LIBERTY3D.COM
Computer Graphic Resources

Liberty3D UberCam

For NewTek's LightWave3D

Version 2.4

March 16th, 2016

Thank you for purchasing UberCam!
(Win32Bit, Win64Bit, MacUB Intel 32 Bit / Intel 64 Bit)

Copyright © 2010-2016 Liberty3d.com (Declaration Pictures Inc.)
Version 2.0 - 2.4 Code and Engineering: Chilton Webb
Design and Concept Version 1.0 - 2.4: Kelly Lee Myers (aka Kat!)

Special Thanks: NewTek, Oliver Hotz at Origami Digital, Enhanced Digital, The Dr Pepper Company, The Liberty3D.com Citizens and all of our supporters Liberty3d.com including our Beta Testers, Customers and Friends around the world.

COPYRIGHT
© 2010-2016 Liberty3D (A Division Of Declaration Pictures Inc.). All Rights Reserved.
Except as otherwise permitted by the authors, this publication, or parts thereof, may not be reproduced in any form, by any means, by any method, for any purpose. Disclaimer THIS PUBLICATION AND THE INFORMATION CONTAINED HEREIN IS MADE AVAILABLE BY THE AUTHORS "AS IS." LIBERTY3D DISCLAIMS ANY WARRANTIES, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE REGARDING THESE MATERIALS.

Liberty3D No-Nonsense License Statement
(We acknowledge inspiration to this in part to the words and the spirit of the Borland No-Nonsense License Statement)
This software is protected by both United States and Canadian copyright law, as well as international copyright treaty provisions. It is software that is licensed only to be used in conjunction with Newtek LightWave3D software (which is described at the URL http://www.lightwave3d.com).

Therefore, you must treat this software just like a part of a legally licensed installation of LightWave. You may copy it onto a computer to be used and you may make archival copies of the software for the sole purpose of backing-up our software and protecting your investment from loss. As such, since the software is license locked to a hardware or software identification device, commonly known as a dongle, any use of the software on a computer workstation in a non-network rendering mode without a legally obtained hardware identification device and the appropriate keys to unlock the software is prohibited.

Further Explanation of Copyright Law Provisions and the Scope of This License Statement
You may not download or transmit the software electronically (either by direct connection or telecommunication transmission) from one computer to another, except as may be specifically allowed in using the software on a computer network. You may transfer all of your rights to use the software to another person, provided that you transfer to that person (or destroy) all of the software, as well as the hardware identification device and documentation provided in this package, together with all copies, tangible or intangible, including copies in RAM or installed on a disk, as well as all back-up copies.

Remember, once you transfer the software, it may only be used at the single location to which it is transferred and, of course, only in accordance with copyright law and international treaty provisions. Except as stated in this paragraph, you may not otherwise transfer, rent, lease, sub-license, time-share, or lend the software, diskettes, or documentation.
Your use of the software is limited to acts that are essential steps in the use of the software on your computer or computer network as described in the documentation. You may not otherwise modify, alter, adapt, merge, de-compile or reverse-engineer the software, and you may not remove or obscure copyright or trademark notices within this product.
There is no rights to redistribution of this software that is included in this License. All licenses, software downloads, and software keys must be obtained via the Liberty 3D website (URL http://www.liberty3d.com).
By obtaining a software key and using this software, the user signals agreement to this license agreement.
If the hardware device is damaged or lost, our support policy again matches that of NewTek. If NewTek support will verify that they agreed to reissue a hardware identification device to a legally obtained licensed version of LightWave3D due to damage or loss, then we will reissue a new key upon request within a reasonable amount of time. The exact content of this agreement in obscure legal jargon may be supplied upon request, and the laws of Alberta, Canada will apply if there is any conflict in the interpretation of the law between our place and your place.
All trademarks are the property of their respective owners.

Chapter 1 Introduction: Liberty3D UberCam v2.4

Liberty3D UberCam is a plug-in for NewTek's LightWave3D that enhances the LightWave platform with additional camera capabilities while expanding into the realm of Virtual Reality and Augmented Reality. Three and a half years of additional research and development, testing and customer feedback since the release of Version 1.9 have been brought forward in Version 2.4.

In version 2.0.1 we introduced additional specific tools and cameras to the artist who wants to explore Virtual Reality content creation using LightWave3D. Virtual Reality over the last 4 years has exploded and Liberty3d.com has been working quietly to bring forward tools to LightWave artists that will give them a competitive edge in this new medium. Since the release of version 2.0.1 we have been hard at work improving these tools. With the advances made in refining the technology present in our cameras and of the tools in UberCam 2.4, you are now in possession of a unique toolset in the industry.

Our development efforts in the areas of Virtual Reality for UberCam continue and we will be providing additional tools and functionality to the package as industry changes drive innovation. This includes support for additional Virtual Reality Headsets beyond the Oculus Rift. As of the writing of this manual, release dates for the CV1 of the Oculus Rift are slated to start shipping March 28th, 2016. HTC Vive headsets are expected to begin shipping at the end of Q2 2016 with pre-orders potentially having sold out (15,000 units in 10 minutes) on February 29th, 2016. The day pre-orders started! So while we wait just a little bit longer for physical units to start showing up in the hands of users and we will introduce enhancements to the existing Rift support and add Vive support. As we mentioned earlier as new head sets come on line we will be adding them on or "around" their availability to the public.

With UberCam 2.4 you get all of the original cameras from previous versions of UberCam with many bug fixes, performance enhancements and the added capabilities we have introduced that open up new possibilities.

About Liberty3D UberCam

As of version 2.4 there are at least 19 camera types included in the UberCam package. Although the cameras will function as described in LightWave 9.3.1 and above, we recommend upgrading to LightWave 11.6.3 at least, and above in order to take advantage of additional capabilities, LightWave 2015 for full stability during use. We have tested UberCam 2.1 using LightWave 3D versions 11.6.3, 2015, 2015.2 and 2015.3. In the future, support for LightWave 9.x may need to be removed in order to advance this product.

In addition to the Camera Plug-ins, UberCam 2.4 ships with a set of additional and very special plug-ins giving LightWave3D artists unique VR production capabilities.

While these additional capabilities are intended for use with the Oculus Rift, we will be expanding and improving head set support from other manufacturers shortly.

First, there is the UberRift Viewer Window Plug-in simply called "L3DRiftViewer64.p" or "L3DRiftViewer32.p" for PC installations. This plug-in provides for a floating window clone of your primary layout viewport and is intended for use with the Oculus Rift VR Head Mounted Display. This cloned viewport applies the correct barrel distortion to your camera view for use with the Oculus Rift VR Head Mounted Display and eliminates the need for you to export your scene to a game engine such as Unity3D or Unreal Developer Kit in order to see your work in VR with the Oculus Rift. It supports both of the currently available Oculus Rift Developer Kits; 1.1 and 2.1. ***but only up to the Oculus Runtime 0.6.0.*** We are working on updating this shortly so that the latest run times are supported. Oculus sadly broke some of the technology that we rely on in builds later than 0.6.0. Again, we are working towards updating these components and expanding support for other headsets.

Second, there is an Image Filter Plug-in called "L3DStereoF64.p" (64bit version, 32bit and OSX versions are named appropriately) that applies the correct barrel distortion to a Liberty3D Stereo Cam rendered image. This plug-in will apply the distortion in F9, F10 and LWSN renders.

Lastly there is the L3DVRHeadset64.p (x64 Windows) or L3DVRHeadset32.p (x86 Windows) or VRHeadset.plugin (Mac). This plug-in provides for Oculus Rift rotational head tracking (positional tracking will be made available at a later time) capability via LightWave's Virtual Studio Tools. It consists of two main components; a VR Headset Manager which is found under the Virtual Studio Tools Device Manager Panel and an Item Info Device node that connects a Virtual Studio Tools ItemMotion:Camera Trait to the VR Headset Manager. These combine to give you the real-time rotational head tracking of the Oculus Rift. It is intended to be applied to the rotation of a camera, but can be added to any item in a 3D scene such as a Null, Light or Object providing you with an additional real-time input device option with Virtual Studio Tools.

PLEASE NOTE: Our UberRift viewer requires at least LightWave3D 11.6.3 running on a Windows PC in 32bit or 64bit at this time. The UberRift viewer is currently not supported on Mac OS systems. It is important to note that Oculus VR has recently dropped support for Mac platforms. While we are working to get a MacOS version of the viewer working for a future update, there is no guarantee that the Oculus Rift will still support Mac in the future.

In addition to this, to use the Oculus Rift rotational head tracking features provided in UberCam via LightWave3D's Virtual Studio Tools which became available in version 11.0 you will need to at least be running 11.0 for LightWave3D on the PC or Mac.
We have tested the UberRift viewer and L3DVRHeadset plug-ins with both the Oculus Rift DK1.1 and DK2 developer kits extensively. Either headset can be used with these plug-ins as expected.

In order to use both the UberRift viewer and the Oculus Rift rotational head tracking functions together you will need to be running at least 11.6.3 on a Windows PC in order to provide a full Virtual Reality experience in Layout as intended. *For more information* *on how to use the Oculus Rift and UberRift Viewer together please see **Chapter 5** of this Manual*

Installation:

WARNING! WARNING! WARNING!

Was that a big enough alert to get your attention? Ok good. Just checking.

This section is specifically geared towards Windows based users of UberCam 2.4. If you are on a Mac, you don't need to worry about this. ***UberCam on windows platforms (x86/32Bit and 64Bit) requires the installation of certain Visual Studio C++ Redistributables from Microsoft.*** ***So read on!***

If you ignore this section you are going to have a bad time!

Microsoft VS C++ Dependencies

UberCam 2.4 requires the installation of the Microsoft Visual Studio C++ 2015 Redistributables in order for the Cameras portion of the UberCam package to work correctly. Before using UberCam in production on any Win32 or Win64 system, including render nodes, we highly recommend that you download and install the redistributables package directly from Microsoft for any machine that will make use of the UberCam 2.4 plug-in package. Microsoft likes to move this stuff around and previous versions of this manual had almost immediately incorrect links in it.

As of March 16th the links below are accurate.

Be advised:
Make sure you select the appropriate version for you OS and LightWave edition. 32bit Redistributables for 32bit LightWave3D must be installed on a 64bit Windows system. With this said, on a 64bit system it doesn't hurt to have both, but on a 32bit system its going to error on you if you try and install or run the 64bit version.

You can download the MS VS C++ 2015 redistributables from this URL.
https://www.microsoft.com/en-us/download/details.aspx?id=48145

The UberCam VR Headset Manager may require the Installation of the MS 2013 Redistributable. We are working to update this to later versions of MS Visual Studio (2015). Please note that ***this is only required to download and install if you intend to use the VR Headset Manager plug-in.***
If you are a render farm or render manager, you don't need to worry about this because

the VRHeadset manager is going to be kind of useless on a render node in the first place.

You can download the MS VS C++ 2013 redistributables from this URL.
https://www.microsoft.com/en-us/download/details.aspx?id=40784

Now that we have that over with.... We can move on.

Plug-in Installation

Liberty3D UberCam is installed just like any other plug-in for LightWave. To install UberCam, simply place the appropriate version of the plug-ins that come with this package into a folder named "Liberty3D" or something similar in your LightWave3D plug-ins folder. Then, run layout and add the plug-ins from your chosen directory location.

This is done by selecting add-plugin in the Utilities menu bar selection in the menu bar system, or by pressing add plug-in in the edit plugins dialog box. In order to correctly recognize all the camera types we recommend un-installing (removing or renaming it so it won't load) the old plug-in first, then re-adding the new plug-in.

This will recount the plug-in types included in the package. We do not recommend attempting to run older versions of the UberCam plug-ins along side the latest versions. Doing so may be a bad time.

The additional plug-ins found in the package should be added in the exact same way. Remember to close Layout down afterwards in order to write the plug-ins to your configs on your system. We recommend doing this immediately once you have all the plug-ins loaded so that you are good to go for the next step.
We also HIGHLY recommend that you write protect your LW plug-in list config files especially if you are using LightWave on a render farm. This is just generally good practice for LightWave users.

Key Activation

When you purchased UberCam from www.liberty3d.com you should have been emailed back a Key number after you entered your Product Lock ID into the store section field on check out. If you didn't do this you need to contact us and provide your Product Lock ID (aka hardware lock ID) number to us as well as your receipt. This will help us to get the key back to you faster. You will then be emailed a key.

We pride ourselves on our fast customer service and will have your key back to you very shortly but it is not an automated process as of this release, so please be patient as we get your key to you.

Once received, this key is then entered into a dialog box that pops up when you first try to access the property panel for any of the UberCam Cameras listed inside of the camera properties panel. Simply paste the number or enter it in manually and hit “OK”. You will get a message stating the key is accepted or not. The box, depending on which OS type of LightWave you are running will either go away on its own or not. If it doesn't, just close it by using the “X” button on the license key dialog box. Keys issued from versions prior to UberCam 2.0 will not work. A new key must be issued. In otherwords, have you bought your upgrade yet?

Need to upgrade and want a discount instead of paying full retail for the current copy?
Email: kat@liberty3d.com for details.

Once you have entered your key and it has been accepted by the software, you should now be able to access the properties of the camera, or switch cameras to any of the others available in the drop down list and make use of them.

Please note: The key can only be entered by accessing the UberCam Camera properties. It cannot be entered through accessing the viewer plug-in, the VR headset manager plug-in or the image filter plug-in.

We highly recommend that you set the permissions of your LightWave3D installation or wherever you store your plug-ins to be writeable by the local user and run Layout in Administrator mode when activating your copy of UberCam 2.4. This will ensure the plug-in has permissions to write the key file to the directory where the plug-ins are stored so you won't have to go through this process again each time you start up layout and want to use UberCam 2.4.

If you have any problems with your key, please contact us immediately via email.
Email: kat@liberty3d.com

A Note on Site Licensing and Render Farms:

For those of you who have purchased a site license or studio issued multi-seat pack (5 seats or more), should you have any problems with your key, please contact us immediately via email. Our key licensing system has changed from UberCam 1.9 to 2.0 and while we encourage you to upgrade immediately we want to make sure you are indeed working with the latest build of 1.9 should you have not upgraded already. There are some issues with the key licensing system clashing with later versions of LightWave past 11.0.3.

Commercial render farms should contact us directly regarding any problems they may have with the use of UberCam 2.x on their systems. We have discovered that many problems are render controller related and not directly attributed to UberCam. We can guide you through the process of ensuring reliable, rock solid render jobs processed through your farm regardless of what controller you may use. We have already assisted render managers and farm operators who have encountered problems that use RenderPal, Dead Line and ButterFly Net Render which seem to the only controllers on the market that have consistently similar issues when working with LightWave + UberCam.

For support on this issue or if you have general set up questions, please contact us. We are here to help!
Email: kat@liberty3d.com

What has changed in UberCam?

Lots! UberCam over the years has evolved significantly, culminating with this UberCam 2.4. For the sake of space this list is not entirely exhaustive but covers the basics.

Changed from v1.0 to v1.5

- Liberty3D UberCam v1.5 was changed to be compatible with LightWave v9.3.1., 9.5, 9.6 and now LightWave3D 10.0 Previous versions ofUberCam required LightWave v9.6.
- The Liberty3D SuperPano Camera and the Liberty3D Overscan Camera was added in UberCam v1.5.
- Some camera names were adjusted for clarity.
- This document was added.
- The Liberty3D Performance Camera was included as a separate optional camera plug-in package in order to judge the performance hit from various camera choices.

Changed from v1.5 to v1.9

- The Radial Shift Camera, and the DPI Camera were added as new camera types.
- The Liberty3D performance measurement camera was included also in order for you to check your performance changes on your own scenes and object files.

- The one point poly and two point poly problem with multi-up and stereo camera combinations has been fixed by using a workaround for a LightWave 3D rendering bug.
- There was an error in calculating the zoom factor in a rendered sequence if the zoom factor was enveloped. This has been fixed.
- The PPC build type was removed from the Mac UB build in order to reduce the size of the plugin.
- This document was updated, although we were too lazy to update the screen captures for the dialog boxes.

Changed from v1.9 to v2.0.1

- The Liberty3D Immersive Camera was added.
- The L3D UberRift Viewer was added.
- The L3D VR Headset Manager was added.
- The L3D VR Headset Item Info/Device Node was added.
- The L3D Oculus Rift Barrel Distortion Image Filter Plug-in called "L3DVR-OculusRift" was added.
- The L3D Surface Baking Camera was added. This is a highly experimental camera type at this time and is not documented in this manual.
- Several bug fixes and performance enhancements were made to the original cameras found in UberCam 1.9 and updated for compatibility with changes in LightWave3D 11.x and up especially when used in conjunction with LightWave's VPR.
- The L3D Key Protection system for UberCam has been upgraded in order to protect both our users and our development team from piracy while ensuring future compatibility with LightWave3D 2015 and up.
- Performance Camera has temporarily been removed for this release.
- DPI Camera has been temporarily removed for this release. It is now somewhat redundant since the appearance of Matt Gorner's DPI calculator in the LW 11.x cycle. We may put it back in later.
- Several bug fixes between UberCam 2.0.0 and 2.1 were made including crashes with LWSN rendering on networks and switching between cameras in Layout rapidly.
- This document was updated.

Changed from v2.0.1 to v2.1

- The Liberty3D Immersive Camera was completely re-written to eliminate any "back seam" artifact.
- The Liberty3D Immersive Stereoscopic Camera was complete re-written to eliminate any "back seam" artifact.
- The Liberty3D Immersive Stereoscopic Camera raytracing code was completely re-designed to eliminate distortion at the poles producing distortion free images in these areas.
- The Liberty3D Immersive Stereoscopic Camera now includes a Convergence Control for stereo convergence point management.
- The Liberty3D Immersive Stereoscopic Camera now renders material using a new algorithm that produces the stereoscopic effect in a more comfortable manner than previously possibly in prior versions of UberCam and this

algorithm rivals the stereoscopic output of any competing camera or shader technology on the market that aims to produce a similar rendering effect.

- This document was updated.

Changed from v2.1 to v2.4

- The Liberty3D Immersive Stereoscopic Camera now provides users with an additional set of feature made possible by our new algorithm for rendering 360 Degree Stereoscopic Immersive format renders. These features are called "Polar Realignment" and "Polar Realignment Fall Off" it is an industry first as far as we are aware. It certainly doesn't exist in any commercially available product where you can control it.
  This combined with changes in UberCam 2.4's Immersive Cameras now provide users with the best, most advanced solution on the market at any price for rendering VR out of a 3D Program. **All the more reason to use LightWave3D!**
- ***We discovered that UberCam seems to be completely compatible with the Kray for LightWave3D*** **render engine, so now you can work with any of the cameras and render in that Kray as well. Very cool.**
- Changed GUI text on the Liberty3D Immersive Stereoscopic Camera to full words instead of things like "IPD" to make it more clear what the function is next to that text.
- Several minor bug fixes in the Immersive Camera and the Immersive Stereoscopic Camera.
- Aligned build number to appropriately reflect which version this actually is. :)
- Update of this manual to deal with Kat's fluffy paws while he types, occasionally making mistakes. He get's distracted sometimes. Little red dots... You know the drill.
- This document was updated.

Why does the camera view not show what I expect?

The OpenGL view ports in LightWave Layout are hardware accelerated assuming that a linear transform can be applied, like that used in an Orthographic camera or Perspective camera. In these, a straight line in the scene will always end up as a straight line in the view port image. This can be done only for the Linear camera types in UberCam. For all other cameras we recommend using a visualization plug-in such as F-Prime from Worley Labs or LightWave 3D's VPR. Fully functional support for how images should appear in a F9 or F10 render compared to those of VPR may be limited at this time and may not perform as expected. This may change in future releases of LightWave 3D. We feel it would be a worthwhile effort by The LightWave Group to make this possible for all camera types when viewed in OpenGL.

Chapter 2 Linear Cameras Vs. Non-Linear Cameras

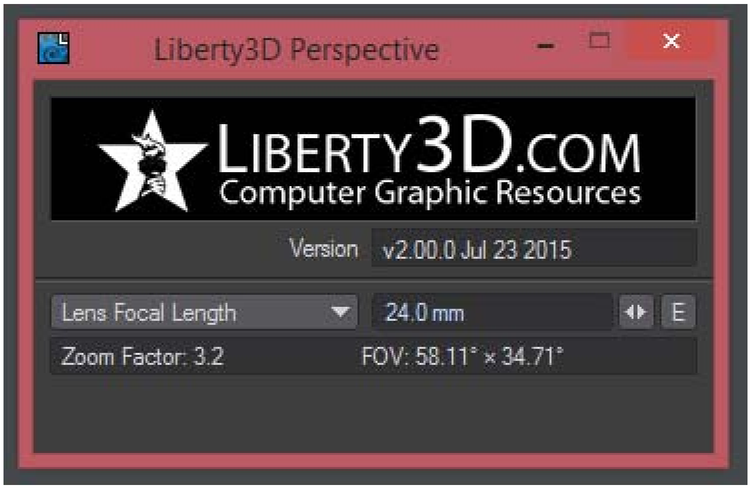
There are 3 linear cameras in the Liberty3D UberCam V2.4 package.

