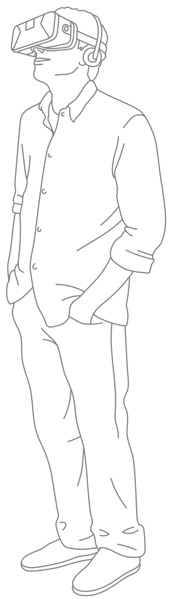


GUIDE TO VR

chaosgroup.com | v-ray.com | labs.chaosgroup.com

CHAOSGROUP | LABS

version 01

contributors: Christopher Nichols, Kim Baumann Larsen, Justin Denton
editor: Henry Winchester, Colin McLaughlin, David Tracy, Dariush Derakhshani
layout and design: Melissa Knight, Andrey Katanski

Understanding and creating pre-rendered Virtual Reality content in V-Ray

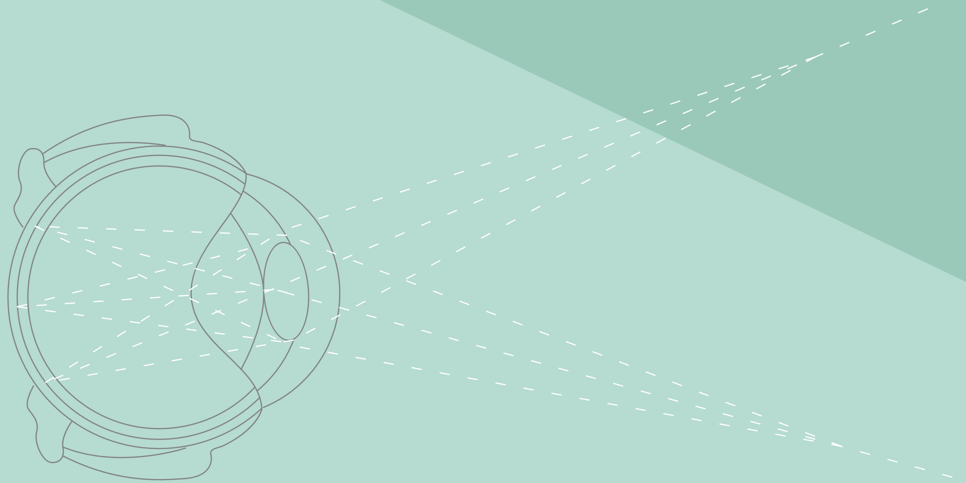
Abstract

This paper examines the process of creating three-dimensional panoramic images and videos for use with virtual reality headsets. We will explain and show how to create pre-rendered Virtual Reality content. We look at the unique challenges of creating content for virtual reality, including camera positioning, the top and bottom poles, scale and interocular distance, and field of view. The paper concludes with instructions on creating VR content with V-Ray for Samsung's Gear VR and Google Cardboard.

Disclaimer: The world of virtual reality is evolving rapidly, and at the time of writing (late 2015) a number of consumer headsets are due for imminent release. Therefore the information in this paper may not be up to date. For regular updates on virtual reality in V-Ray, we recommend visiting the Chaos Group Labs site: https://labs.chaosgroup.com

If you have a solid understanding of virtual reality, and want to jump straight to a quick guide on using V-Ray to create VR stills or video, skip to appendix A or B respectively.

Understanding stereo panoramic images and videos



VR then and now

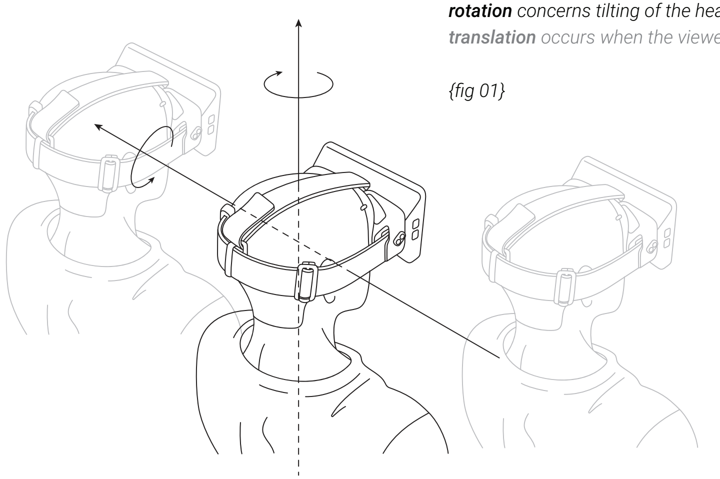
VR stands for Virtual Reality. VR has been around since the early 90s. The term implies that the goal is to replace many of your senses with virtual ones, so that the virtual environment will feel as real as possible. While we think of the five senses as vision, hearing, smell, taste, and touch, the VR headset combines vision with another important sense: balance.

Once the headset is on, vision can be replaced by feeding the user an image that is tied to the motion of their head. If you turn your head to the right, sensors in the headset can recognize this motion and then apply it to the viewer's current point of view. For this reason, any delays between natural head motion and a shift in virtual perspective will result in a disruption in balance. This can lead to motion sickness.

For years, the latency gap between vision and headset made VR unusable. But today, we are in the middle of a VR revolution primarily because we have overcome this issue. There are many types of VR headsets these days, and they all provide very good results with very low latency. As a primer: Latency is tracked in measures of frames per second or hertz, and often refers to how quickly an image refreshes when you turn your head. The higher the hertz, the lower the latency, the better the feedback . Generally, you are looking for at least 60 Hz for smooth virtual experiences, though many headsets - Oculus, for example - recommend somewhere between 75 to 90 Hz.

