Fulldome Content Development

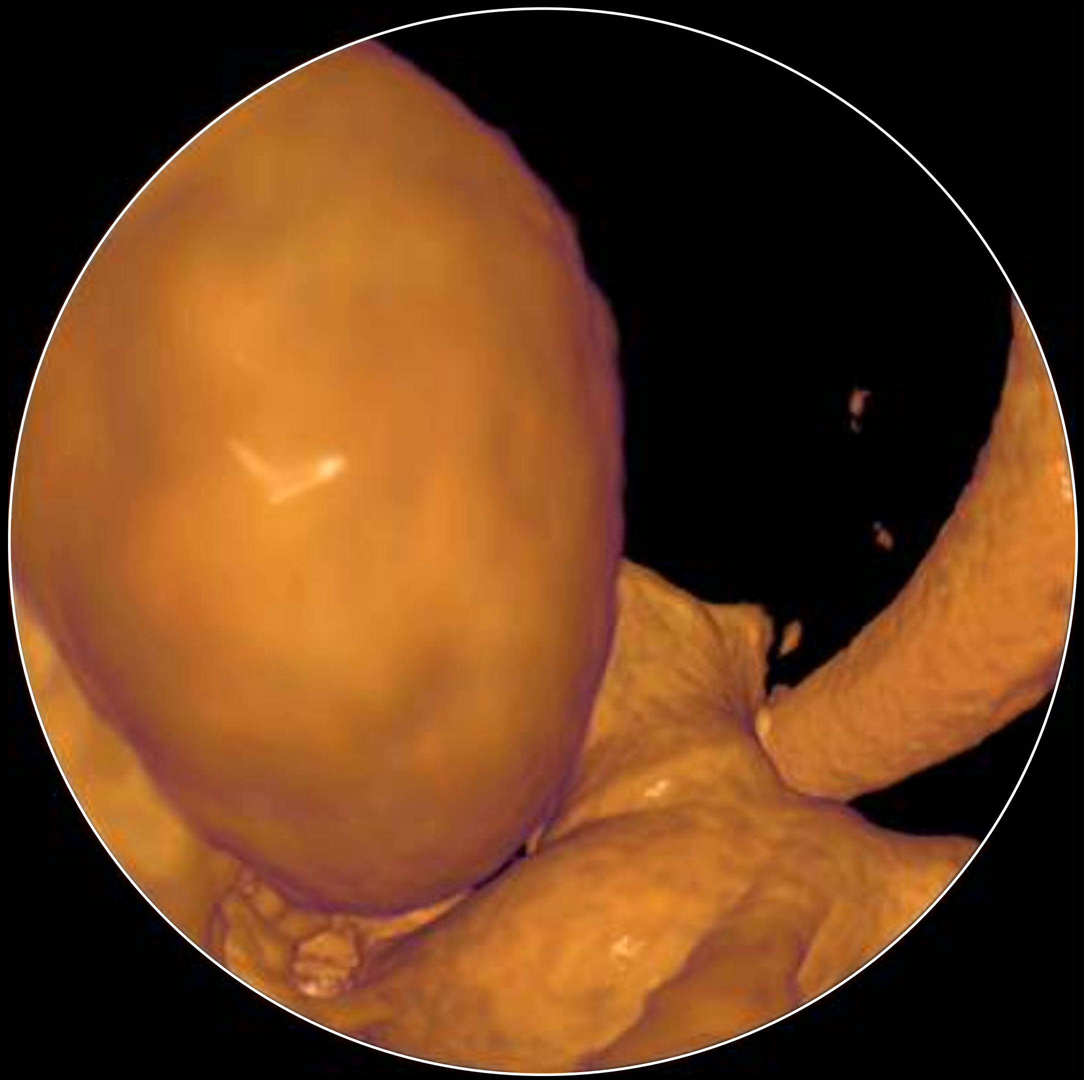
Everything you wanted to know, and more

Paul Bourke
EPICentre, UNSW

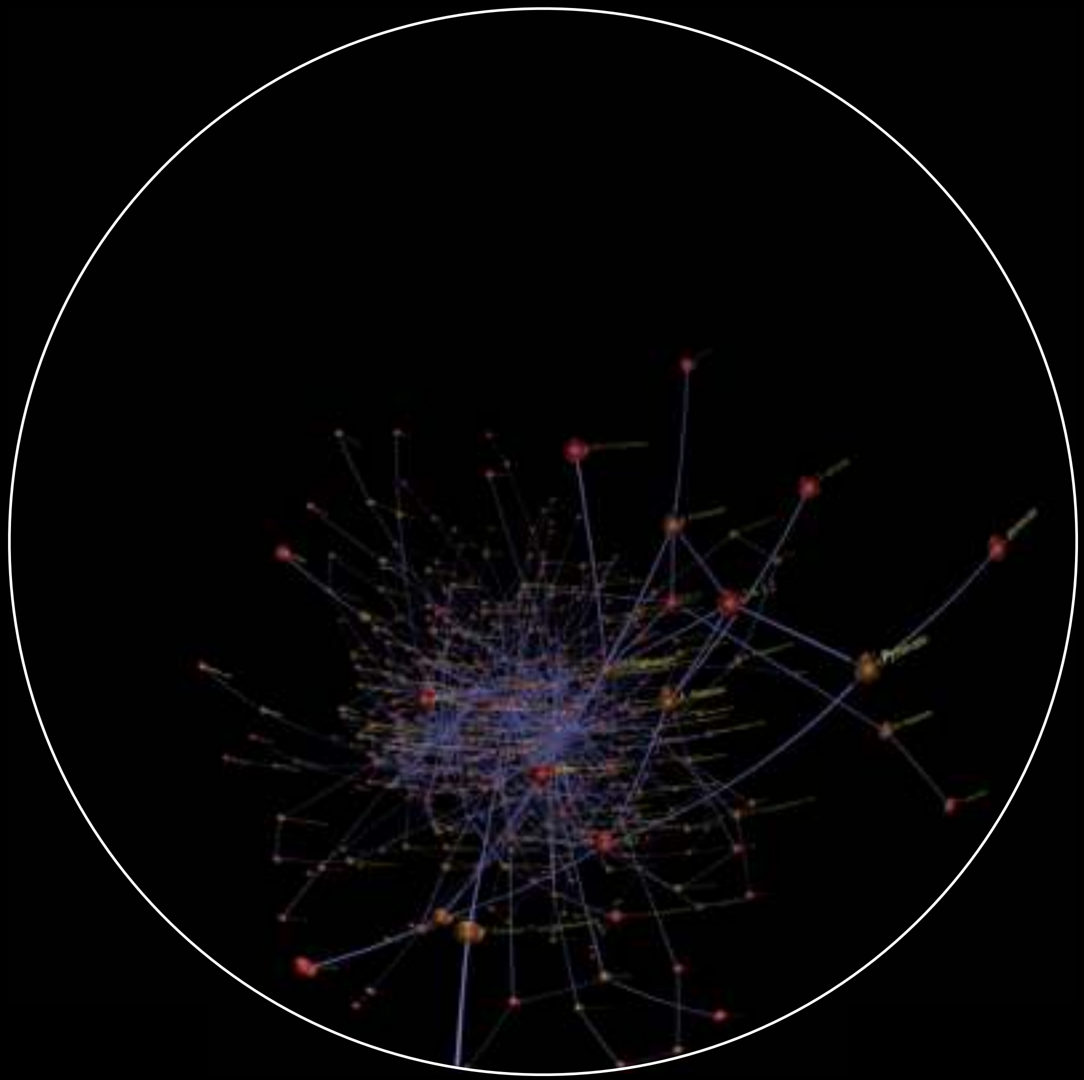
Contents

- Introduction, why I’m presenting on fulldome
- Projection theory
- Types of content: CG, photography, video, realtime
- Projection
- Considerations
- DomeLab standards
- Sample workflows for Protector point clouds and panoramas
- Questions and discussion

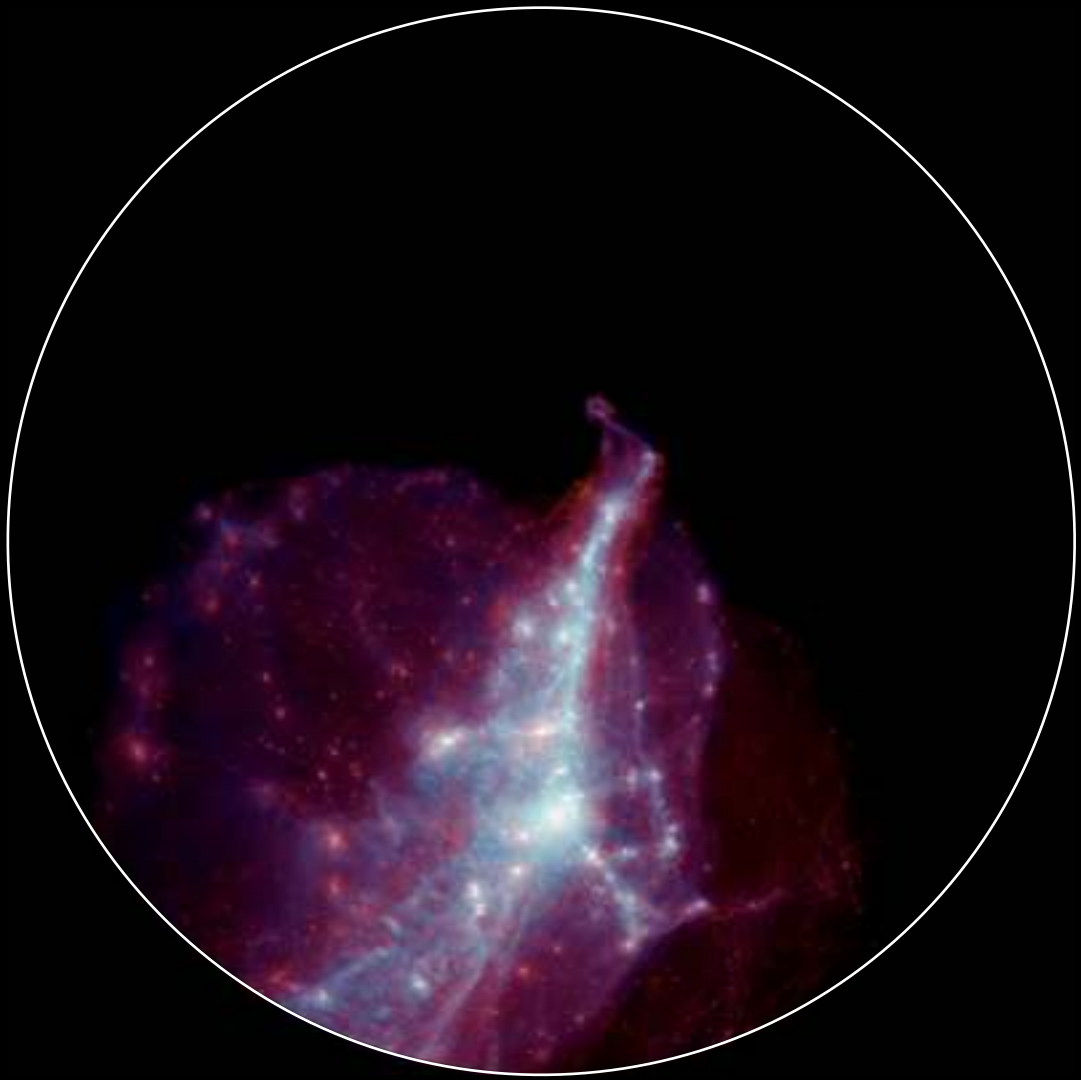
Introduction: Movies



Volumetric visualisation



Biological network visualisation

Introduction: Spherical mirror

Introduction: iDome

MMK Museum für Moderne Kunst Frankfurt am Main

Wollongong Science Centre

NTU, Singapore

Ngintaka, South Australia Museum

Introduction: Capture

LadyBug-3 & 5 camera

Red Scarlet + Sigma fisheye Lens

Canon 5K MK III + Canon 8-15mm fisheye lens

Lumix + Sigma fisheye lens

Introduction: Gaming


ASKAP "walk about"

