iDome workshop

Prepared for School of Creative Media,
City University of Hong Kong

Paul Bourke
WASP/iVEC
University of Western Australia

Outline

- Introduction to surround displays and hemispherical displays.
  - Summary of planetaria and projection systems.
- Image projections.
  - How objects in the 3D world are mapped to a 2D image.
- Introduction to the iDome.
- Content creation.
- LadyBug-3 camera.
  [This includes running through the whole pipeline from filming to presentation on iDome]
- Quartz Composer.
  [This is largely hands on]
- Further reading.

Example of surround display technologies

- Multiple flat panels.
  - Becoming popular for gaming.
  - 3, 5, 6 display port cards available.
- Multiple flat walls, usually seamless.
  - Lots of combinations have been tried.
  - CAVE.
- Cylinders, partial or complete.
  - AVIE.
  - 120 degrees popular in VR centres.
  - A number of personal displays available.
- Planetaria, hemisphere or other section of a sphere.
  - iDome.
- Monoscopic and stereoscopic

Brief history of domes for planetaria

- 1500BC: Earliest known depiction of the night sky on Egyptian tomb of Senenmut.
- 500BC: First known domed building, called the The Dome of Heaven.
- 1923: First planetarium built in Munich, Germany. Projection using the Zeiss Mark 1 star projector.
- 1949: Spitz demonstrated their first star projector at Harvard College in the USA.
- 1959: First planetarium and star projector by GOTO of Japan.
- 1965: First star projector by Minolta of Japan.
- 1973: First OmniMax (iMAX) opened in Reuben Fleet Science Centre, based upon 70mm film.

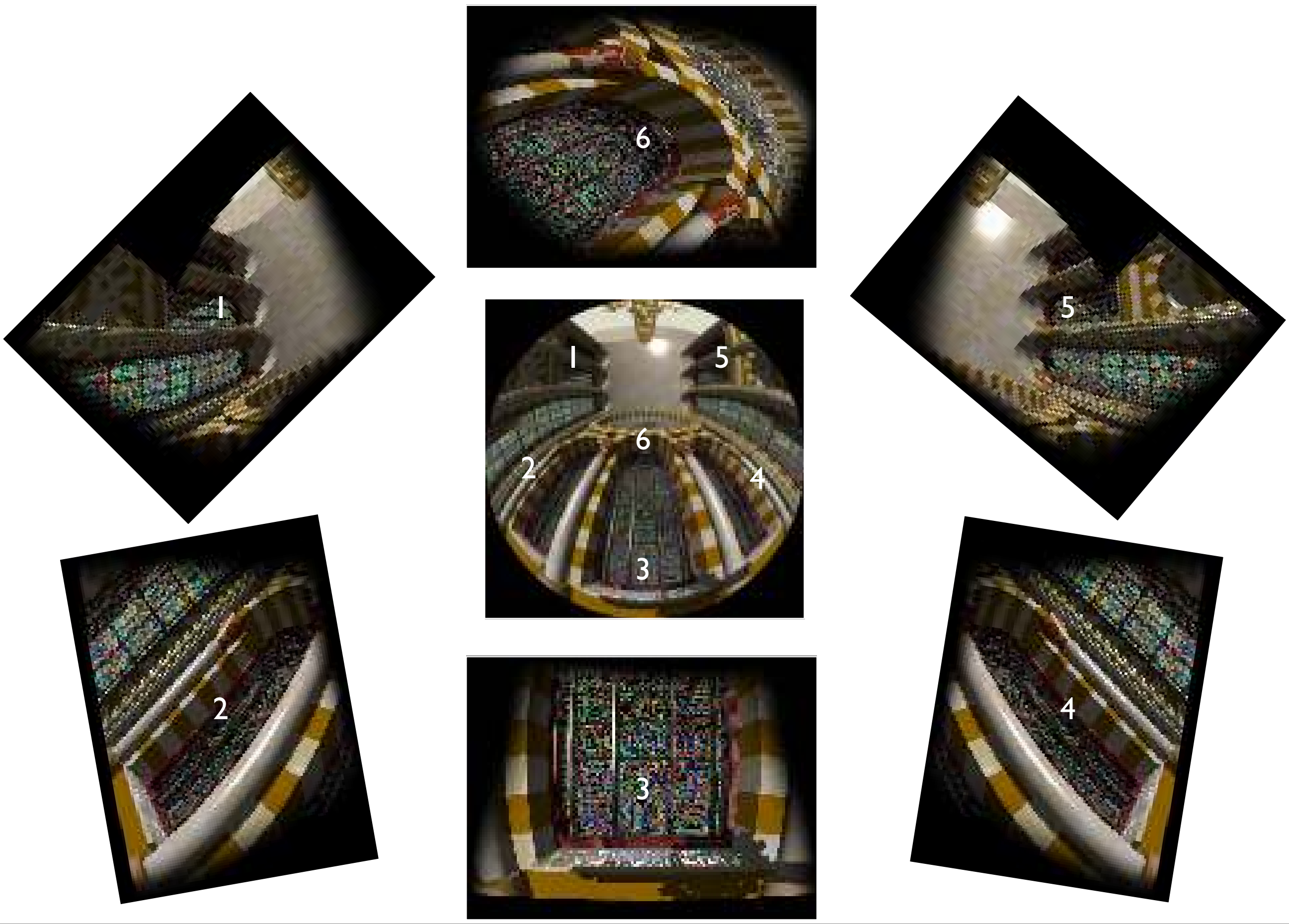
Brief history of digital domes (fulldome)

- 1983: Evans and Sutherland develop a vector graphics style projector capable of creating points and lines at the Virginia Science Museum.
- Alternate Realities released the VisionDome in 1994. Elumens produced "personal domes", primarily as simulators for the military.
- 1997: Spitz install the first ElectricSky system in Canada comprising of 4 CRT projectors and edge blending.
- 1998: SkySkan demonstrates their digital projection system. The first digital video content not reliant on custom projection hardware.
- 2002: First laser projection system by Zeiss demonstrated in the largest digital dome at the time, 24m diameter.
- 2005: GOTO of Japan create the first full sphere projection system.
- 2008: SkySkan installs the first 8Kx8K projection system in the Beijing planetarium.
- 2010: SkySkan installs first stereoscopic 8Kx8K planetarium in Macau.

Summary of projection technologies for domes

- Multiple projectors: Traditionally CRT due to perfect black capability, today other digital projector technologies are used. Often 5, 6, or 7 projectors.
- Laser based, not very common. Generally multiple units, very high colour fidelity and dynamic range.
- Two projectors with wide angle lenses. Popular configuration for high resolution, generally use the 4K projectors from Sony.
- Single projector with a full or partial fisheye lens. Was the standard solution for small and portable planetariums for many years.
- Spherical mirror (developed by myself). Quickly becoming the standard for single projector installations. With care it is as good as single projector and fisheye systems (4K projectors excluded).
- Comparing projection systems is largely a matter of resolution (number of pixels on the dome). Price rises rapidly with resolution.

Data projection example for 6 projector dome



Fisheye data projection

- Single or dual projectors and fisheye lens is the simplest most natural way to project fisheye images within a hemispherical dome.
- Has been very popular for small dome based simulators.
- Inscribed fisheye only uses 59% of the pixels, truncated option uses 84% of the pixels. For planetarium style configurations the truncation is usually at the back. For upright domes it is usually at the floor.
- Hard not to have chromatic distortion near the rim, extreme angles through the fisheye lens. Quite a bit of light can be "lost" through the lens.
- Most full fisheye projection system use SXGA+ and 4:3 (no significant advantage with HD since the inscribed circle is limited to the image height, 1050 vs 1080).

Inscribed fisheye in a 4:3 frame

Back truncated fisheye in 4:3 frame

Fisheye data projection

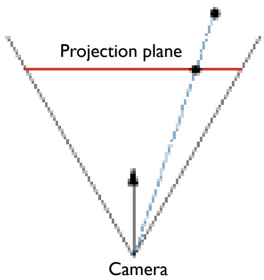
- Full fisheye in 16:9 aspect ratio projector only uses 44% of the pixels.
- Truncated 16:9 uses 80% of the pixels.
- Dual projectors are best suited to 16:9 aspect ratio projectors.
- WUXGA gives maximum height of 1200 pixels.

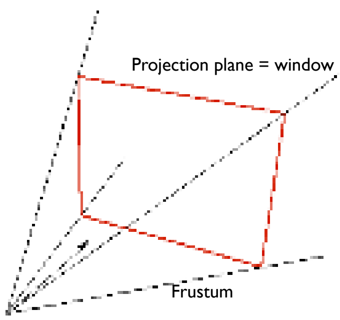
Inscribed fisheye in a 16:9 frame

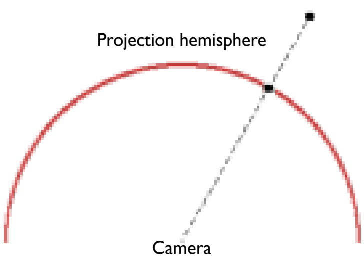
Twin fisheye pairs in a 16:9 frame

Introduction to projection of 3D to a 2D surface

- Fundamentals of a perspective projection, the point on the projection plane is the intersection of a line from the 3D object to the camera.
- Model of looking through a window.
- A perspective projection is the simplest that captures the required field of view for presentation onto a rectangular region on a plane.
- Why isn't a perspective projection sufficient for a dome?
- Intersection of world objects with a sphere defines a fisheye or spherical projection.
- Fisheye projection is the simplest that captures the required field of view for subsequent presentation onto a hemispherical surface.