What is a Linear Camera?

A linear camera is a camera type that can be calculated using a 3D linear transform matrix. In this camera type, a straight line in the scene will always result in a straight line in the rendered image.

Liberty3D Perspective Camera

This camera operates identically to the perspective camera that is the default camera in LightWave 3D 9.3.1 - LightWave3D 2015 but has been optimized for speed.



Liberty3D Perspective

Version v2.00.0 Jul 23 2015

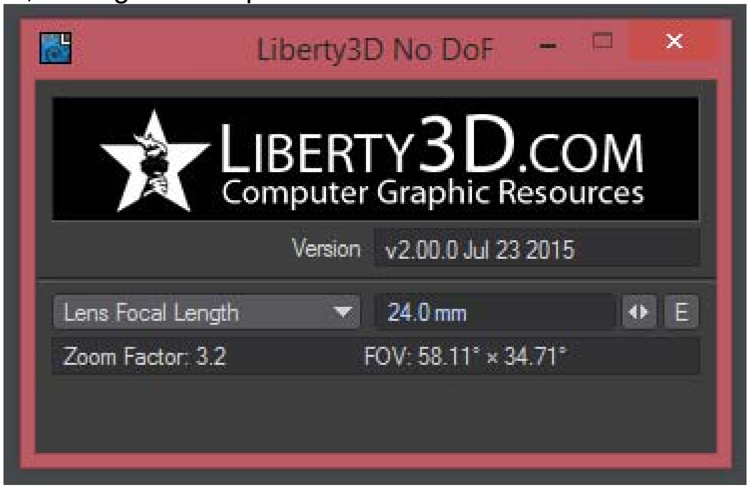
Lens Focal Length 24.0 mm

Zoom Factor: 3.2 FOV: 58.11° × 34.71°

The Liberty3D Perspective Camera Option Panel

Liberty3D No DoF Camera

This camera operates identically to the perspective camera that is the default camera in LightWave 9.3.1 - LightWave3D 2015, but assumes that there is no Depth of Field being used. By knowing this, the calculations that the Depth of Field rely on can be skipped as they are not used, adding further speed to the final render.



Version v2.00.0 Jul 23 2015

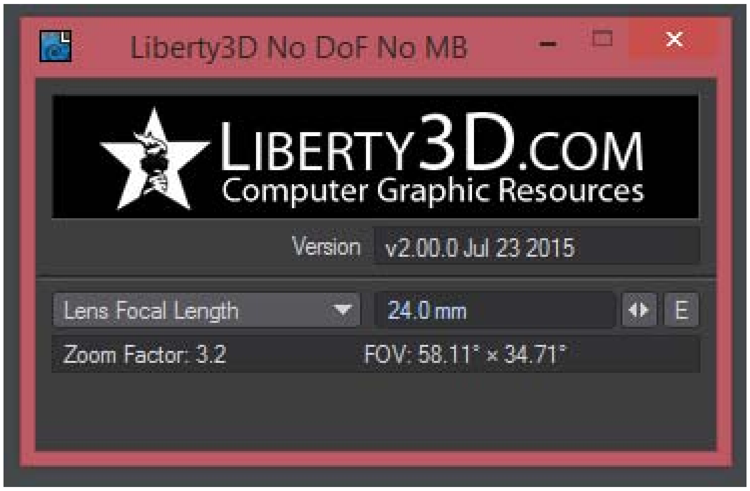
Lens Focal Length 24.0 mm

Zoom Factor: 3.2 FOV: 58.11° × 34.71°

The Liberty3D No DoF Perspective Camera Option Panel

Liberty3D No DoF No MB Camera

This camera operates identically to the perspective camera that is the default camera in LightWave3D 9.3.1 - LightWave3D 2015 but assumes that there is no Depth of Field or Motion Blur being used. By knowing this, the calculations that the Depth of Field rely on as well as the Motion Blur calculations can be skipped as they are not used, adding further speed to the final render.



Version v2.00.0 Jul 23 2015

Lens Focal Length 24.0 mm

Zoom Factor: 3.2 FOV: 58.11° × 34.71°

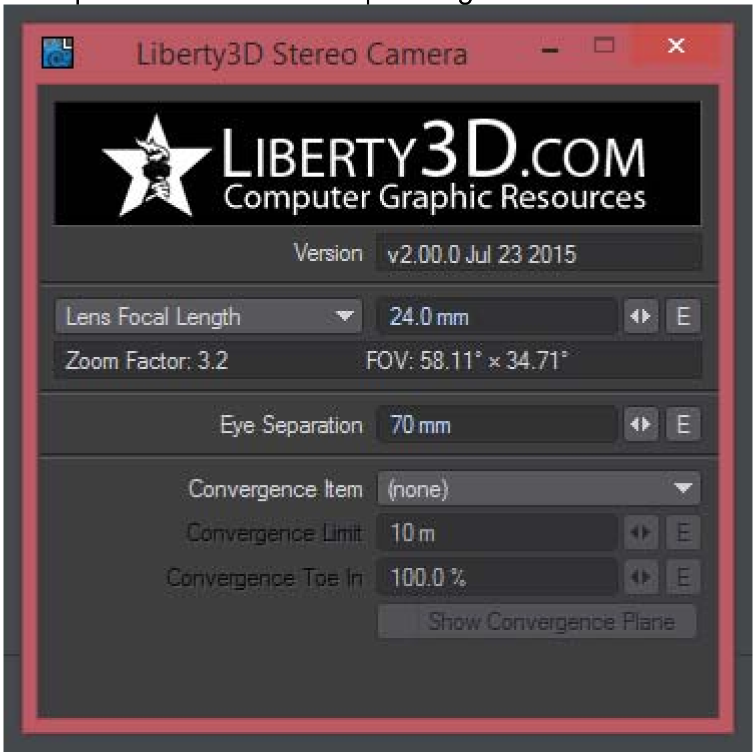
The Liberty3D Perspective No DoF No MB Camera Option Panel

What is a Non-Linear camera?

A linear camera is a camera type that CANNOT ALWAYS be calculated using a 3d linear transform matrix. In this camera type, a straight line in the scene will NOT ALWAYS result in a straight line in the rendered image. There are 8 non-linear cameras in the Liberty3D UberCam v2.1 package. These are the Stereo Camera, the Skybox Camera, the FishEye Camera, the Cylinder Camera, the Panoramic Camera, the SuperPano Camera, the Spherical Camera, and the two Immersive Cameras (mono and stereoscopic).

Liberty3D Stereo Camera

To make use of this camera in your scene you will need to change the width of the camera resolution to twice that of the resolution you wish to have a single Eye frame size at. For example - If you want to output in HDTV format which is 1920x1080, you will need to render your scene at 3840x1080. This will give you a horizontally wide image twice that of 1920 pixels (assuming square pixels). This will give you 1920 pixels per eye when recombined to produce the stereoscopic image.



Liberty3D Stereo Camera

LIBERTY3D.COM
Computer Graphic Resources

Version: v2.00.0 Jul 23 2015

Lens Focal Length: 24.0 mm

Zoom Factor: 3.2 FOV: 58.11° × 34.71°

Eye Separation: 70 mm

Convergence Item: (none)

Convergence Limit: 10 m

Convergence Toe In: 100.0 %

Show Convergence Plane

The Liberty3D Stereo Camera Option Panel

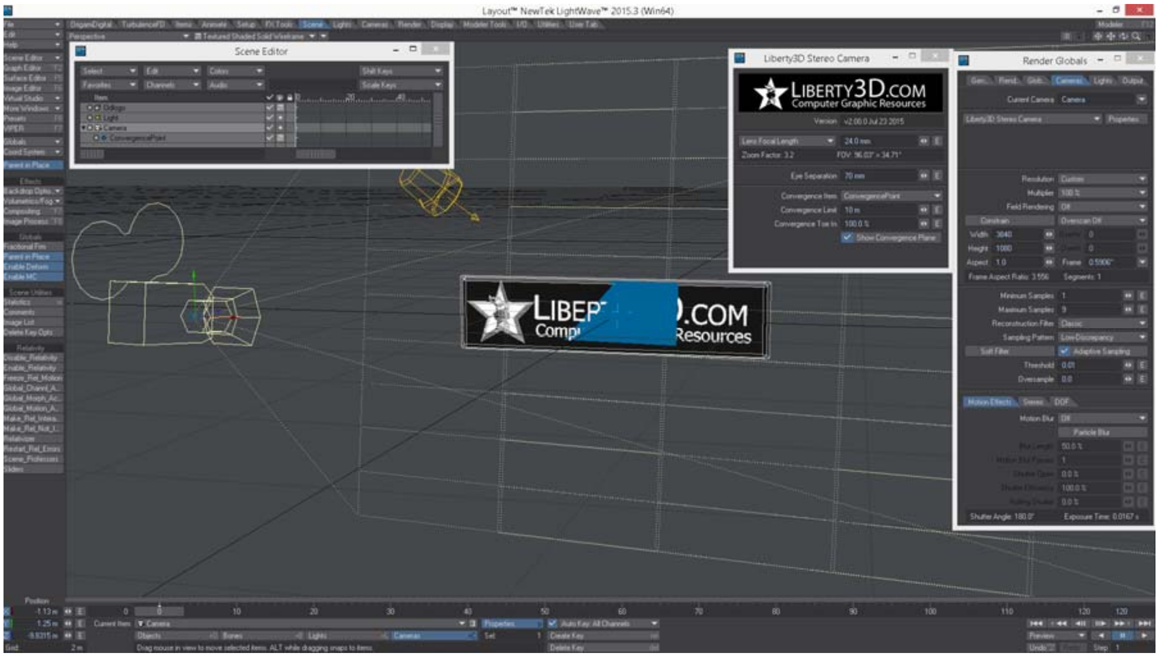
Our Camera Renders both eyes at the same time which is, as far as we know a capability unique to LightWave 3D and this plug-in. The advantages to this are obvious. You no longer need to render a scene twice, once per eye in order to get the images you are looking for. This is why our camera renders the images side by side horizontally.

This is very advantageous when working with modern compositing packages like BlackMagic Designs Fusion 7.0 or higher (Please note that the free version of Fusion 7.x and up does not have a set stereoscopic tools, only the paid for Fusion Studio edition does) as it has a stereo image combing function that specifically can make use of horizontally stacked images and combine them to produce the stereoscopic image result at what is more or less a push of a button.

There are other advantages such as scene file management complexity being reduced and render times over the network are reduced on multiple fronts as well. In UberCam 2.4 this camera has been additionally performance tuned and as a result in most scenes it will very closely perform to that of traditional LightWave Stereoscopic rendering operations. We aim to actually go faster than rendering using the built-in stereoscopic rendering function in LightWave. Actual performance improvements may vary from scene to scene in comparison to the native stereoscopic camera in LightWave. The biggest noticeable advantage of the UberCam Stereoscopic Camera over that of the native LightWave stereoscopic camera is that operations such as "Moving objects" or "Radiosity Render Calculations" or any pre-rendering calculations that take place. By using our Liberty3D Stereo Camera these calculations that take place at render time are done once instead of each time per eye.

A note on frame display in Layout itself: If you need to accurately preview the framing of your scene through the camera view, you can use the Alternative Aspect Display function do so. You can enable this through the Display Options Panel (press d). As a good starting point, use the width of your resolution for your Stereoscopic target frame resolution. An example of this would be if your target stereoscopic frame resolution (the final combined stereo image) is to be 1920x1080, enter in a value of 1.92. This will loosely match the actual framing but not exactly. It's a good starting point, but even the default of 1.85 works right out of the box. Great stuff!

As you can see in the image below we have an example of the UberCam Stereo Camera set up in a scene where a Null Object, called "ConvergencePoint" has been parented to the camera. This Null Object is then selected as the convergence point in the camera options panel. The Convergence Point in stereo is represented by the blue rectangle drawn in the OpenGL view.


Layout™ NewTek LightWave™ 2015.3 (Win64)

Liberty3D Stereo Camera

Version v2.00.0 Jul 23 2015

Lens Focal Length 24.0 mm

Zoom Factor: 3.2 FOV: 96.07° × 34.71°

Eye Separation 70 mm

Convergence Item ConvergencePoint

Convergence Limit 10 m

Convergence Toe In 100.0 %

☑ Show Convergence Plane

Render Globals

Current Camera Camera

Liberty3D Stereo Camera

Resolution Custom

Multiplier 100 %

Field Rendering Off

Overscan Off

Width 3840

Height 1080

Aspect 1.0

Frame 0.5906"

Frame Aspect Ratio: 3.556 Segments: 1

Minimum Samples 1

Maximum Samples 9

Reconstruction Filter Classic

Sampling Pattern Low-Discrepancy

☑ Adaptive Sampling

Threshold 0.01

Oversample 0.0

Motion Blur Off

Blur Length 50.0 %

Motion Blur Passes 1

Shutter Open 0.0 %

Shutter Efficiency 100.0 %

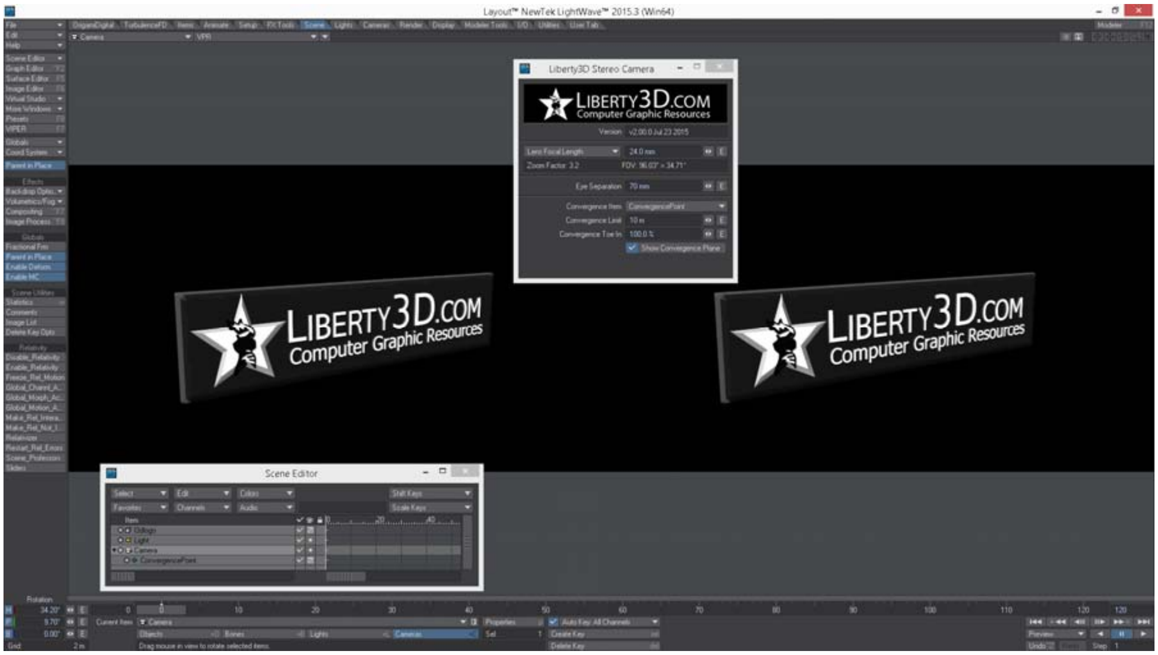
Rolling Shutter 0.0 %

Shutter Angle: 180.0° Exposure Time: 0.0167 s

Current Item Camera

Proper convergence point management and eye separation settings, animated over time are key to making great looking stereoscopic imagery. We highly recommend making use of these functions and understanding how they work together to produce the stereoscopic effect.

An Example of the Liberty3D UberCam 2.4 Stereo Camera with LightWave's VPR enabled is below.


Layout™ NewTek LightWave™ 2015.3 (Win64)
Liberty3D Stereo Camera
LIBERTY3D.COM
Computer Graphic Resources
Scene Editor

Version v2.00.0 Jul 23 2015
Lens Focal Length 24.0 mm
Zoom Factor 3.2 FOV 96.07° × 34.71°
Eye Separation 70 mm
Convergence Item ConvergencePoint
Convergence Level 10 m
Convergence Toe In 100.0 %
☑ Show Convergence Plane

There are additional features and functionality improvements that have been made in UberCam starting with version 1.5 going forward to match improvements made to the native Stereo camera system in LightWave3D 10.1 and above.

The Liberty3d 2.4 Stereo Camera and VPR are for the most part completely compatible except in one odd case and that is if you enable the UberRift viewer window you will see a stereo image pair per eye on each side. The UberRift viewer is specifically intended for use with OpenGL (Point, Wireframe, Shaded, Texture Shaded, etc.) due to interactive performance reasons. If you want to use your Oculus Rift as an inexpensive stereoscopic viewing device, we recommend using these modes exclusively.

For production renders via F9 or F10 (and LWSN), where the OculusVR Barrel Distortion image filter plug-in that ships with UberCam 2.4 is applied will produce renders as expected when combined with our Stereo Camera. Should you wish to use the UberRift viewer with VPR, working in stereoscopic mode with the Oculus Rift, we recommend using our L3D Perspective No MotionBlur, No DoF camera with the stereoscopic mode turned on in the stereo tab under the camera options panel. This will give you best performance results. However...

IMPORTANT: A crash *WILL* happen if one renders either F9 or F10/LWSN using a native LightWave or UberCam Camera is stereoscopic mode combined with the L3DVR-OculusRift Image Filter applied in the processing tab in Layout. Do not use the L3DVR-OculusRift Image Filter in conjunction with any other camera but our own Stereo

Camera. Always make sure in the camera options tab, that "Stereoscopic" is turned off. LightWave when told to render in stereoscopic through this tab option renders one eye and then the other independently. It is during the switch from one eye frame render completing to the start of render of the other eye that the crash will occur.
You have been warned!

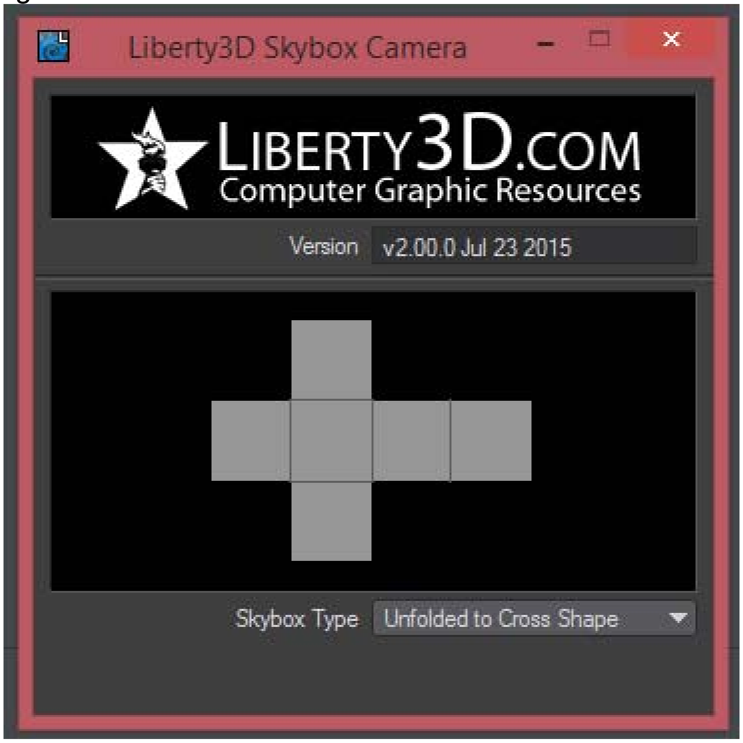
Liberty3D SkyBox Camera

This camera provides a simple way to create cross shaped skybox files, which can then be cut up into individual files as a post-rendering step. Saves time from having to set up and rotate 6 cameras every time a skybox render is required. The forward camera orientation is always rendered in the square that is the intersection square of the cross. In order to get a near perfect SkyBox Render which can then be split up into separate images for use in a game engine package such as Unity3D or Unreal3D we recommend using a 4:3 resolution.