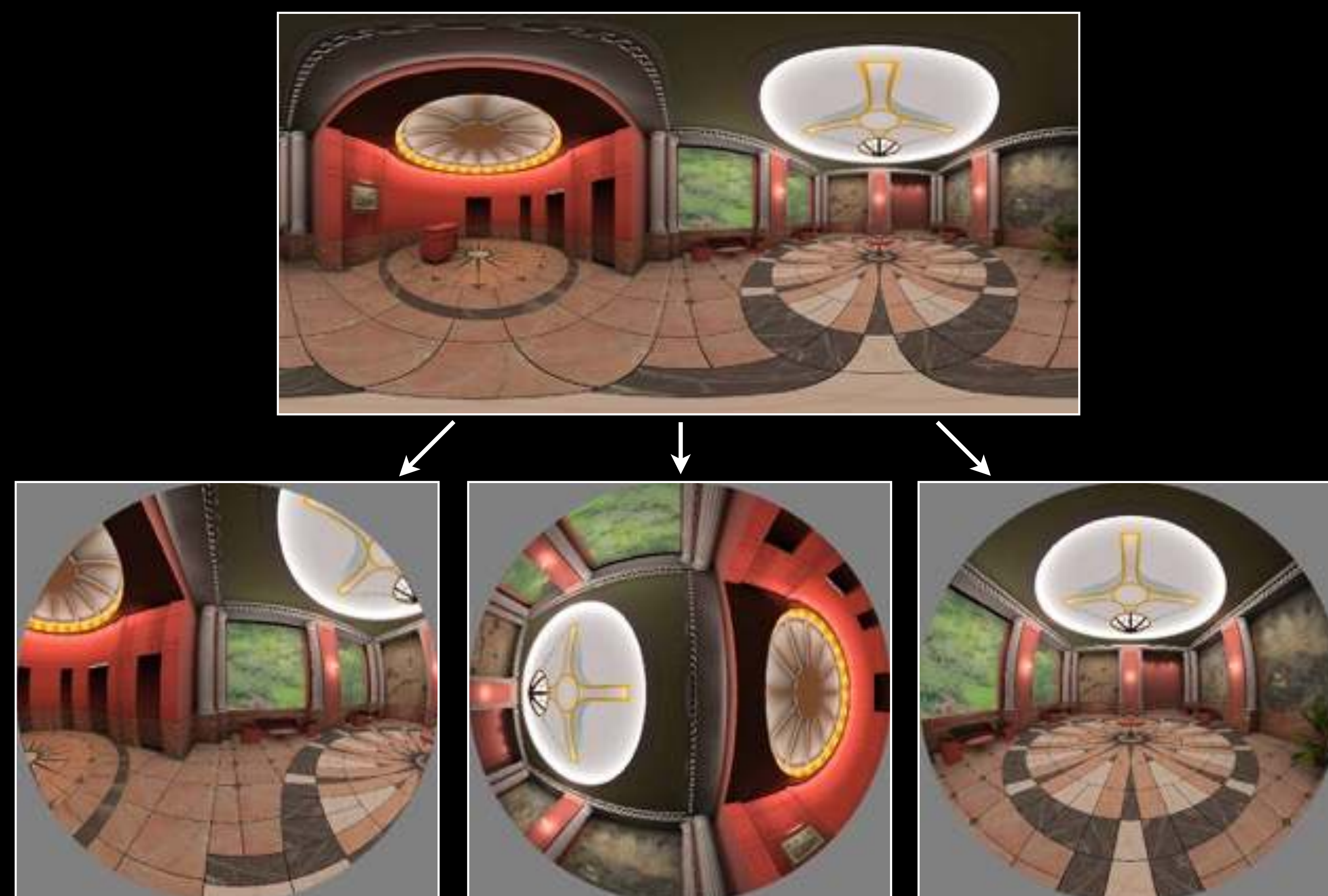

Mawsons Huts


Island, Unity 3D


Yo Frankie (Blender)

Introduction: Software

- cube2dome + dome2cube
- sphere2fish + fish2sphere
- pano2fish + fish2pano
- meshmapper (calibration)
- pbmesh (Unity and Vuo)
- warpplayer + VLCplayer
- offaxis fisheye
- shaders (various)



Projections

- Most familiar with rectangular frustum perspective projections.

- Cannot create a180 degree perspective projection, this is required to fill a dome.


FOV: 110 degrees


FOV: 130 degrees


FOV: 150 degrees

- Most common projections encounter in the dome industry are: cube maps, spherical (equirectangular) and fisheye.

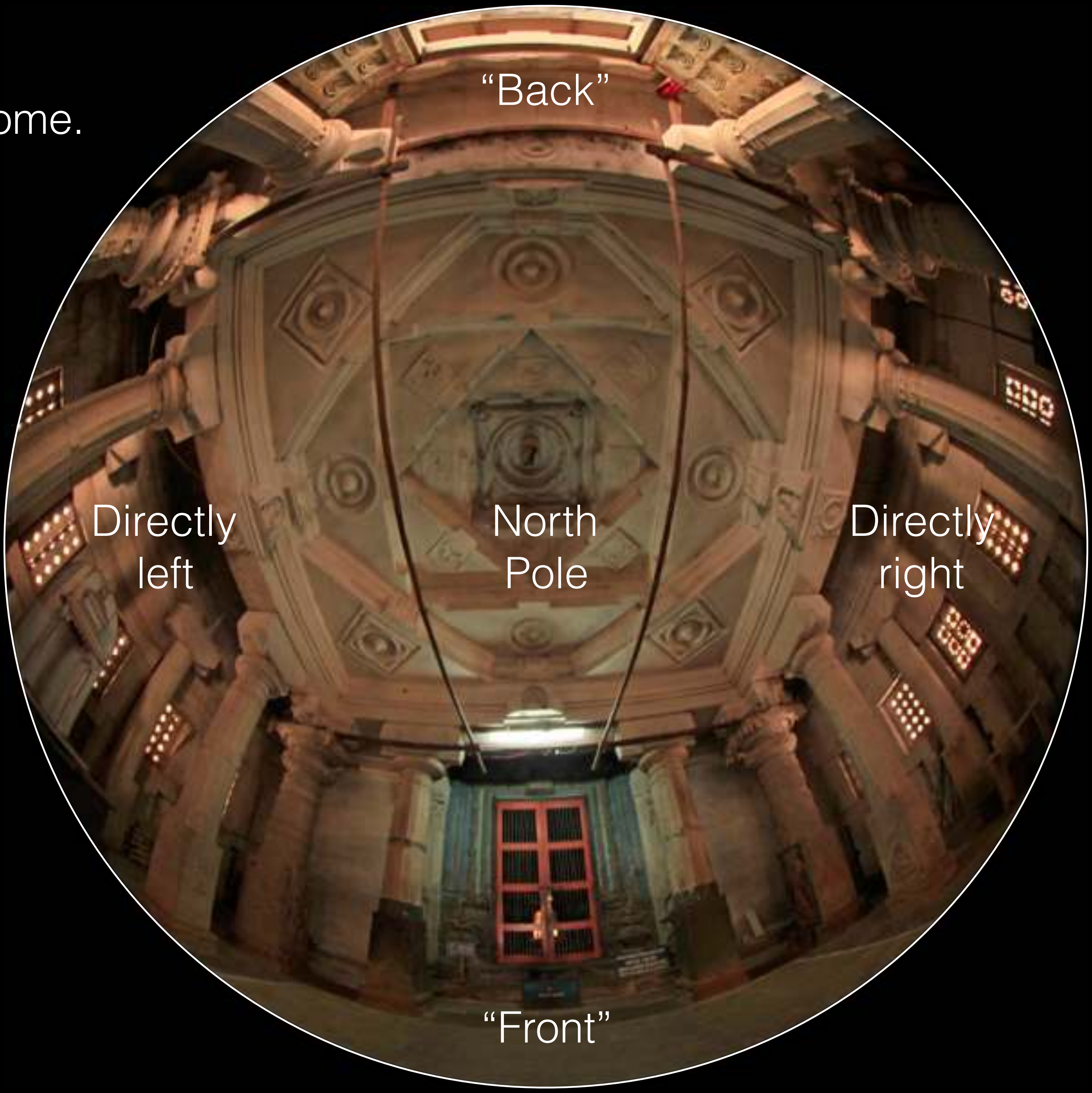
- These are not “distorted”, are all precisely defined methods of mapping a 3D scene to an image plane.

Fisheye

- Captures half the world.
- The most natural image for a dome.
- Not limited to 180 degrees, can be more, or less.
- See later for topics:

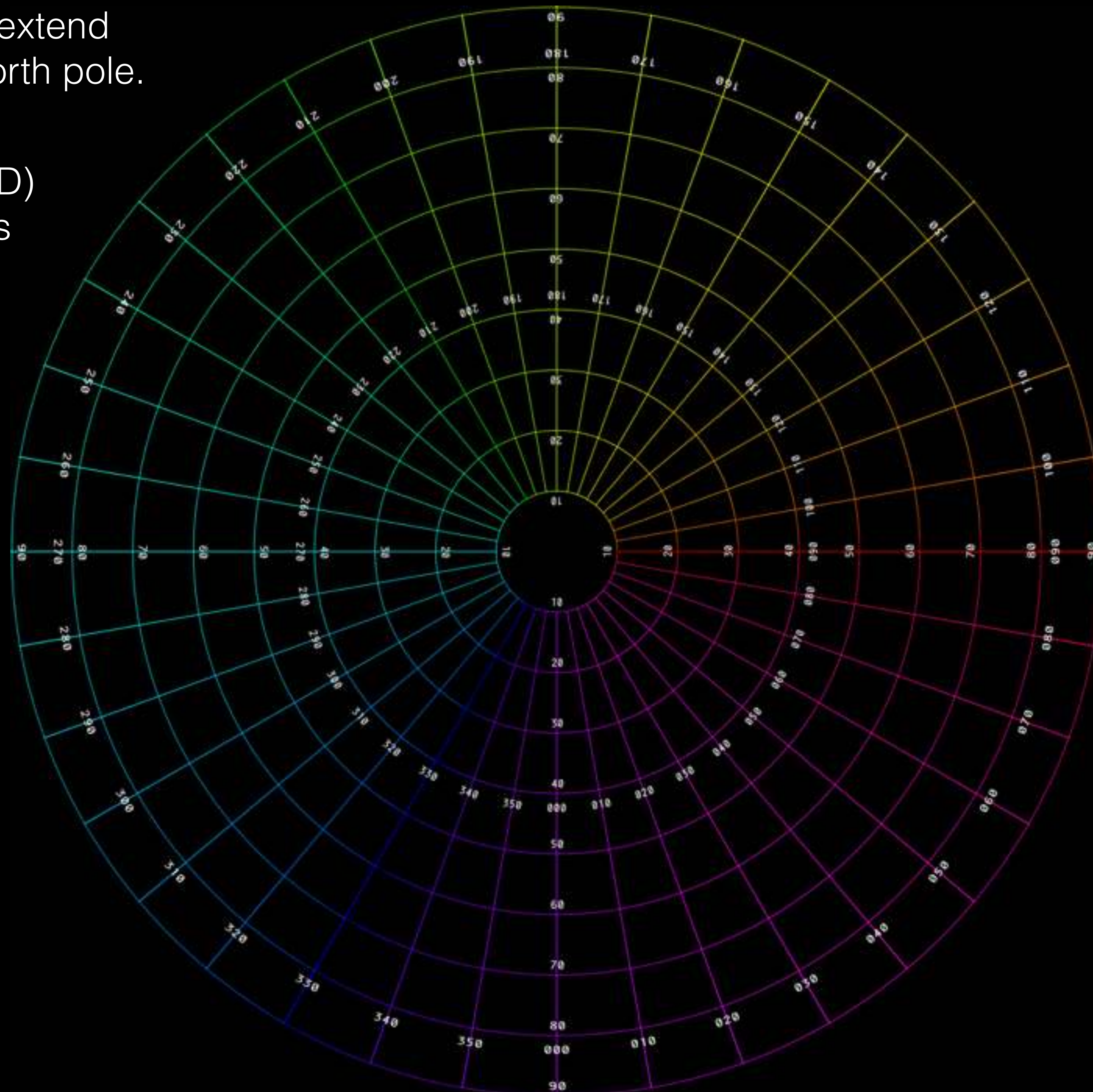
  - dome orientation

  - omnidirectional



- Lines of longitude extend radially from the north pole.

- Lines of latitude (3D) create equal radius lines in the fisheye (2D)



Cube maps

- Captures the whole world.
- Projection of the scene onto the surface of a cube.
- Each face is a 90 degree FOV vertically and horizontally.
- Often shown with the cube folded out.
- See later with regard to realtime generation of fisheye.

Spherical projection

- Captures the whole world.
- Most commonly used texture map for a sphere.

North pole

South pole

The right way of thinking

- Whatever the display surface one should consider the viewer and their relationship to the display surface.

- Where an object appears on the display depends on the viewers eye(s), the display surface and the location of the object in 3D space.

- This way of thinking (window on the world) is the “only” way to correctly understand and problem solve for any interesting display type.

- Includes
  - stereoscopic displays
  - multiple planar displays, stereoscopic or not
  - head mounted displays
  - cylindrical shaped displays
  - hemispherical shaped displays
  - projection mapping
  - … everything else

- Thinking this way also informs what field of view is required for projection, computer generation and capture (photographic or video).