The different types of VR

There are two types of head motion to be aware of: rotation and translation. Gyroscopic sensors and accelerometers, such as the ones found in phones today, do a great job of sensing head rotation. That, in itself, can give the user an interesting VR experience if all they need to do is look around. If they want to move their head side to side, up or down, or even walk around, additional sensors are required. These sensors are usually set up outside the headset to track its position in the room and translate that information back to the user's position in the virtual world. ***{fig 01}***



rotation *concerns tilting of the head up and down / side to side*
translation *occurs when the viewer moves through the space*

{fig 01}

The Oculus Rift Development Kit 2 (DK2) VR headset, for example, uses a small webcam-like camera – that is usually attached to the monitor on the desk – that tracks infrared markers on the headset. Since this is only a single camera, it will only give the user about a meter of overall head motion information - enough to move your head side to side or look around while sitting. The HTC Vive uses two cameras that can track a space approximately 4.5 m squared i.e.the size of the average living room. This allows the user to walk around in a small area.

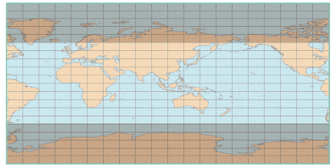
When it comes to content, there's a crucial difference between a VR headset that does rotation and translation and one that only does rotation: a headset that does both requires a full 3D scene, run either through a 3D game engine, or a deep imaging system such as lightfield or point cloud. For the latter, the author has to know how to build a full real-time environment, or have a system on hand that can play back very large and complex lightfield renders. If the only goal is to enable a user to look around, rotation is all that is needed. You can accomplish this by feeding two spherical images (one for each eye) to the headset. Real-time is unnecessary for this application, and you can work with pre-rendered material.

Many argue content that does not allow for translation (where the viewer can move in the space) does not provide a true VR experience. However, as of this writing, creating content that simply feeds left and right spherical images to each eye (either stills or video) is a much simpler way to tell a story. While the user can not walk around the environment, they can still look around and feel a part of the space. It also allows the author to create the content using technology that they are already familiar with, as well as play it back on a wider range of devices.

Spherical images: latlong vs cubemap

When making pre-rendered content, you need to generate a spherical image for each eye – this looks a lot like the domed projection surface of a planetarium. However, a common issue that results when making an image that represents a full 360 degree snapshot of the world around you, is that the resulting image is inherently 2D. To fix this, you have to unwrap the map, and represent it in a flat manner. The traditional way of doing this is called a latlong, which stands for latitude-longitude.

The nice thing about latlongs is that they are able to represent the entire world in a single image, only with one seam running down an edge of a longitude line, and another two on the top and bottom latitude lines, where the image turns into a single point. ***{fig 02}*** This format makes it much easier to see the environment as a single image, and this in turn makes it easier to edit. The issue with latlong images is that they are very distorted; 66% of the image represents the top and bottom poles, which are quite distorted and generally not the area of focus. ***{fig 03}***



***latlong** seams are along the top and bottom plus vertical edge*

{fig 02}

***66%** of image form the poles once wrapped into 360^o view*

{fig 03}

Cubemaps are much more efficient at representing 360 degree images. As their name suggests, Cubemaps place the images on the inner surfaces of a virtual cube. The six square images are then placed side by side into a single, 6:1 aspect image. There is little distortion with cubemaps since each of the six side-by-side images appear asa 90 degree FOV image, with one square facing in each direction. ***{fig 04}***

Compared to the latlong, only two of the six images are used for the top and bottom poles, which accounts for only 33% of the entire cubemap image. ***{fig 05}*** The drawback with cubemaps is that, because each edge of the cube represents a seam, there are far more seams compared to the latlong - 12 in all. Generally, you won't see edges in the images if the VR player is good, and the images are high quality. But if you have compression artifacts such as with the popular video codec H264, those edges will tend to show up as seams in the VR playback.

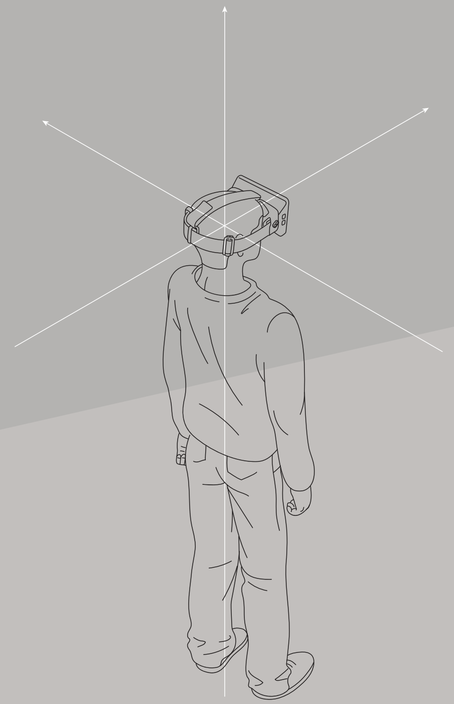
***cubemap** seams are along the edge of each square*

{fig 04}

***33%** of the cubemap form the poles*

{fig 05}

Understanding stereography for a 360° image or video

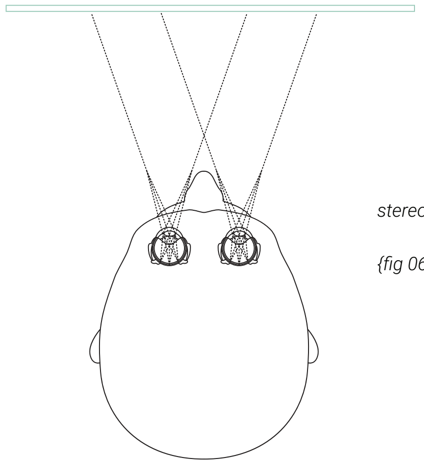


Stereography is the act of delivering two unique images from two different cameras. These cameras are slightly offset from each other to mimic what each eye (left and right) would see, essentially giving the viewer an experience that appears to be 3D. We will cover how to actually do this in a later section of this paper. This section explains how stereo images for a 360 degree VR view are different from stereo images for a flat screen.