For example, 1000x750 for each SkyBox side to be 250x250 pixels each. There are two modes supporting the two standard skybox shapes.

- Liberty3D Skybox Camera - Unfolded to Cross Shape
- Liberty3D Skybox Camera - Unfolded to Alternate Shape

You can switch shapes by using the dropdown box next to the text "Skybox Type" as shown in the image below.

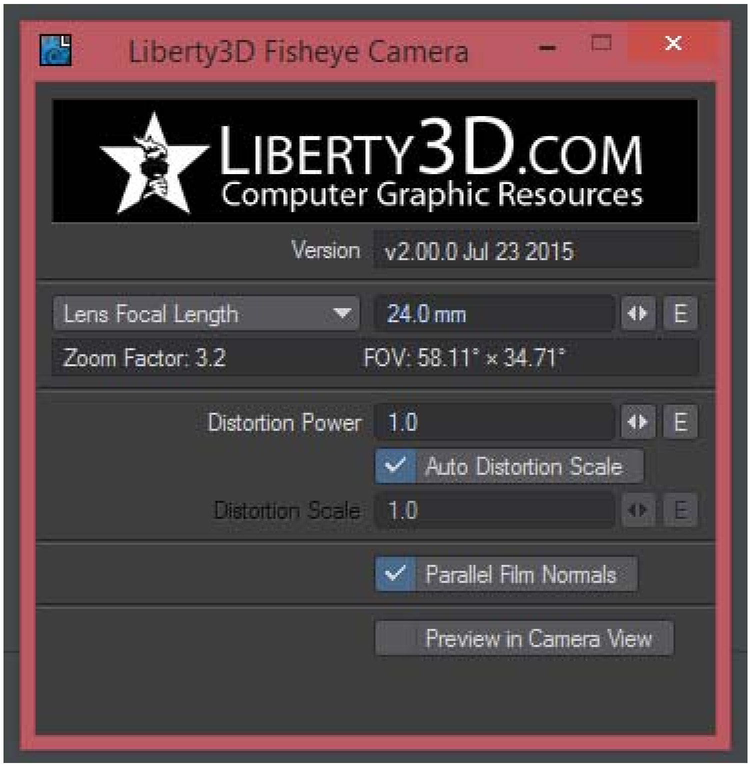


Version v2.00.0 Jul 23 2015

Skybox Type Unfolded to Cross Shape

Liberty3D FishEye Camera

This camera presents spherical lens distortion similar to that found in actual lenses. If the Distortion power is set to 1 and the Distortion Scale is set to 1, then the resulting image is identical to the default LightWave Perspective Camera. A distortion Power greater than 1 pincushions the field of view frustum inward. A distortion power less than 1 barrels the field of view frustum outward. That means parallel lines within the scene show up bending in the opposite way in the final render. As the power value increase results in a smaller field of view, a second control called distortion scale is provided to scale back up the field of view frustum without changing the focal length. A check box control is provided to automatically calculate the appropriate distortion scale in order to perfectly match the selected Focal Length and Zoom Factor.



Version v2.00.0 Jul 23 2015

Lens Focal Length 24.0 mm

Zoom Factor: 3.2 FOV: 58.11° × 34.71°

Distortion Power 1.0

☑ Auto Distortion Scale

Distortion Scale 1.0

☑ Parallel Film Normals

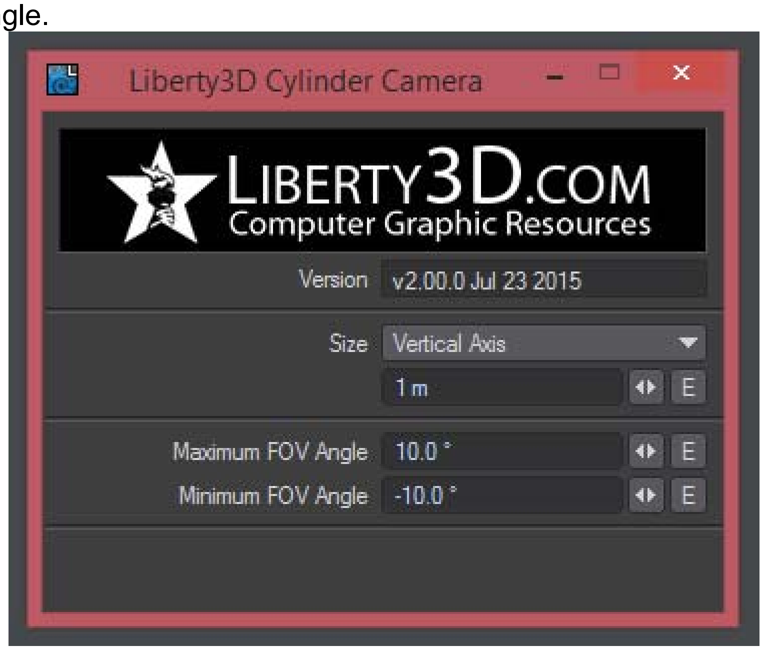
Preview in Camera View

Finally, as there is no way to accurately represent non-linear cameras like this in the camera view when working with OpenGL, we alternatively draw the camera view rectangles in order to allow you to match camera shots within the camera view. FishEye Camera in action with Fprime 3.51 or VPR (in non-draft mode) being used to view the results.

A quick note: While we have worked hard on the elimination of crash bugs dealing with VPR and our cameras working together at the same time, we recommend that you save your scene before making use of the mini-sliders in the options panel for the FishEye Camera. In the past, VPR has been very touchy when used in this scenario. With later versions of LightWave and improved code work having been done on our side for this has combined into crashes being a rarity as of March 1st, 2016 where LightWave 2015.3 is being used with UberCam 2.4. This comes from an almost complete re-write of these cameras.

Liberty3D Cylinder Camera

This camera presents a panoramic view, but with the vertical axis having orthogonal perfectly horizontal direction of views from all points along a vertical axis. The controls are the length of the imaging axis, the maximum Field of View Angle, and the minimum field of view angle.



Version: v2.00.0 Jul 23 2015

Size: Vertical Axis

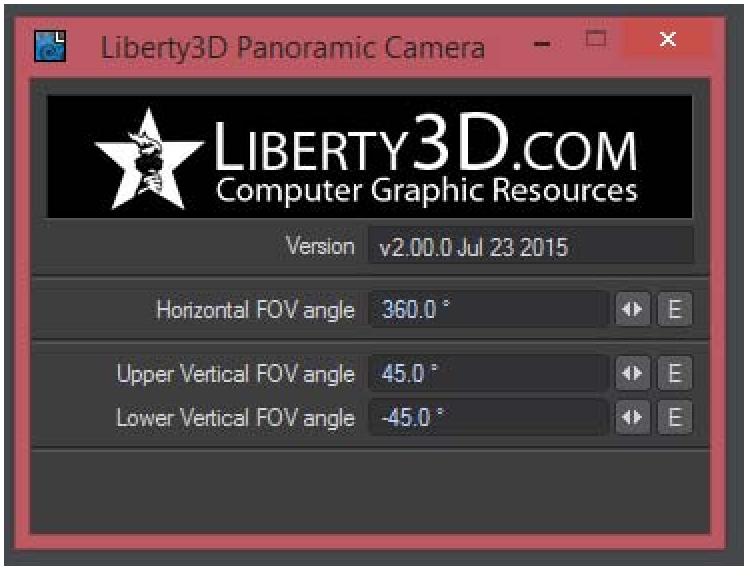
1 m

Maximum FOV Angle: 10.0 °

Minimum FOV Angle: -10.0 °

Liberty3D Panoramic Camera

This camera presents a perfectly hemispherical panoramic view presented from spinning around the vertical axis. The controls are the horizontal Field of View Angle, the minimum vertical field of view angle, and the maximum vertical field of view angle. The horizontal field of view assumes that half is on either side of the direction the camera is pointing.



Version: v2.00.0 Jul 23 2015

Horizontal FOV angle: 360.0 °

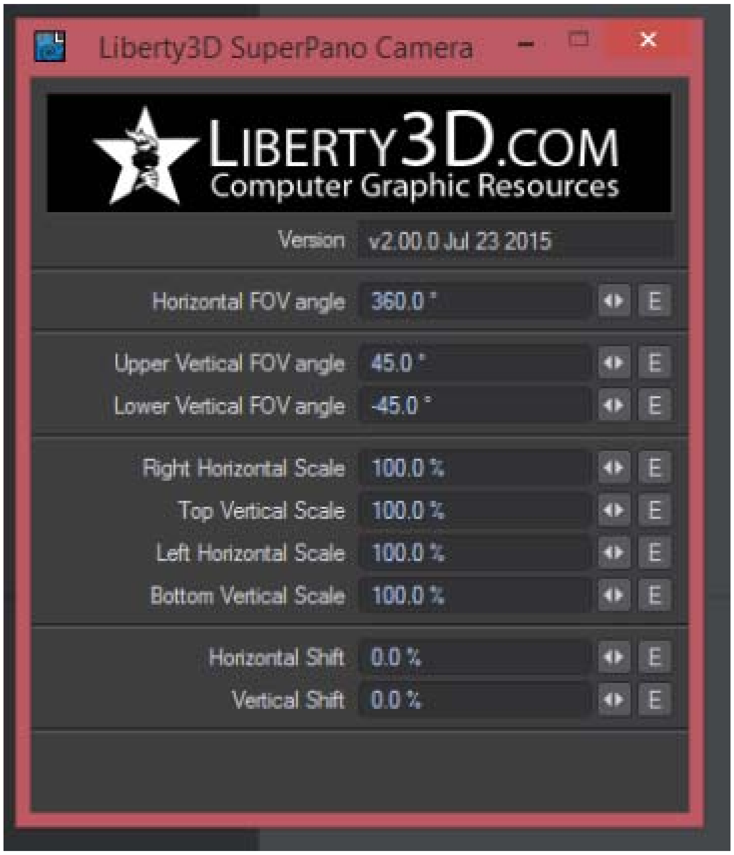
Upper Vertical FOV angle: 45.0 °

Lower Vertical FOV angle: -45.0 °

As with many of these camera types, they cannot be accurately represented in OpenGL through the camera view port. However they can be viewed correctly working with Fprime 3.51 or LightWave's native VPR in non-draft mode.

Liberty3D SuperPano Camera

This camera presents an imperfect hemispherical panoramic view. In addition to the controls on the panoramic camera there are additional controls to provide anisotropic scaling and shifting of the final image. These controls were added in order to help match existing panoramic camera footage, where the program that stitched the still pictures together did so in a non-linear fashion.



Version v2.00.0 Jul 23 2015

Horizontal FOV angle 360.0 °

Upper Vertical FOV angle 45.0 °

Lower Vertical FOV angle -45.0 °

Right Horizontal Scale 100.0 %

Top Vertical Scale 100.0 %

Left Horizontal Scale 100.0 %

Bottom Vertical Scale 100.0 %

Horizontal Shift 0.0 %

Vertical Shift 0.0 %

The Liberty3D SuperPano Camera Options Panel

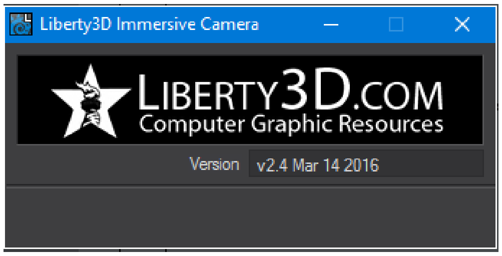
Welcome to the New World:
The Liberty3D Immersive Cameras

Now before we get into the part about how to "use" these cameras, we have to review what they do, where they came from and why. This will assist you in being able to use them effectively, but if you don't care you can skip this part. We highly recommend you don't though.

Let's begin...

New cameras in our collection introduced in UberCam 2.0.1. which have been further enhanced for 2.4 are the **Immersive Camera** (mono) and the **Stereo Immersive Camera** (Stereoscopic).

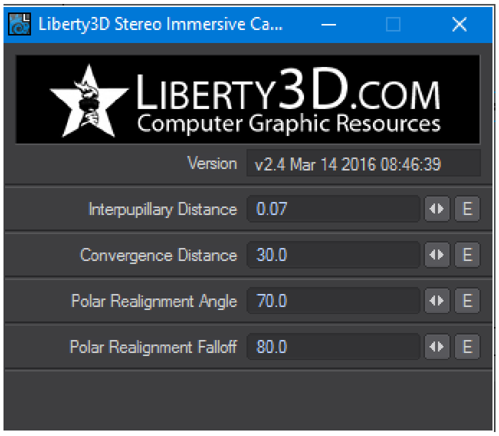
These are two of the most exciting features introduced in UberCam 2.x. We created this camera along with its big brother, the Immersive Stereoscopic Camera, to be really easy to use by being as straight forward as possible. Don't let that simplicity fool you. Make no mistake, these two cameras are right on the bleeding edge of advanced VR production in the CG Industry.



Version: v2.4 Mar 14 2016

Thankfully you are on the other side of that blade in full control.

We wanted LightWavers to have tools that allowed for them to enter this new medium and start kicking ass. Yes, that means you. We know you are capable of amazing work and now its time for that work to show through in a medium that is practically wide open with no single software application or artist type dominating it. We can fight back against the onslaught of Autoplex minions and now is the time!



Version: v2.4 Mar 14 2016 08:46:39

Interpupillary Distance: 0.07

Convergence Distance: 30.0

Polar Realignment Angle: 70.0

Polar Realignment Falloff: 80.0

Our new Immersive Cameras are specifically intended for use in the production of 360 degree spherical VR mono or stereoscopic videos that have taken the VR world by storm over the last 6 months. You will find videos like this now on facebook.com, Youtube.com as well as several boutique VR video portals like littlstar.com. Immersive 360 Degree Format video, photos or renderings have some cool uses that have not been possible until recently.

Head over to www.round.me for a 360 degree experience that lets you jump from place to place in a scene. You can produce content for these platforms in LightWave via the Immersive Cameras found in UberCam 2.4 achieving results that at least from the "camera/lens" quality stand point are second to none in the business. No one has the power of UberCam with respect to Immersive VR Video creation in the CG industry accessible to them, beyond LightWave artists right now. That's special. That means you can pull in clients who want to do Immersive VR video projects or something of that nature, leveraging this information to your advantage.

Let us be perfectly clear about what we are saying here.

The Immersive Cameras ***are unique in the industry***. While there may be other stuff out there from competitors that will do similar things and it may seem that you could use a SuperPano camera from UberCam 2.4, enabling stereo and there you go! Poof! Instant Immersive Stereoscopic VR render out of LightWave natively... but you would actually be very wrong.

The way we have designed and implemented the "camera/lens" system for our Immersive Cameras in UberCam 2.0.1 and up, is absolutely proprietary. They are not repurposed cameras that have existed previously to UberCam 2.0's release. Furthermore, we didn't get this stuff from some Siggraph Whitepaper or open sourced project and it is not possible to duplicate their output using other solutions for LightWave at all. We know. We've tried and so have many other LightWave users. None of these attempts have come close. To do Immersive VR properly we need to create new Cameras to do it.

The design principle for the Immersive Cameras was to make them produce renders that are distortion and artifact (back seams, other weirdness) free as possible in a consistent manner. They should also be really easy for the user to understand and operate while still giving advanced users control that they demand for stereoscopic VR productions. We set out to hit that goal and did it for the Immersive Camera fairly quickly.

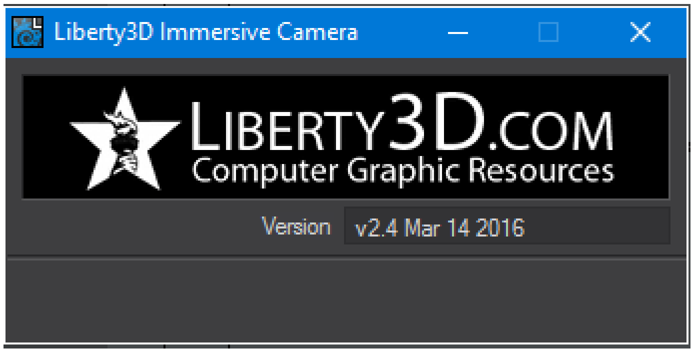
The Stereoscopic Immersive Camera however would prove to be vastly more complex and thus took longer to complete its development and feature set. Here is why. The simplicity of the Immersive Camera would have to transfer over to the Immersive Stereo Camera yet be capable of producing the most comfortable to view image results in terms of stereoscopic effect, easily. We wanted something that could be considered "point and shoot". When it comes to stereoscopic in VR though, artists needed more control. We had to find a way to make these cameras a combination of a Sony EZ Shot and a high end model DSLR from Nikon or Canon. Finding the right balance was critical and this has been a daunting challenge to say the least. In other words, we wanted UberCam and these Immersive Cameras by extension to be something special; something that no product on the market has done in this area and be considered a complete solution at any price point to date. What we have ended up with thus far in both cases is something more along the lines of a high end Spherical Camera system like Nokia's Ozo Spherical Camera (https://ozo.nokia.com) system that works inside of LightWave without any form of stitching or post processing required to get the image results artist are after.

"Ok that's really cool... but why are there two cameras?"

First off, the mono Immersive Camera was made for website or non-stereo VR application use where stereoscopic isn't practical or available. Any other camera solution on the market for LW, including several attempts by users to produce a 'poor man's' Immersive Camera has failed to produce an image that you can get in UberCam 2.4, requiring practically zero set up. The Immersive Camera is essentially distortion and artifact free.

This is thanks to an advanced algorithm that was specifically design for Immersive VR Video content production. All other "solutions" on the market that we have studied thus far including those for some very expensive software have problems somewhere in the image - resulting in distortions, artifacts in the form of seams and more. Stand alone render engines on the GPU and CPU with an Immersive Spherical rendering option or camera as well as plug-ins for other 3D applications offering the same, all seemed to have some kind of "gotcha" in one or more of these areas. Even the closest attempts done using the LW Advanced Camera have been blown out of the water by what the UberCam Immersive Camera produces; distortion and artifact free images, 360 Degrees, all the way around, top and down. Not only that, UberCam's Immersive Cameras are way easier to set up and with rendering speeds due to efficient ray tracing math and full multi-threading in first place they really are the only choice to go with if you are serious about doing Immersive 360 Degree rendering in mono or stereo.

While this may not sound like such a big deal - it is. Other solutions were kind of painful to deal with one way or the other. To make use of the Immersive Camera one only has to place it in a scene. No funky set ups post work, stitching required. A "point and shoot" VR camera for the CG industry basically is what we wanted, and that's exactly what the Immersive Camera needed to be. As you may have noticed, from the Immersive Camera interface there are no options in it. It just works. Here is the GUI again incase you missed it.



Version v2.4 Mar 14 2016

"Point and Shoot" with the worlds easiest Immersive Camera solution for the CG Market. UberCam 2.4 Immersive Camera (mono) gives you everything you need and nothing you don't. It just works.

The Immersive Camera is a single camera to keep things clean and easy for users but also for us while we continued development on a stereo solution. Thus the reason for two cameras but it also came down to demand.

We have always been very responsive to our users. People needed the Immersive Camera right away so it existed as soon as possible and we shipped it as part of UberCam 2.0.1. While we could have stopped there we wanted to go further than just the extra mile with what else might be out there from another developer regardless of the 3D host application or render engine used. We wanted to do stereoscopic VR image rendering in LightWave and not screw up anything that could break someone's scene by adding stereoscopic features to camera that was already being used in production for mono VR rendering. Thus we decided to make this a completely separate Immersive Camera focusing on stereoscopic VR.

With people cooking out renders in the Immersive Camera it was time to tackle the task of making it possible for LightWave users to do stereoscopic VR rendering.

Enter the Dragon:
The Liberty3D.com Immersive Stereoscopic Camera

Ok maybe not a dragon, but it certainly is the elephant in the room. Ok, maybe it's Trogdor. But it is certainly ahead of the current state of the art.

The Immersive Stereoscopic Camera is very much the same if not twice as impossible to replicate with native LW cameras or 3rd party ones available that may do spherical rendering and "work" with LW's stereo rendering option. Go ahead, try it! Our competitors did and failed in one example that was rather interesting.

Indeed, simply firing up a SuperPano-like camera and turning on "stereoscopic" seems to be the "state of the art" approach in some cases with other applications that charge more than UberCam. WAY more. ***But this approach absolutely doesn't work.*** There are solutions out there yes; yet not one of them that we have been exposed to thus far, appears to go "all the way" in terms of solving the problem of rendering immersive VR format in mono let alone in stereo while making it "controllable". "Point and Shoot" is what you want for a mono immersive camera, but the moment you introduce stereoscopic you absolutely need significant amounts of control over that effect. This is super important and what sets our Immersive Stereoscopic Camera apart from the rest. We have been working with artists at top studios as well as independents on the cutting edge of VR content creation to ensure we get it right.