Content types: CG

- 3DStudioMax, Maya, Cinema4D …
  + Data visualisation

- Pretty much all rendering packages today have a fisheye lens type or a third party plugin.

- Fallback position is cube maps which only requires
  - 90 degree perspective camera
  - scriptable or multiple camera rig

Content types: Photography

- There is a distinction (in some circles) between a wide angle fisheye and a circular fisheye.

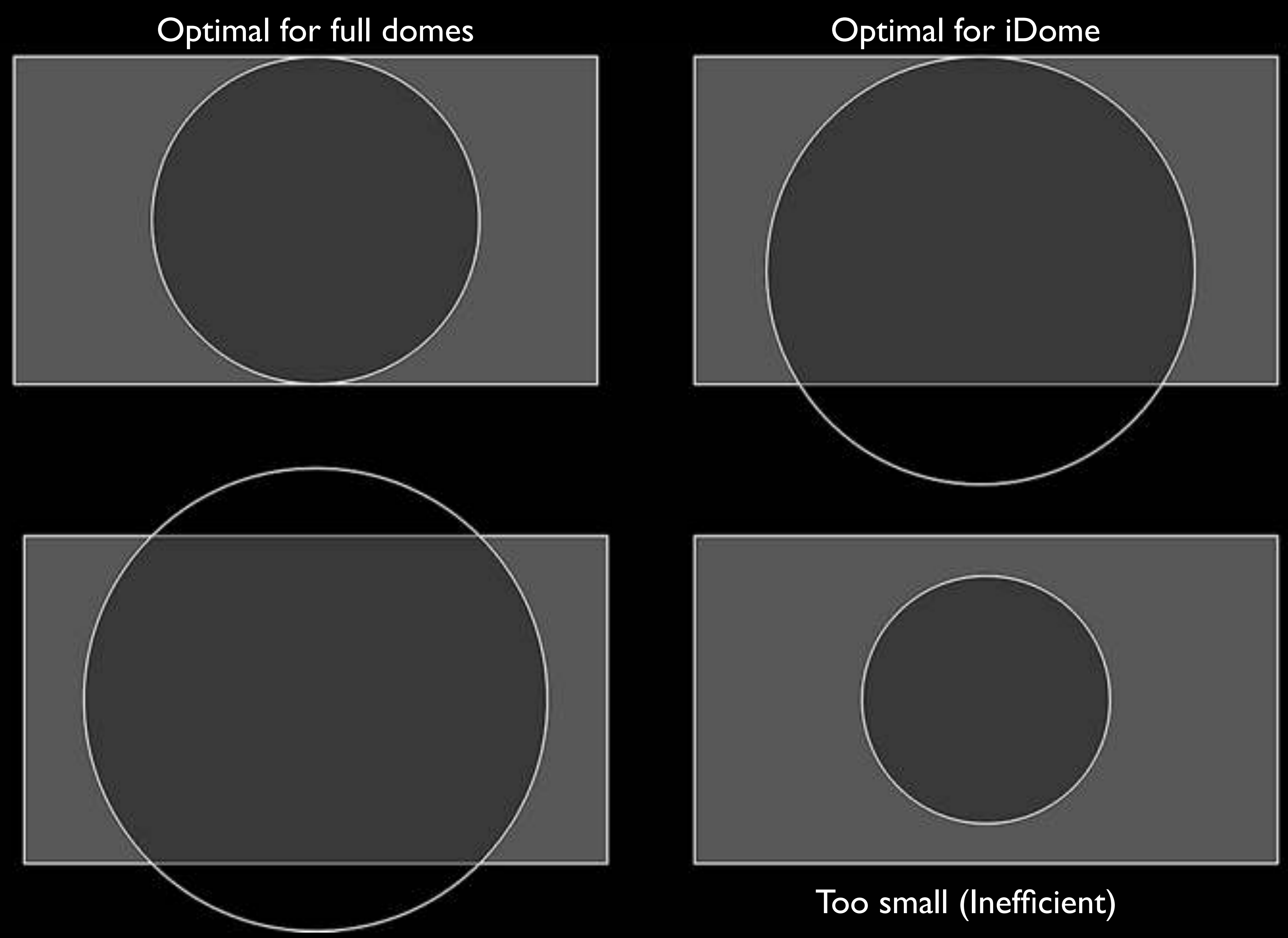
- Even a 170 degree wide angle fisheye (eg: GoPro lens) covers a very small part of a hemispherical dome.

170 degree wide angle fisheye

Circular fisheye

Realities of real lenses

- Need to consider the location and size of the fisheye circle on the camera sensor.
- Generally a match between sensor size (eg: full frame, APS-C, etc) and the lens.



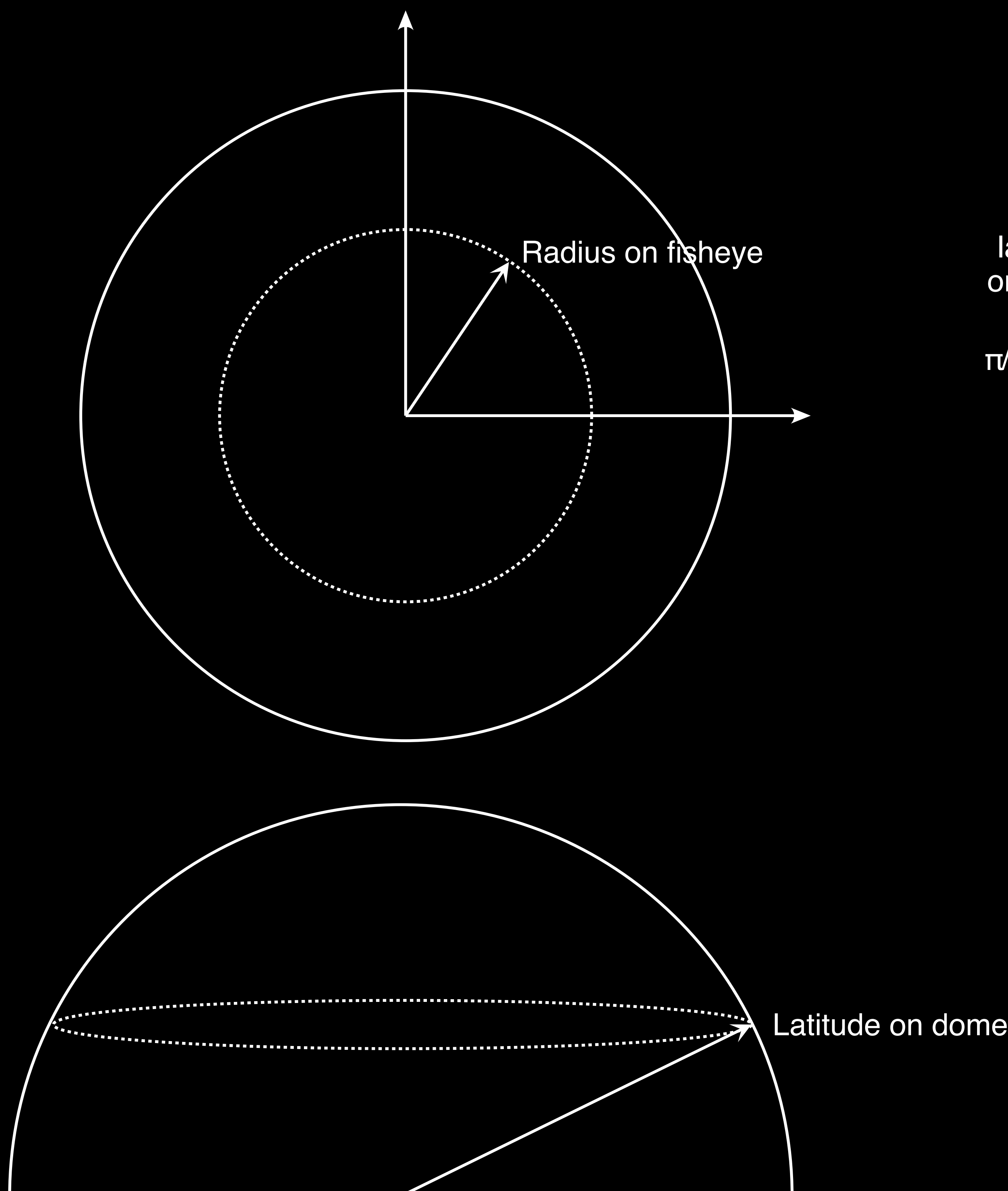
Example of a full frame fisheye on a 2/3 sensor

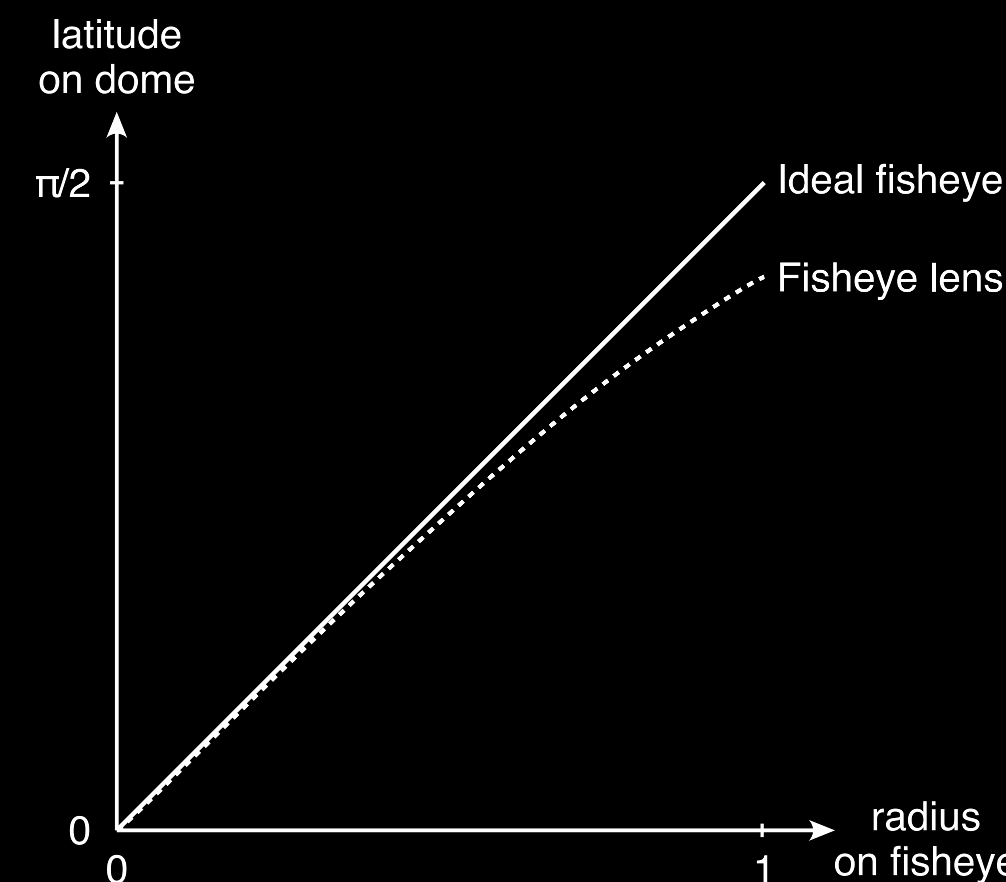
Example of a 2/3 fisheye on a full frame sensor

Ideal, 2/3 fisheye on a 2/3 sensor,
or full frame fisheye on full frame sensor

Nonlinear radius vs latitude

- Idealised fisheye projection has a linear relationship between radius on the fisheye image and latitude on the dome. Real lenses rarely do.