When rendering stereo for a flat screen, such as a 3D movie, setting up a stereo camera simply involves rendering from two properly offset cameras: left and right. Generally the offset is set at 6.3 cm apart, and this figure is referred to as interocular distance, or IOD. It is sometimes called interpupilar distance. However, this offset value is often cheated in feature films, even on a shot-by-shot basis. The reason for this is that a close-up created with a long lens may give a false sense of scale (making a person's head seem giant) if a standard 6.3 cm offset is used. Stereographers will reduce the interocular distance to a value as low as 2 cm or less for such close-ups.

For a VR experience, there is no such thing as a "close-up." The viewer experiences what is happening as if through their own eyes. For that reason, the interocular distance needs to stay at the correct physical scale for the scene. It should only be cheated if you want the user to feel as if their own scale has changed within the scene.

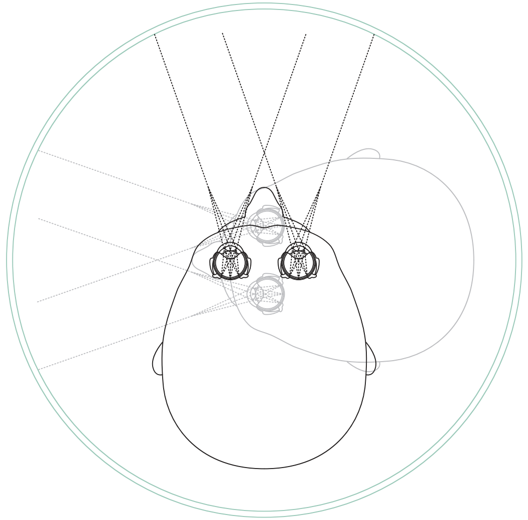
The next point to discuss about watching stereo on a flat surface, such as a theater screen, is that the viewer is looking forward the entire time. In essence, the viewer is looking through a window, with little rotation to their head as they watch. In this case, creating a stereo image with the offset cameras pointing straight ahead at the action works well. ***{fig 06}***



stereo image viewed on a ***flat screen***

{fig 06}

In contrast, with a VR experience the stereo effect needs to work no matter which way the viewer is looking. By using the standard offset cameras, when looking forward, your left and right eyes are correct. However, if you look backward, and the offset is the same, your left and right eyes are swapped. Therefore, a special type of stereo camera needs to be adapted for 360 degree images. Not only do the cameras need to be offset from each other, but they have to maintain that same offset as the camera rotates to view the whole 360 degree image. ***{fig 07}***



stereo image viewed in a ***360° VR environment***

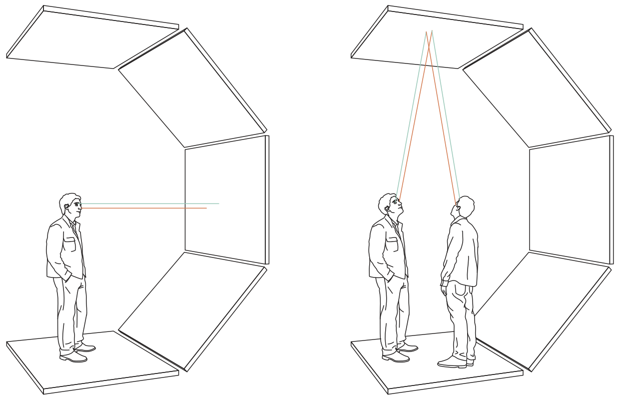
{fig 07}

Imagine doing a live action still in stereo. If you were to try and capture the experience, you may shoot in four directions (left, front, right, back; each with a stereo offset). Since the cameras not only rotate, but are also offset, this can lead to some difficult stitching. To make this easier you can add more directions so that the angular difference between each pair of stereo cameras is less. Going to six, eight, or 12 directions instead of just four would make the 360 degree stitching more accurate, and the overall effect would be better. Now imagine a special camera rig with nearly infinite cameras, each capturing a vertical column of pixels equivalent to each column in the rendered scene. When using raytraced rendering, this is actually possible. The result is that when rotating your head horizontally, the correct stereo effect is visible all the way around.

The challenge with the poles

If you simply only looked forward, side to side, or back, this stereo effect would work well. However, for a true VR experience, the user should be able to look up and down as well. But looking up or down represents a problem when rendering stereo 360 images or videos. These issues occur around the “poles”, which, like the North and South poles of the Earth, are found at the top and bottom of the VR image.

Let’s take an example where we have a stereo 360 image. You, as the viewer, look forward and up toward the top pole. The stereo offset works fine; your left and right eyes are where they’re supposed to be and at the correct distance. Now you tilt your head back down to look straight forward. Turn your head to the left and then look up at the same top pole. Now, in relation to the previous view of the pole (straight and up), your left and right eye have a different orientation than before. ***{fig 08}***

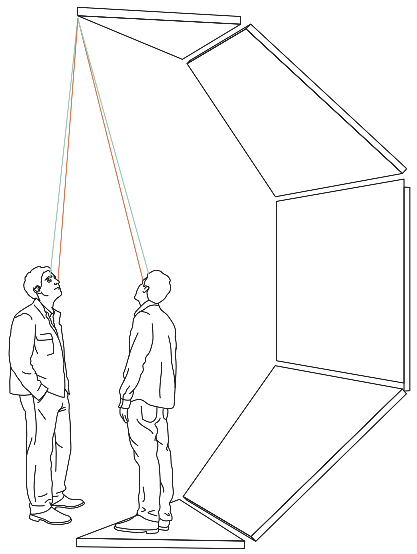


if the offset is kept the same at the poles as it is in other directions, the viewer will see and inverted image at the poles should they turn around

{fig 08}

For the stereo offset to work as it did when you first looked forward and up, your left eye needs to be at the bridge of your nose, and your right eye in the middle of your forehead.This disparity in the viewer’s eye positions makes stereo offset at the poles very tricky. The same is true when looking at the bottom pole of the stereo 360 image as well: your eye positions differ at different head rotations.