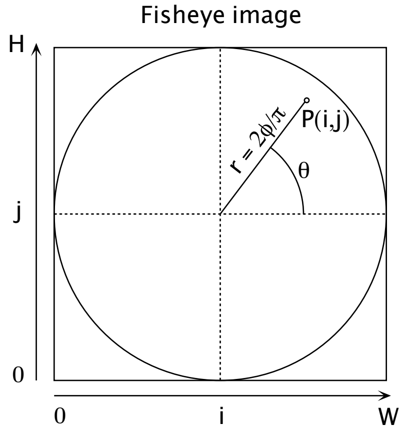


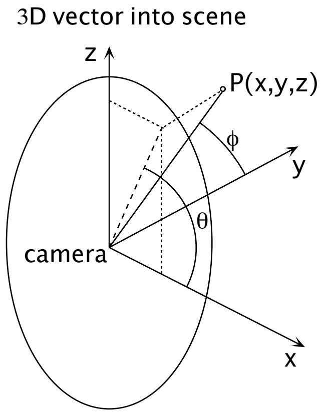




Fisheye projection equations

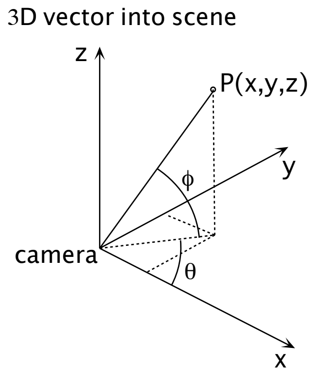
- Typically need to relate the mapping to/from fisheye image coordinates (2D) to a world vector (3D).
- 1. Given a point P(i,j) on the fisheye image (in normalised image coordinates), what is the vector P(x,y,z) into the scene?
  $r = \mathrm{sqrt}(Pi^2 + Pj^2)$
  theta = atan2(Pj,Pi)
  $\Phi = r\,\pi / 2$
  $Px = \sin(\Phi)\cos(\theta)$
  $Py = \cos(\Phi)$
  $Pz = \sin(\Phi)\sin(\theta)$
- 2. Given a point P(x,y,z) in world coordinates what is the position P(i,j) on the fisheye image?
  $\Phi = \mathrm{atan2}(\mathrm{sqrt}(Px^2 + Pz^2), Py)$
  $\theta = \mathrm{atan2}(Pz, Px)$
  $r = \Phi / (\pi / 2)$
  $Pi = r\cos(\theta)$
  $Pj = r\sin(\theta)$
- Traditional to limit the fisheye image to a circle but it is defined outside the circle.

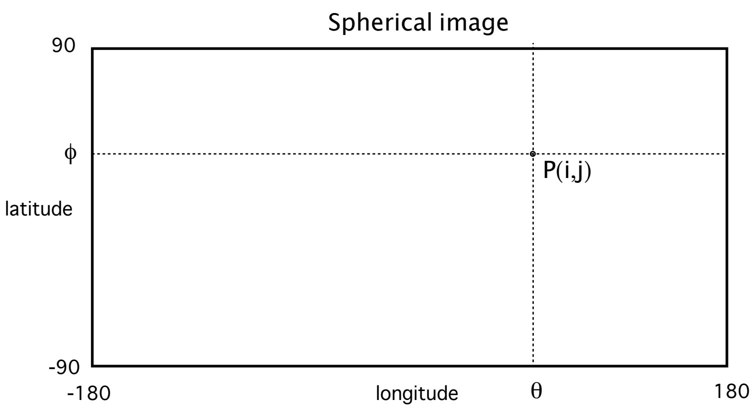




Spherical projection

- Also contains sufficient visual information for a presentation into a hemisphere, actually captures more than required.
- 1. Given P(i,j) in spherical projection, what is the 3D vector into into the scene P(x,y,z)
  $Px = \cos(\Phi)\cos(\theta)$
  $Py = \cos(\Phi)\sin(\theta)$
  $Pz = \sin(\Phi)$
- 2. Given 3D vector P(x,y,z) what is the corresponding point on the spherical projection.
  $\Phi = \mathrm{atan2}(Pz,\mathrm{sqrt}(Px^2 + Py^2))$
  $\theta = \mathrm{atan2}(Py,Px)$





iDome

- A small, personal, upright hemispherical surface.
- The problem with a fisheye lens projection system is that the projector is ideally located at the same position that is ideal for the viewer.
- The spherical mirror ensures light is reflected onto the whole dome surface.
- Projector is conveniently located at the rear base of the hemisphere, largely out of sight.

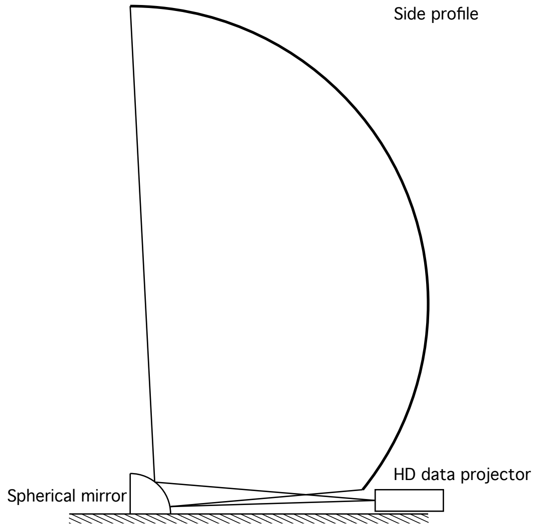


Image warping for spherical mirror projection

- Fisheye lens and data projector just accepts a fisheye image, nothing more to do.
- Spherical mirror projection requires a fisheye warping (distortion) to give the correct result in the dome.
- Describe the mesh that performs the warping.
- Each configuration slightly different so requires a site specific calibration.
- Variables: projector/mirror/dome position, projector optics (throw, offset), mirror/dome dimensions.

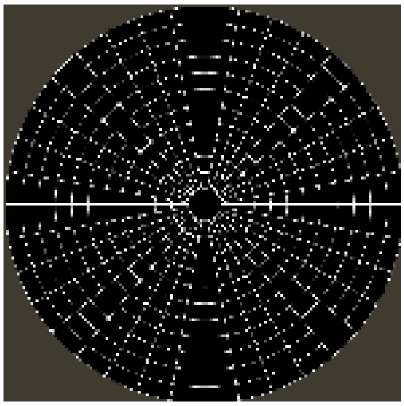
Fisheye

Warped fisheye

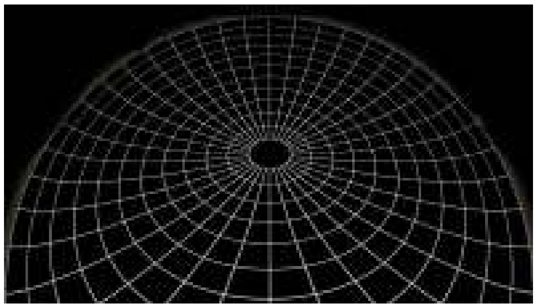
Image in the iDome

Calibration

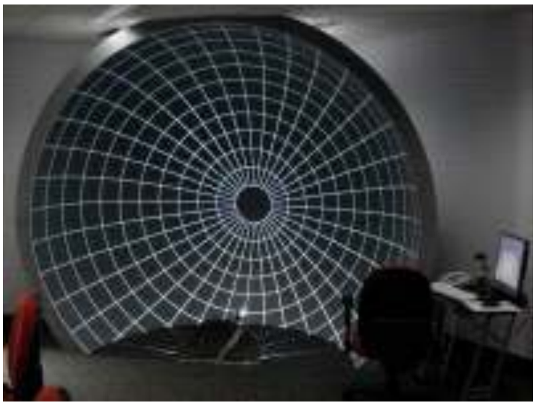
- A series of lines of latitude and longitude is the usual calibration image.
- The lines of longitude should be straight.
- The lines of latitude should be circular rings.
- The north pole should be at the centre of the hemisphere.



Fisheye polar grid



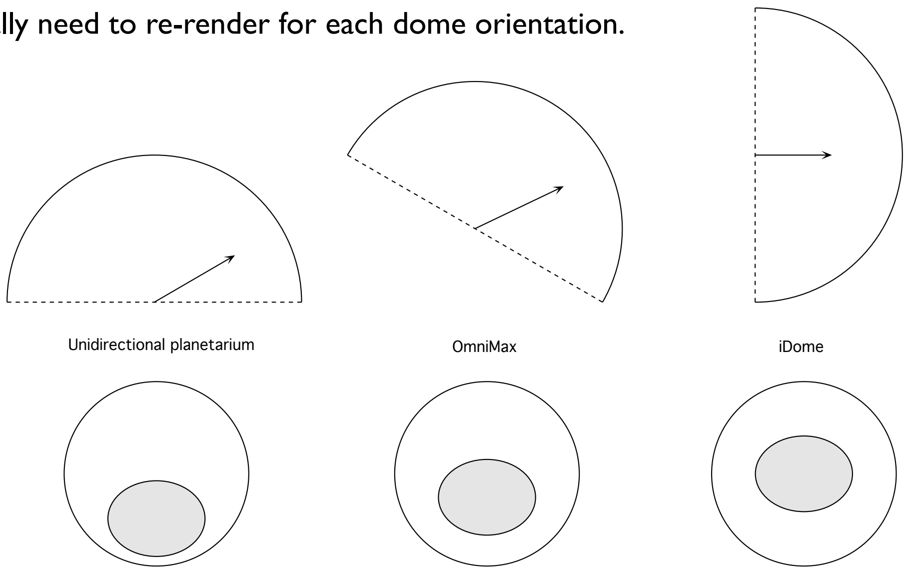
Warped fisheye



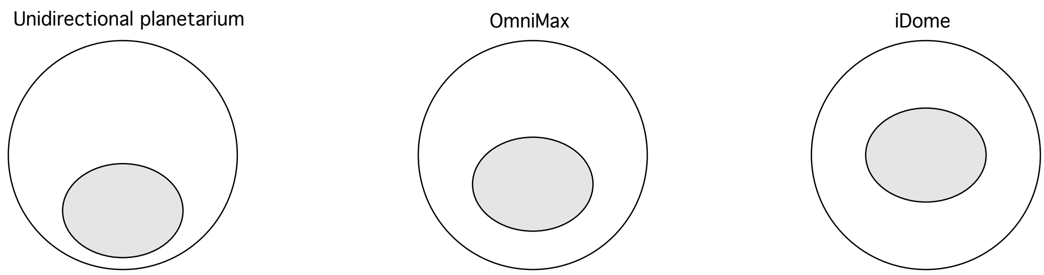
Result in iDome

Dome/seating orientations and implications

- Need to consider the angle of the dome, dome range from 0 degrees to 90 degrees of the iDome. Most iMax domes are at a 60 degree angle. The angle of the dome with respect to the seats defines the 'sweet spot", the position on the dome the viewers had defaults to.
- Need to consider whether the installation has omni-directional or uni-directional seating.
- Why most directional content for planetariums isn't suited to the iDome.
- Generally need to re-render for each dome orientation.