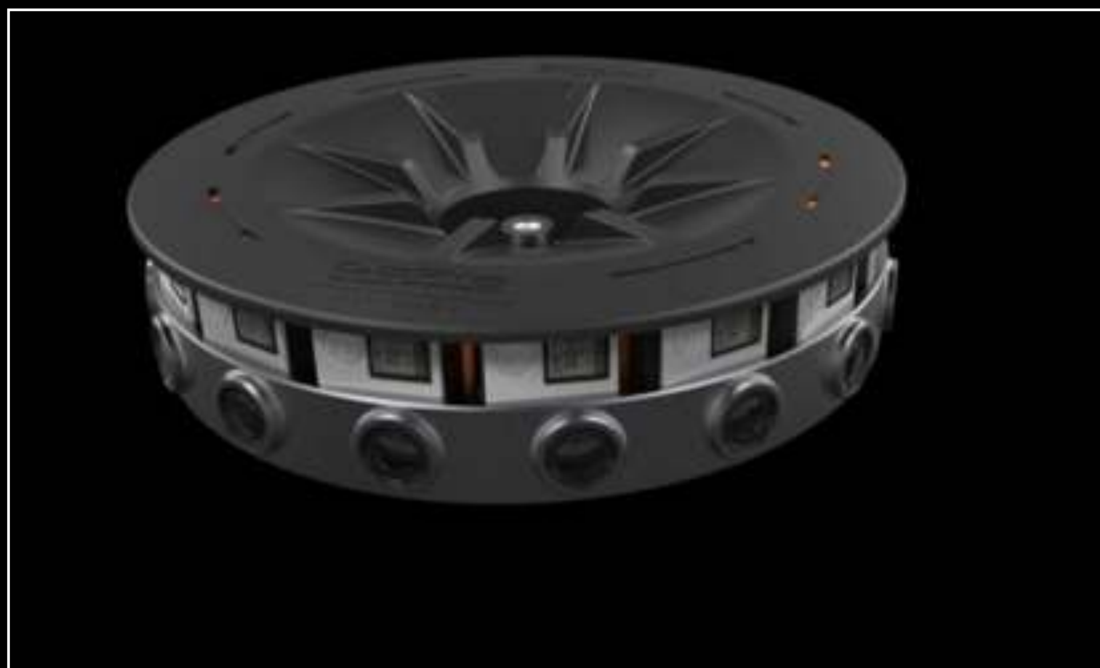


Content types: video

- Same comments regarding lenses apply here.
- Huge industry at the moment trying to satisfy the HMD market, although that requires full 360 capture which is a little harder.
- The problem with just using a fisheye lens with a video camera is that for fulldome you only end up using the height of the sensor.

Multiple camera rigs

- The issue with multi camera rigs is that there are fundamental parallax issues if the camera nodal points are not co-linear.



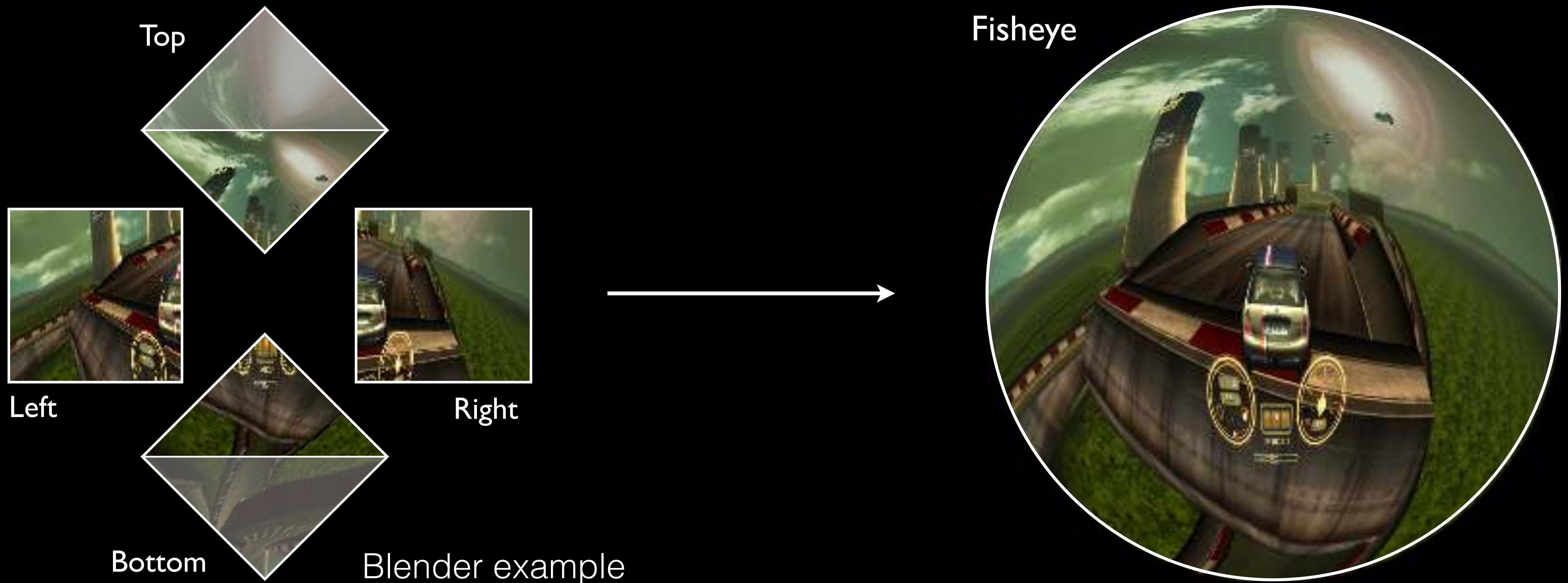


360 video

- Advantage is that one can extract a fisheye at the desired orientation in post.

Realtime

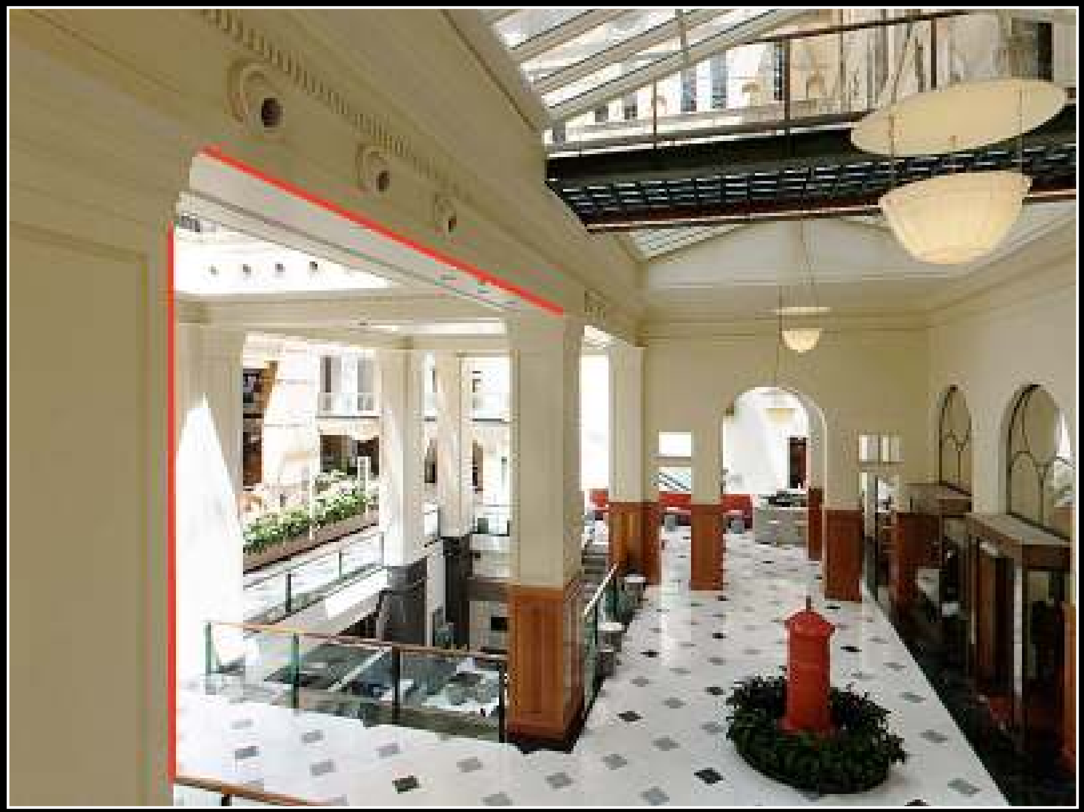
- Realtime APIs don't support fisheye directly.
- Two approaches
  - multi-pass rendered cube maps
  - vertex shader
- Each has relative merits, most implementations choose cube maps.
- Unity3D, Crystal Quest and Blender have proven fisheye generation.

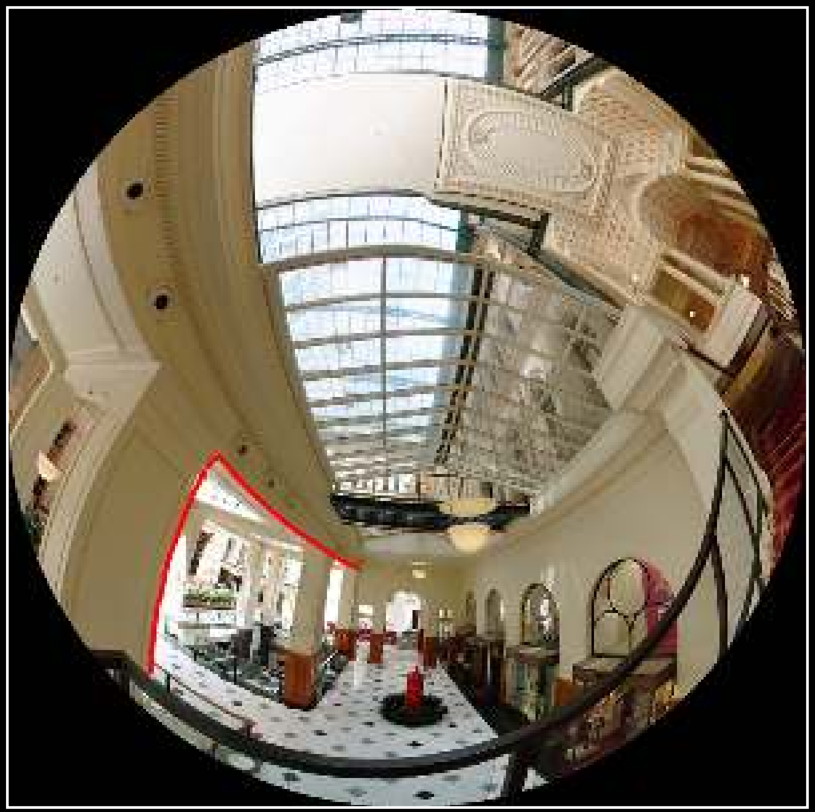


Blender example

Vertex shaders

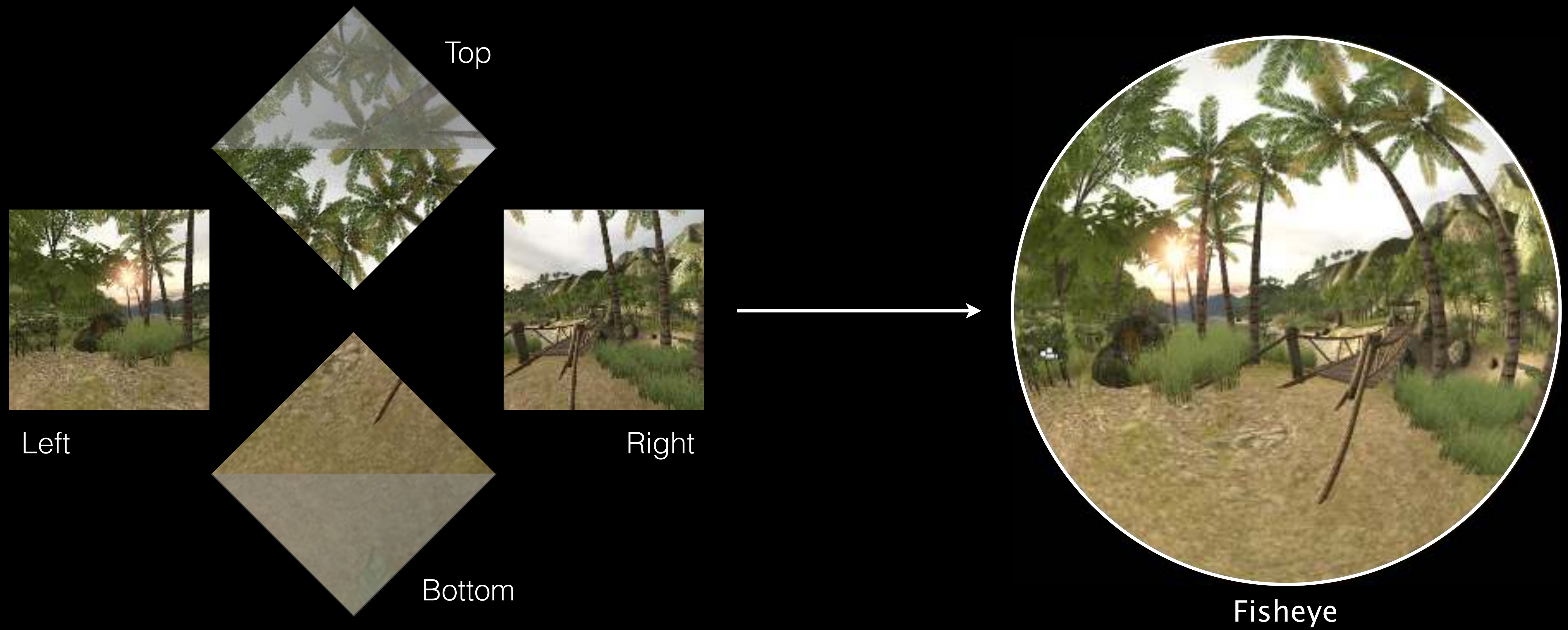
- Other approach is single pass using vertex shader.
- A cunning trick: modify the position of each vertex such that the result when viewer with an orthographic camera is a fisheye image.
- A straight line in a standard perspective projection only requires knowledge of the two end points. A straight line is not “straight” in a fisheye projection.
- The solution is to tessellate all the 3D geometry being drawn. The optimal algorithm to do this is not at all trivial.