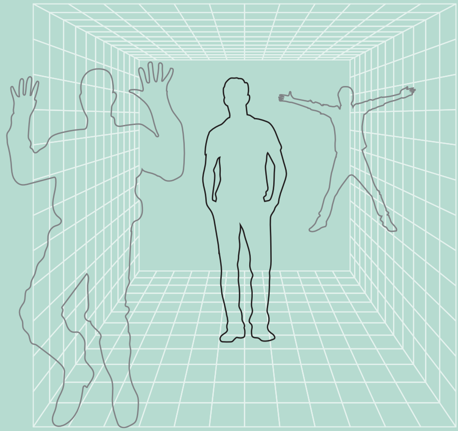
The work around for this problem is to taper off the stereo effect as you look towards the poles. The logic is that 90% of the action or interest to the user is happening near the horizon, and not at the poles, so there is no real loss of content. Using this tapering technique, when set up with proper values to give a smooth transition for the shot, the viewer will hardly notice that the stereo effect is tapered off to be fully removed at the poles. ***{fig 09}***



the solution is to reduce the offset at the poles so the interocular distance converges to a single point which looks the same no matter what angle it is viewed from

{fig 09}

How to render VR in V-Ray: Stills



interocular distance *determines how big the viewer feels within the environment;*
too big and the world seems minaturized, too small and they float within the space

{fig 10}

Understanding scale

Since VR allows the viewer to experience what it is like to be inside an environment, scale is a critical part of making that experience feel correct. This is true for environmental scale and viewer scale. To that extent, there are two dimensions that are critical to making this work. The first is the location of the camera (placed at the correct height), and the second is the interocular distance or IOD.

Interestingly, the IOD is one of the most important factors in determining a scene's scale in relation to the viewer. In part, because it determines the distance between the two cameras used to render stereo. You naturally use that distance to determine how big your head and body are in the scene. If that distance is too small, (also known as hypostereo), your body is essentially scaled down in the scene and the world appears too large. If the IOR is too big (hyperstereo), then your body is enlarged and the world around you looks miniaturized. *{fig 10}*

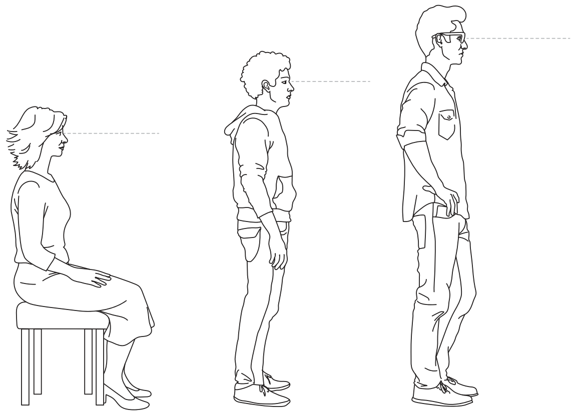
So what is the proper scale to use? IOD will vary with both age, gender and ethnicity. A default eye distance needs to be based on statistics. The most cited reference is the US Army study (ANSUR) from 1988. The median or mean IOD value in this study is 63 mm (the min and max are respectively 52 and 78 mm). Furthermore, other studies such as the Dictionary of Optometry cites 64 mm for males and 62 mm for females, indicating that 63 mm is the average. Then there is the study by Moffit giving a mean or median value of 63 mm and Waack of 63.5 mm.

The next dimension is height. For most cases, you may want to viewer to be either standing or sitting. Either way, you will have to set the camera at the proper height.

There is a significant variance between the female and male anthropometric data for both eye height from the ground in a standing and seated position. Additionally, a seated position could vary greatly depending on the type of chair used, from a low sofa-type setting to a workstation chair adjusted to a high position.

The statistical sizes of people also vary with age and ethnicity. For instance, the average height for an elderly person can be up to 80 mm lower than that of a younger person. Height also varies significantly depending on nationality. Dutch people are generally among the tallest, and Americans are generally larger than the British, who are larger than people from India or Spain and so on and so forth. Lastly, the relevant target group for the VR experience may be significantly different to the general population. For instance, if you are creating content for use in high school, or for a pro basketball experience.

The company Ergotron references US anthropometric studies that suggest a seated eye height average of 117 cm, with a female sitting eye height of 112 cm and a male sitting eye height of 123 cm. For a standing experience the figures are an average of 157 cm, with a female standing eye height of 151 cm and a male standing eye height of 164 cm. *{fig 11}*



***eye height levels** vary depending on gender, age, ethnicity and viewing position*

{fig 11}

At the same time, these are not a hard and fast rules. You may have scenes where you would want to allow the user to experience at different scales. For instance, if you want your viewer to experience a room as if they were a fly, you'd place the camera very close to a surface and set the IOD to 0.15 cm.