Centre of attention ("sweet spot")



Single seating position for a perfectly correct view

- There is only one position for the viewer that results in a perfectly undistorted view of the scene, generally the center of the hemisphere.
- In this position straight lines should appear straight, in all other positions straight lines will appear to curve.
- It is possible to move this correct viewing position to anywhere, outside the scope of this workshop but the technique is generally called "offaxis fisheye".
- When projecting into a dome using a fisheye lens the exact same method can be used to located the fisheye lens away from the center of the dome.
- The iDome is intended for a single viewer, for a planetarium audience everyone gets a distorted view except for those near the center.
- [The same applies to viewing stereoscopic content even on a flat wall].

Peripheral vision

- Peripheral vision is one of the capabilities of our visual system that is not engaged when looking at standard flat or small displays. (Similarly stereoscopy, sense of depth arising from our two eyes, is another unused capability).
- For all practical purposes our horizontal field of view is 180 degrees, vertical field of view is approximately 120 degrees.
- Note we don't necessarily see colour or high definition in our extreme horizontal field, it has evolved to be a strong motion detection mechanism. Our visual system does "fill in" the colour information for us.
- A hemispherical dome allows our entire visual field (vertically and horizontally) to be filled with digital content.
- We are used to seeing the frame of the image which anchors the virtual world within our real world. In a dome one often doesn't see that reference frame.
- The "magical" thing happens when one doesn't see the dome surface, more common in high quality domes with good colour reproduction. Our visual system, without any physical world frame of reference, is very willing to interpret representations of 3D worlds as having depth. More difficult to achieve this without a high quality seamless dome surface.

Content creation

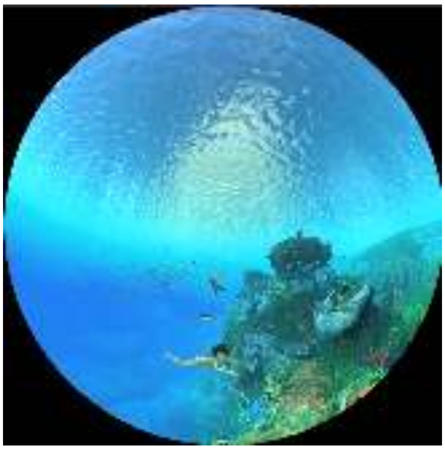
- Computer generated
  - Fisheye lens
  - Cubic maps
  - Mirrored sphere (Approximation)
  - Spherical projections

- Photographic
  - Fisheye lens
  - Spherical projections from fisheye lens
  - Spherical projection from Gigapan
  - Video from single camera
  - Video from multiple cameras

- Realtime CG
  - Multipass textures
  - Unity3D game engine
  - Blender Game Engine (BGE)
  - Vertex shader

Fisheye lens

- This is the simplest of all, the software renders a fisheye projection directly.
- Supported sometimes as a third party plugin.
- This can be the most efficient for a targeted display but may not be the best approach for content developed for different environments, eg: different dome orientations. If a full spherical or 6 cubic maps are rendered then adjusting the fisheye angle is simply an image manipulation exercise in post production.



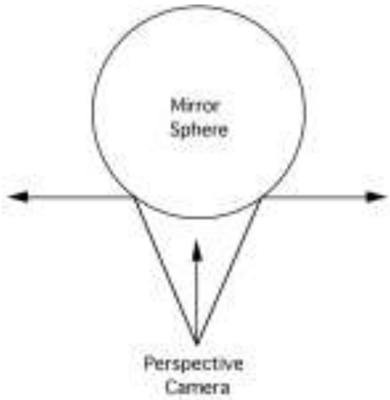
Children of the Water

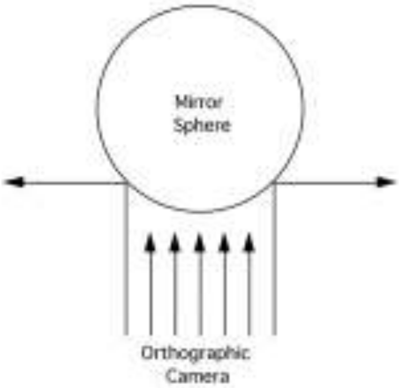
Dinosaur Phophecy

Cubic maps

- Only need 4 views to create a fisheye, using 6 views allows one to create a fisheye orientated in any direction.
- Software: cube2fish (see later), GLOM.
- Also amenable to realtime exploration of images or movies (see later).
- Sometimes also known as an environment map.
- Typical resolution of each cubic map is 1/2 the diameter of the final fisheye image.

Mirrored sphere

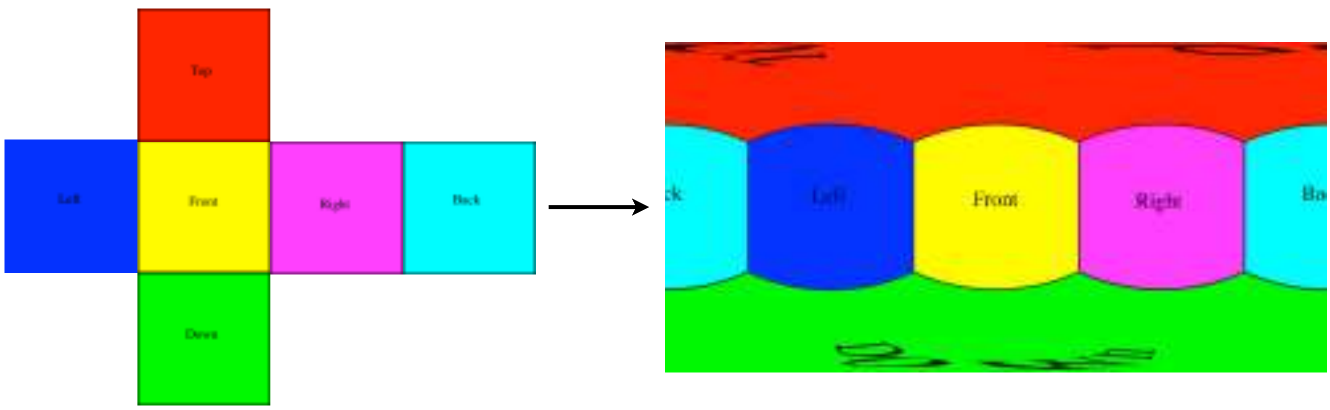




- Pointing a camera at a perfectly mirrored sphere gives an approximation to a fisheye projection.
- Two ways of doing this
  - perspective camera
  - orthographic camera
- Not strictly correct (the radial dependence of latitude lines is not perfectly linear) but the error is often acceptable.
- The reflective sphere is usually made very small so the camera is very close ... reduces the chance of world objects coming between the mirror and the camera.
- The mirrored ball must not cast shadows on the world objects.
- Not terribly common any more since fisheye support is more prevalent than in the past.

Spherical projections

- These allow one to create interactive dome movies.
- The entire visual field is captured.
- This is the projection captures by the LadyBug camera (see later).
- Sometimes supported in software or through third party plugins.
- Can also be created from 6 cubic maps.



Photographic - Fisheye

- SLR camera and fisheye lens.
- “Fisheye lens” for photographers is often just a very wide angle lens.
  Term used in photography circles is “circular fisheye”.
- Nikon D300 and Sunex 185 fisheye, capable of 2800 pixel square 180 degree fisheye images.
- Lens uses 85% of the sensor height.
- Issues of lens quality, chromatic error and focus issues on the rim.

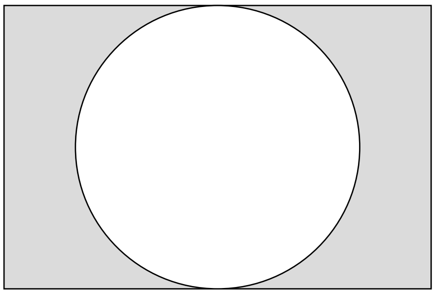
Fisheye lens sensor utilisation

- When choosing any fisheye lens and camera it is important to consider the circular fisheye image size and the sensor size.
- May be truncated, ideally it uses the maximum height of the sensor.
- For the iDome is may be acceptable if the fisheye is truncated across the bottom.
- One wants to use as much of the available pixels as possible.
- It is possible to rotate a truncated fisheye in post production to (partially) remove the truncation at one extreme (top or bottom).

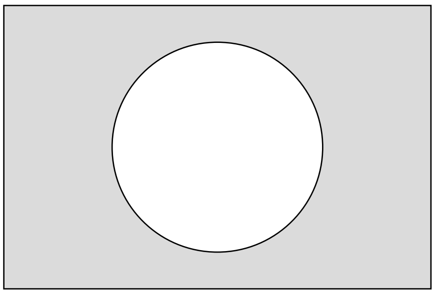
Truncated fisheye (top and bottom)

Canon EOS 5D MkII + Sigma 8mm fisheye lens

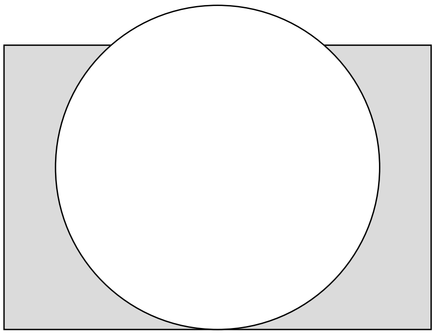
Fisheye sensor utilisation



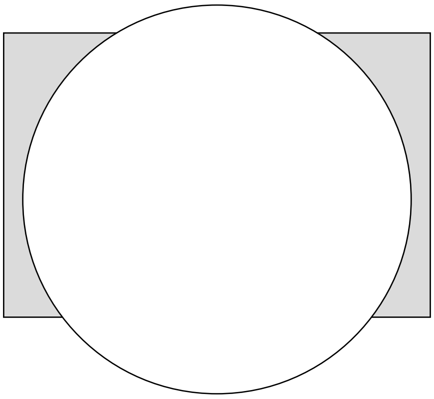
Ideal, full sensor utilisation.