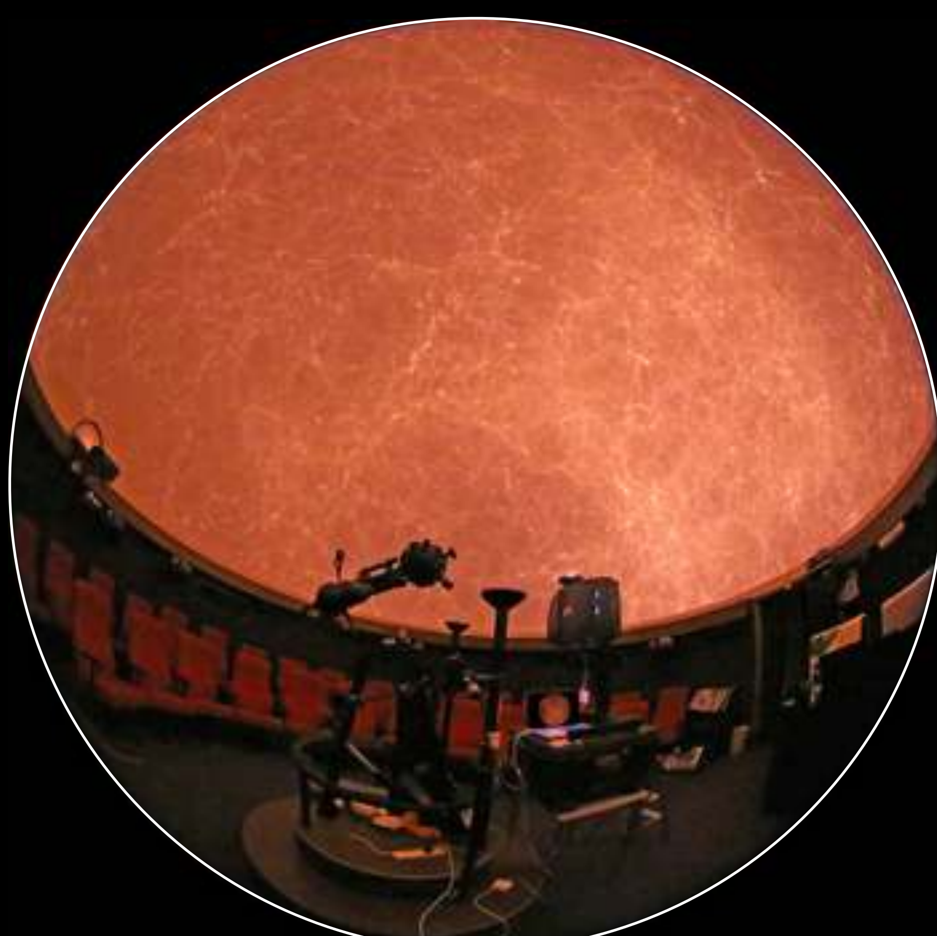
Unity

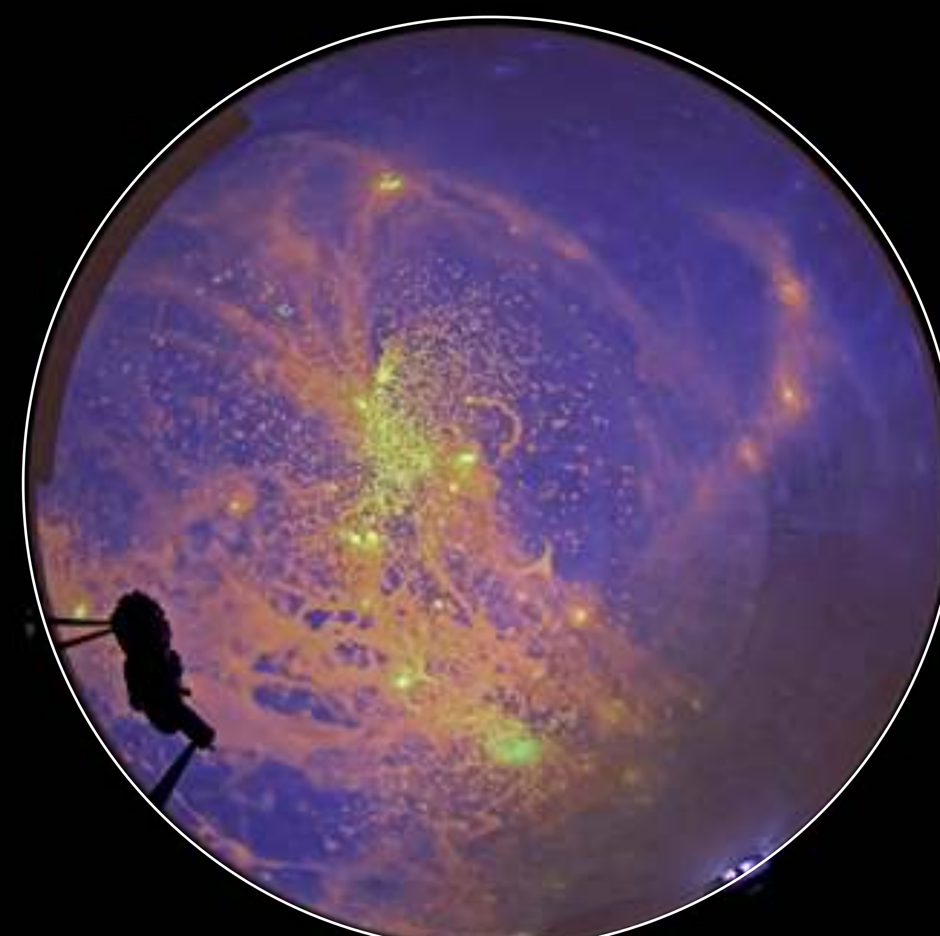
- Proposed solution for Unity is a 4 pass render to texture to create sufficient field of view.
- Apply to correctly crafted meshes to create a fisheye at a resolution suitable for the projection system being used.
- Typically each render texture would be 1/2 the final fisheye width, so 2K for DomeLab.



Digital projection - Single projectors

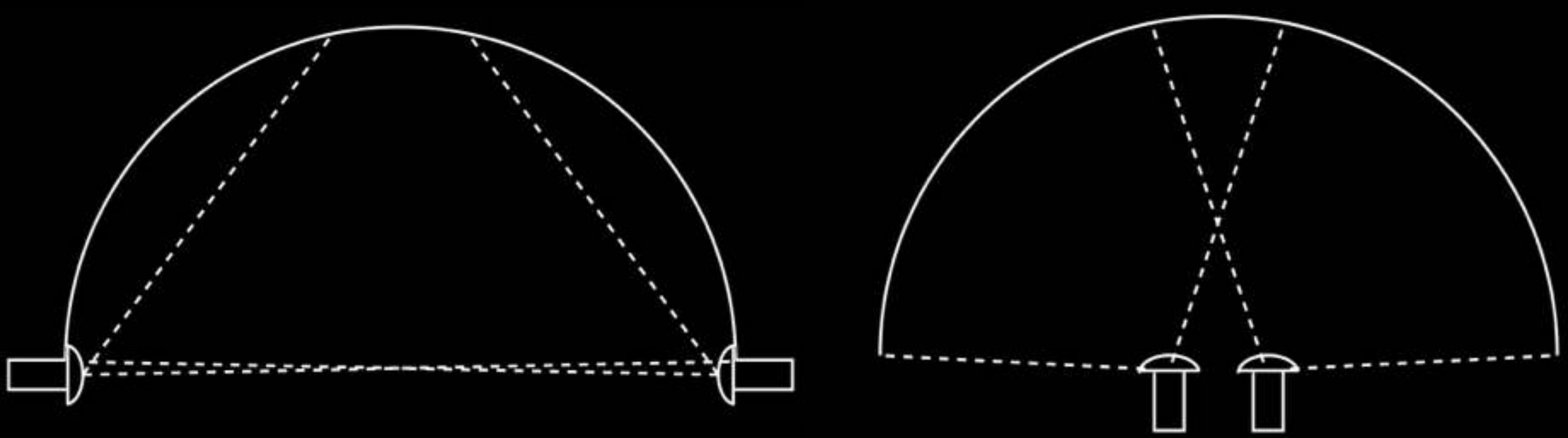
- There is a hierarchy of digital projection options for fulldome.
- 1. Simplest: Single projector and fisheye in the middle of the dome.
  - Main issue is the hardware is occupying best seats in the house.
- 2. If (1) is too expensive then a single projector and spherical mirror.
  - Lowest cost, hardware on rim of dome, complicated by warping required.





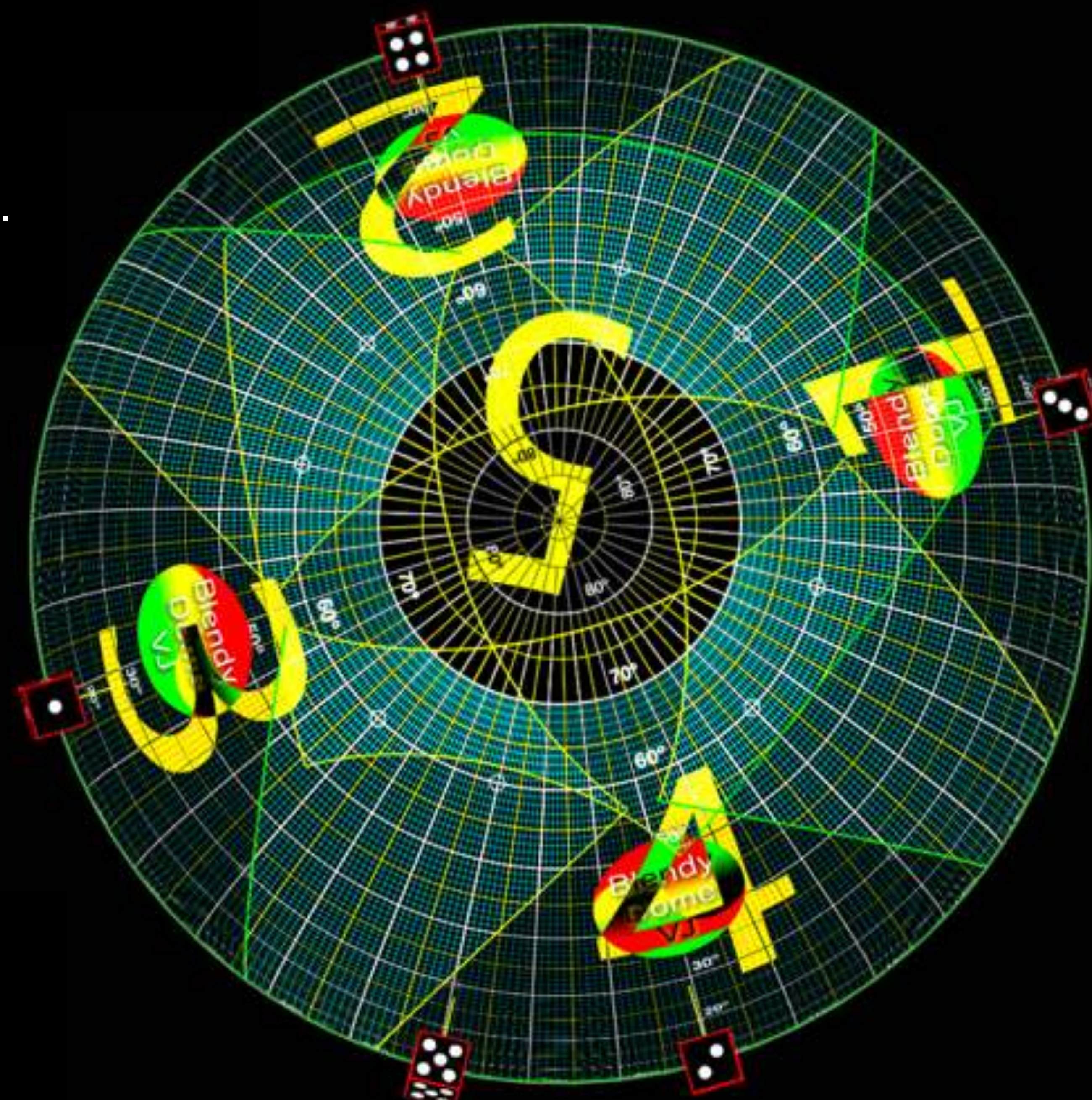
Digital projection - Dual projectors

- 1. Projectors located on the rim of the dome with wide angle lenses.
- 2. Projectors in the center with truncated fisheye lenses.
- (1) is lower resolution than (2) for the same resolution projectors.
  (2) occupies the center, the best seats in the house.
- (2) is often acceptable for planetariums since they often already have a mechanical star projector.