Reference:

"Variation and extrema of human interpupillary distance"
by Neil A. Dodgson, University of Cambridge Computer Laboratory, 2004
http://www.cl.cam.ac.uk/~nad10/pubs/EI5291A-05.pdf

"Synthetic stereoscopic panoramic images"
by Paul Bourke, WASP, University of Western Australia, 2006
http://paulbourke.net/papers/vsmm2006/vsmm2006.pdf

Ergotron Ergonomics Data & Mounting Heights:
https://www.ergotron.com/Portals/0/literature/whitePapers/english/ergonomics_arms_data.pdf

The Ergonomic Center Anthropometric Data (excerpts from the 1988 ANSUR study):
http://www.theergonomicscenter.com/graphics/Workstation%20Design/Tables.pdf

Placing the camera

Besides placing the camera in the correct location relative to the scale, there are other aspects to consider. First is that you are not trying to match a specific camera or lens - you are trying to match what a specific eye sees. Therefore, you want to use a basic camera that simply has a direction of where it is looking. Use a standard camera in your 3D application.Don’t use a V-Ray Physical Camera since it is designed to match physical lenses.

The next thing to consider is the field of view (FOV). This refers to the amount of the image you see at any one time. Since you will be rendering a full 360 degree image, the FOV attribute will be ignored. But if you want to have a general idea of what someone will see when they put the headset on and look without turning their head, use a 90 degree FOV, which for a 35 mm film back represents the look of an 18 mm lens. Again, the camera will render the full 360 degree, this is just to assist you as you place your camera in the scene.

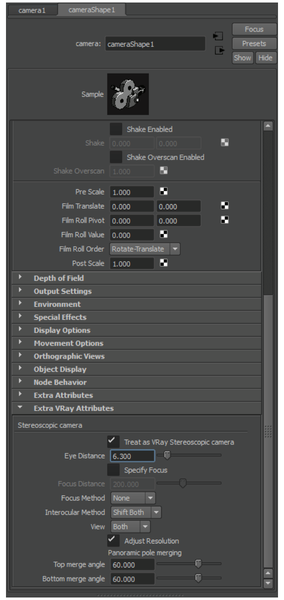
Some VR experiences initialize the viewer in a specific direction every time they place the VR headset on. This may be something to consider. They generally initialize what they consider to be the “front” position, which is important if your scene is set in a vehicle, for instance. If this position is important to your VR experience, place the camera to point in that position to set what is considered front. Also consider whether you want your user to have a seated or standing experience, and then position the camera accordingly.

Setting up the stereography

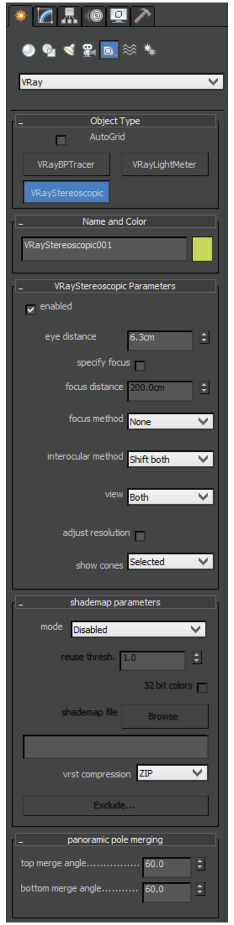
V-Ray provides a stereographic helper which is ideal for setting up stereo cameras. Using this tool allows you to set the parameters of the stereoscopy. It also tells V-Ray to render the scene as a single side-by-side image with the left eye on the left side of the image, and the right eye on the right. Here is how you use this tool:

- In Maya *{fig 12}*
  + First place the camera as described above.
  + Go to *Attributes -> VRay -> Stereoscopic Helper*

- In 3ds Max *{fig 13}*
  + First place your camera as described above.
  + Go to *Create -> Helper -> VRay -> VRayStereographic*
  + Place the helper anywhere in your scene. Its location does not matter, it is simply there to set up parameters.



{fig 12}



{fig 13}

- The eye distance will be set to a default of 6.3 cm. Only adjust this if it is appropriate for your scene.
- Set your focus to none. This sets your stereo to parallel cameras. You will want to keep it as such. Using rotation or shear will set the cameras to converged, which focuses on a certain distance in the space, which has a cross-eyed effect, and can create a very negative experience in VR.
- Keep the interocular method to shift both. This means that your current camera in your viewport is between the left and right eye and that you shift half the interocular distance left and right for each eye. If you are working in a scene where you need the right or left eye to be the dominant eye, you can select it to shift left or right.
- The Adjust Resolution checkbox tells V-Ray to automatically double the horizontal resolution of your render. As an example, if you are rendering a 2048 x 1024 latlong image, checking this checkbox will make the render 4096 x 1024 with left camera rendered on the left side of the image and the right on the other side.
- The last parameter to adjust is the Top and Bottom merge angles. This is where you control the tapering of the stereoscopic effect at the poles. The default is 60 degrees, which is generally good. If you need to see more stereo effect at the poles, you may want to lower this, but this can have a negative effect of seeing a strange depth transition when looking up. Raising this number too high will remove too much of the stereo effect and flatten your image.

Once these parameters are set you are ready to render.

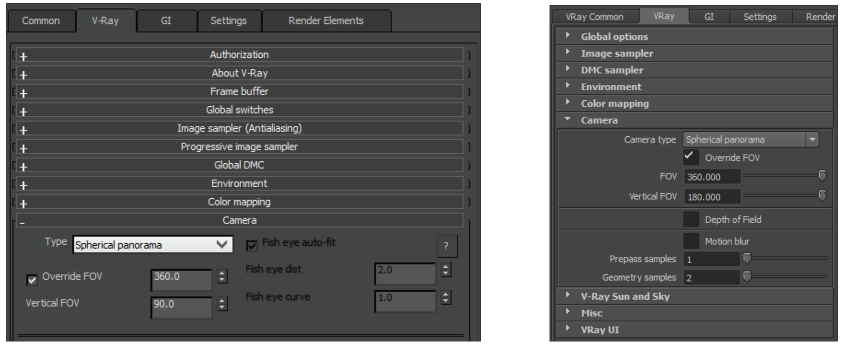
Rendering

We are now ready to render. Based on the previous discussion, you will need to pick if you want to render a latlong or a cubemap.