Moving at the speed of Virtual Light

VR is evolving quickly. This mostly comes from the fact that this is all completely new stuff. In general many of the things we are seeing happen in the advancements of VR especially with regards to Immersive imagery have never been done before. There is no standard or example to follow or borrow from. Earlier versions of our cameras had issues too, but we were able to quickly resolve them and now we feel that the Immersive Cameras are an incredibly strong tool set that VR content creators at any level, can rely on.

Since their introduction, the Immersive Cameras based on customer reaction thus far have been justifying the price of UberCam 2.4 on their own. Some have gone as far as to call it 'Game changing". We really appreciate that response and its been amazing but we want to say thank you by letting you know we think we can do more on this front, helping you as a LightWave artist to stay in the game and this time around - be ahead of the curve. LightWave is incredibly powerful in the right hands and we know that it is now time for it to shine again. This new medium brings forward an opportunity that hasn't existed for LightWave artists and the software in general for many, many years. Let's make it count. We are doing our part by continuously advancing this technology and quickly.

The Power of Presence: The Key to Impactful VR experiences

Ok you are ready to kick ass right? Good! Now for some technical stuff.
Some might ask the question: "Why is the stereoscopic portion of an Immersive Camera's technology so important?" The answer is simple. Very simple. Surprisingly thought our competition doesn't seem to get this.

Bare with us as we explain...
To watch in VR a mono immersive format video is one thing. Yes, it's very cool but it does feel like an image projected onto sphere that's playing back for you as you sit in the middle. This mono Immersive 360 Degree Format is better suited for use with Facebook, Youtube.com (via a webbrowser), the web version of littlstar.com as well as Round.me or in situations where Immersive Stereoscopic could be limited in some way such as bandwidth requirements or storage availability.
The above mentioned scenarios have another commonality that is these are examples of uses where you don't' have access to a VR head set but still want make use of the spherical nature of the format in some way. Typically this means panning around in a browser with your mouse or changing the direction of your smart phone as you look around.

No matter where it ends up, your renders have to be looking sweet and the use of our Immersive Camera in UberCam 2.4 nails the "spherical" part so you don't have to worry about it. People are really taking advantage of it lately and it awesome to see the work that's coming out from our users now that they have had some time to play with it so don't count out mono Immersive 360 format presentation down the road. Not by a long shot! It's here to stay. ***HOWEVER...***

When it comes to Immersive VR formats, if a project is done in stereoscopic and done right, you really do feel like you are there when watching an Immersive Stereoscopic 360 degree VR video. The sensation of "presence" is realized incredibly well. We go from the "cool" factor into the realm of "that is fraking awesome!".

Of course, that is where we want to be. Fraking awesome. The level of "presence" or that fraking awesome factor is what absolutely "sells" the VR experience and the only real way to get there is via stereoscopic presentation of imagery.

This has been a primary focus of VR developers on the gaming side since Luckey Palmer, the founder of Oculus coined the phrase in a speech at Oculus Connect 1.0 back in the fall of 2014. In that speech he referred to establishing presence and to not do anything to that would break it, spoiling the effect. Well, gee Luckey... Thanks for that awesome tip. Duh!

While he has a point, at the time he was referring to real-time game engine style presentation in VR as we mentioned. Immersive format video was still in its infancy and to a degree remains that way today. But that's quickly changing. Game engines have a much easier time of doing this, but delivering content via a real-time engine still remains a barrier for content creators for a variety of workflow and technical issues. It's also limiting factor in a lot of ways in terms of audience accessibility.

There was also a big lesson to be learned from the 3D era from a few years ago. While 3DTV and 3D films are neat, the effect has been a let down for most. People want holographic objects in front of them (in the case of AR technology) that they can almost reach out and touch much like has been demonstrated in movies over the last 15 years (Minority Report, Iron Man) or what Hollywood had us hooked on back in the 90s with representations of VR where we could lose ourselves in the experience. This is what the audience wants, and this is the holy grail. With Immersive VR video in stereo we are getting to where we need to be very quickly. We are no longer hearing "Access Denied" in our ears so to speak, especially as LightWave Artists. Indeed, we now have an advantage in this new medium made possible through UberCam.

VR Production Challenges

The production of Immersive VR content is still having some growing pains though. While it goes through these pains the images results are going to be all over the place for some time before a "standard" is kind of settled on. Here is why; It's freaking hard and there are a lot of variables. Inside the computer these variables are reduced greatly but for those working outside the computer... Yikes!

Shooting in VR is a rather difficult, if not an expensive endeavor to try and pull off in the real world. You may have seen various rigs using these cameras such as the GoPro around the net or some of the new first generation spherical camera systems that have hit the market just this past month. All of these real world approaches are attempts to make shooting VR practical, but it still remains rather daunting, confusing and time consuming with final output results varying greatly from solution to solution, approach to approach. It is this way because 360 degree camera rigs are still evolving and improving while dealing with some physical limitations making it difficult to achieve stereoscopic image capture in the most effective way. Real world based VR film makers are then facing another difficult challenge of stitching the results together in stereoscopic where the resulting video material is free of distortions, artifacts while trying to keep the stereoscopic effect comfortable. These factors make the goal of achieving "presence" a rather difficult one especially if you are trying to make it consistent shot to shot, show to show.

One brutally obvious problem with live action VR shot using conventional camera technology is the presence of unwanted stuff. You know... stuff like the camera itself, lights, cables, microphone booms, directors, PAs who won't move when they are told to, make -up trailers, craft service (which of course has to be within 50 feet of the director and your director hates CG or Post) and last but not least some grip or gaffer somewhere with saggy pants showing off the latest fashions in underwear. Not fun times! Humor aside, these are problems on non-VR shoots and it just gets worse with a camera that that can see everything around it. Even specialty designed cameras that do all the stitching and camera synchronization work internally can't completely eliminate these issues and the stereoscopic effect is largely fixed.

In the computer, many of these issues, if not all of them vanish. Fully CG produced content for this reason leads the way as we can make thing consistent and we have control on a level not possible in the real world. The Immersive Stereoscopic Camera does what is, at this time physically impossible in the real world even if those issues we described above were eliminated some how, on a real world set.

Utilizing a highly proprietary Camera Ray Tracing algorithm we designed over the course of the last 7 months and introduced in UberCam 2.1 just 10 weeks ago, we have been able to go even further with the product by providing a control that handles the gradual collapse of the stereoscopic effect ending at the +/- angle near the poles set in our Stereo Immersive Camera.

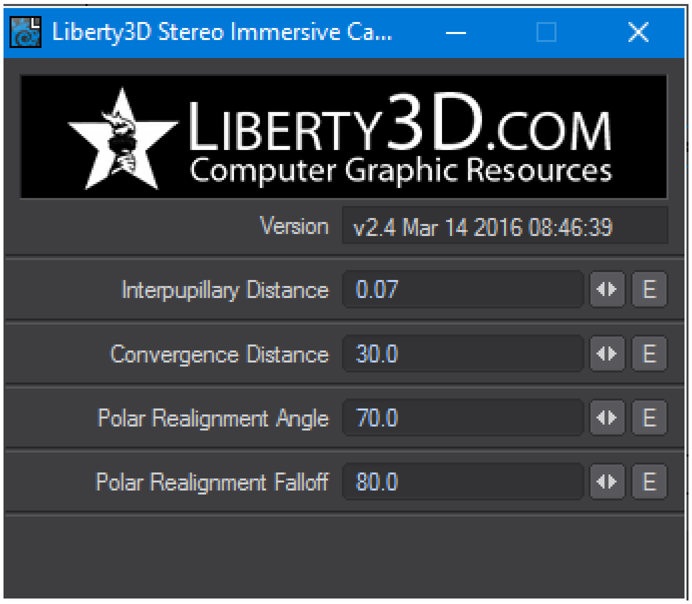
Something that plagues practically all Immersive Stereoscopic images, whether shot on real cameras or produced in other applications from a variety of companies are the numerous, distracting, and sometimes nauseating distortions found at the poles on the Y axis. Before we introduced this advanced functionality via our algorithm; earlier versions of our cameras had similar problems with these artifacts and distortion. This was unacceptable.

Stereo Immersive Camera Controls

Our algorithm is a solution that provides for control of the stereoscopic effect in full. By providing access to it through animation controls a LightWave artists is used to, one now has the ability to control the behavior of this stereo collapsing effect precisely.

This is a world's first for CG based Immersive Stereoscopic Camera technology and it is called "Polar Realignment". It can be found under the IPD (Interpupilary Distance a.k.a Eye Separation) and Convergence Point controls in the Immersive Stereoscopic Camera shipped with UberCam 2.4. While some other competitors solutions may do this under the hood we have not encountered one that gives you access to that function with a control over it that you can animate.

Let's take another look at the Stereo Immersive Camera GUI again and familiarize ourselves with it.



Version v2.4 Mar 14 2016 08:46:39

Interpupillary Distance 0.07

Convergence Distance 30.0

Polar Realignment Angle 70.0

Polar Realignment Falloff 80.0

The Stereo Immersive Camera in UberCam 2.4 with its default settings.

Having access to the Polar Realignment Angle we feel is very critical to creating the best Immersive 360 Degree Stereoscopic VR images and animations possible. We have also introduced a control for how this effect behaves as well in terms of fall off. This is called Polar Realignment Falloff.

Now as mentioned, the purpose of Polar Realignment is to "collapse the stereoscopic" effect so that when it reaches a certain angle value determined by the Polar Realignment Falloff value, we will have reached monoscopic.

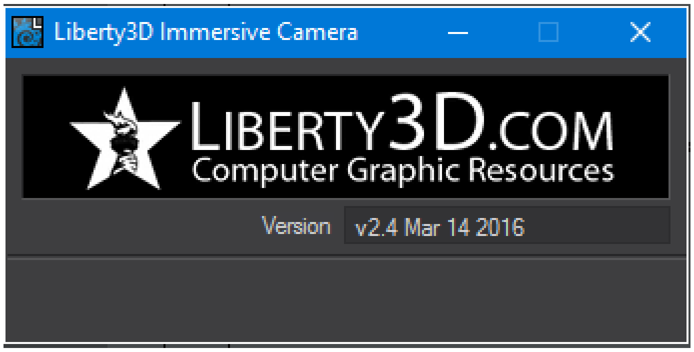
As we introduce more functions that give you access and precise control over what our algorithm is doing, we ask that you fully understand that from one version to the next certain things that we do to improve the product may change the look of your renders.

We don't want you to be discouraged from using the Immersive Stereoscopic Camera by any means. We just want to make you aware of this so that should you do comparisons of renders from identical scenes that used older versions of UberCam with using a newer version than 2.4, that you will be able recognize what the differences in the images are. This way you can take note of the enhanced capabilities and exploit them in your work.

Our goal is to make things look better, more comfortable and not worse with these advanced features. So you rest assured we won't step backwards in this area.

Now then with all that said, let's take a look at the interfaces for these two cameras, what their image render results are and what do with your renders so you can see them properly on various Immersive VR platforms. It may seem like we are going over territory again here but we want to be as clear as possible about the technology behind them which we have covered in this section and now how to make use of them effectively inside of LightWave Layout while also guiding you through areas of interest once you have your finished renders and how to handle them. We have to rewind a little bit to the Liberty Immersive Camera but this is a worth while read.

Liberty3D Immersive Camera

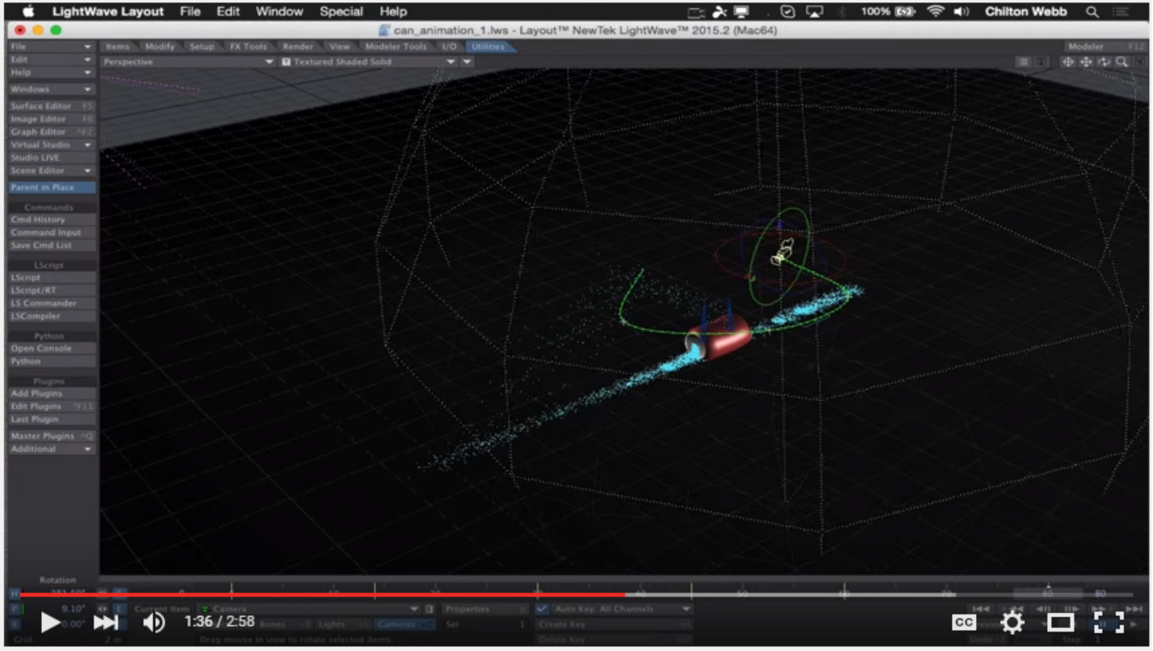


Version v2.4 Mar 14 2016

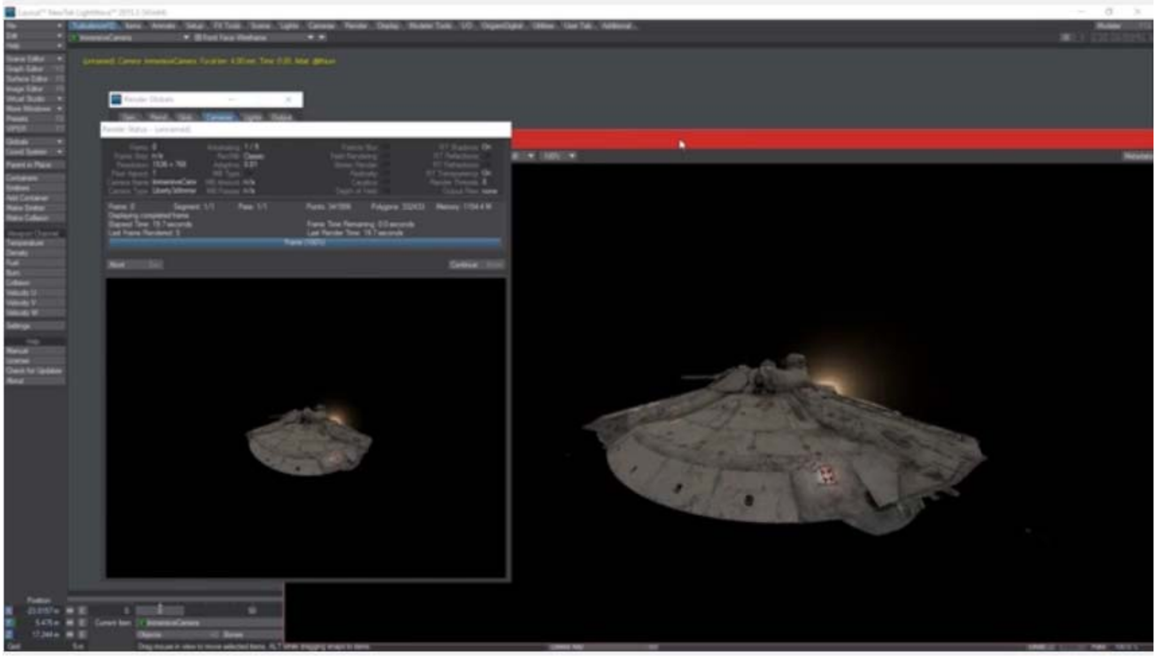
A super simple interface but this camera can make you money!

Yup. That's it for the Immersive Camera that renders in mono. Like we said, we wanted it to be "Point and Shoot". Chilton Webb has produced a quick video here that you can watch that shows this camera's functionality.
https://www.youtube.com/watch?v=uDc679zKW50&feature=youtu.be

This video was recorded in July of 2015, and we have improved the render results that you get from this camera but the GUI hasn't changed. It's still the same.



Kelly "Kat" Myers has also produced a video that should help you with being more accurately able to judge the camera's relation to your subject when working in OpenGL wireframe or shaded modes. This will help you while working to frame your shots. Please watch this video for further details. This concept applies to either the Immersive Camera or the Stereo Immersive Camera. Check out the video below. https://www.youtube.com/watch?v=1rtQJ5I7Iw8



What to do with your renders!

The Immersive Camera in UberCam 2.4 on its own allows you to produce these spherical renders. That much is clear, but there is a trick to getting a resulting video file onto youtube.com so that it is understood to be a 360 degree spherical video for use with cardboard and your android phone or be able to look around using your mouse via a web browser.

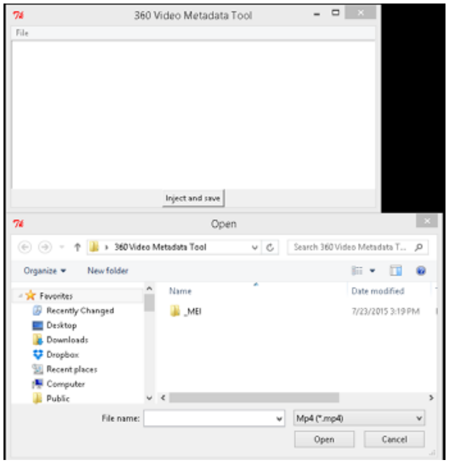
First of all we suggest you use Fusion 7.x or QuickTime Pro to make your MP4 (MPEG-4) video from an image sequence render out of LightWave regardless of where or how you decide to present it. If you are having difficulties getting a good compression to file size ratio, another compression step via an application like Hand Brake may be warranted. Keep in mind that any time you compress already compressed video you're significantly reducing your final presentation quality. This can severely impact audience enjoyment especially when it comes to stereoscopic material. Certain video editors (Adobe Premiere) prior to CS5 may have problems processing material into MP4 at higher resolutions. Believe it or not, cheap video editing apps and even free ones seem to be better at dealing with these things.

If you have the hard drive space, Hand Brake may permit you to work with uncompressed material directly so you can avoid a compression step in your work flow. Other options would include rendering to professional formats such as Apple's Prores or a number of options available from Avid as an intermediate format. Before you go and spend any money on a professional codec, make sure there isn't a free option already available (such as Lagarith) that may work for you and make sure these encoders work with your preferred applications for doing editing and saving your work out at these resolutions.

YouTube MetaData Injection

Once you do have your video ready for upload to Youtube we need to do the "trick" and that is to inject MetaData into the MP4 file through a special tool that tells YouTube this wonderful creation of yours is 360 degree video. For Youtube presentations you need to do this regardless of it being Mono or Stereoscopic.

Google has produced a MetaData file injector that is super easy to use with video files destined for its Youtube service. You can download the tool for Mac or PC here and find out more about how it works.



Google and Youtube.com are both pushing 360 Degree VR videos on youtube heavily. At Comicon 2015 several companies, studios and productions handed out over 1.1 Million (or something like that) Google Cardboard units. This year its going to be much bigger and events like SXSW are being dominated by VR where even more CardBoard is being given away. What this means is there is a very big audience that needs some stuff to watch and now you can make that stuff. Google recently announced that they will start injecting commercials produced in 360 Degree spherical VR video format. This means you can get in on the business of producing advertisements for clients who want 360 degree spherical videos for Youtube.com using LightWave3D and the UberCam 2.4 package. As we said, these cameras can make you money. So get to work!