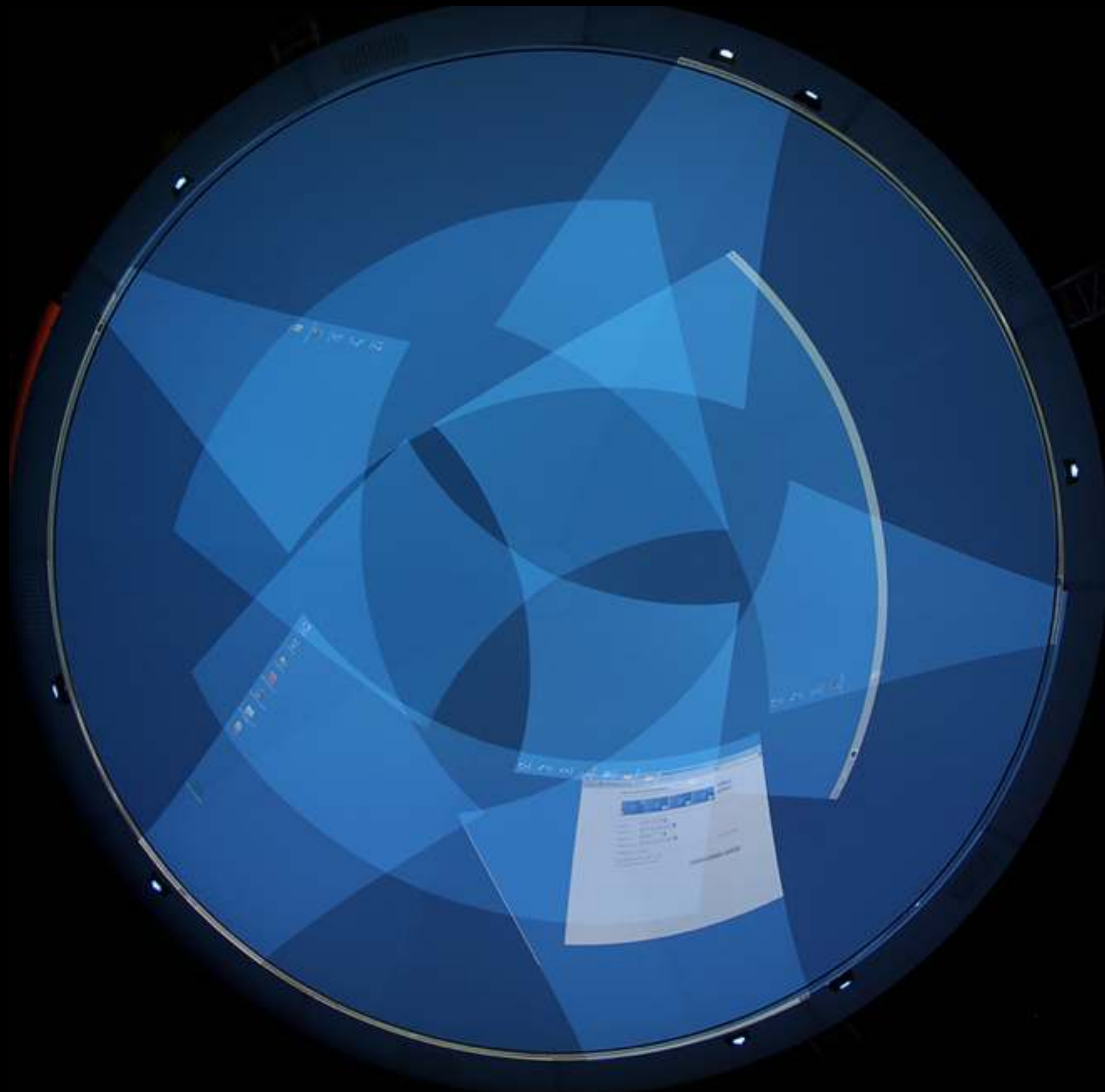


Digital projection - Multiple projectors

- This is the option where resolution scales, just a function of the number of projectors and narrowness of lens.
- Similar concept as resolution scaling for gigapixel photography.
- Projectors generally around the rim.
- Generally employs a cluster of computers. At some point a single machine cannot support enough graphics ports or the performance is insufficient.



DomeLab has 8 projectors



Digital projection - Imaging

- For 2 or more projectors the fisheye image needs
  1. To be diced into N pieces, one for each projector
  2. Geometry corrected to deal with the geometric and optical projection (warped)
  3. Edge blending mask applied to create a seamless image across the projector overlap.

- In the case of movies it is conventional that the producer supplies the 4K frames to the dome operators. The dicing, warping, blending is applied by the operators.

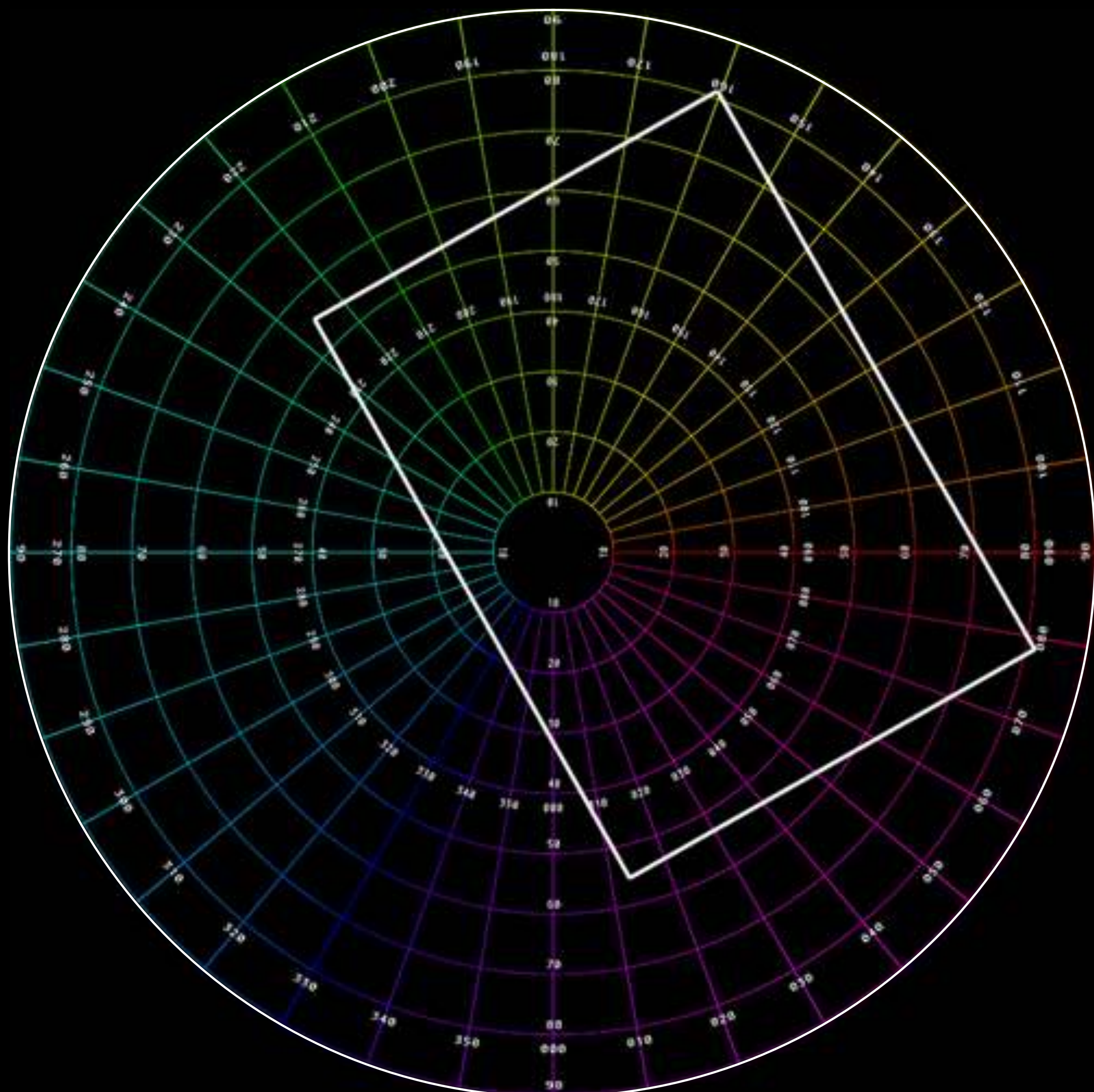
- In the DomeLab case we will provide the software to perform the dicing.
  We have a solution for Windows, Mac and Linux (source code for the later 2).

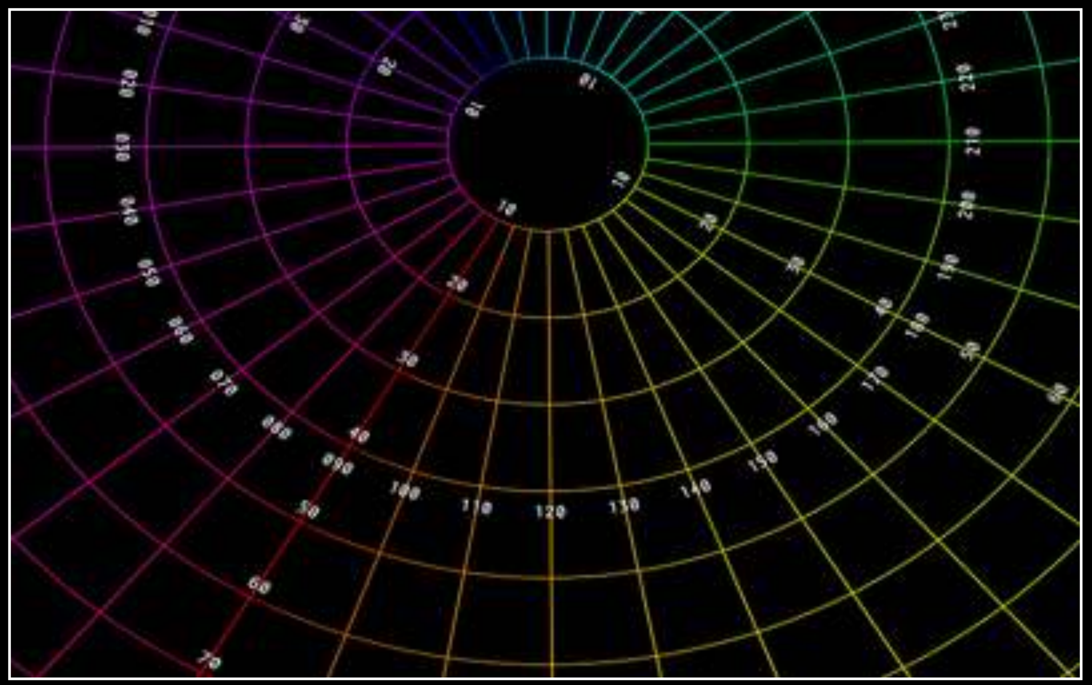
- The warping and blending is performed by the movie playback software (Watchout).

- The diced frames need to be encoded into movies.