Full fisheye but inefficient use of sensor.



Truncated fisheye
May be OK for some planetariums,
or the iDome if truncated at the bottom.



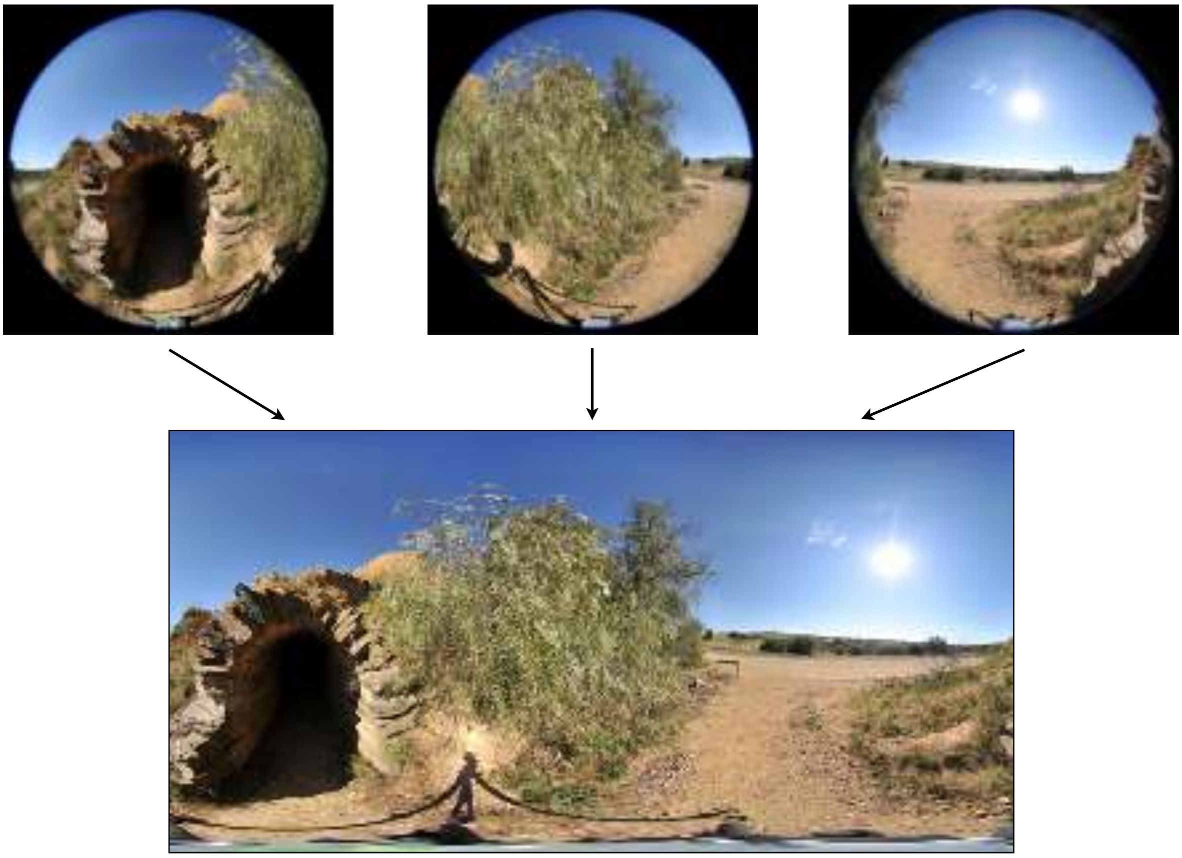
Fisheye truncated to and bottom.

Photographic: spherical projection using fisheye lens

- Multiple shots captured horizontally that are stitched and blended together to form a spherical projection.
- With the Nikon and Sunex 185 degree lens, just 2 images are required for full spherical projection, but generally 3 are taken for larger overlap and less strict camera alignment.
- As with most multiple image compositions, the shooting conditions need to be "fairly" static between shots.
- Require constant camera settings: exposure, aperture, white balance, etc.
- Camera should be rotated about its "nodal point" (see discussion of problems with multiple camera video later).
- The resolution of the final spherical projection is limited by the diameter of the fisheye image. As such there is a limit to the resolution of the spherical projection.

Nikon D300 and Sunex fisheye lens

Spherical from 3 fisheye photographs, an example



Gigapixel images

- Higher resolution spherical panoramic images can be acquired using a large number of shots.
- One takes multiple overlapping images at different latitudes and longitudes. These are stitched together and blended to form a single seamless spherical projection.
- This can be automated with devices such as the Gigapan mount.
- In general a regular scan in longitude and latitude is performed even though this is inefficient towards the north and south pole.

Mawsons hut (Antarctica), courtesy Peter Morse.

Video: Camera

- Camera with fisheye lens.
- Very difficult to get sufficient resolution.
- Canon EOS 5D MkII and Sigma 8mm lens, HD 30fps video.
- A fisheye image within a HD video camera creates at best a 1080x1080 fisheye image. This is generally not high enough resolution.
- Solutions exist with the Red camera that give 4K fisheye movies.

Video - Multiple cameras

- In order to increase resolution the solution, as with still photography, is to employ additional cameras.
- Main issue is the lack of portability due to disk arrays required for storage and associated cables.

Micoy camera

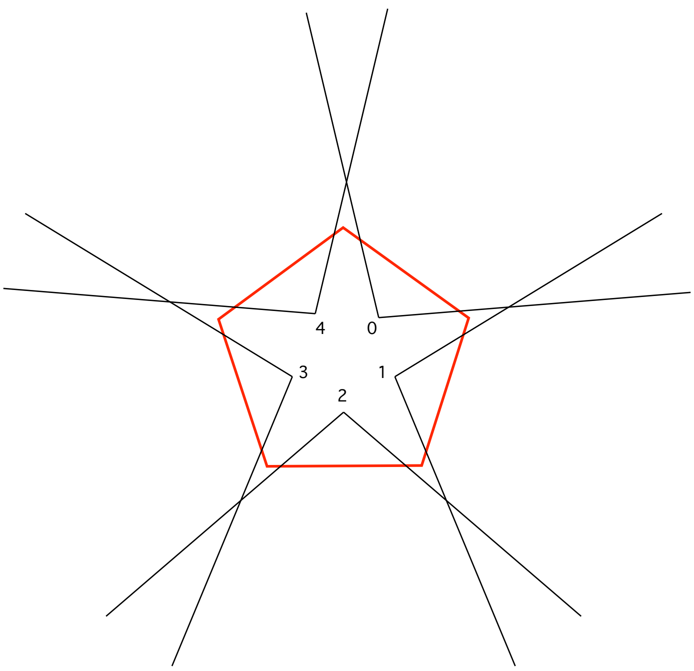
iCinema, UNSW

Video: LadyBug-3 camera

- Captures almost a whole spherical projection.
- 5400x2700 pixels.
- 16fps.
- 360 degrees in longitude
  and about 140 degrees in latitude.
- Can be powered from a laptop, full frame rate and full quality requires an external disk drive.
- See later for more details ...

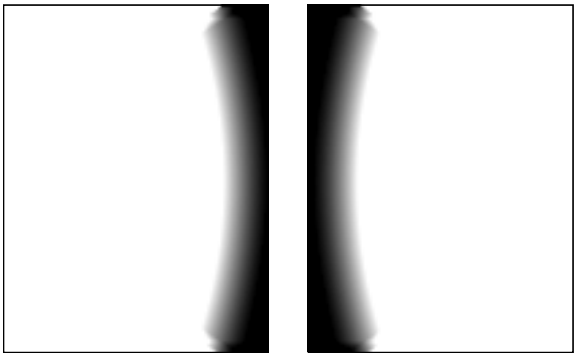
Overlapping field of view

- Top view showing overlapping cameras field of view.
- Similar overlap occurs between the top and side cameras.



Blending

- In addition to geometry correction between the cameras, the overlapping regions are blended together.
- See “Blending width” later in the discussion of the LadyBug-3 operation.
- There is a blend between each pair of cameras (0 to 4), and a blend between the top camera (5) and each camera (0 to 4).

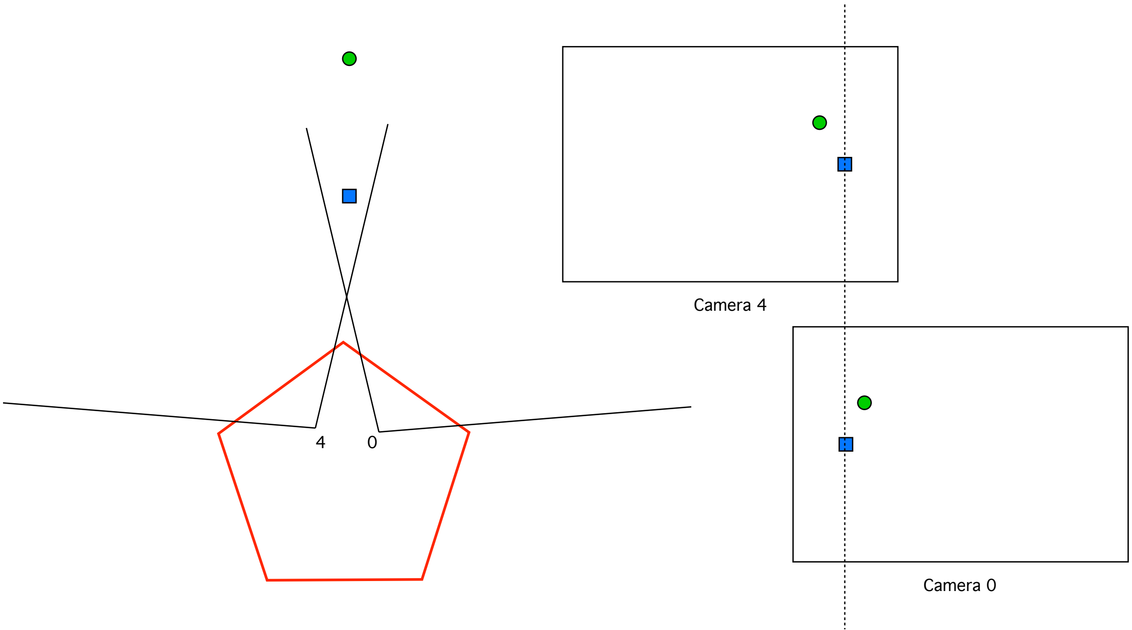


Blend masks between camera 0 and 4

Applied between camera 0 and 4

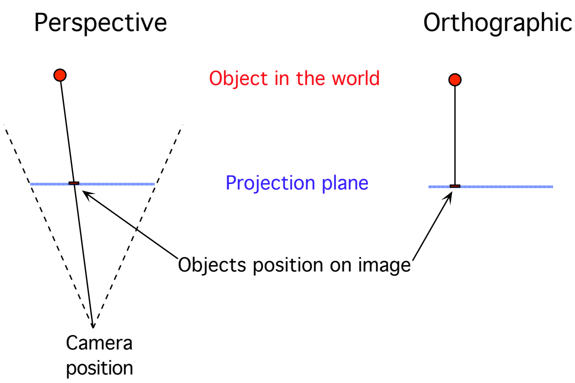
General problem with multiple cameras

- The fundamental problem with multiple cameras is the lack of a single projection plane.
- Consequence is that perfect stitching is not possible across all depths.
- If the stitching is adjusted for blue square then it will be incorrect for the green circle.
- See “Sphere stitching size” later in the discussion of the LadyBug-3 operation.