VERY IMPORTANT NOTES AND TIPS:

First and foremost, it is important to note that you should use resolutions that are 2:1 (twice as wide as it is tall) when working with the Immersive Camera. This is critical if you want to produce absolutely distortion free material for use on online such as at Youtube.com, with their app or in a variety of spherical video players out there. If you don't do this, you will end up with strange stretching or a seam at the back of your spherical presentation. We have had users in the past come back to us saying that we had broken our camera or it wasn't working properly because they were encountering a seam up the back of their spherical presentation looking like a fine line or a large streak of pixels as well as the poles being mashed or nasty looking in some way. This typically results from an incorrect resolution setting either in LightWave or some kind of change to the resolution in a compositing or editing program or encoding tool that breaks the image being a perfect 2:1 aspect ratio.

Now when we say absolutely distortion or artifact free such as this back seam problem we mean the image result itself coming directly from LightWave's render engine with no post processing image effects applied. This includes LightWave's glow effect on surfaces. Any kind of post processing filter inside of LightWave could be a problem, so our recommendation is to avoid using them if at all possible. Also be careful with glow or blur effects in a compositing package. Many of these tools do not understand how to properly "wrap" the effect from edge to edge and thus you get a seam.

Considering Compression

Additionally, what a streaming platform does or compression does is beyond our control. We have heard of and seen problems on some websites including youtube.com that, through their method of displaying a 360 sphere or compression method creates a seam up the back. Indeed, youtube.com recently corrected a problem with their HTML 5 player component for spherical video to address a bug that was creating a seam when there was none in the image to start with.

Whenever possible, aim for high the highest resolutions you can. Suggested resolutions are 2048x1024 (minimum) or 3840x1920 (median, currently accepted "standard" in most situations) or a derivative of this multiplied or divided by a factor of two 2 Youtube.com will process material even in these high-resolution formats and beyond. Of course the higher you go, the more rendering time is involved, but if you can, aim for 4K by 2K. This will make your videos stand out and look amazing. Google is watching! Not only that, but there is more to life than the church of Google. Several VR centric streaming services such as Littlstar.com support 3840x1920 by default.

Almost always your material is going to cap out at 4K for playback on youtube.com and similar services as that is the currently max supported resolution for playback (but not for upload) that makes sense for them at this time to provide.

If your are looking to present video material on a smart phone like an Android directly via a file located on the device, you shouldn't exceed 4000 pixels wide (or high). The reason for this is due to a ***current*** OS limitation that restricts hardware from playing material beyond that size in either pixel dimension if exceed. Keep your video material at 4000 pixels or lower in either direction to avoid this problem. For this reason we suggest something just under 4000 pixels that also translates well in other ways like our suggested 3840x1920 image resolution.

Always use the best source material you can produce to feed their encoding technology and you will get the best results out on the other end. If you upload heavily or even modestly compressed MP4 format video you will regret the final results for sure, especially in this format. With that said, there is a point of where increased frame rate can have a seriously negative impact during final presentation.

Hi-Speed Frame Rates: Don't believe the hype!

One other recommendation is that when you produce your content, frame speeds like 24fps or 30fps (not 29.97, this isn't broadcast television!) will be better in terms of fluid playback and cleaner compression once YouTube's' encoders step on it. Going beyond this is largely insane.

Here is why...

60fps (or higher) seems to be an obsession with some VR film makers, but there are problems with that frame rate sucking up valuable data rate that would normally go to image quality which leads to more compression artifacts and this can be horribly distracting in VR. If you think it looks bad in mono it gets worse in stereo. A lot worse. We recommend using a tool like HandBreak for your MPEG-4 compression step as it provides the best range of options to control the compression and is fast to encode material to MPEG-4 format files. Don't be sucked into the "higher frame rate is better" game. This is basically the "loudness war" in the audio realm but applied to video. Another reason why we generally lean away from producing material at excessively high frame rates is that it doesn't make sense when you consider what is displaying it and how when working with video of this nature.

In a real-time game engine it may make sense for higher frame rates because the headset is driving the engines draw calls for graphics and you want those calls to be fed the latest information all the time as fast as possible because those graphics draw calls and other pixel rendering instructions are incredibly fast and if they have to wait for data, fluidic motion is interrupted. Game engines do this so they can efficiently present only that which is in the field of view of the player. But with video in VR currently, the entire presentation sphere is being rendered consistently all the time at the max speed the device can handle and a higher frame rate isn't going to do jack squat for you if that device can't play it back in the first place reliably. Excessive frame rate just makes the problem worse.

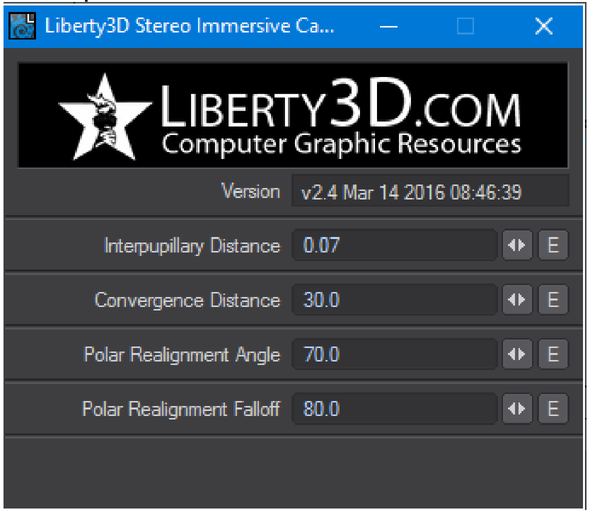
While your device might be able to handle 60 frames per second play back for video at 1920x1080 or even 3840x2160 (4K UHD), its doubtful it can handle playback of 60fps material while producing a spherical object to project the material on, then handling the lens distortion for VR all the while working to update the direction you are looking in getting data from the sensors at a refresh rate pushing 120hz consistently. It's either going to choke, or try to compensate by reducing how nicely it decompresses the material in order to maintain frame rate. This basically translates a horrible experience but now you know why from a technical stand point. This gets even more difficult to do with stereoscopic material because of the addition resolution, bandwidth consumed from the device storage, the graphics processor demand increase... The list goes on.

So when some "VR" game tech geek who thinks he or she knows everything says you should produce and encode your material for 60fps, tell them to get bent and why.

Liberty3D Immersive Stereoscopic Camera

For those of you who want to go all the way with your Immersive rendering, the Liberty3D Immersive Stereoscopic Camera is the ultimate weapon for creating cutting edge VR Immersive Video. Based on similar principles as our Immersive Camera and Stereoscopic Camera, when these two elements are combined, the Immersive Stereoscopic Camera gives LightWave artists access to a rapidly expanding and lucrative market opportunity through a presentation format that is being hailed as the future of the entertainment. It also presented major technology challenges that had to be over come. These challenge have driven the evolution of the Immersive Stereoscopic camera quickly since is introduction in UberCam 2.1.

Immersive Stereoscopic Camera GUI in UberCam 2.4 with Default Settings



Version v2.4 Mar 14 2016 08:46:39

Interpupillary Distance 0.07

Convergence Distance 30.0

Polar Realignment Angle 70.0

Polar Realignment Falloff 80.0

As you can see from the image above there are 4 controls in the Immersive Stereoscopic Camera as of UberCam 2.4.

They are as follows:

1. Interpupilary Distance (Formerly know as Eye Distance in previous versions).
2. Convergence Distance (Also known as CP or Convergence Point)
3. Polar Realignment Angle
4. Polar Realignment Falloff

All of these controls can be animated over time by accessing the "E" buttons next to them and we have provided mini-sliders for easy, quick adjustments.

The defaults in the GUI for the camera as shown in the example above are represented in meters for the first two functions in the list above. For Interpupilary Distance (or IPD for short), 0.07m is a typical distance for the space between the centers of the pupils for adult humans. It is possible to use negative values for IPD. This can produce some very interesting results but we generally recommend avoiding negative values.

Now what is Interpupilary Distance? What does it do?

Well, IPD is essentially "stereo" effect control. The higher the value moving away from 0.0 (which would be no stereo effect and thus mono) in either direction (+/-), the more drastic the stereo effect. Excessive values beyond what is appropriate for your scene will cause your audience to have difficulty in viewing your material. The key to presence is maintaining a comfortable presentation of the stereo effect. Too much or too little stereo effect breaks this or defeats the purpose completely of rendering in stereo. The best way to test this is on yourself. If it feels wrong it is probably wrong for someone else. If it feels right, go with it. Remember, less is more.

The idea here is to make everything comfortable, yet still gently nudge your viewer towards your intended area of focus.

Convergence Point Control

One of the ways to help you achieve comfortable stereo that is dramatic is by working both the IPD and Convergence Point or CP values together, animating them over time if need be.

The value for Convergence Point (or CP for short) is 30.0m represented in the example above. This value is actually rather suitable for real world scale scenes as people generally focus on points in space approximately 30m in front of them while outdoors or in large room settings. It is why we have chosen it as a default but like anything, this maybe changed from scene to scene. It is also a distance that most say stereo stops being effective in real world scale scenes. Accurate depth perception beyond this range is largely lost. But that's the real world. In the computer we can play with that and that's why we give you access to this control and others to maximize creative options.

As a production tip, we recommend that you base this distance off of what is the subject that you want your audience to pay attention to in the scene first and foremost and then adjust appropriately the controls around that. You want people to look at your subject and it has to look good, but you also need to ensure generally that any other points of interest in your scene feel comfortable for the audience.

The Convergence Point control highly impacts the stereoscopic effect. A Convergence Point too far away (from the camera) from your closest subject can lead to it being difficult for the viewer to look at. Same goes for it being too close, but in both situations, working the IPD and Convergence Point Distance together can bring things into "focus" leading to a nice natural, easy to look at stereoscopic image in VR.

The default of 30.0m is a good place to start, however some artists are of the impression that the best place for convergence would be just slightly behind the front of the edge or tip of the nearest object to the camera. This would include the ground or a ceiling in a normal room for example as these are typically the closest objects found to a camera in a lot of situations. However these are not directions the audience is generally interested in looking at and these directions can create problems visually for the viewer. this is part of the reason why we provide for the Polar Alignment Angle And Falloff Controls which we will get into shortly.

This is simply a different school of thought but the logic here is that by doing it this way, objects closest to you would always be easy to look at without straining your eyes and limit the need to adjust IPD to reduce the stereoscopic effect to inhuman values. When they get too close, they will give you a "doubled" appearance for the part of the object that is crossing over this point and it looks confusing and strains the eyes. This area closest to you, in front of the of the convergence point is called positive space in conventional 3D stereoscopic terms. Put things too much in positive space (nearest to you before the convergence point) and it hurts to look at. By not putting enough things into positive space everything in your scene will appear as if it is artificially beyond your reach, separating you from "being there". This would be considered breaking the "presence" we want to establish and maintain in our work. It can be used to dramatic effect however, like an "out of body" experience or a "fly on the wall" point of view. A great example would be that of the POV of a security camera or a drone and perhaps a bird looking down or an ant looking up.

With all of these factors involved it may seem daunting but use your best judgment and trust your instincts. Part of the reason why some artists prefer the approach we have just described is that it comes off as being natural for human point of view scenes.

When things are extremely close to you in VR, just as in real life, say closer to you than your arms reach, you eyes have to do more work with your eyes to bring it into focus. Therefore by maintaining or at least limiting the minimum distance that the convergence point is from camera using this as a guide you should be able to limit the need to adjust the IPD in order to maintain a good stereo effect that is comfortable to view for all.

Beyond that the next suggested distance to land the convergence point on and keep near until an object comes closer that you want to direct the viewers attention to is that of the ground as it would commonly be the closest object to your point of view which is essentially the camera origin acting as a human head for someone of an average height. In the scenario where someone might be sitting in a chair or a cockpit, the closest edge or surface to the camera would be your idea starting point for the placement of the convergence distance allowing for a little bit of that object or surface to exist in the positive space between the camera and that object. The closer you get however to the camera with your convergence point the more likely it is that you will have to reduce the IPD to keep things natural looking.

Again, this is a guide not an absolute rule. This "school of thought" in dealing with distance is going to be similar to that of what humans usually put between themselves and other objects at a minimum without physically interacting with them via their hands or with in hands reach. Things that are closer then this will penetrate into the positive space but not so much as to be difficult to look at if the still generally they remain just beyond the "arms reach" range. For objects that get closer than this you will find that you will need to reduce the IPD to compensate, keeping that "doubled edge" issue under control. It's a good bet that you will need to adjust your IPD to a smaller value if your convergence point is closer than the middle of your forearms or wrists if you think of the camera as being your head.

You will have to play around to find what works most comfortably and every shot is different and will require some experimentation and we encourage you to take advantage of the animation controls for each function as they will help you achieve and maintain the "presence" that you want in order to wow your audience.

Polar Realignment Angle and Falloff Settings

As mentioned earlier the only exception to values in the GUI being represented in meters are the Polar Realignment Angle and Polar Realignment Falloff values. These are actually a representation of the degree of pitch, for both +/- angles on the axis (Y).

"What the heck do these two functions do? Why should I care?"

Let's answer the first question without getting into any hardcore math.

In short, the Polar Realignment Angle and Polar Realignment Angle Fall off mechanisms work together to control internally our algorithm and how it deals with Interpupilary Distance and Convergence Distance simultaneously when the camera ray traces areas near or at the poles of the Y axis. What this translates into is control over algorithm to "collapse" the stereoscopic effect in these zones of the image.

You should care because it is in these areas of the image that stereo can easily confuse the viewer and looks incredibly unnatural. Having control over this allows you to further create the most comfortable stereoscopic experience possible for your viewers.

With this said, we have determined that the default settings of 70 degrees for the start of the Polar Realignment effect to begin and 80 degrees to where it is fully collapsed the stereo effect represented by the Polar Realignment Falloff are suitable for most cases. Think of these as start and stop points for the collapsing of the stereoscopic effect.

Lower values set for the Polar Realignment angle will make the stereo collapsing function work in a more gradual way but chew into the stereoscopic effect looking more towards the equator of your VR spherical image. The opposite is true for higher values.

The default of 70 degrees was chosen as a "comfortable" suggestion for the effect to start from. Using values much higher than this essentially negates the point of it being available as the fall off to mono looks incredibly drastic and the stereo effect near the poles will still cause viewers to have problems when looking these direction. It can also introduce what looks like a "tear" in the render when values are used that are so close to the poles that the "fall off" to mono is so compacted it because visually looks as to be applied in a linear fashion.

Extremely low values will in turn impact the stereo effect beyond that which is desirable, stepping on your ability to achieve and maintain presence in VR.

We suggest that artists generally avoid putting this number below that of the intended or normal vertical field of view range in a scene. Think of this as your "stereoscopic" framing and one can use the Polar Realignment Angle as a way to direct the viewers "focus" into area you want them to look at because its the most comfortable, pleasing portion of the image that is in stereoscopic. Use this wisely! It's not intended to be used a cattle prod against the eyes of your audience whenever they look up or down beyond a certain level.

While you can set the Polar Realignment Falloff value to a higher number including 90 degrees, it will increase the stereo confusion and distortion of the image at the poles to the point of being distracting. You want at all times to keep these distances wide enough from each other and from the pole angle of 90 degrees so as to not produce any form of artifacting or distortion while still maintaining the stereoscopic effect as deep into the pole directions as possible. This will take some experimentation and it will change from scene to scene which is why we have included these controls in the first place.

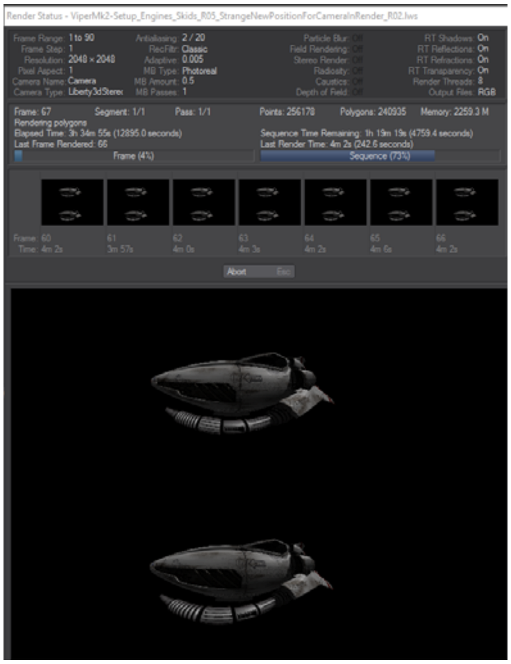
As we mentioned before, Virtual Reality film making generally has dealt with real world scales. Our recommendation is to adjust Convergence as needed first, when working in real-world scale scenes as you would normally, leaving the IPD to default of 0.07 unless your scene warrants adjustment. Examples of this being in the extreme sense are macro and micro scale based scenes where the default IPD value would need to be adjusted to keep the stereoscopic effect comfortable. No matter what, the idea is to keep your stereo effect comfortable and maintain that presence. With all of this said, we have found in our experience that the old saying applies more often than not and that is "Less is more".

Don't treat the stereoscopic effect as a gimmick that is on full blast constantly. This is isn't 3DTV or film and besides who do you think you are? James 'Jim' Cameron!!? Less is more. Remember that. We will get into some technical reasons why in a bit.

This is virtual reality and you are a pioneer in this new medium. Leave the dinos in the dust. Your powers are stronger.

Resolution and Aspect Ratio:

Much like our Stereoscopic Camera, the Immersive Stereoscopic camera renders both Left and Right eyes at the same time to a single image. It is therefore necessary to set your camera resolution appropriately. Just as we demonstrated with the Immersive Camera resolution settings, an aspect ratio of 2:1 is required per eye. However, since the Immersive Stereoscopic Camera renders the Left Eye on the top of the frame and the Right Eye on the bottom of the frame, we need to set our frame resolution to 1:1. Suggested resolutions would be similar to that of the Immersive Camera as we described earlier for frame width represented in pixels such as 2048 or 3840 but since we need to maintain a 1:1 aspect ratio for the entire image saved you can set your height resolution to be the same as your width. This will give you for example a 3840 x 3840 resolution square image. In this image you have left and right eye frames represented, each with an aspect ratio on their own of 2:1 and a pixel resolution of 3840 x 1920.



Render Status - ViperMk2-Setup_Engines_Skids_R05_StrangeNewPositionForCameraInRender_R02.lws

Frame Range: 1 to 90
Frame Step: 1
Resolution: 2048 × 2048
Pixel Aspect: 1
Camera Name: Camera
Camera Type: Liberty3dSterex

Antialiasing: 2 / 20
RecFiltr: Classic
Adaptive: 0.005
MB Type: Photoreal
MB Amount: 0.5
MB Passes: 1

Particle Blur: Off
Field Rendering: Off
Stereo Render: Off
Radiosity: Off
Caustics: Off
Depth of Field: Off

RT Shadows: On
RT Reflections: On
RT Refractions: On
RT Transparency: On
Render Threads: 8
Output Files: RGB

Frame: 67 Segment: 1/1 Pass: 1/1 Points: 256178 Polygons: 240935 Memory: 2259.3 M
Rendering polygons
Elapsed Time: 3h 34m 55s (12895.0 seconds)
Last Frame Rendered: 66
Sequence Time Remaining: 1h 19m 19s (4759.4 seconds)
Last Render Time: 4m 2s (242.6 seconds)
Frame (4%)
Sequence (73%)

Frame:	60	61	62	63	64	65	66
Time:	4m 2s	3m 57s	4m 0s	4m 3s	4m 2s	4m 6s	4m 2s

The Immersive Stereoscopic Camera rendering in the Render Status window in LightWave Layout.

For optimal results we recommend a resolution of 3840 x 3840. Presented in Stereoscopic VR this will really look stunning. If you want to go higher you certainly can, however there is a catch in some cases.

At 4096x4096 one can get stunning results when viewed in VR, but some devices on the market are unable display such resolutions and are capped at 4000x4000. We have mentioned this before but again incase you missed it, certain current generation Android phones GPU hardware are limited in this way and thus the OS prevents their playback of material beyond 4000x4000. Additionally certain video editing applications are unable to produce renders to MP4 video at sizes larger that 4000x4000. So what do you do?

In cases where you run into this, your next best bet is to use the emerging standard of 3840 x 1920 per eye. Many of the graphics processor systems on mobile devices are optimized to deal with this resolution and as such its easier to hand them that type of material in the first. By understanding this, we can make stereoscopic renders that are 3840x3840 (Remember, 1:1 aspect ratio for the final stereo image saved to disc), giving us our 3840x1920 per eye resolution. Even if your target device cannot play back this material effectively, you can always downsize it lower resolutions should it warranted. Keeping in mind the "perfect math" for mobile processors and displays is the idea here. Black Magic's Fusion (remember, the basic version is free now! no excuses!) is great for this kind of work and it's super fast.