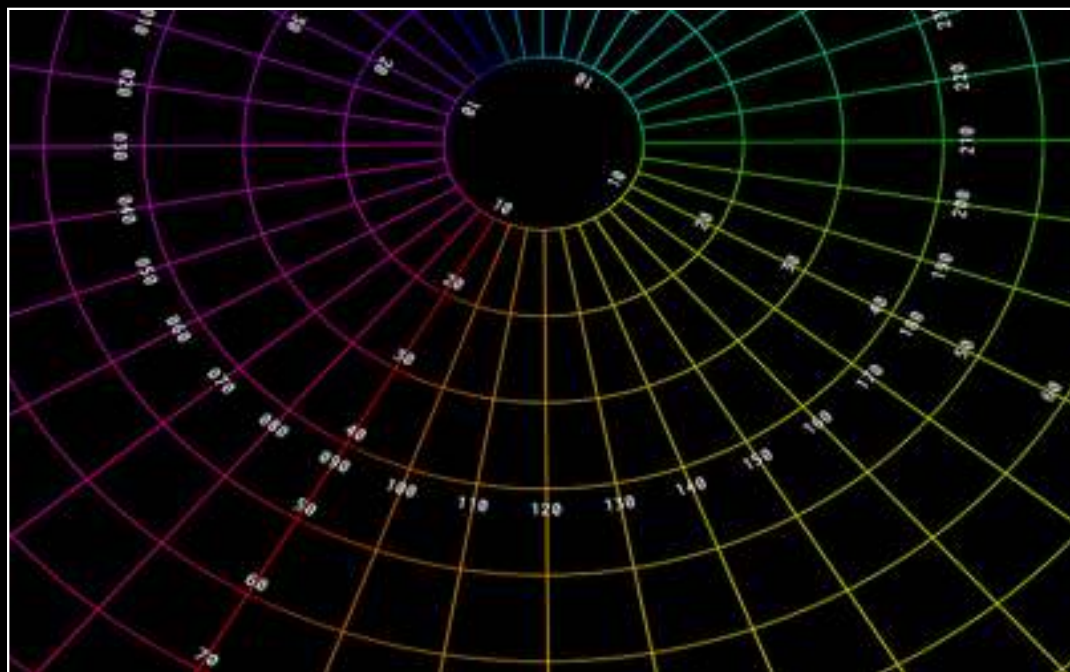
  ffmpeg -threads auto -r 30 -i "/Volumes/Drobodome/Ocean_cut/0/%07d.png" -f vob -vcodec mpeg2video -b:v 50000k -minrate 50000k -maxrate 50000k -g 1 -bf 2 -an -trellis 2 "/Volumes/DomeLab_1_1/Ocean_50k_g1/Display_0.m2v"

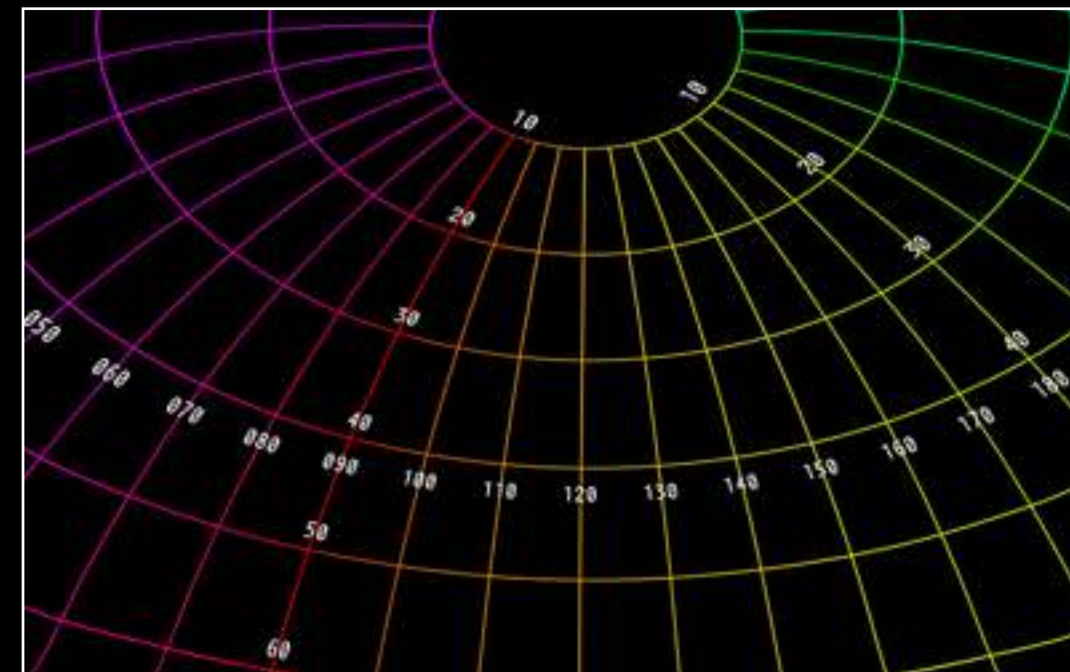
Image processing pipeline

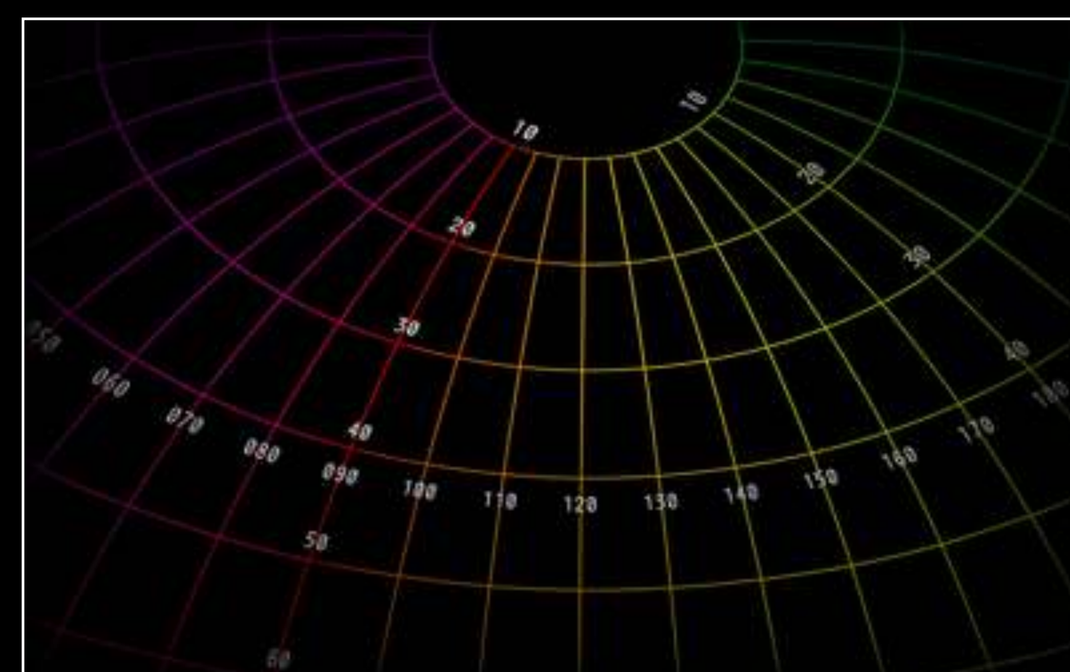




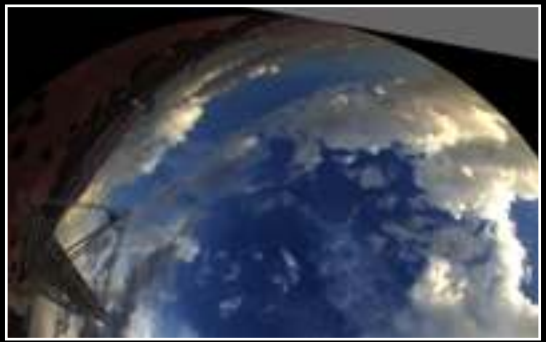
Extract segment for projector N

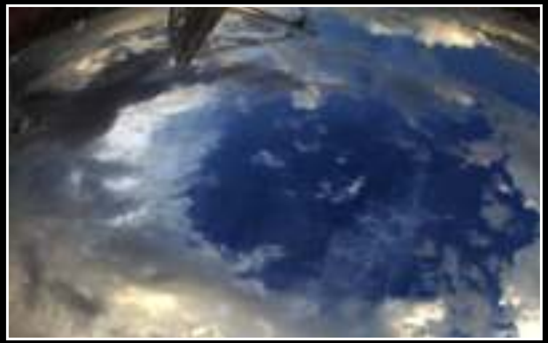






Warp and apply edge blend mask









Realtime

- For generative (realtime) and interactive content the dicing, warping and edge blending needs to be all computed in realtime.

- We have a solution for Unity.

- More importantly we understand every detail of the process so can implement elsewhere.

- The pipeline is similar to the one for movie playback
  - a 4 camera rig arranged to create 4 faces of a cube map
  - 4 crafted meshes to turn those faces into a fisheye
  - 8 crafted meshes with fisheye applied create a warped view for each projector
  - blending applied to those 8 views using simple shader

Considerations: 2D elements

- In general 2D elements, eg: text, needs to be included in the scene as a 3D textured primitive.

- If rectangular elements are applied in fisheye image space they will appear curved in the dome.

Dragon Gardens, Hong Kong

Considerations: Colour saturation

- Common to create higher saturated images to compensate for inter-reflections/cross-bounce.

- Also, bright areas will tend to desaturate the rest of the image due to interreflections. Dome surfaces are a compromise between reflectivity and projector brightness.

Vibrance adjustment

Considerations: Dome angle

- Fisheye angle depends on dome angle and audience arrangement (directional vs omnidirectional).
- Most planetariums are referred to as 0 degrees. iDome is one extreme, 90 degree tilt. Scitech dome in Perth is 30 degrees.
- Most planetarium content has a tilt, due to directional seated audience.
- Unlike movie production where the angle is locked in, for realtime the camera rig angle can be adjusted for optimal angle.

(90 degrees) iDome

30 degrees tilt

0 degrees

Considerations: Speed of movement

- As with most immersive (surround displays) the speed at which one moves or objects move may need to be limited.
- Affects sense of balance once peripheral vision is engaged, conflicts between visual cues and sense of body.
- Momentum / inertia is important for movement by the viewer / player.
- Frame rates even more important than on limited FOV flat screen displays.
- Similarly, smooth playback is more important.

Considerations: Position of viewer(s)

- There is only one position in the dome where the view is correct.
- It may be any position, but only one at a time.
- Creating fisheyes for positions other than the centre of dome outside the scope of this discussion, called “offaxis fisheye”.
- For all other positions straight lines will not be straight for the viewer.
- As an aside, this is true of all displays.
We are simply insensitive to it for flat images (photographs, video ...)
More important for stereoscopic material, tolerable if seated (stereoscopic movies)
Increasingly problematic as the display becomes more capable (stereoscopic and surround).

Considerations: Miscellaneous

- No such thing as zoom.
  Normally achieved by changing the FOV of the camera, in all immersive environments the FOV is locked to the relationship between the viewer and the display surface / geometry.
  To see something up close one needs to move towards it.

- Dome experience works best when one is inside something.

- Be aware of interreflections from bright imagery and subsequent loss of contrast.

- For a 0 degree dome, there is the possibility of an omnidirectional experience.
  How to keep it interesting around the dome?
  Without head tracking the viewer may be looking anywhere.

Recommendations

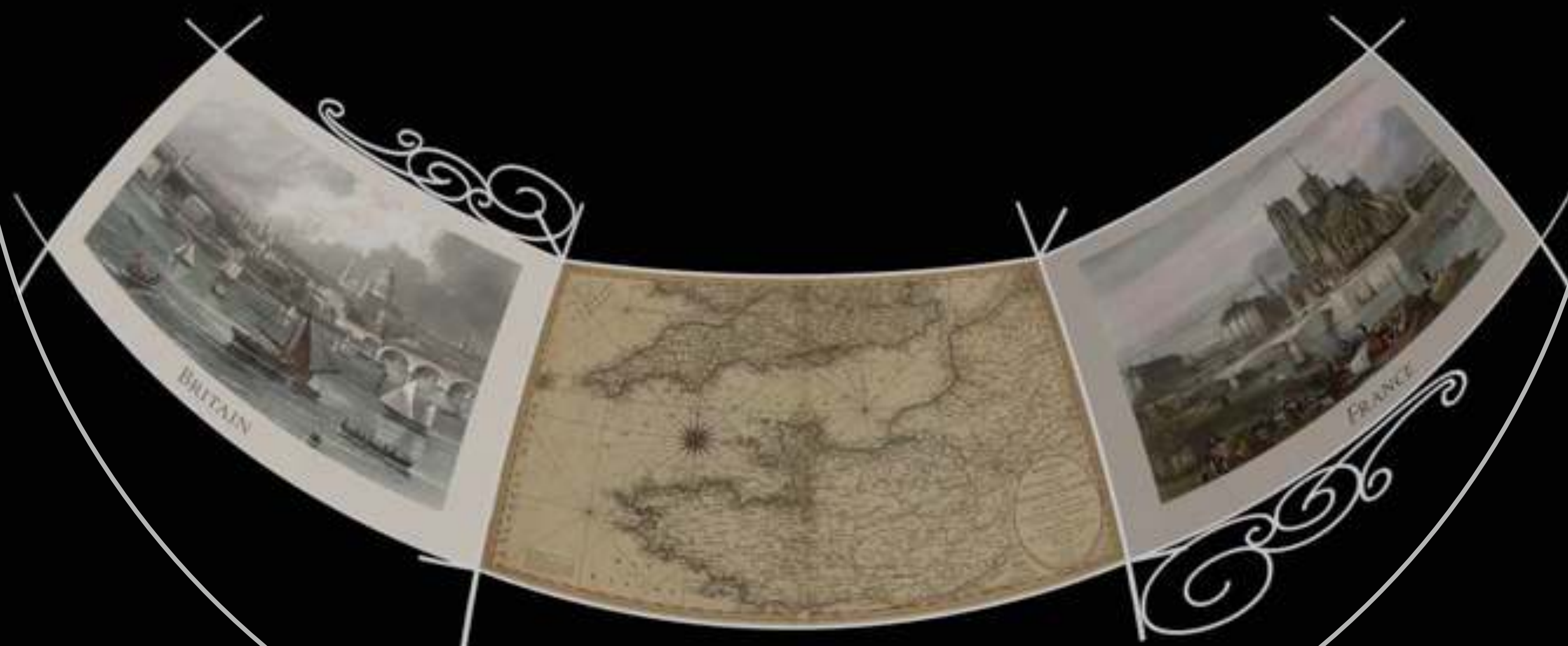
- Develop with access to a dome for preview purposes, especially during the early stages of development.

