etc.
  - Graphics API e.g. DirectX, Metal, OpenGL, Vulkan etc.
  - Graphics Processing Unit (GPU) and driver

Vulkan is an open standard for cross-platform 3D graphics which succeeds OpenGL
Rendering Pipeline

Geometry
Vertex Shaders

Vertex Processing → Rasterization → Pixel / Fragment Processing → Post Processing → Rendered Frame

Textures
Fragment / Pixel Shaders
VR Rendering Pipeline

- **Vertex Processing**
  - Geometry
  - Vertex Shaders

- **Rasterization**
  - Textures
  - Fragment / Pixel Shaders

- **Pixel / Fragment Processing**

- **Post Processing**
  - + Lens Distortion
  - + Timewarp / Spacewarp

- **Rendered Frame**
Lens Distortion

- Lenses used in headset to achieve wide field of view at close range
- Distortion must be corrected for in the frames send to headset
- Usually carried out as post processing
- Process specified by VR runtime

Timewarp & Spacewarp

- Predictive interpolation of frames
- Distorts previously generated frame based on movement of user and scene
- Low cost tricks to maintain high framerate and low latency
- Process specified by VR runtime

Image credit: Neo222
https://xinreality.com/wiki/Asynchronous_Spacewarp
An Open Standard for VR Runtimes?

- The Khronos Group is a consortium of industry partners who develop open standards for 3D graphics
- **OpenXR** is working group attempting to develop a VR runtime standard
- All the big players in the VR industry are involved
- Like glTF and Vulkan, could also benefit preservation efforts... if it is adopted

Image credit: Khronos Group [https://www.khronos.org/openxr/]
Next Steps: Exploring Preservation Strategies

<table>
<thead>
<tr>
<th></th>
<th>General</th>
<th>360 Video</th>
<th>Real-Time 3D</th>
</tr>
</thead>
</table>
| **Migration** | ● Headset lens distortion algorithms | ● Moving between projection types  
● Variability in players | ● How reliable are the open 3D asset standards  
● Re-creating a 3D scene in different engines |
| **Emulation** | ● Emulating VR runtimes | ● Emulating 360 video players | ● Paravirtualization and passthrough for real-time 3D VR |
| **Documentation** | ● Field of view video capture  
● Accounts of experience | ● Documenting projection type | ● 360 video capture |
Contact

Tom Ensom
tom.ensom@tate.org.uk
@Tom_Ensom

Jack McConchie
jack.mcconchie@tate.org.uk
@DataPotatoes