# Huge Panoramas

*There are several ways to render a spherical panorama from an artwork in Bryce. The document width is limited to 4000 pixels but there are means to create larger panoramas.*

## Introduction

The *Document Size* for a render within the Bryce workspace is limited to 4000 pixels wide, the height can exceed 4000 pixels. A document can be rendered to disk larger but the *Render to Disk* option is not very reliable and it only uses one processor core on a computer featuring a multi-core processor.

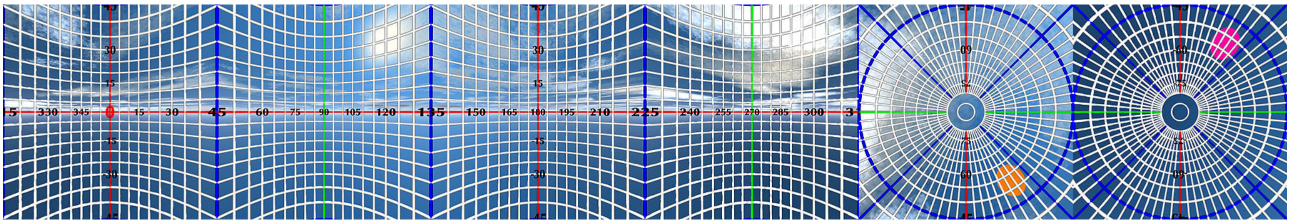
With Bryce come four tutorials that show four methods to make renders that can be used to create spherical panoramas (`\Bryce\Content\Tutorials\Horo Wernli\Start.html`): Cube, Hemispheres, Mirrorball and Skydome. All renders are limited to 4000 pixels.

There is the *Spherical Mapper* and the *Scene Converter* — lenses that can be bolted in front of the Bryce camera; however, both are limited to 4000 pixels. On the plus side, the *Spherical Mapper* renders a finished panorama in the spherical projection and the *Scene Converter* in the Angular Map projection. Both renders can be used directly as panoramas.

To create larger panoramas, several renders must be combined and an additional program to stitch the parts together and transform the result into the desired projection is needed.

## Cube Method

With this method, the six faces of a cube are rendered. Each face can be 4000 by 4000 pixels at most. Because the panorama is a sphere within the cube, the biggest spherical panorama that can be created is 10,026 x 5013 pixels.



*From left to right: North (0, Y=0°), East (1, Y=90°), South (2, Y=180°), West (3, Y=270°), Zenith (4, X=-90°, Y=0°) and Nadir (5, X=90°, Y=0°). The numbers are the image numbers to identify the individual faces when assembling and converting into the spherical (also called equirectangular) projection.*

There are a few programs (some free) that can assemble and transform the cube faces to a panorama. I experimented with several programs in the past and now use *Pano2VR*.

For the maximum spherical panorama size of 10,000 x 5,000 pixels 96,000,000 square pixels must be rendered to get a panorama of 50,000,000 square pixels. This is not very efficient. Of course, the panorama can be created bigger, but there are only those 50 Mega pixels. This means you have to render almost double as much pixels as you finally get.

The surface that has valid pixels is half the width of a side squared, then multiplied by 4 x PI:

$$A = (a/2)^2 \times 4 \times PI = (4000/2)^2 \times 4 \times PI = 2000 \times 2000 \times 4 \times 3.14159 = 50,265,440.$$

The Camera FOV must be set to 112.5° at Scale 100% to get a horizontal and vertical angle of view of exactly 90°.