- Use fisheye imagery where possible. Incorporate 2D photographs as panels on some 3D geometry.

- Many of the perceptual effects one observes in a dome (eg: depth from relative motion with distance) arise from the correct mapping of the 3D world into a 180 degree fisheye.
  This of course assumes camera or objects are always moving. Principle in movie content for dome is to always have movement to leverage this effect.

- Ensure your pipeline (software and underlying hardware) can handle 4Kx4K images.

- Think about why you are using a dome.
  Ensure you are going to leverage the media rather than using it just as a curved flat screen.



12773

Domelab movie standards

- 30fps (not NTSC 29.97)
- “Front” is at the bottom of the fisheye.
- 5.1 audio
- 4K (4096x4096 pixels)
- PNG format for frames

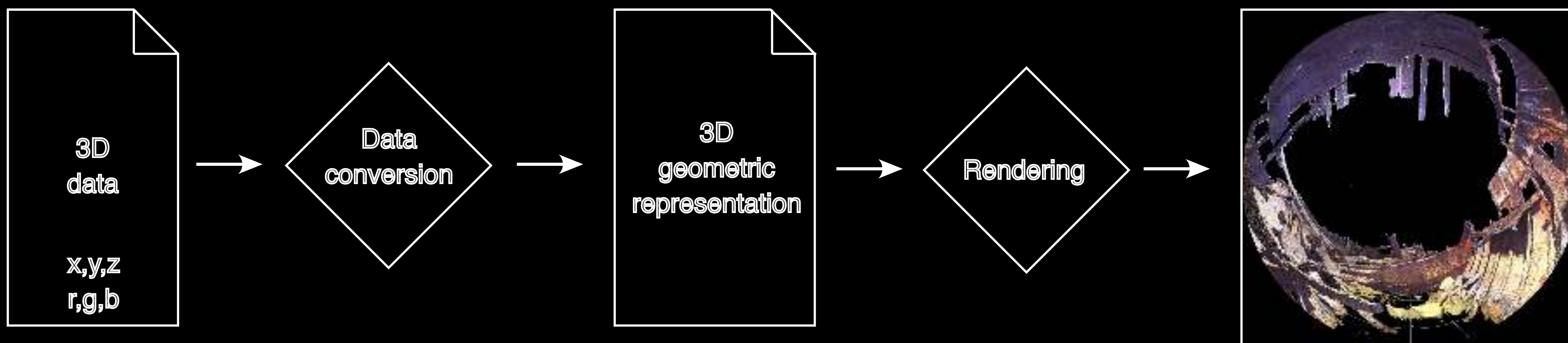
Frame number and time

- File naming should be sssss_nnnnn.png
  Number nnnnn should be zero padded, for example, myshow_00123.png
- Place frame numbers and logos around the unused portion of the frame.

Protector example: point cloud

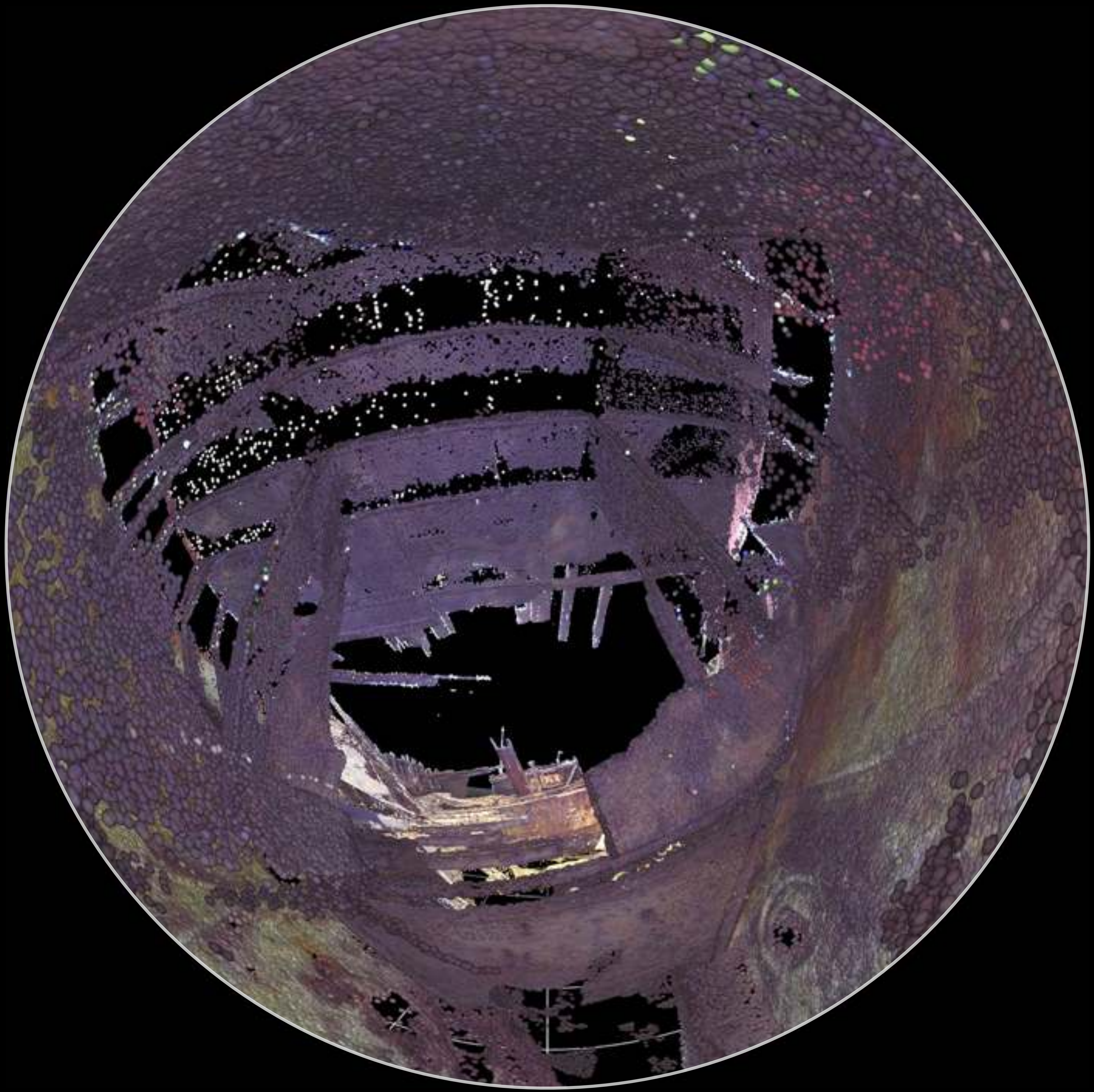
- Example of 3D geometry that can be rendered with a fisheye lens.

- Same general workflow for all 3D data: turn into geometry in your favourite 3D package that supports a fisheye lens.



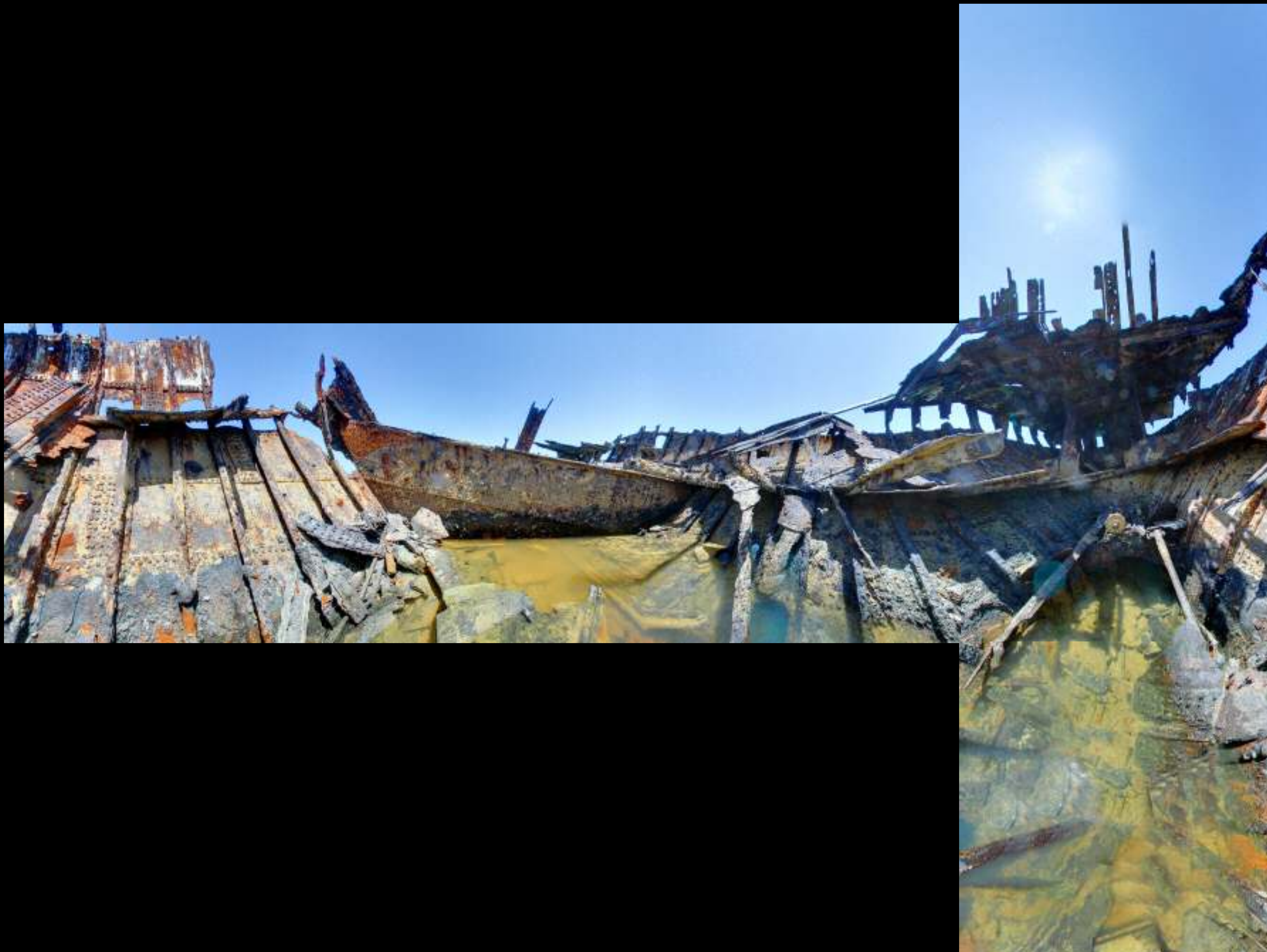

```
550.60 7039.82 1.56 139 116 92
554.94 7035.64 1.62 127 82 66
552.46 7039.99 0.48 55 40 37
553.28 7041.29 2.37 98 71 63
554.66 7040.56 0.45 48 38 34
558.73 7035.80 2.56 242 217 212
555.33 7041.93 1.93 78 54 56
556.75 7040.89 0.54 43 29 34
   :
   :

120 million points

   :
   :
```

```
sphere {
 <19.0017,7.43784,1.46943>,  RR*vlength(VP-
<19.0017,7.43784,1.46943>)*tan(0.5*MAXANGLE)
 texture {
  pigment { colour rgb <0.94902,0.85098,0.831373> }
  finish { thefinish }
 }
}
   :
   :

120 million spheres

   :
   :
```



Protector example: panoramas

- Panoramas are incomplete, missing top and bottom 15 degrees.
- Bottom cut-off isn’t so important, but the top means a hole in the sky.



Questions and discussion