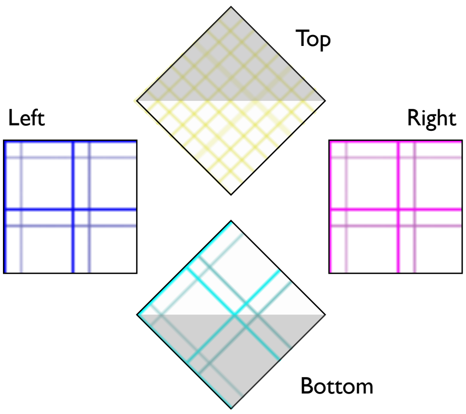
Realtime

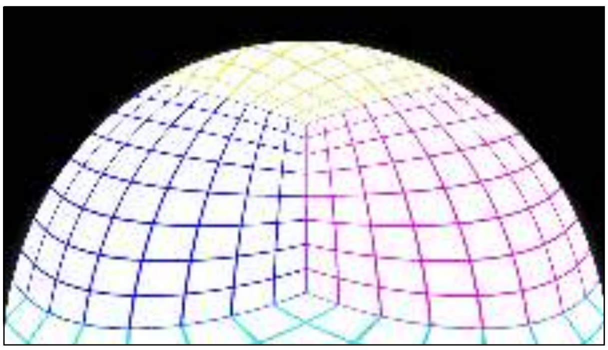
- OpenGL and DirectX only support perspective and orthographic projections.
- As before need to somehow create a fisheye projection.
- Two main approaches
  - Multipass textures.
  - Vertex shader.
- Each has their relative merits.



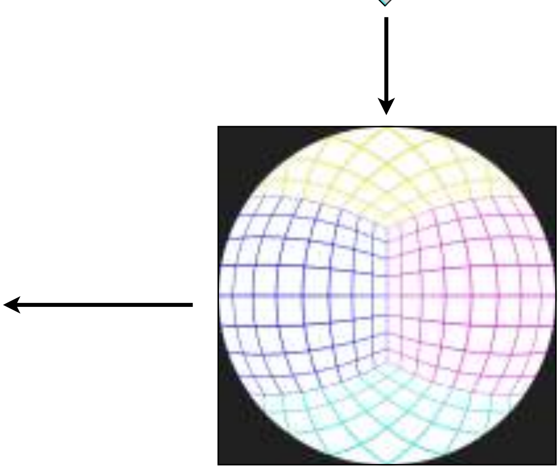
Multipass textures

- The most straightforward approach and often the easiest to integrate into existing software.
- The scene is rendered 4 times and the resulting images, instead of being displayed, are mapped onto a mesh that creates a fisheye projection when rendered with an orthographic camera.





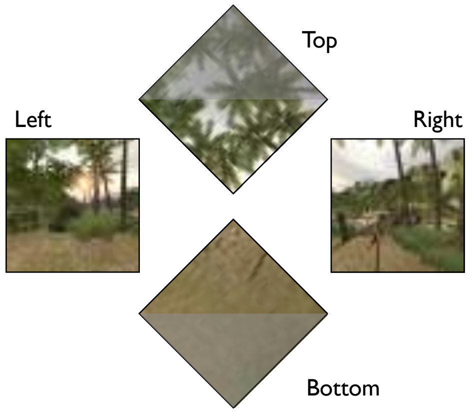
Warped fisheye



Fisheye

Example from Unity3D game engine

- Four initial passes implemented as “render-to-texture”, so requires Unity Pro.
- Possible to skip the fisheye step and apply the 4 textures directly to the warped texture mesh but the performance for the texture warping phase is negligible, less than 1 fps. This direct warping has some tricky implications for the design of the required texture meshes.



Warped fisheye

Fisheye

Texture dimensions

- What size textures to use in each stage? Two high and there are performance and aliasing effects. Too low and the full resolution of the iDome isn’t being exploited.
- Cube face textures: 1024 pixels square. Fisheye texture is 2048 pixels square. Final image to be projected is HD, 1920x1080 pixels.
- Typical performance penalty: each cube face individually render at between 36 and 52 fps (old ATI X1900 graphics card). Combined to form fisheye: 24 fps. Warped fisheye (final frame rate): 23.2 fps.

Issues

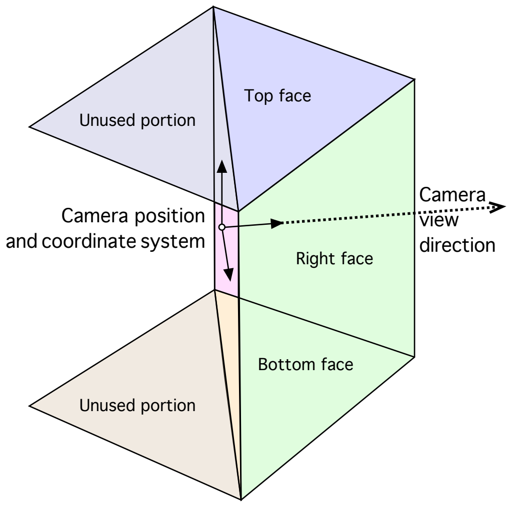
- Effects that are essentially 2D effects on the image plane are problematic, for example, sun halos. The sun near an edge of the cube faces only appears in one face so that is the face the flares are added to and not the adjacent faces.
- These effects are also problematic with vertex shader approach since the effect will generally not be in fisheye coordinates.
- Another problem occurs with effects or geometry that have camera angle dependence. For example billboarded textures, the billboard will be at different angles between cube cameras leading to a discontinuity.
- Billboarding for grass is particularly noticeable since the camera for the bottom cube face is parallel with the grass billboards. [Grass billboarding can be turned off in the current release].

Sun effects as it crosses the cube faces.

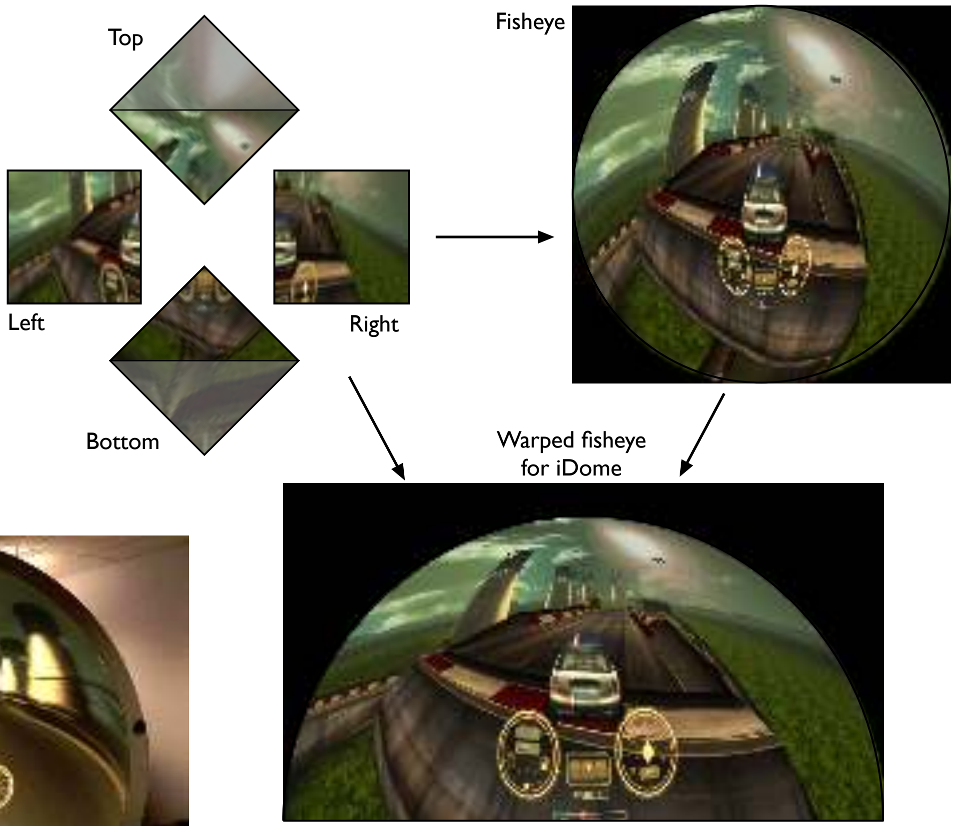
Billboarded grass.

Example from Blender Game Engine (BGE)

- Support for fisheye and warped fisheye is now built into the standard build of version 2.49 onwards.
- Uses the same mesh files as other software: warpplayer, pbmesh, etc.



Example: Club Silo



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.
- Very simple vertex shader, renders in a single pass.
- However there is a complication. 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, inefficient tessellation results in a high geometry load on the graphics card.

Other considerations

- Speed at which things move across the fisheye dome.
  Generally need to move objects much more slowly than traditional animation. Similarly it is important not to have sudden jarring start and stops in the animation.