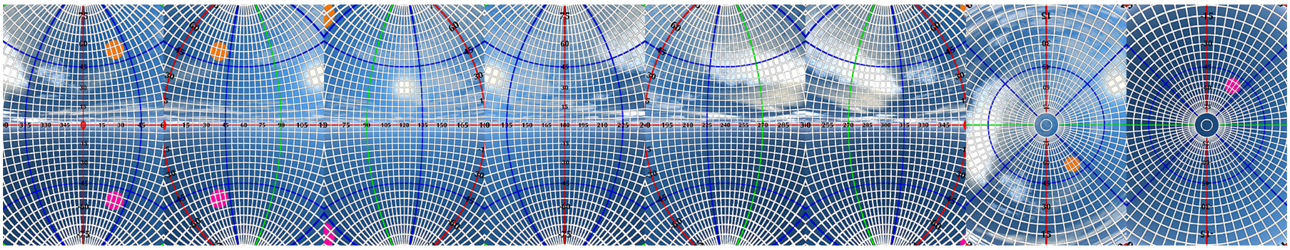
## Fisheye Lens Object (FLO) Method

With the *Lenses and Filters* product comes (among other lenses and filters) the FLO. This lens is a close approximation to a photographic full frame fisheye lens. A full frame fisheye lens has a diagonal angle of view of 180° (a circular one horizontal and vertical 180°).

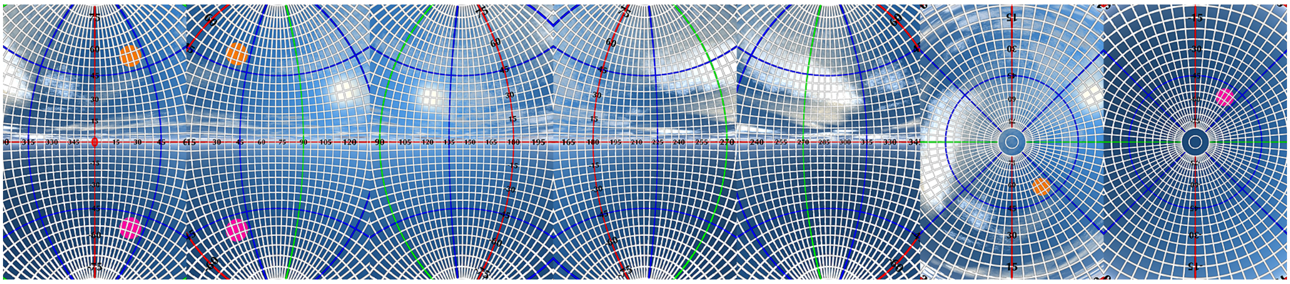
To acquire photographic spherical HDRI panoramas I use an FX camera with 24 MPx and a 16 mm full frame fisheye lens. The image aspect ratio is 2:3 vertical (4016 x 6016 pixels) and I take horizontal six images in 60° steps, and additionally several zenith and nadir shots. The 60° interval gives an overlap of ±12°.

I set up the document size to 4000 x 6000 px (no limit vertical) to have the same image size as my photographs. The document that comes with the *Lenses and Filters* product sets the Camera *FOV* for this aspect ratio and the FLO to 74.2° at *Scale* 100% and says that the FLO is roughly a 13 mm full frame fisheye lens; actually, the focal length corresponds to 12.14 mm.

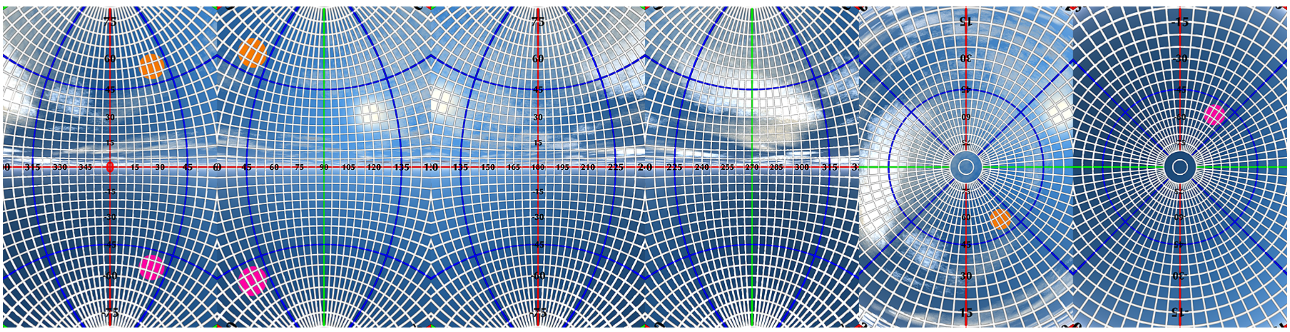
I first rendered six images at 60° steps horizontally but I got an overlap of ±22.5°, which is very generous (8 renders).