Latlong

In your render settings (both in 3ds Max and in Maya) go to *VRay -> Camera -> Spherical panoramic* and select 180 degrees for the vertical FOV. This will render a full 360 degree image with the vertical covering 180 degrees (top to bottom). ***{fig 14}***

An image of this type has an aspect ratio of 2.0, where the horizontal is twice the vertical. Generally it is recommended to have at least a 2k image (2048 x1024) but 4k (4096 x 2048) or higher gives better results. Remember that this is stereo, so the image will end up being double for the left and right eyes.



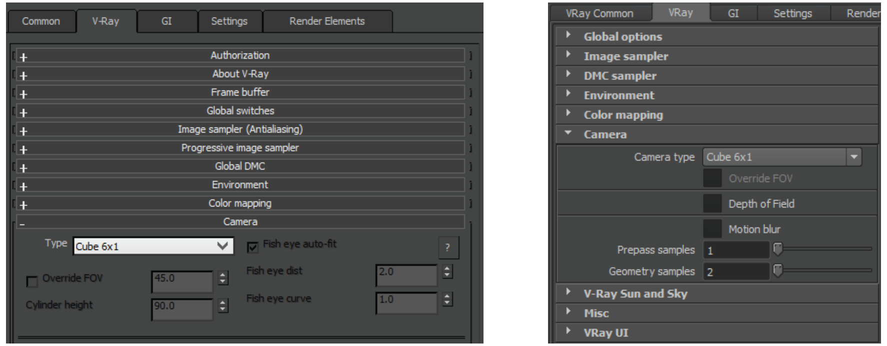
latlong render settings *in 3ds Max and Maya*

{fig 14}

Cubemap

There are many ways to render cubemaps, but in general the VR community has embraced the 6x1 cubemap. In both 3ds Max and Maya, under the render settings select *VRay -> Camera -> Cube 6x1*. ***{fig 15}***

This format will render six sides to a cube, beginning with the front, then moving to the back, top, bottom, left and right. Each image will be a square, and there are six squares, so the aspect ratio will be 6.0. It is recommended that if you render for Samsung's Gear VR, each square is 1536 x1536. So you need to select a resolution of 9216 x 1536 for an aspect ratio of 6.0. Since this is stereo, it will render this for each eye which will make the aspect ratio 12.0. This means that the final image in stereo will end up being 18432 x 1536. Some VR players will view smaller images, but you will want to render the full 18k image for the correct full resolution on current high-end displays for phones (2560 x1440).



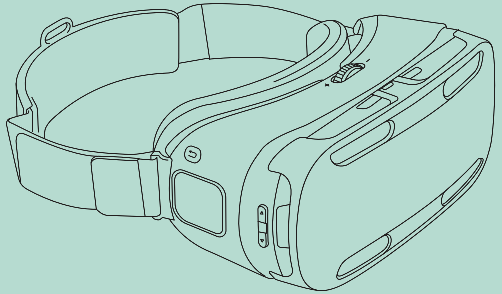
cubemap render settings *in 3ds Max and Maya*

{fig 15}

It is also important to note that you need to turn off Image Filtering (called AA Filtering in V-Ray for Maya) when rendering. Otherwise, the edges of each side of the cube could get some cross-contamination, which can lead to visible edges when viewing in VR.

Image format

It is recommended that you save in a full-res, high bit rate, uncompressed image format such as EXR, so that you can edit your image. However, most VR players today will play JPG or PNG formats. While PNG formats have less compression, if you are rendering a very large image such as an 18k stereo cube map, use JPG. The file size is much smaller, and the load time is much faster.



Samsung Galaxy Gear VR

{fig 16}

Viewing a stereo panoramic still in VR

It should be noted that VR technology is constantly changing. In fact, as we write this paper, new and different ways of viewing VR are constantly being developed. It should also be noted that much of the technology that people think of when they think of VR, such as the Oculus and the HTC Vive, are not available to the public yet. They are tools designed to give developers something to work with as consumer products are being developed and changed. Therefore, many of these products do not have out of the box tools that allow you to view these images. For this reason, we will focus on a few tools that do exist out of the box.

Viewing stills on the Samsung Galaxy Gear VR

Samsung has a VR headset called the Gear VR Innovator's Edition. ***{fig 16}*** The device itself costs around $200 and comes in two models, for the Samsung Galaxy Note 4 and the Samsung Galaxy S6. If you happen to have one of these phones, it will work well. If you don't, you'll have to buy one, which will cost up to $800 for an unlocked phone. A new Gear VR is now available for only $99 and will work with all Note5, S6 edge+, S6 edge, and S6 products.

The advantage of the Gear VR is that the head mounted display is very comfortable, is hands-free, has a touchpad built-in (D-Pad for the new version) to assist in the interface, and has no cables attached to it. It also has a native application that allows you to view your stereo panoramic stills as well as videos. These steps assume that you have already purchased your phone and the Gear VR and have gone through the initialization and setup process.