As with the Immersive Camera renders, once you have your final results in an MP4 file ready to upload to a service like Youtube, or you want to watch on your Samsung GearVR or Google Cardboard you will need to make use of the Google MetaData Injection tool as described in the Immersive Camera section. This is really only needed if you want to watch through the Youtube App on your mobile device.

Some services like Littlstar.com assume all video material uploaded to them is Immersive, but don't know how to deal with stereoscopic automatically. Remember that Immersive Strereoscopic is VERY cutting edge stuff right now and only now are sites playing with this for streaming or download delivery.

Please review your preferred site's instructions on how they handle different material carefully when planning your productions. For those of you who don't yet have a GearVR, Rift, Cardboard or whatever it might be, we highly recommend you check out KolorEyes Desktop player. They have recently released a new version and its perfect for use with renders from either our Immersive Stereoscopic Camera or Immersive Camera. You can find out more information here. http://www.kolor.com/2016/01/11/360-video-player-kolor-eyes-desktop-1-6-beta-final/

*******Some important notes when using the Immersive (mono) and Immersive Stereo Camera:**

Both of our Immersive Cameras works great in VPR with it clearly giving you a WYSIWYG output. However OpenGL wireframe or shaded views are very misleading, in a way. What we mean by this is that while VPR will show you what the physical render will look like, it is nowhere a clearly understood representation of what it will look like inside of a viewer. Things can seem very "pushed back" or "sucked into lens" depending on what the camera is directly "looking" at when compared to the OpenGL viewports. However, once you compare the OpenGL viewport view to what you would actually see in a GearVR or Cardboard, it's quite similar although the FOV is rather limited. It is for this reason that we recommend that you test and experiment with your camera placement and "framing" to accommodate this phenomena frequently throughout your production.

Shooting Techniques

Many people also feel that it is a "no no" to rotate or move an Immersive Camera. This is completely untrue. However it has to be done in way that may feel quite alien to what you are used to as a 3D artist or even as an artist who has worked with conventional cameras. Moves such as static "pans", where the camera is planted in a single spot and then pans in a certain direction may cause the viewer to experience a form of motion sickness, this is true. However if the move is motivate by action in the scene, the viewer's brain has an easier time of maintaining spatial relationships greatly alleviating this phenomena. As George Lucas once said, sound is 50% of the experience so when you go to the sound side of your production keep this in mind. Audio cues can help give people that "grounding" that they need and this will help to reduce any form of camera induced nausea.

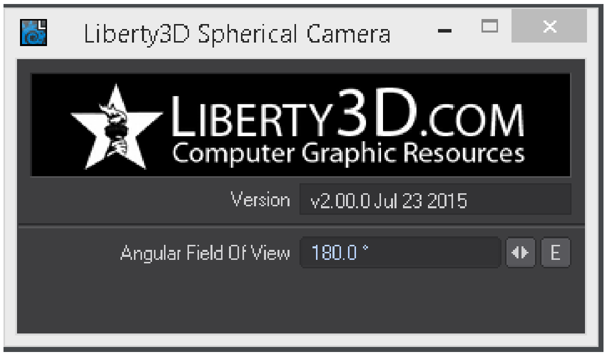
Certain panning moves can be replaced by maintaining the camera's fixation or direction focused on the subject currently in the intended field of view and sweeping the camera around in the direction you wish to establish a new fixation or focus on a subject, finding a new subject and then "handing off" to the viewers interest to this new subject. Anyone who has every operated a steady-cam will understand this principle. These are just suggestions of course. Give this method a try. We would love to see your work.

One of the primary benefits of this type of camera movement approach is that it reduces the amount of "work" that a viewer has to do to look around to find the subject that you as a director or CG cinematographer wants them to focus on. "Lean back" style entertainment can be interactive in Immersive VR video but it's not intended to give you a work out. So be gentle on your audience. Lead them to the eye candy and they will follow but don't make them rubberneck constantly and you will find this produces a much more satisfactory viewing experience.

Liberty3D Spherical Camera

Like the Immersive Cameras, our Spherical Camera is another specially camera that while niche in its use gives the impression of "being there". This camera has found use in the world of hemispheric dome film production. The kinds of movies you would see at amusement parks or at planetariums. We have a few customers around the world that make use of this camera specifically for those purposes.

The only input function on the Spherical camera is the Angular Field of View input value. This function has a default value of 180 degrees which is pretty much standard for dome film projection set ups. This function can be enveloped over time to create some really wonderful effects.


Liberty3D Spherical Camera

Version v2.00.0 Jul 23 2015

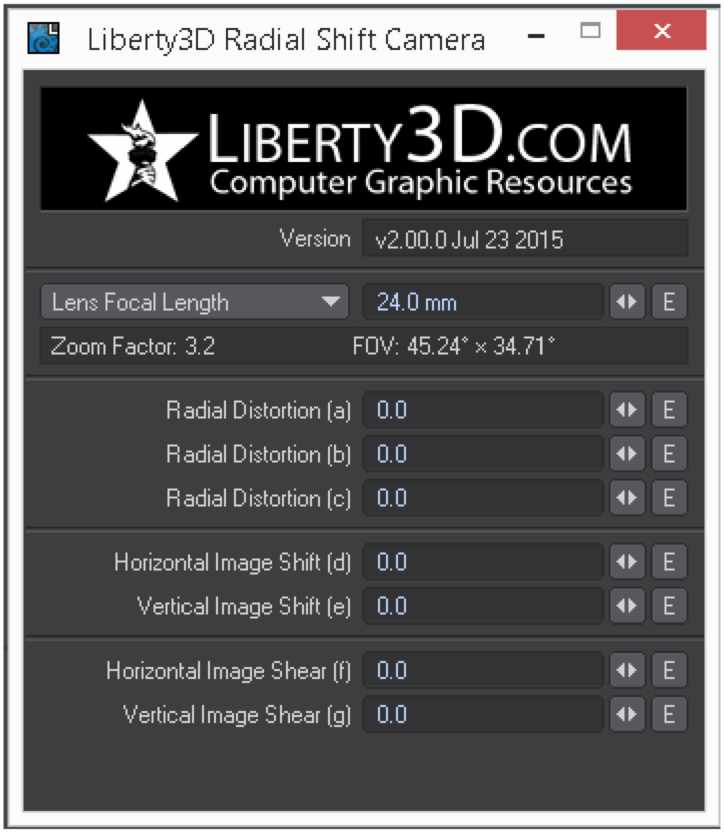
Angular Field Of View 180.0 °

The UberCam Spherical Camera options panel.

Liberty3D Radial Shift Camera

This camera presents controls to match actual camera bodies and lenses, such as used in panoramic stitching software like Hugin or Panotools, or in Denis Pontonnier’s Radial Shift camera. It uses a 4th order polynomial coefficient values to take into account various lens barrel or pincushion effects, including the difficult to match mustache effect.

This can be used so that rendered output files for compositing can be made to match existing footage at render time, rather than going through a distortion matching step during composition. Although the steps involved in calculating the radial distortion values are fairly complicated, there are databases of existing lenses and camera bodies that have already been determined and published on the Internet. If you have a certain camera and lens combination that you need information about, please refer to either the Panotools website, the Hugin tool website, or contact us directly here at Liberty3D.com. Chances are that the information regarding the lens and camera combination can be found fairly quickly, or be calculated from a captured still image.



Version v2.00.0 Jul 23 2015

Lens Focal Length 24.0 mm

Zoom Factor: 3.2 FOV: 45.24° × 34.71°

Radial Distortion (a) 0.0

Radial Distortion (b) 0.0

Radial Distortion (c) 0.0

Horizontal Image Shift (d) 0.0

Vertical Image Shift (e) 0.0

Horizontal Image Shear (f) 0.0

Vertical Image Shear (g) 0.0

The Radial Shift Camera options panel.

Chapter 4 Multi-up Cameras

There are 4 Multi-up cameras in the Liberty3D UberCam v2.4 package.

What is a Multi-Cam camera?

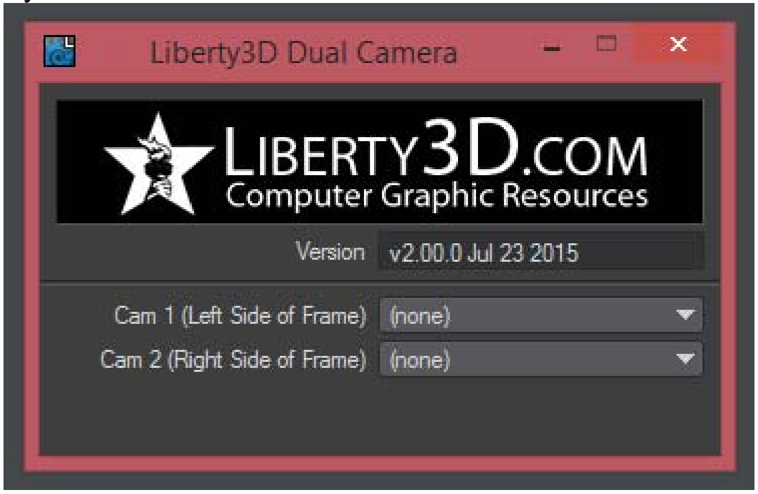
These cameras allow for multiple existing LightWave cameras in a scene to be rendered at the same time. In addition to this there is an OverScan Camera that allows a single existing LightWave camera to be rendered with additional control modifications like OverScan and jitter. Also, there is a DPI camera that it is really useful for people who need to go to print and works wonderfully. Other uses for Multi-Cam Camera set ups are pretty numerous. Working with "Holographic" reflective pyramids or boxes where reflections of renders are produced from a tablet or smart phone creating a "holographic" effect for product displays are a snap with a Multi-Cam Camera.
Here is an example: http://thefutureofthings.com/5151-holho-make-your-tablet-a-video-hologram-projector/

There are many uses, so many that we can't name them all but here are some more ideas. Multi-Cam cameras are useful not only in outputting an existing stereoscopic camera rig that you have built in side-by-side stacked stereo files but also as a pre-visualization tool, where the scene can be adjusted and viewed from multiple camera locations. The 12Up camera can also be used to create custom skybox camera rigs.

Only one instance of Multi-Cam is allowed in a scene, and it is not permitted to select the Multi-Cam as one of the sub cameras. There is only one instance of an Overscan Camera allowed in a scene, which can then select a multi-up camera.

Liberty3D Dual Camera

This camera allows for two existing cameras to be rendered simultaneously side by side on the same render. The main use for this is for custom stereo camera rigs, although this is not the only use for such a camera.



Liberty3D Dual Camera

LIBERTY3D.COM
Computer Graphic Resources

Version v2.00.0 Jul 23 2015

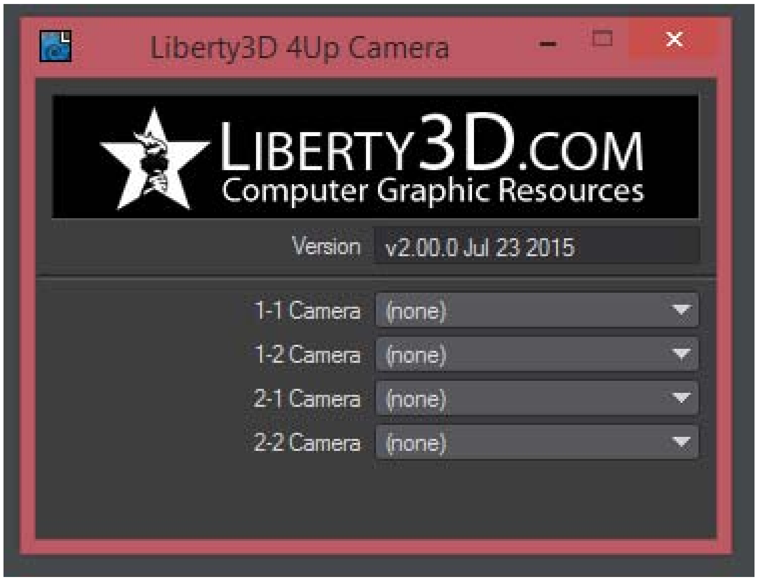
Cam 1 (Left Side of Frame) (none)

Cam 2 (Right Side of Frame) (none)

Dual Camera set ups can be used with our L3DVR-OculusRift Image Filter to produce Oculus Rift compatible still images or animations. You can also use our UberRift viewer should you wish to combine real-time viewing of your camera view with the Oculus Rift.

Liberty3D 4Up Camera

This camera allows for up to four existing cameras to be rendered simultaneously in a 2 by 2 grid on the same render.



Version v2.00.0 Jul 23 2015

1-1 Camera (none)

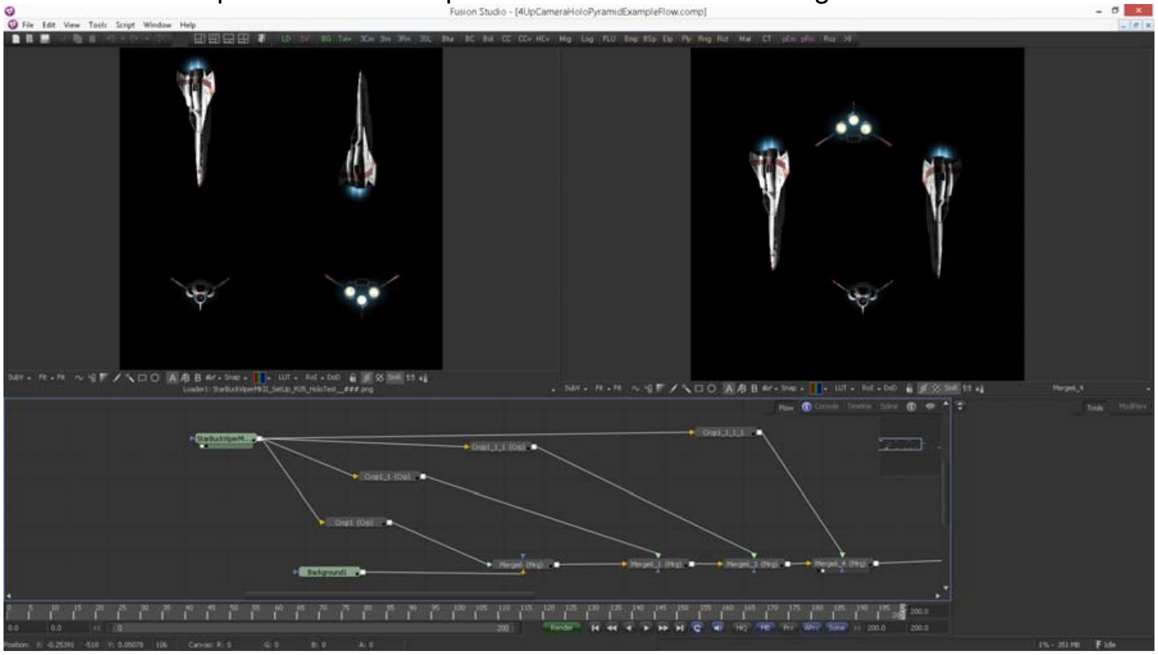
1-2 Camera (none)

2-1 Camera (none)

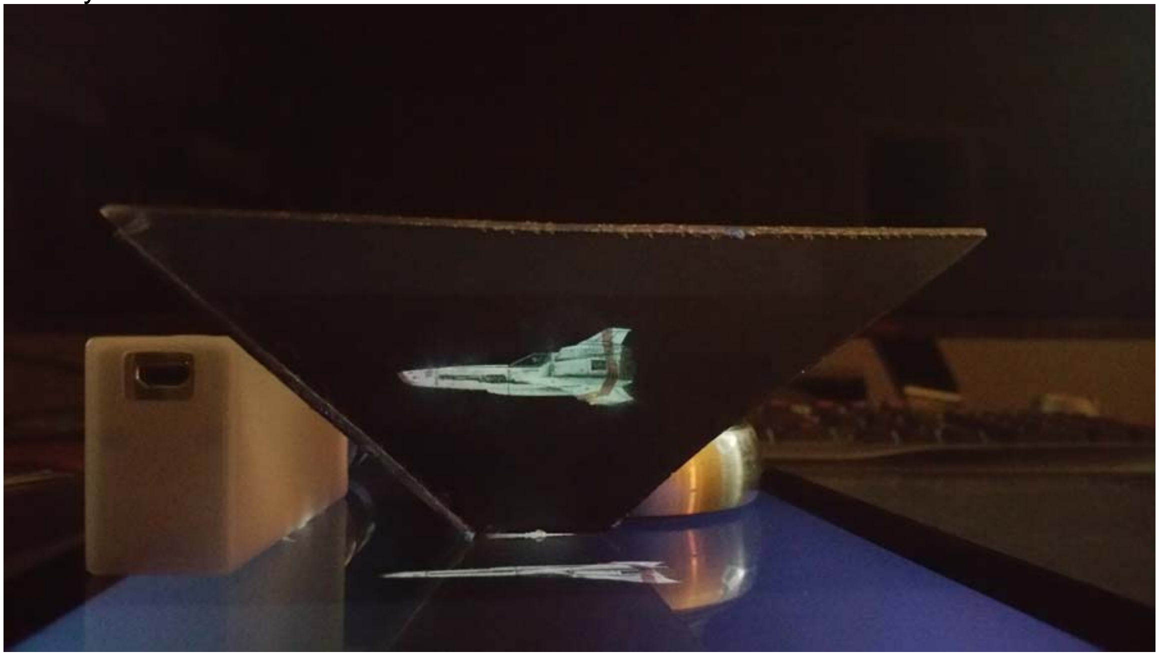
2-2 Camera (none)

As mentioned before, one of the great uses of a 4 up camera is to produce "Holographic" pyramid style renderings for use with tablets and smart phones. You may have to flip your cameras upside down to get it to work right. Also, since our 4Up Camera renders each camera view in a quadrant, you will need to use Fusion or another compositing application to reposition these areas into a checker board arrangement with each image in a cross shape.

Here is an example of a flow set up in Fusion 7.x demonstrating what this looks like.


Fusion Studio - [4UpCameraHoloPyramidExampleFlow.comp]
File Edit View Tools Script Window Help
Loader1: StarBuckViperMKII_SetUp_R05_HoloTest___###.png
Merge6_4
Flow Console Timeline Spline
StarBuckViperM...
Crop1_1_1_1
Crop1_1_1 (Crp)
Crop1_1 (Crp)
Crop1 (Crp)
Background1
Merge6 (Mrg)
Merge6_1 (Mrg)
Merge6_3 (Mrg)
Merge6_4 (Mrg)
Render
HQ MB Prx APrx Some
200.0

A 4 Up Camera Pyramid Example render is shown on the left, the flow in Fusion 7.x produces the arrangement of the Viper render as required to be reflected correctly into a HoloPyramid.



The image above shows the results of this workflow using a primitively built holo-pyramid display made out of old flat screen TV material and a Microsoft Surface tablet as the source video device.

If you choose to build one of these things yourself here are some links for you to check out.

http://www.instructables.com/id/Reflective-Prism/

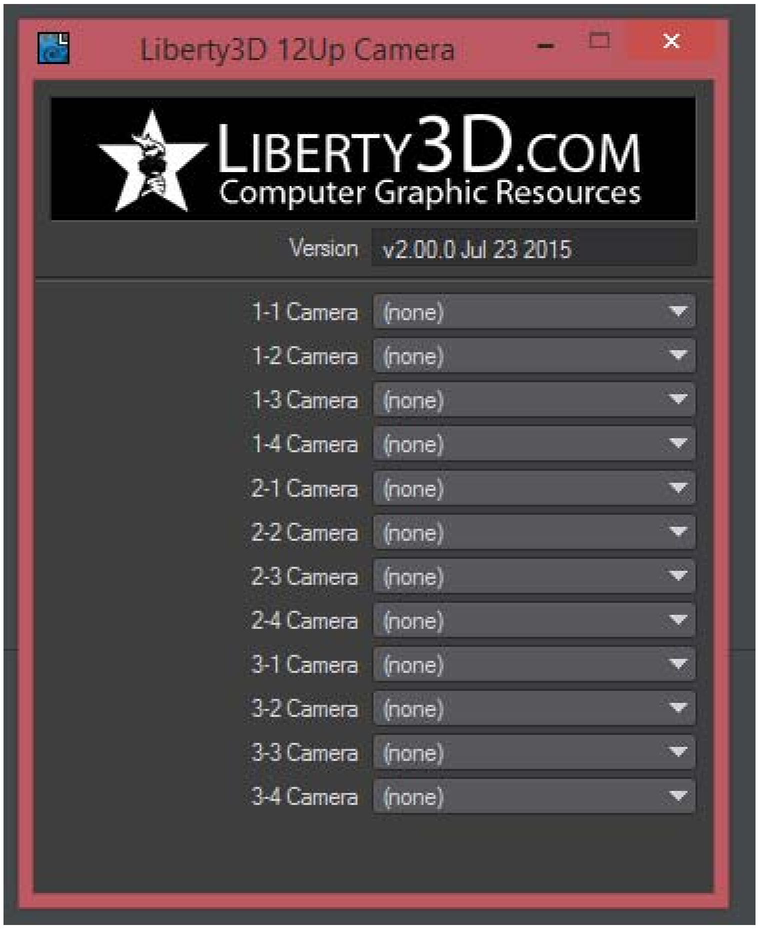
https://www.youtube.com/watch?v=UFhhIOaGW7U

https://www.google.com/search?q=making+your+own+hologram+pyramid&espv=2&biw=1057&bih=682&tbm=isch&tbo=u&source=univ&sa=X&ved=0CDAQsARqFQoTCMGr_YWt8sYCFQg2iAodcAUG4Q&dpr=1

There is almost no limit to how large you can make one of these displays. You can even use a video projector or a large flat screen TV as your video source. This is another great opportunity to make money using LightWave3D and UberCam 2.1.
We have included the example Fusion flow showing the Viper set up in this package for you to work with as a start.