I tried five images at 72° steps and still got ±16.5° overlap (7 renders).



I finally settled at four images at 90° with a very narrow overlap of ±7.5° (6 renders).



*From left to right: North (Y=0°), East (Y=90°), South (Y=180°), West (Y=270°),*
*Zenith (X=-90°, Y=0°) and Nadir (X=90°, Y=0°).*

When photographing, the camera may be tilted a bit in any or all of the three axes, each time a bit different. Some space must be allowed for that and the overlap, ±12° works fine, it is not too much. The Bryce camera is rock stable and always at the same position and there is no need for much overlap.  Besides, there is neither a tripod nor a shadow of it on the ground.

The stitching software I use (PTGui) made a mess when creating the *Control Points* automatically to align the individual renders and I had to set the control points manually to connect the pictures, a tedious work which took very long. Then the *Control Points* were *Optimized* several times until the maximal error was below 2 pixels.

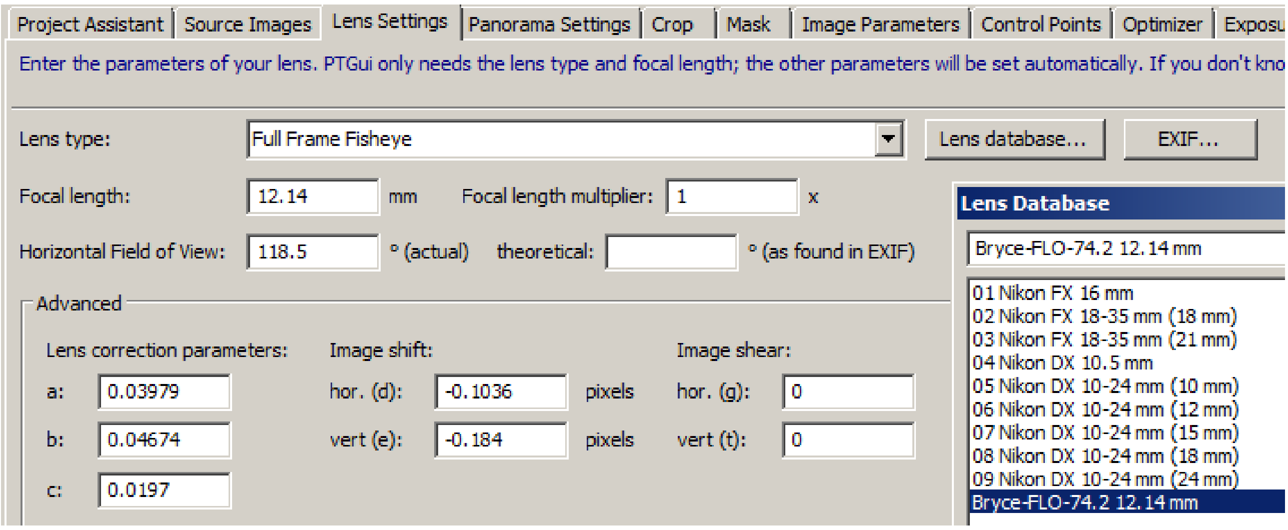
By optimising the control points, the stitching program could figure out what sort of lens was used and what errors it has. The lens parameters can be saved in the *Lens Database*. This database is pure text and can be edited (i.e. sort the order if the lenses stored). It can be found here: C:\Users\<you>\AppData\Roaming\PTGui\lensdb.ptl. Below the line:

```
"Bryce-FLO-74.2" 118.5181392736657 3 0.03979213257924842 0.0467436025274227 0.01969627307839617 -0.1036051324250293 -0.1839524823989604 0 0
```