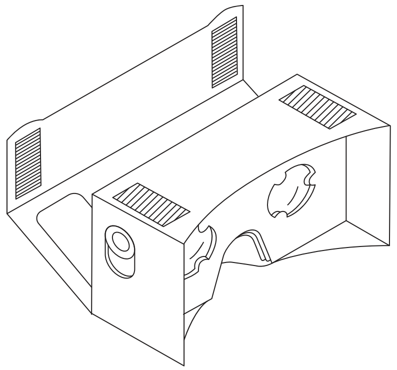
Here is how you view the stills:

- Connect your Galaxy S6 or Note 4 to your computer via USB.
- Open a file browser on your computer.
- Navigate to root directory of the phone.
- If there is a directory called “Oculus” open it, otherwise, create one and open it.
- If there is a “360Photos” directory open it, otherwise, create one and open it.
- Within this directory you can make any directory you want. This will appear as a project in the Gear VR interface, and will list all the images that you have made in there.
- Place your images in that directory. Remember that they need to be either JPG or PNG, and that JPG will load faster.
- Disconnect your phone and place it in the Gear VR.
- With the Gear VR on, navigate to the Oculus 360 Photos application.
- Use the touchpad on the side of the Gear VR to scroll up and down until you see your project listed. If you are putting this on for the first time, the thumbnails may appear gray as they are being generated. You will notice a red and blue box on the thumbnail if your image is stereo.
- Tap once you have the image selected.
- You can now look around.
- If you have several images that you have done, you can swipe forward or back to skip to the next image.

Viewing stills with a Google Cardboard

Google wanted to see if it could give people the ability to experience some level of VR using nothing but a box, a few plastic lenses, and the smartphone that they already have in their pocket. ***{fig 17}*** They have even opened up the design of the Cardboard so that it is available from many different manufacturers. You can make your own branded version of Google Cardboard for only a few dollars. While the experience may not seem as fluid and integrated as the Gear VR, it does allow for content creators to easily distribute their content to whichever audience they would like at a fraction of the cost.

Cardboard itself does not have a native application to view stereo panoramic images, so we will focus on third-party applications that will allow you to view these images. Also note that, while Cardboard supports iOS, some applications do not.



Google Cardboard

{fig 17}

Google Cardboard third-party applications

Holodeck by Standard3D

This application is available on Android, with iOS coming soon. Currently, it uses low-level OpenGL code that allows for extremely fast and responsive playback. To use this app, you can download it from the Google play store. There are two versions of Holodeck:
Holodeck (free, with a time limit for viewing images)
Holodeck Pro (paid, without restrictions)

Here is how to view your images in Holodeck:

1 Take all the images you want to view and rename them with _pano.<ext>. at the end. As an example, image_01.jpg would get renamed to image_01_pano.jpg
2 Connect your phone to your computer.
3 Open a file browser and navigate to your phone. If your phone has an SD card, navigate to that, otherwise, navigate to the root of the phone.
4 At this point you are free to place your images anywhere on the phone. For simplicity, make a directory and place your images in there.
5 Disconnect your phone from your computer.
6 Open the Holodeck application.
7 Once open, on the left side of the screen, you will see three folders. Click on the left directory. This will list all files on the SD card (or phone if you don't have an SD card) that have the _pano.<ext> file name structure.
8 Select the filename you want to load.

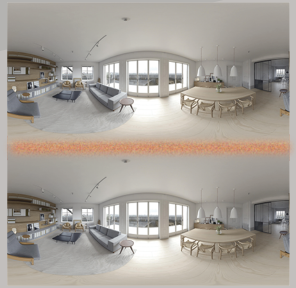
9 Click on the double arrow icon to load that image.
10 Now select the Google Cardboard icon to put it in Cardboard mode.
11 You can now look around your image.
12 To view the next images, remove the phone, tap the screen to get out of cardboard mode and repeat steps 8 through 11.

Iris VR

The IrisVR app works a bit differently. Here, the images are saved on the web and can be downloaded to your phone. The application supports both iOS and Android. Here are the steps to getting your images to play in IrisVR:

1 Navigate to http://app.irisvr.com/signup.
2 Create an account.
3 Click on the "Access Now" button on the Mobile Viewer.
4 You will be given a user ID number. Create a pin.
5 Download the app from the iOS or Android app stores.
6 Launch the application and place your phone inside of Cardboard.
7 Below the sample images you will see a button called Login. Click on it using the trigger with Cardboard.
8 Enter your user ID number and pin.
9 You will now see your images as thumbnails.
10 Click on the image you want to load.
11 Use the trigger again and you will see the thumbnails again.

How to render VR in V-Ray: Video



***compression artifacts** run along seams;*
latlongs [bottom] have fewer seaams than cubemaps [top]

{fig 18}

We will assume that you already have learned everything about creating stills for VR. All the points and concerns for stills are the same for video. Here are some additional concerns to be aware of when creating video:

- When moving forward do not move too fast as this can lead to discomfort. If you are supposed to be walking, move the camera at a walking speed or slightly slower. If you are driving, move at the appropriate driving speed.
- Turn the camera only if absolutely needed, and if you do, turn very slowly. Remember that you need to give the viewer his or her own choice where to look. That is the whole point of VR.
- Keep the camera level with the horizon at all times. Do not look up or down, or roll the camera. This can also cause discomfort for the viewer.

Panoramic stereo formats

Compression artifacts on videos are much greater than they are on stills, in order to achieve optimal playback at appropriate frame rates. This means that any time two parts of the image overlap, the compression artifacts can affect both sides of that seam making it visible when playing back. For this reason, cubemaps have many seams which can make the seams very visible. In order to minimize this issue, the best and preferred method is to use a latlong format where the two images are top bottom. That way, with the seams being at the poles, all artifacts are compressed down to a single point. *{fig 18}*

Resolution and frame rate

The resolution and framerate of the video you create is largely dependent on the player and device you are planning on using. It was recently announced that a new technology, which uses multiple JPG or PNG frames, will allow for much higher resolution playback on some devices. For now, here are some guidelines. As of this writing, the technology is not yet available to the public.