Liberty3D 12Up Camera

This camera allows for up to 12 existing cameras to be rendered simultaneously in a 4 by 3 grid on the same render.



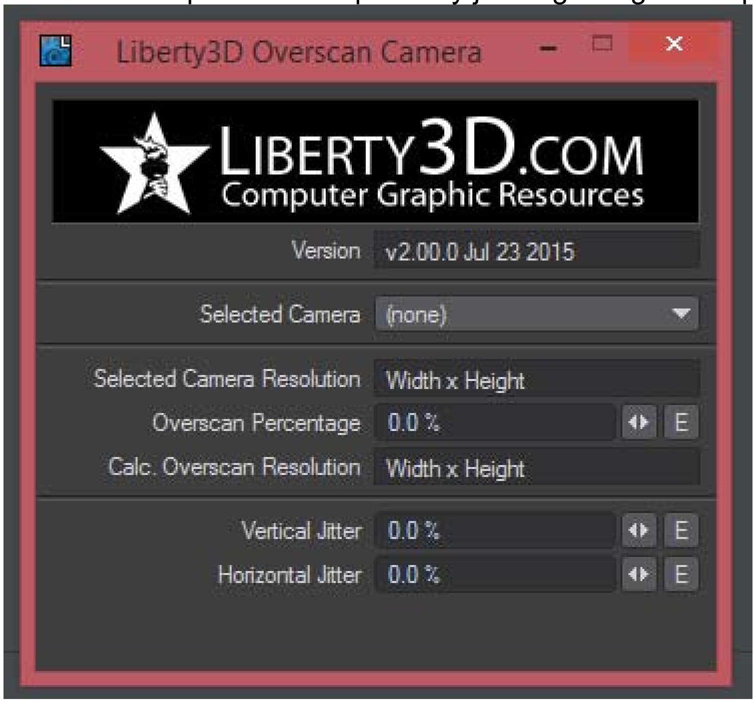
Version v2.00.0 Jul 23 2015

1-1 Camera (none)
1-2 Camera (none)
1-3 Camera (none)
1-4 Camera (none)
2-1 Camera (none)
2-2 Camera (none)
2-3 Camera (none)
2-4 Camera (none)
3-1 Camera (none)
3-2 Camera (none)
3-3 Camera (none)
3-4 Camera (none)

The 12 up camera can be used to build highly specialized stereoscopic spherical rigs similar to those used in the real world to shoot 360 degree video. There are other uses for a camera of this type that we haven't even thought of and we encourage users to experiment with it. We would love to hear from you and see what you come up with.

Liberty3D OverScan Camera

This camera allows for an existing camera to be rendered with an overscan percentage and / or horizontal and vertical jitter. If the overscan is used in addition to modifying the render resolution of the overscan camera output then the render step can recover the original rendered output in a compositing step from the overscan camera output, but would have additional pixels available if something just out of frame turns out to be needed during composition. In addition the shaky camera type of shot can be applied to existing cameras without complex camera paths by jittering using envelopes.



Version v2.00.0 Jul 23 2015

Selected Camera (none)

Selected Camera Resolution Width x Height

Overscan Percentage 0.0 %

Calc. Overscan Resolution Width x Height

Vertical Jitter 0.0 %

Horizontal Jitter 0.0 %

NOTE: If horizontal shredding occurs in the final renders, try using a single thread for the render. This is a known issue. Alternatively however this can be corrected for by baking out key frames on the envelopes and the camera will multi-thread without shredding the image.

Chapter 5 Oculus Rift Functionality and Use with UberCam 2.4

A big part of UberCam 2.4 is the added support for the Oculus Rift Virtual Reality Headset. Ubercam 2.4 provides for real-time rotational head tracking a special viewer known as UberRift and a image filter that let's you produce properly distorted renders for viewing in the Oculus Rift headset. This section details the use and functionality of these tools.

Before we get started though we need to cover off some things concerning functionality and performance.

Oculus Rift DK1.1 and DK2.1 headsets were used in the development and testing of UberCam 2.0 - 2.4 and while the head tracking functionality for rotation is supported from either unit, at this time the positional head tracking (depth, side to side, up and down motions) possible using the DK2.1 head set is not supported. We left this functionality out for this release but it is something we will be adding in once the Oculus Rift SDK evolves a couple of steps further as Oculus prepares for the release of the Consumer version of the Oculus rift in Q1 or Q2 of 2016. You won't have to wait that long however to get positional head track in Ubercam 2.4. We just want to see a couple more updates done to the SDK before including it along with units shipping in bulk.

In addition, while every effort has been made to optimize the performance of the UberRift viewer, the overhead involved in producing the floating view in UberRift combined with the real-time data monitoring of the Oculus Rift and LightWave Layout's OpenGL/Geometry engine performance means that ***UberCam 2.1 is not intended to be used as a platform for VR experiences as you would have in a game engine such as unreal or unity, but instead as a content creation tool.***

Performance of the tools when used in combination with each other highly depends on your system and the limitations of LightWave's performance itself. For example, the OpenGL viewports in Layout are single threaded. It is for this reason that you will experience sluggish response while the UberRift viewer is in use even in lightly populated scenes on the average system. The barrel distortion mathematics used to distort the stereoscopic view is also fairly heavy in terms of processing time. For now in the interest of maximum compatibility the way that these tools are used in combination can put a heavy load on your system ***and real-time feed back may be impeded.*** We are working on a faster solution that reduces some of the overhead to a bare minimum and this will be provided as an update to UberCam 2.1.

Needless to say, we encourage users of UberCam 2.1 to stay up to date by making sure they check our website at www.liberty3d.com and staying subscribed to our mailing list. When we make updates you will be notified through these channels.

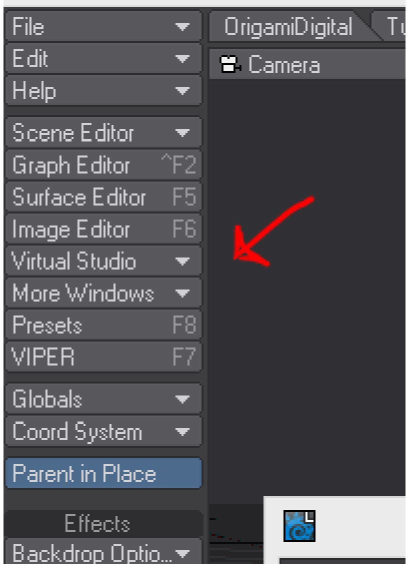
With these limitations understood, we will now show you how to set up UberCam 2.1 using Virtual Studio Tools so you can get the real-time head tracking rotation of the

Oculus Rift to drive the rotation of a UberCam 2.1 stereo camera and see the scene via the UberRift viewer through your Oculus Rift.

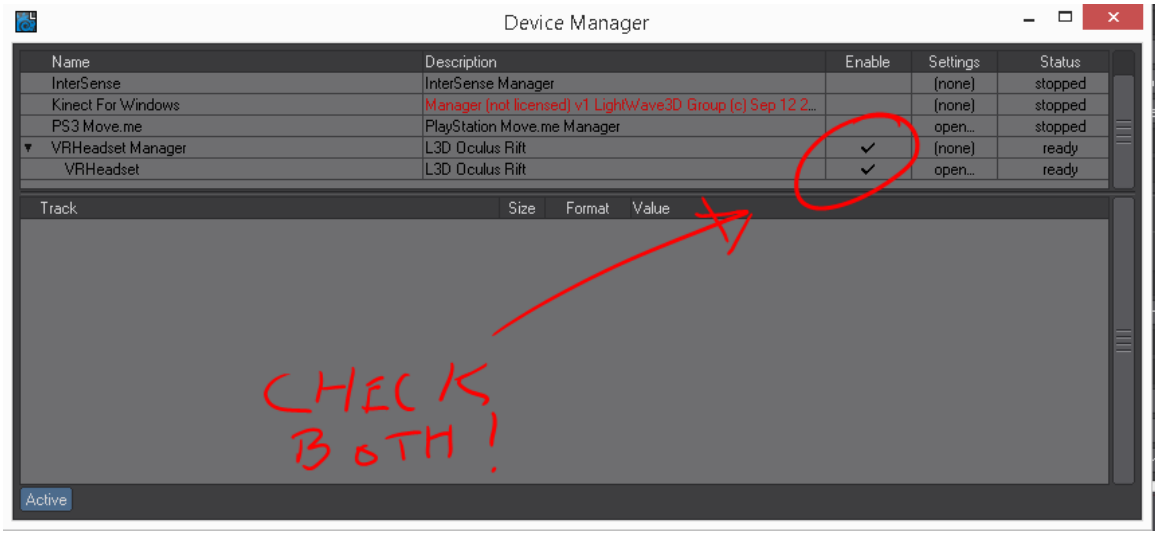
Step 1: The Virtual Studio Tools Device Manager and Setting up a camera

In order to get the rotational head tracking working, ensure that the plug-ins are installed and you have fired up a fresh instance of LightWave Layout. Your Oculus Rift should be connected, turned on and calibrated before you begin. For both DK 1.1 and DK 2.1 users make sure the Rift configuration utility is set up so that you are in extended display mode. This means that the Oculus Rift is acting as an attached display to your video card. For further information on this, please consult the Oculus Rift Configuration Utility Help and the documentation that comes with the Oculus Rift DKs.

Now then... In layout we need to start up the Virtual Studio Tools "Device Manager Window" before proceeding any further. This is found under "Virtual Studio Tools" dropdown menu commonly located just below the Image Editor button on the left hand of Layout's interface



Yup. It's there, just click "Virtual Studio" and you will see a Device Manager button. You want that. When you select that button you will get a window much like this that comes up below.



Name	Description	Enable	Settings	Status
InterSense	InterSense Manager		(none)	stopped
Kinect For Windows	Manager (not licensed) v1 LightWave3D Group (c) Sep 12 2...		(none)	stopped
PS3 Move.me	PlayStation Move.me Manager		open...	stopped
▾ VRHeadset Manager	L3D Oculus Rift	✓	(none)	ready
VRHeadset	L3D Oculus Rift	✓	open...	ready

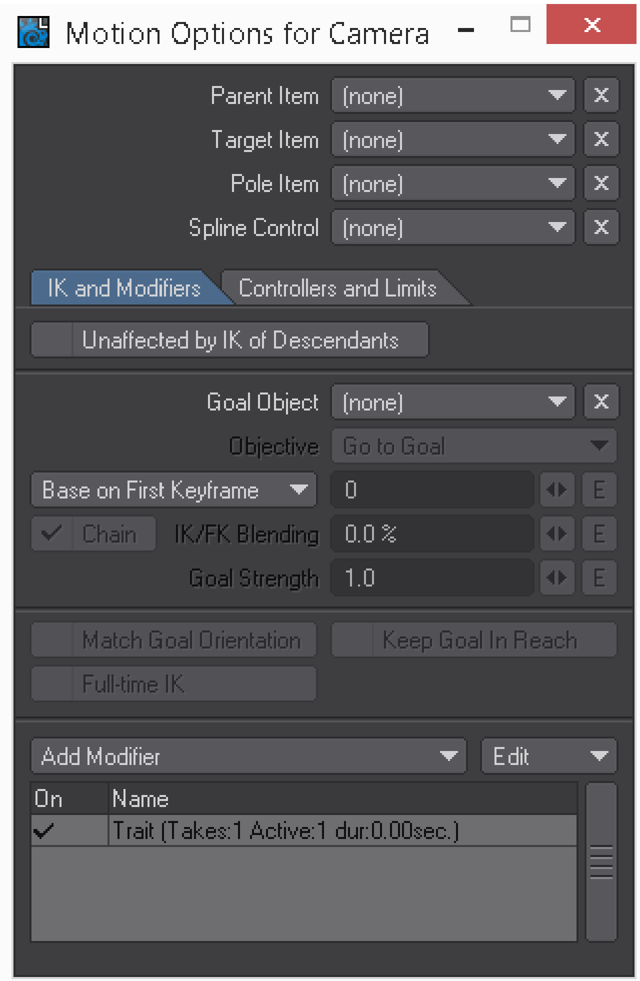
Track	Size	Format	Value

CHECK BOTH!

Enable the L3D Oculus Rift check box as shown.

Once you have this window up, you need to check both of the "Enable" boxes for the VRHeadSet Manager. This essentially "activates" the Oculus Rift and Virtual Studio Tools is now listening for the data coming off the unit. You can minimize this window for now but we will come back to it later.

Next, select your camera that you want to make work with the real-time rotational head tracking being driven by the Oculus Rift in Layout. From here we want to add a motion modifier. Hit "m" to access the motion modifiers panel. An example of this is shown below.



Motion Options for Camera

Parent Item (none)

Target Item (none)

Pole Item (none)

Spline Control (none)

Goal Object (none)

Objective Go to Goal

Base on First Keyframe 0

✓ Chain IK/FK Blending 0.0 %

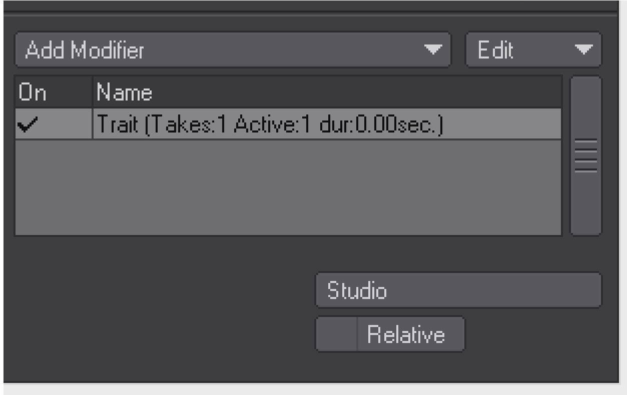
Goal Strength 1.0

On	Name
✓	Trait (Takes:1 Active:1 dur:0.00sec.)

An example of the Virtual Studio Trait Motion Modifier added to the Motion options of a camera.

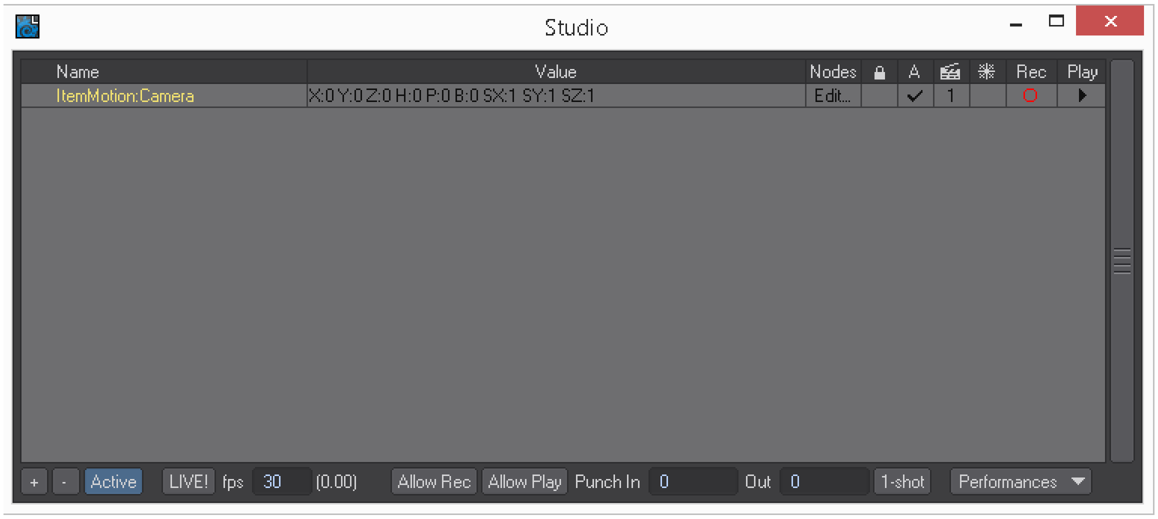
The motion modifier you want to add is called "Virtual Studio Trait". Once you add this, you should see the Trait much like it is shown above. From here double click on the area where it says "Trait (Takes:1 Active:1 dur: 0.00sec.)".

This will expand the window displaying a button called "Studio" and a check box called "Relative". Click the Studio button to continue to the next step.



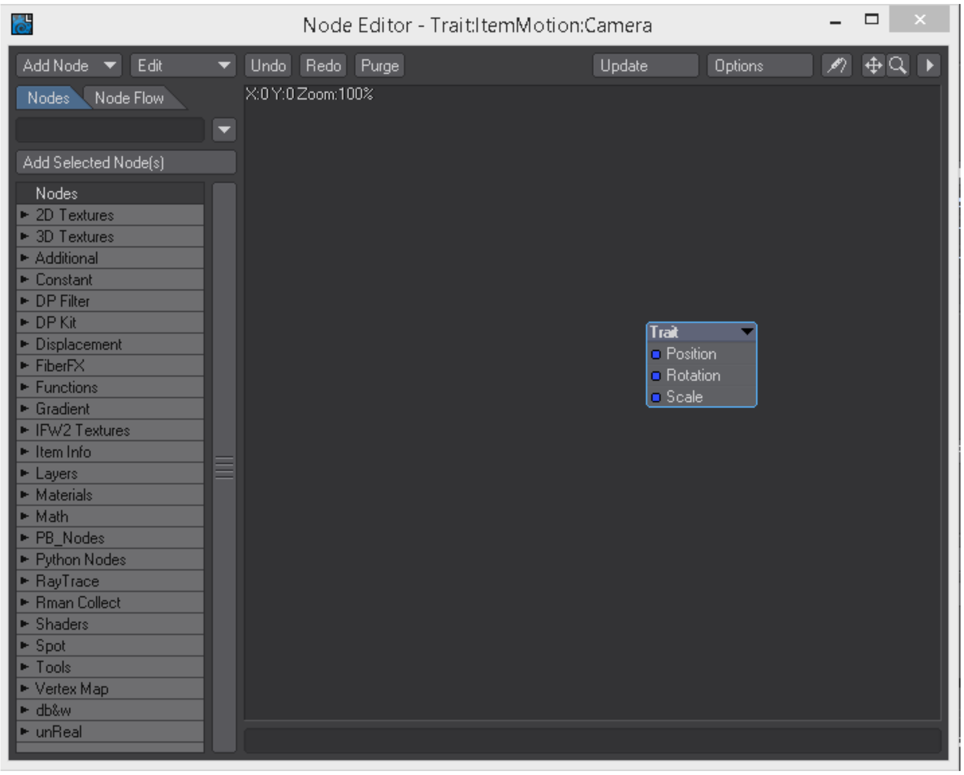
On	Name
✓	Trait (Takes:1 Active:1 dur:0.00sec.)

Step 2: Setting up nodes for the ItemMotion:Camera Trait.
We know that this area of LightWave can be daunting but its really not that bad. Once you click studio you will be presented by a window called "Studio". Yeah... Which looks like what we have below.