- Camera motion needs to start/stop smoothly, similar to real life where everything has momentum. This is equally true for filmed footage where very careful attention needs to be paid to any camera movement.

- Text size and the amount of the dome a single piece of text occupies, difficult to read across a wide field of view. Generally should keep a text token within a narrow enough field of view that head panning isn't required.
  Animated text (because it is generally across a larger distance than usual) tends to be more difficult to read in a dome environment.

- As a general rule higher levels of spatial antialiasing (to overcome visible pixel borders across between high contrast transitions) are required for dome content.

Challenges for content creators

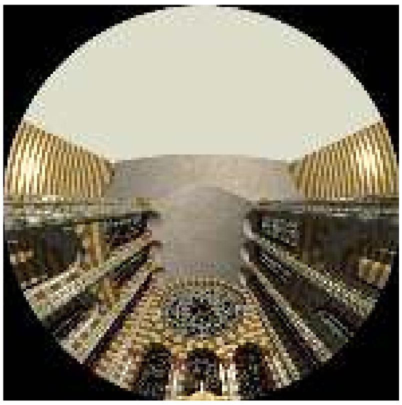
- Need to model in detail much more of the scene than needs to be done for a "normal" limited field of view perspective projection.

- The requirement to render very large frame sizes compared to more traditional media. Perhaps the most common format for widely distributed content 3600x3600 pixel fisheye images is the standard. Compared to a XGA (1024x768) resolution animation that is 16 times the number of pixels per frame.

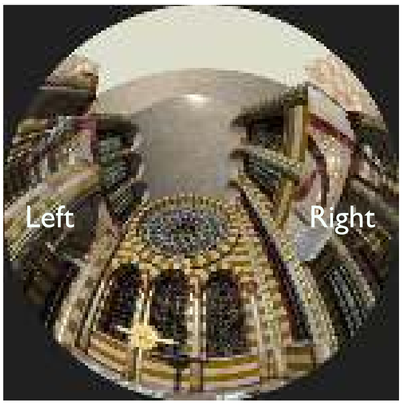
- Access to render farm with your rendering engine of choice.

- Lack of easy preview options. Small domes and projection systems are rare and relatively expensive, time in a digital planetarium is valuable for a number of reasons. Difficulty capturing live footage, so CG is the norm.

- Polar coordinate compositing, not necessarily available in your favourite compositing software.



Fulldome standards



- 30 fps (not NTSC standard of 29.97 fps).
- Frame resolution has been increasing over the years from 2400, 3200, 3600, ... current standard is 4096 pixels square = 16 MPixels.
- Square pixels, reflects the CG (instead of filmed) history of fulldome.
- Frames are usually created/archived in a lossless format, historically TGA. Better more powerful images formats are starting to be used, in particular, PNG and less commonly TIFF.
- Audio: generally surround 5.1, quite a bit of variation between installation capability ranging from simple stereo to 7.1.
- Fisheye orientation with “front” at the bottom of the image. This is a view of the image from within the dome rather than from above the dome.
- Not uncommon to place information within the unused portions of the circular fisheye frame. Eg: frame number, sequence name, author ... no standards.

Software tools

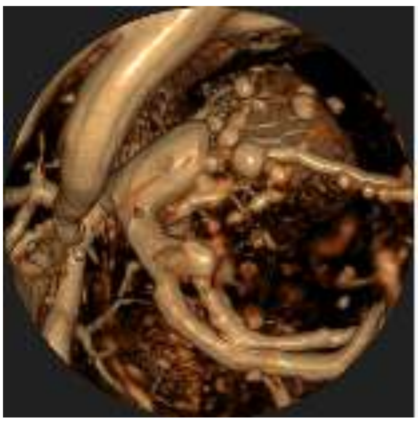
- cube2dome
- cube2sph
- sphere2fish
- Image stitching
  AutopanoPro, ptgui, Gigapan ... many others.
- warpplayer
- Quartz Composer and pbmesh [Will discuss QC in more detail later]
- Format of warp mesh files
- meshmapper

cube2dome

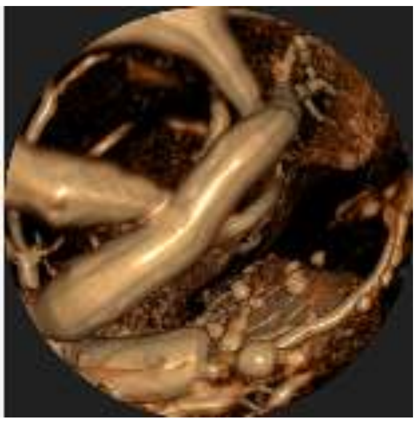
- Unix (eg: Linux, Mac OS-X) command line utility intended to create fisheye frames from a sequence of cube maps.
- Employs a very general file naming convention.
- Allows the fisheye to be orientated in any direction (see -ht and -vt).

```
Usage: cube2dome [options] filemask
       filemask    A C style filename mask, must contain %c then %d for the
                   face name [l,f,r,b,t,d] and frame number. For example:
                      frame_%c_%04d.tga
                   will look for cube faces with one of [l,f,r,b,t,d] substituted
                   for %c and the frame number substituted for %04d (zero padded)
                   New from version 1.06: file mask can now contain two %c fields.
                   This allows frames to be split by directories, for example:
                      sample_%c/frame_%c_%5d.tga
Options
       -w n        sets the output image size to n, default: twice image width
       -a n        sets antialiasing level to n, default: 2
       -n n        starting frame number, default: 0
       -m n        frame number stepping count, default: 1
       -s n        last frame (inclusive and optional)
       -j n        save as jpeg with specific quality (default: TGA, quality 100)
      -bc r g b a  sets the background colour, default: 32 32 32 255
      -mc r g b a  sets the colour for missing faces, default: 0 0 255 255
      -ht n        rotate camera about up vector, default: 0
      -vt n        rotate camera about right vector, default: 0
      -fa n        fisheye angle in degrees, default: 90
      -vp x y z    sets the view position (x,y,z) for offaxis fisheye
       -o s        overlay with the tga file "s" (expects alpha channel)
      -bf n        fade towards the back of the fisheye, n = fadepower
       -d          verbose, debug mode
```

cube2dome example



-vt 0

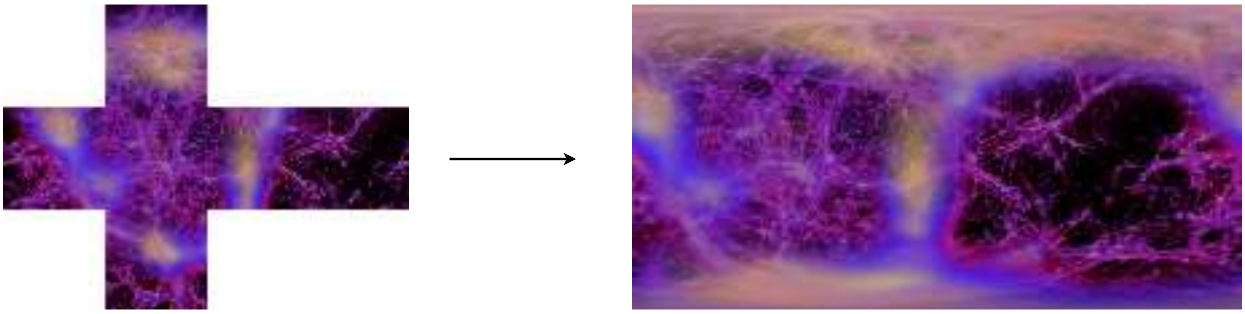


-vt 90

cube2sph

- Similar concept to cube2dome.
- Only applies to a single frame so need to script it to use with multiple frames of a movie.

```
Usage: cube2sphere [options] filemask
filemask should contain %c which will substituted with each of [l,r,t,d,b,f]
For example: "blah_%c.tga" or "%c_something.tga"
Options
   -w n     sets the output image width, default = 4*inwidth
  -w1 n     sub image position 1, default: 0
  -w2 n     sub image position 2, default: width
   -h n     sets the output image height, default = width/2
   -a n     sets antialiasing level, default = 1 (none)
   -s       use sine correction for vertical axis
```

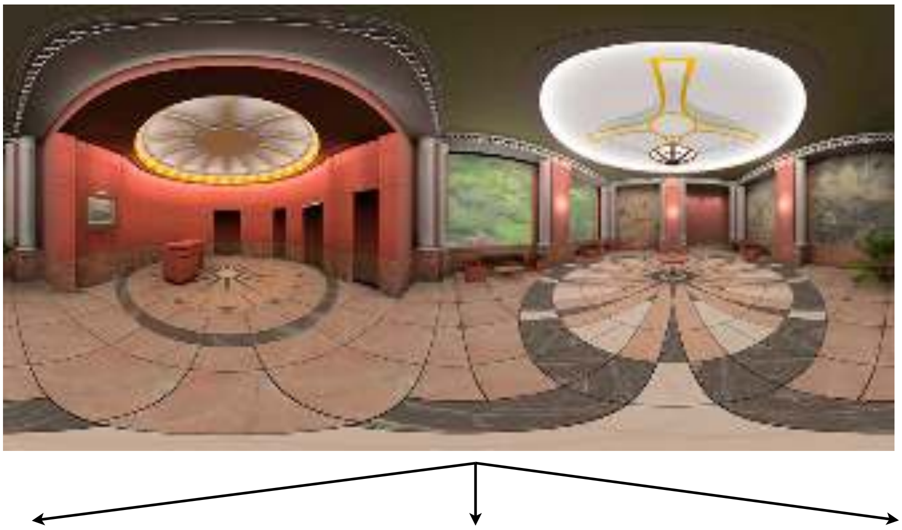


sphere2fish

- Unix (Linux and Mac OS-X utility).
- Takes a spherical projection as input (eg: frame from LadyBug-3 camera) and creates a fisheye projection.
- There are of course an infinite number of possible fisheye images, the exact orientation of the fisheye is controlled by -c option.

```
Usage: sphere2fish [options] sphereimage
Options
   -w n        width and height of the fisheye image, default = 500
   -t n        fisheye aperture (degrees), default = 180
               maximum is 360 degrees
   -c x y      center longitude and latitude of fisheye, default = 0 0 degees
   -u x y z    up vector, default = (0,0,1)
               required for latitude = +- 90 degrees (poles)
   -f          rectangular fisheye instead of circular
   -a n        antialiasing level, default = 1 (no antialising)
```