Playing back on a phone

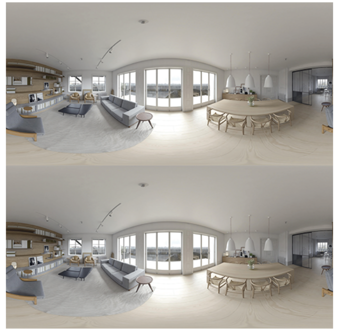
If you plan on playing video on a phone, such as for the Gear VR, limit your video size to around 20 Mb/s. Keep in mind that battery level on some of the devices can also affect playback. Additionally, you will need to render latlong, as it minimizes the seams.

At this point you will have to choose if resolution or frame rate is more important: If you have some fast motion in your scene, such as flying, a higher frame rate may be desirable to make the experience smoother, but this may also mean you need to sacrifice resolution. On the other hand, if motion is fairly slow and you have a lot of rich textures, you can reduce the frame rate and increase the resolution. It should also be noted that the OS on a Gear VR has a limit of 2160 pixels for the vertical resolution. You can do an anamorphic "squeeze" of your top and bottom images to fit within these limitations, which can cause a bit of vertical softness, but the detail is maintained horizontally.

Based on these examples, here are some good choices to use depending on your content:

- Video with faster motion: 2048 x 2048 top/bottom (each eye is 2048 x 1024) at 60 frames per second. *{fig 19}*
- Video with lots of detail: 3840 x 1920 top/bottom (each eye is squeezed at 3840 x 960) at 30 frames per second. *{fig 20}*

Lastly, in order for the Gear VR to know that your video is formated as a Top / Bottom stereo, add "_TB" to the name of the video. For example: "myVRvideo.mp4" would become "myVRvideo_TB.mp4"



fast motion video *- 2048 x 1024 each eye*

{fig 19}

high detail video *- 3840 x 960 each eye*

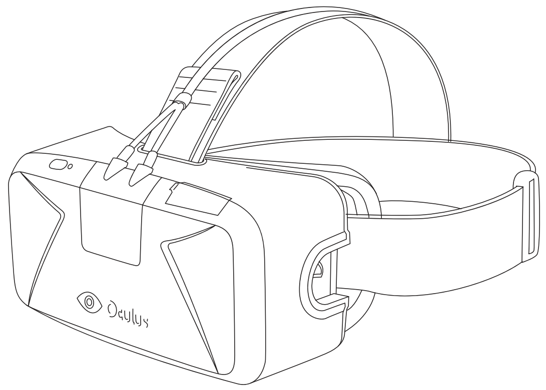
{fig 20}

Playing back on an Oculus

Since the Oculus *{fig 21}* is attached to a computer which can generally playback much higher bandwidth video, you are not as limited as you are on the phone. You can push the bandwidth limit to 40 Mb/s or so. However, it should be noted that the Oculus DK2 screen is not as high resolution as the screen on the Galaxy S6 in the Gear VR. At some point, going to very high resolution on video will not deliver a better viewing experience because you will have surpassed the resolution of the screen.

Therefore a good high benchmark resolution and frame rate would be 4096 x 4096 at 60 fps. Keep in mind that you will need a fairly powerful GPU and possibly a solid-state drive on the attached computer to play back at those rates.

Oculus
{fig 21}



There are several tools that people have written that can be used to view stereo 360 videos, including:

LiveViewRift
https://share.oculus.com/app/liveviewrift

VR-Player
https://share.oculus.com/app/vr-player

Whirligig Player
https://share.oculus.com/app/whirligig-player-and-film-vortex

Conclusion

When making pre-rendered content for VR there are several things to consider that can affect the choices in terms of format and content. It is generally best to work backwards and start by considering what you will be showing your content on. You may also consider if you will be showing stills or videos. If you plan on showing your content on multiple platforms, it is generally best to create the highest quality content and down-res for it to be playable on other devices.

Appendix A:
Quick guide to rendering VR stills in V-Ray

1 Camera and scale
- Make sure your scene is in the correct scale.
- Place your camera where you want your viewers head to be in the scene.
- Keep the camera level.

2 Add stereoscopic
- 3ds Max: created a stereoscopic helper.
- Maya: added a stereoscopic extra attribute.

3 Adjust stereoscopic
- Keep intraocular distance at 6.3cm.
- Focus Method: None.
- Check Adjust Resolution.

4 Go to render settings
- Go to *Camera* -> *Type* and select Cube 6x1.
- Select resolution of 9216 x 1536 (final will be: 18432 x 1536 for stereo)
- 3ds Max: Under the *Render Settings* -> *V-Ray* -> *Image Sampler* turn off Image Filter.
- Maya: Under the *Render Settings* -> *V-Ray* -> *Image Sampler* turn off AA Filter.

5 Render your image

Appendix B: Quick guide to rendering VR video in V-Ray

1 Camera and scale: follow the same rules as with VR stills.
2 Stereoscopic settings: follow the same rules as with VR stills.
3 Frame rate: Pick a frame rate appropriate to your scene
    - For faster motion use 60 fps.
    - If motion is slow, you can go as low as 30 fps.
4 Go to render settings
    - Go to *Camera* -> *Type* and select Spherical.
    - Select a resolution appropriate to the desired devices.
        - 2k: 2048 x 1024 (final will be 4096 x 1024)
        - 4k: 4096 x 2048 (final will be 8192 x 2048)
5 Once rendered, use a compositing application to convert your 'side by side' to a 'top bottom' stereo.
6 In order for the Gear VR to know that your video is formated as a top bottom stereo, add "_TB" to the name of the video. For example: "myVRvideo.mp4" would become "myVRvideo_TB.mp4"

by