The line above contains the data for the Bryce FLO lens at FOV 74.2° and Scale 100%, below it is shown in a better readable size.

```
"Bryce-FLO-74.2" 118.5181392736657 3 0.03979213257924842 0.0467436025274227
0.01969627307839617 -0.1036051324250293 -0.1839524823989604 0 0
```

The *Focal Length* 12.14 mm is not included and must be entered manually. The first entry after the name is the *(actual) Horizontal Field of View*, which matches the 118.5° in the render. 3 is the pre-defined lens type, the list starts with 0 so it is the fourth. Then follow the *Lens correction parameters a*, *b* and *c*, the *hor*izontal and *vert*ical *Image shift*s and finally the *hor*izontal and *vert*ical *Image shear*. The picture shows the *Lens Settings* with these parameters loaded.




I rendered some other panoramas and stitched them by using these camera settings and the resulting panorama turned out perfect — after all, the camera and lens were set exactly the same way. So it turns out that using these camera settings, the panorama stitches perfectly and can be directly created, without going through the time consuming control points and optimisation steps.

For the maximum spherical panorama size of 12,148 x 6,074 pixels 144,000,000 square pixels had to be rendered to get a panorama of almost 74,000,000 square pixels. This is not very efficient. Of course, the panorama can be created bigger, but there are only those 74 Mega pixels. This means you have to render almost double as much pixels as you finally get.

## Comparing the Cube and FLO Methods

From a render efficiency point of view, both methods must render nearly the double surface than makes it into the final spherical panorama (Cube: 1.92, FLO: 1.95). The FLO panorama gets 1.47 times more pixels than the Cube one, but renders 1.5 times as many pixels as the Cube one. Therefore, it can be concluded that the methods have matching efficiency but the FLO method results in a bigger panorama.

## Camera and Document Settings

The Bryce camera must be set according to the method used. The *X*, *Y* and *Z Position* can be set as desired. This is the position where the nodal point of the camera lens is and it must remain the same throughout the render series.

The *Y Rotation* must be set to 0°, 90°, 180° and 270° for the four horizontal renders. For the zenith *X Rotation* -90 and nadir *X Rotation* 90°, keeping *Y Rotation* at 0°.

### Cube Method:

*FOV* = 112.5°, *Scale* = 100%; document aspect ratio 1:1, max width 4000 px.

### FLO Method:

*FOV* = 74.2°, *Scale* = 100%; document aspect ratio 2:3 (width x height), max width 4000 px.

## Links

This list is not exhaustive and only gives some examples.

### Tutorial:

BRYCE 6 Mini Tutorial 21: Render > 4000 pixels.

https://horo.ch/raytracing/tuts/pdf/minitut21_en.pdf

(or https://horo.ch > Raytracing > Tutorials > Bryce Page 2)

### Calculator (Spherical, Mirrorball, horizontal and vertical Cross, Cube Faces)

https://horo.ch/science/calc/panocalc_en.html

(or https://horo.ch > Science > Panorama)

### Products mentioned:

Spherical Mapper: http://www.daz3d.com/bryce-7-pro-spherical-mapper

Scene Converter: http://www.daz3d.com/bryce-7-pro-scene-converter

Lenses and Filters: http://www.daz3d.com/bryce-7-1-pro-lenses-and-filters

### Selection of programs that can handle Cube Faces:

Pano2VR: http://ggnome.com/pano2vr

GoCubic: http://ccm.net/download/download-18492-gocubic

AutoPano: http://www.kolor.com/autopano/

AutoPano: http://autopano.de.softonic.com/

### Selection of Stitching Programs:

PTGui: https://www.ptgui.com/

Hugin: http://hugin.sourceforge.net/

April 2016/horo