sphere2fish example



sphere2fish lobby.tga

sphere2fish -c 90 0 lobby.tga

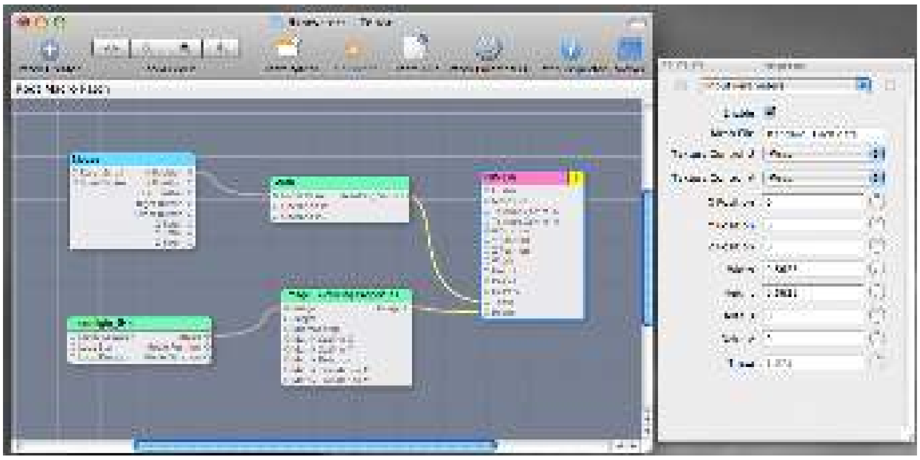
sphere2fish -c 0 45 lobby.tga

warpplayer

- Takes a movie and a mesh file (looks for "default.data" by default) and plays the movie with each frame applied to the warping mesh.
- Optionally launchs fullscreen.
- Totally abstracts the input movie projection type and the output projection type from the application. Contained entirely within the warping file.
- The image type specified in the warp file header only indicates what types of navigation is appropriate.
- Ability to navigate within movie frames, eg: spin a fisheye frame about it's center.
- Geared mainly towards movies consisting of cylindrical or spherical projections, provides horizontal panning.

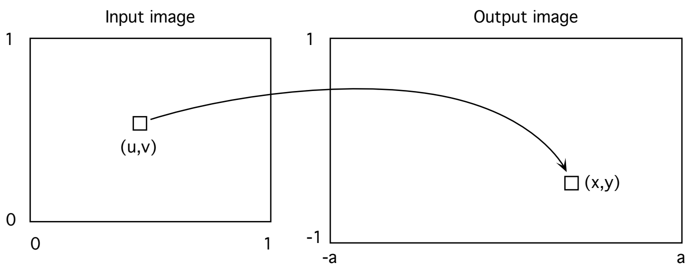
Quartz composer and pbmesh

- Implements warping within Quartz Composer.
- Uses the same warp mesh files as warpplayer, (and other tools).
- Ideal for scripting exhibitions with interactive elements, dynamic content, randomised components, transitions, etc.
- See later...



Format of a warp mesh file

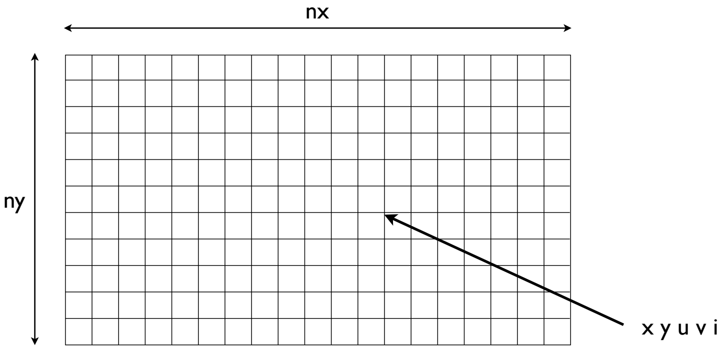
- A regular grid.
- Each vertex consists of a position (x,y), texture coordinate (u,v), brightness (i).
- The positions are in normalised screen coordinates, -aspect to aspect horizontally, -1 to 1 vertically. Will appear full screen when viewer with the appropriate orthographic camera.
- Texture coordinate are from 0 to 1, these refer to the input image.
- Brightness values are from 0 to 1 (0 is black). Implemented as a multiplicative factor eg: glModulate().



Format of a warp mesh file

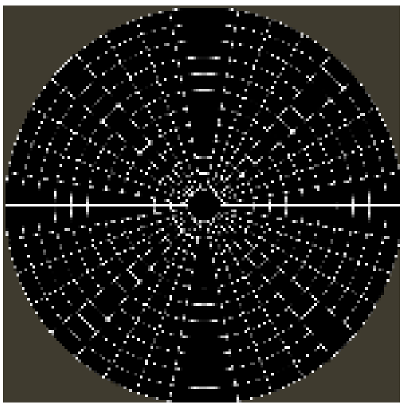
- First 2 lines consist of a header.
- First line is the type of input image.
  1 = planar image
  2 = fisheye image
  3 = cylindrical panorama
  4 = spherical panorama
  5 = cubic map
- Second line is the dimensions of the grid (nx,ny).
- Remaining lines are the node values (x,y,u,v,i).

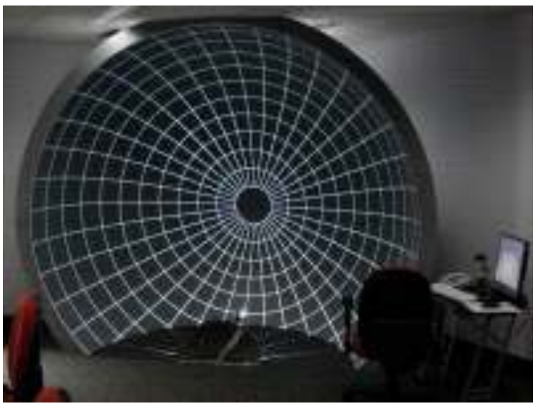
```
2
100 60
-1.77778 -1 0.1608250 0.867370 0.530521
-1.74186 -1 0.1433430 0.850423 0.59831
-1.70595 -1 0.1264890 0.832400 0.670399
-1.67003 -1 0.1103900 0.813376 0.746497
-1.63412 -1 0.0951536 0.793427 0.826292
-1.59820 -1 0.0808694 0.772634 0.909462
-1.56229 -1 0.0676125 0.751080 0.995681
-1.52637 -1 0.0554441 0.728845 1
-1.49046 -1 0.0444134 0.706012 1
-1.45455 -1 0.0345590 0.682660 1
-1.41863 -1 0.0259108 0.658869 1
-1.38272 -1 0.0184902 0.634716 1
    :
    :
    :
    :
    :
    :
    :
    :
    :
    :
1.63412 1 0.545315 0.959740 0.0238356
1.67003 1 0.544729 0.961201 0.0225102
1.70595 1 0.544156 0.962582 0.0212905
1.74186 1 0.543600 0.963886 0.0201703
1.77778 1 0.000000 0.000000 0.0206006
```



meshmapper

- Given a knowledge of the geometry of the system it creates a warp mesh.
- The user enters the geometric information as best they can and then adjusts the less certain parameters until a test patter looks correct.
- Test pattern is usually a polar grid.
- Parameters includes
  - the position of the components: projector, mirror (dome defines the origin).
  - radius of dome and spherical mirror.
  - optics of the projector: throw, aspect ratio, offset.

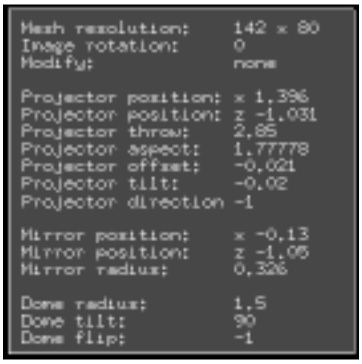

Polar grid


Result in the iDome

meshmapper

- Current settings are stored in a file "last.cfg"
- Keyboard commands allow the user to change the parameters.
- Current settings displays in the top left of screen.

```
projector_pos 1.396 0 -1.031
projector_throw 2.85
projector_aspect 1.77778
projector_offset -0.021
projector_angle -0.02
mirror_pos -0.13 0 -1.05
mirror_radius 0.326
dome_radius 1.5
dome_angle 90
dome_flip -1
```




```
Usage: meshmapper [options] tgaimage
Options
                  -h     this text
                  -f     full screen
                  -d     verbose/debug mode
                  -r s   read existing geometry file
Left mouse button for popup menus

Keyboard
          h      camera home
      p,d,m      choose projector, mirror, or dome
    X,Z,x,z      modify position (projector or mirror)
        R,r      modify radius (mirror or dome)
        A,a      modify tilt angle (projector or dome)
        T,t      modify throw (projector)
        O,o      modify offset (projector)
          f      modify dome top (dome)
          w      windowdump
      esc,q      quit
```

LadyBug-3 camera

- 6 cameras with wide angle lenses.
- Each camera is 1600x1200 pixels.
- Supersedes the earlier LadyBug-2 camera.
- Main difference, besides resolution, is the LadyBug-2 was 30 fps while the LadyBug-3 is maximum of 16 fps.
- Supplied with standard tripod mount and a car roof mount (aka Google street view).


Ladybug-2

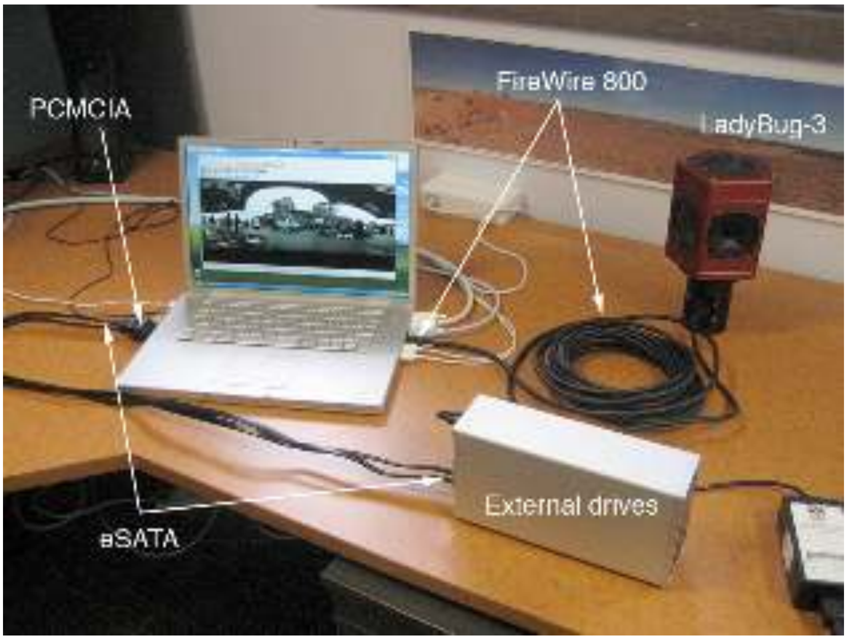

Ladybug-3

Capture extents

- Captures 360 degrees horizontally (longitude).
- Captures from the north pole to approximately -50 degrees vertically (latitude).

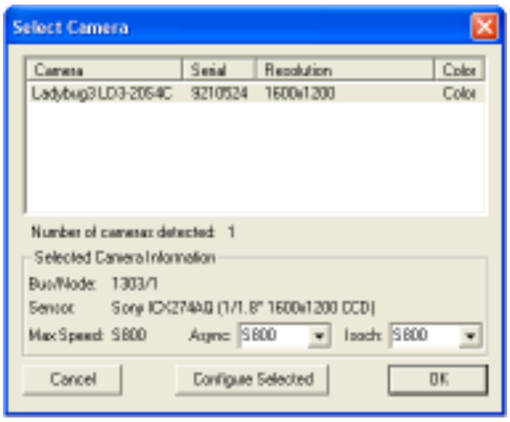
Data transfer information

- Interface from the computer to LadyBug-3 camera is Firewire 800. So 800MBits/sec.
- For 95% jpeg compressed data to disk requires around 70MBytes/sec. Generally not achieved with standard laptop disks.
- Generally use external PCMCIA or eSATA drive.

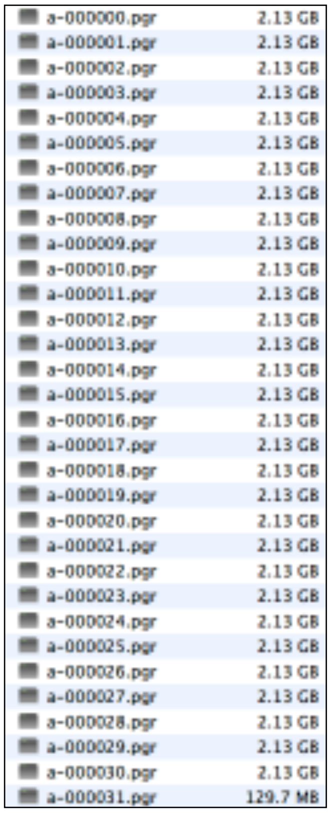


LadybugCapPro

- The LadyBug-3 camera is supplied with various bits of software including software API and sample code for programmers who wish to develop their own applications.
- Otherwise everything can be accomplished with the LadybugCapPro software.
- This includes controlling - configuring- capturing using the camera, as well opening existing recordings for movie frame exporting.
- When operating the camera it is important that the maximum speed is selected as S800 and not S400 (FireWire-800 vs FireWire-400)



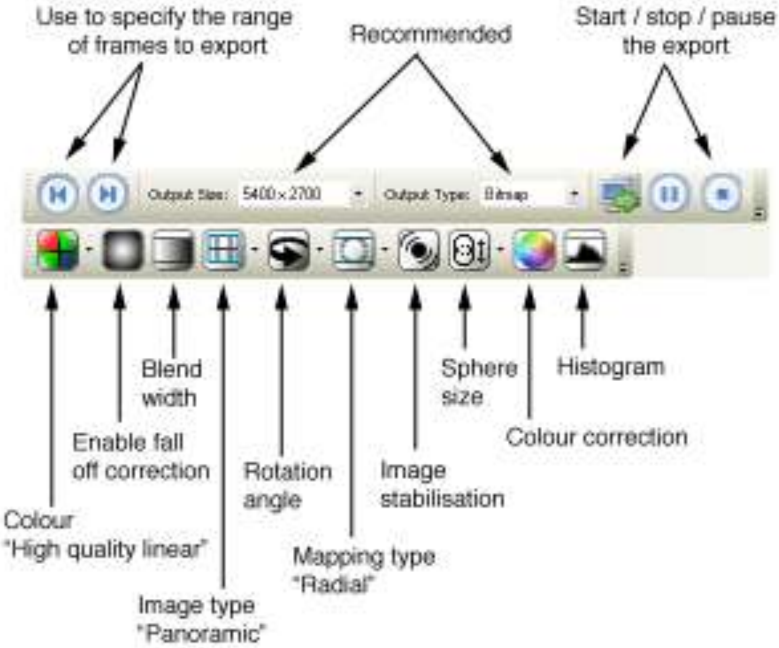
Opening an existing recording



- When opening an existing recording, choose one the first PGR file in the sequence.
- Each recording is split into 2GB portions, the last portion will be some other size.
- The LadybugCapPro will automatically read all portions.
- “pgr” files are a proprietary format, sample code is provided for reading those files.

Exporting frames from an existing recording

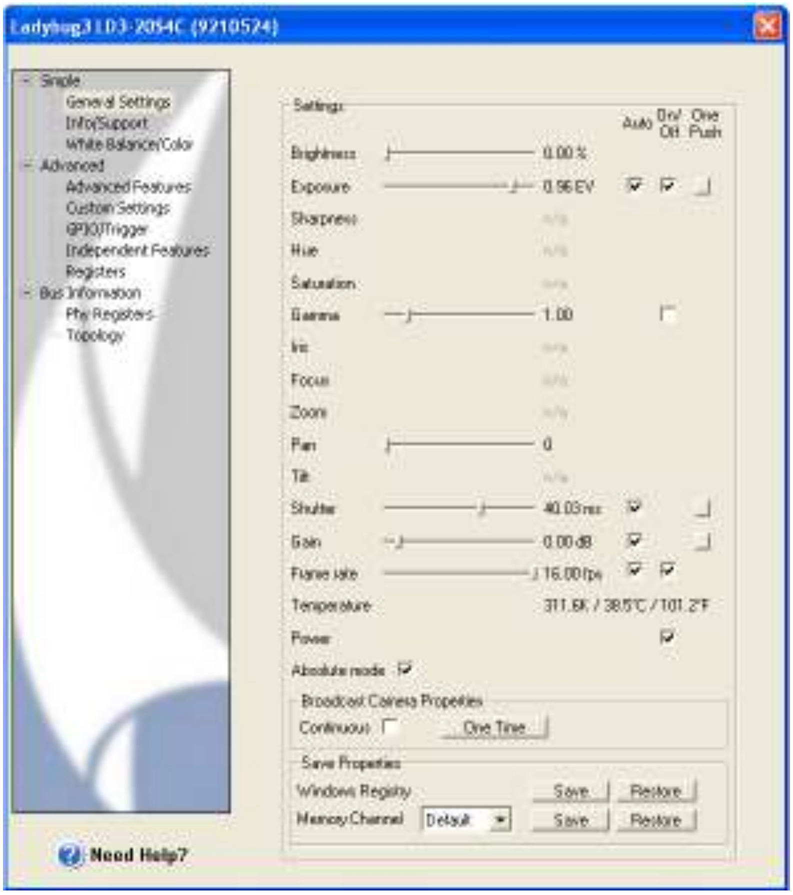
- The most important settings are: colour quality, fall off correction, image type, mapping type, and sphere size.
- The default blend width of 100 is generally ok.
- Rotation angle identifies where “front” is.
- Colour correction can generally be performed best in post production.
- The histogram is largely for ones information.



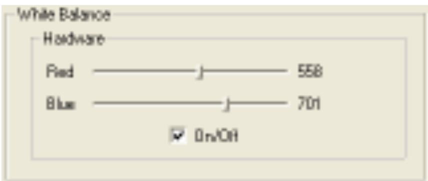
LadyBug-3 recording

- Most of the controls such as colour, falloff correction ... etc can adjusted in image export, see previous sides.
- The important camera controls are all accessed from the camera configuration button.

Camera configuration



- Important to get a feel for these settings before any actual filming.
- Besides the "General Settings" the next most important is the "White Balance/Colour".



Miscellaneous comments

- Generally want to position the Ladybug-3 camera at the same height as the intended viewer.
- For high quality recordings the storage is approximately 4GB per minute, or 1/4TB per hour.
- When recording for the iDome a target resolution of 4096x2048 is appropriate. Note it is generally better to export at the full 5400x2700 and scale down to 4096x2048 to introduce some antialiasing.
- There are a number of technologies for intelligently upscaling the time base from 16 to 30fps. (Note 30fps is more standard than 25 for domes and cylindrical projection). The author has used Kronos in FinalCutPro but there are others, some are built into compositing packages.
- In dark environments and long shutter setting there may be excessive motion blur artefacts for "fast" moving action.
- The sun, or other bright lights, are always in shot on one camera.
- When compositing in other elements, they also need to be spherical projections.

Quartz Composer

- Tool for presenting (optionally interactive) iDome content.
- Mac OS-X only.
- Free part of the Apple Xcode package, an optional install on (Snow) Leopard DVDs.
- pbmesh patch implements image warping.
- Additional patches (free and commercial) available from kineme.net web site.
- Examples of fisheye and spherical playback.

Further reading

- Spherical mirror FAQ:
  http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/domemirror/faq.html
- Yahoo group: small_planetarium
  http://tech.groups.yahoo.com/group/small_planetarium/
- Yahoo group: fulldome
  http://groups.yahoo.com/group/fulldome/
- International Planetarium Society
  http://www.ips-planetarium.org/
- Wikipedia page on fulldome
  http://en.wikipedia.org/wiki/Talk:Fulldome
- DomeFest - International (juried) fulldome festival
  http://www.domefest.com/