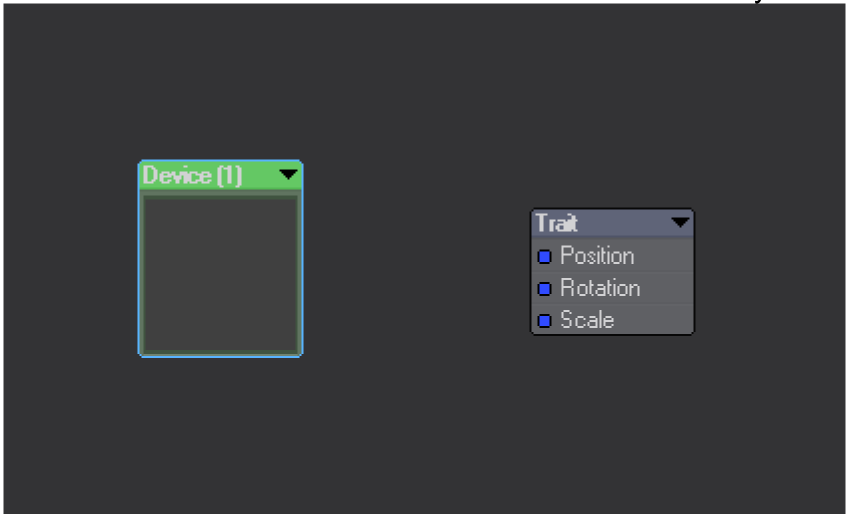
Name	Value	Nodes	🔒	A			Rec	Play
ItemMotion:Camera	X:0 Y:0 Z:0 H:0 P:0 B:0 SX:1 SY:1 SZ:1	Edit...		✓	1		○	▶

This is the Virtual Studio Tools Studio Window and while it doesn't look very interesting, its where a lot of magic happens. From here, we want to double click the "Edit" text under where it says "Nodes". You can actually double click anywhere along that line to continue on. Once you have done that you will be presented with a very dry looking window entitled "Node Editor - Trait:ItemMotion:Camera".



The Node Editor. Don't worry, it's not evil.

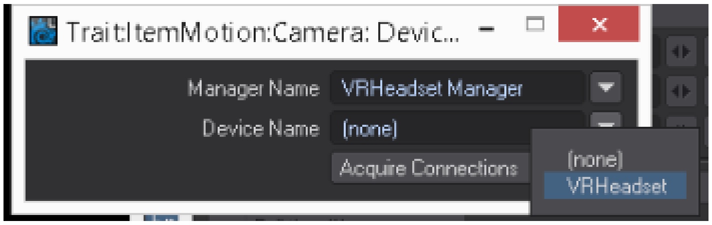
Next, you want to add a "Device" node. You can do this by typing "Device" into the search box and then double clicking it to load it into the editor window or by finding it under the dropdown list "Item Info" and then double clicking the word "Device". This will add it to the into the nodal editor window area. It should like this once you do that.



The "Device" node added to the editor window.

On its own this isn't going to do anything for you and as you can see it doesn't have any "connections" on its right hand side that you can feed to the "Trait" node just yet. First we need to get this device node to be "something". To do that, double click in the gray area inside of the Device Node itself and you will find a new window that pops up called "Trait:ItemMotion:Camera:Device...". This is a Device Manager component. From here we need to make some selections using the drop down arrows for "Manager Name" and "Device Name". Don't worry, this is all standard Virtual Studios Tools stuff that has been around since the release of LightWave 11.0. We didn't do this to make your life complicated. It's just how it needs to be done.

From here we need to get this Device Node to do some work for us. In the dropdown menu for Manager Name, select "VRHeadset Manager" as shown below.

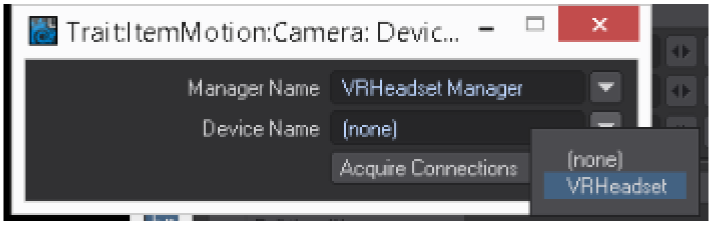

Trait:ItemMotion:Camera: Devic...

Manager Name VRHeadset Manager

Device Name (none)

Select VRHeadset Manager

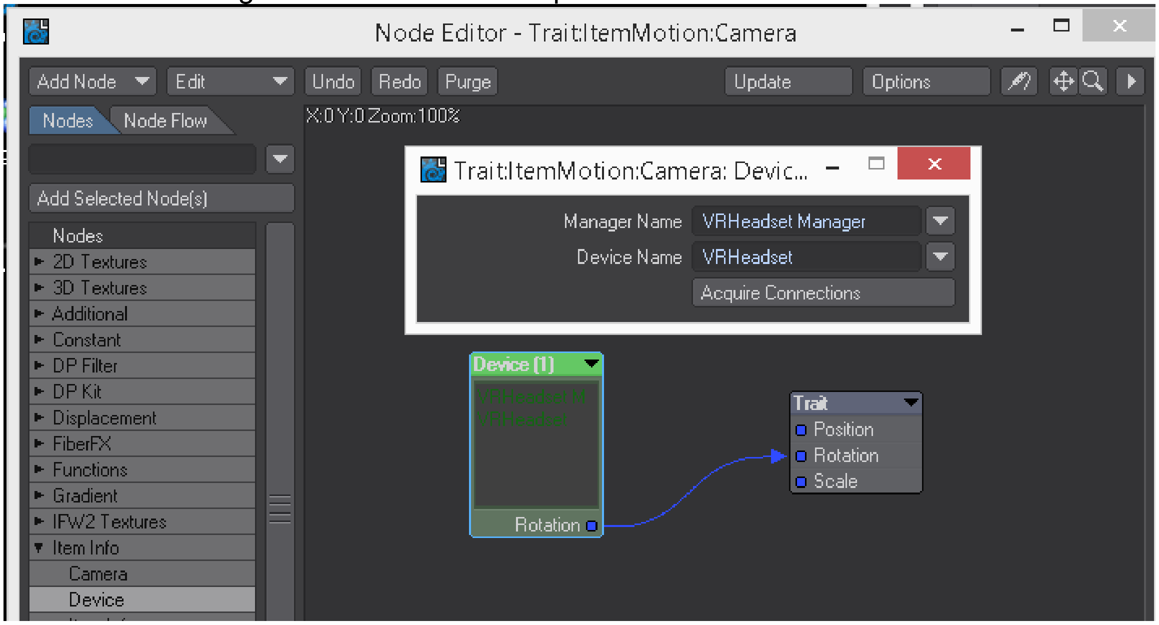
Next we need to do this for "Device Name".



Manager Name VRHeadset Manager

Device Name (none)

When you are done both slots will be occupied much like the example below and you can now make your connection in the node editor taking the Rotation output from the Device and feeding that into the rotation input of the Studio Trait.



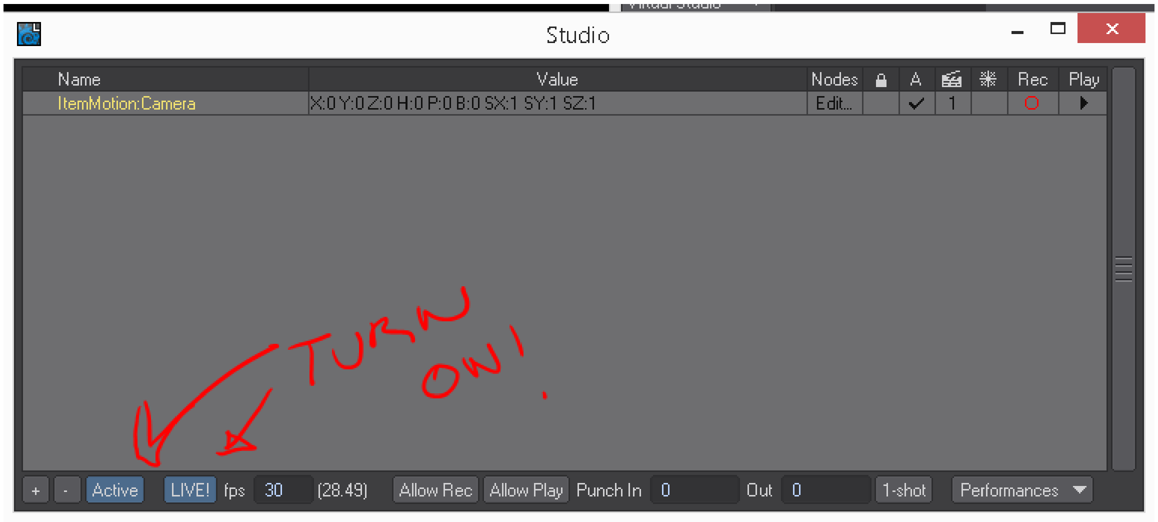
Manager Name VRHeadset Manager

Device Name VRHeadset

Everything is good.

Notice in the example above that the text displayed in the grey area in the "Device (1)" node is green. This is a good thing as it means everything is set up here correctly and you can move on.

Step 3: Making it move.
From here Virtual Studio Tools can be used as normally would be done with any other input device but in our case the Oculus Rift headset is driving the rotation of your camera. If your moving your Rift around and you don't get any movement, make sure the "Active and Live" buttons turned on in the Studio Panel.



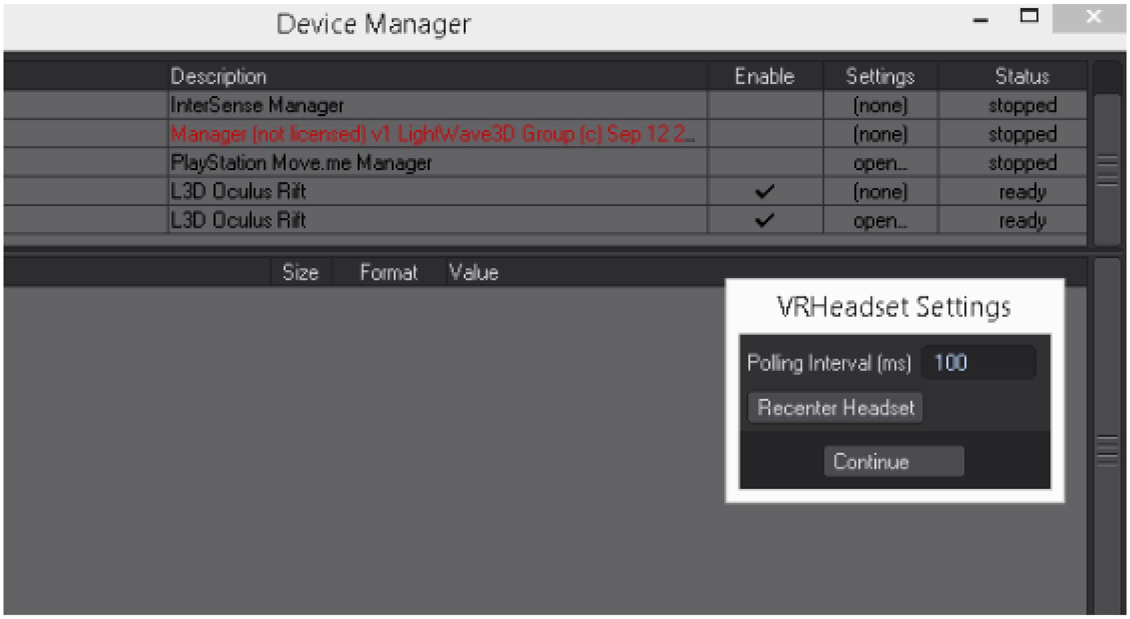
Name	Value	Nodes	🔒	A			Rec	Play
ItemMotion:Camera	X:0 Y:0 Z:0 H:0 P:0 B:0 SX:1 SY:1 SZ:1	Edit...		✓	1		○	▶

fps 30 (28.49) Punch In 0 Out 0

Make sure those buttons are on if you want your headset to rotate the camera when you are moving it.

If you want to turn the Oculus Rift Virtual Studio functionality off for a moment, just disable the "Live" button. Recording takes with your Rift works in exactly the same way as any other input device that works with Virtual Studio Tools. Please refer to the LightWave 11.0 documentation regarding further information regarding Virtual Studio Tools and how the different panels work, recording and playing back takes, etc.

Performance Tuning and Centering:
There is one more small feature to this area that we want you to be aware of and that is the VRHeadset Settings Function. This is found in the Device Manager window for Virtual Studio tools.



Description	Enable	Settings	Status
InterSense Manager		(none)	stopped
Manager (not licensed) v1 LightWave3D Group (c) Sep 12 2...		(none)	stopped
PlayStation Move.me Manager		open...	stopped
L3D Oculus Rift	✓	(none)	ready
L3D Oculus Rift	✓	open...	ready

VRHeadset Settings

Polling Interval (ms) 100

In the image above, next to the "Enable" check boxes where it says "open..." on the bottom line saying "L3D Oculus Rift" you can click that and get a little box. In this box are two options that are very handy. Polling Interval (ms) and a default value of 100 and "Recenter Headset" button.

The Polling Interval deals with how often we ask the Oculus Rift Driver for an update on what it's rotational position values are. The default is 100ms but this is actually quite high. Internally in the Rift, all of its gyros, magnetometers are updating very fast and the Rift is sampling these updates at over 1000 sampling cycles per second. Because we can't synchronize LightWave's Virtual Studio Tools, the Layout OpenGL drawing rate and the Oculus Rift hardware and driver all together, we came up with a way to help users dial things in, although loosely. This value can be reduced to smaller numbers like 30ms or even less for increased responsiveness, but it can also be increased to higher numbers so as to reduce the interruption that we are making to the process which can impact other things such as LightWave's interface menus drawing themselves, OpenGL updates, etc.

We recommend that users play around with this number moving up or down in 10ms steps when they dial in their set ups.

As for the "Recenter Headset Button" this is fairly self explanatory. This button recenters the camera relative to the direction that the headset is pointed. We should point out that you may want to do this from the get go when setting up your scene. Experiment with this function and you will get the hang of how it works in seconds. Also, while this VRHeadset Settings options box is open, Layout stops doing anything until you hit continue. So no, Layout didn't lock up on you and there is no need to swing your Rift all about trying to get it to respond. You have to hit continue in order to get things going again and interactive with Layout's interface.

Once you have your Oculus Rift driving the camera motion, we want to see something through it.

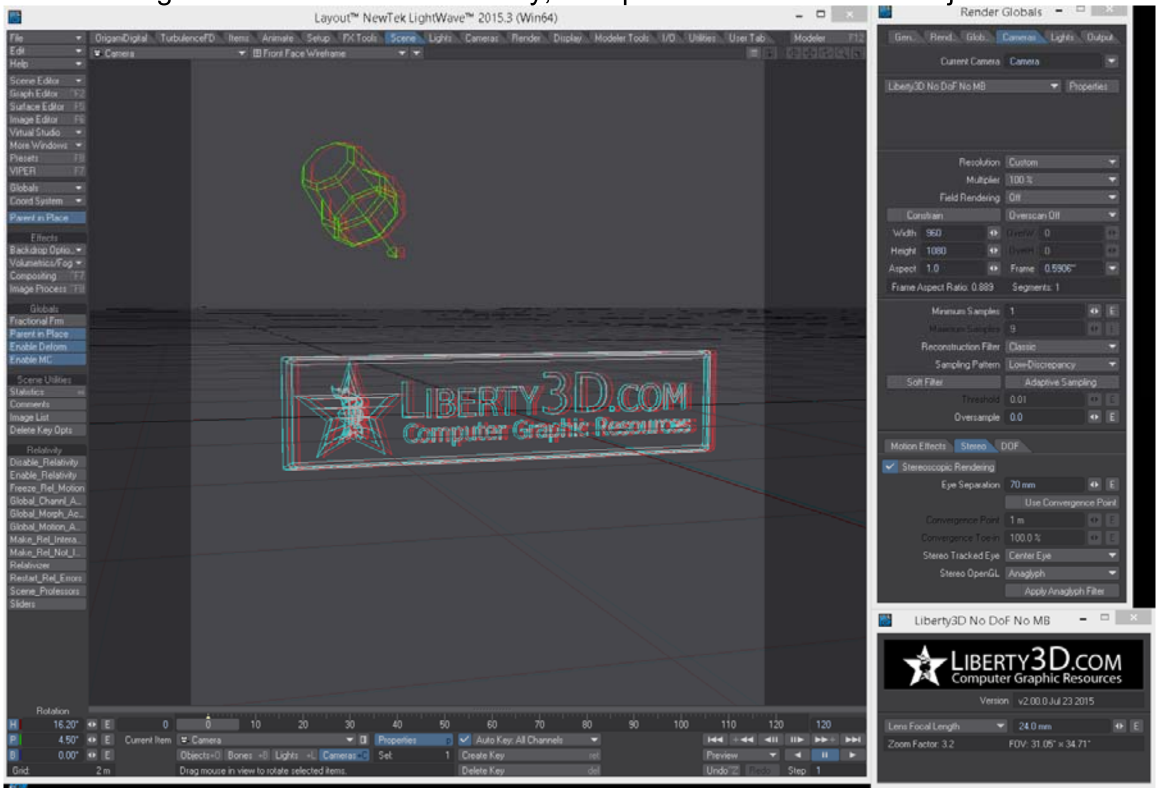
Let's move onto the next step.

Step 4: Configuring your camera

The appropriate camera to use in real-time Oculus Rift set ups for UberCam 2.4 is our own Liberty3D No DoF No MB camera. Alternatively you can use the native perspective camera. However, if you do have the system power available to you to run VPR at the same time, we recommend our No DoF No MB camera because it turns off the checking for motion blur and depth of field. This check is done per pixel in LightWave's native perspective camera (as well as other cameras even if not supported) and this will slow VPR down considerably; something we are already trying to avoid in general.

Again, we don't recommend people try and use their Oculus Rift with the UberRift viewing window and expect to be able to look around in a fluid motion. This is incredibly tasking on even the most powerful systems. It was our intention from the start to make this all work using OpenGL views like bounding box, vertices, wire frame, etc. Still we allow for VPR to be used but please understand what our design parameters were in the first place for this update to UberCam.

With this considered, here in the example below we have our Liberty3D No DoF No MB camera in a scene pointed to an object in layout. Take note of the camera resolution settings of 960x1080. This is the native resolution of the Oculus Rift DK2 VR headset per eye. You can also see that the Stereoscopic Rendering option in the Stereo tab of the camera is enabled. This will by default produce an anaglyph representation of the scene through the camera but don't worry, it stops there. More on that in just a second.


Layout™ NewTek LightWave™ 2015.3 (Win64)
Render Globals
Gen. Rend. Glob. Cameras Lights Output
Motion Effects Stereo DOF
Liberty3D No DoF No MB
LIBERTY3D.COM
Computer Graphic Resources

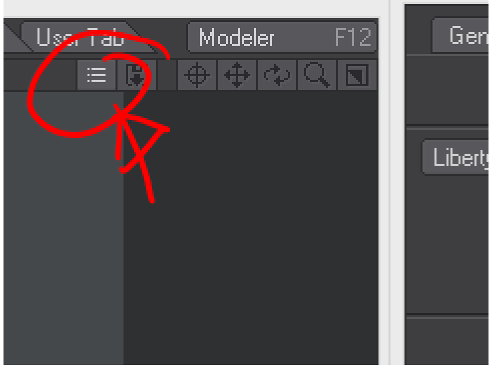
Current Camera: Camera
Liberty3D No DoF No MB
Resolution: Custom
Multiplier: 100 %
Field Rendering: Off
Constrain: Overscan Off
Width: 960
Height: 1080
Aspect: 1.0
Frame: 0.5906"
Frame Aspect Ratio: 0.889
Segments: 1
Minimum Samples: 1
Maximum Samples: 9
Reconstruction Filter: Classic
Sampling Pattern: Low-Discrepancy
Threshold: 0.01
Oversample: 0.0
☑ Stereoscopic Rendering
Eye Separation: 70 mm
Convergence Point: 1 m
Convergence Toe-in: 100.0 %
Stereo Tracked Eye: Center Eye
Stereo OpenGL: Anaglyph
Version: v2.00.0 Jul 23 2015
Lens Focal Length: 24.0 mm
Zoom Factor: 3.2
FOV: 31.05° x 34.71°

For DK 1.1 set ups, the recommended resolution, matching that of the DK1.1 display is 640x800.

Now you are probably wondering why not full HD at 1920x1080 for use with the DK2 or 1280x800 for the DK1.1 unit? Well in short this is because the width is representative of each eye in stereoscopic inside the displays for each developer kit. Multiply the width of each eye by two and you get the full resolution of the display for each unit. Simple. **Additionally you will need to set the FOV of the camera to match that of your headset. This is important!**

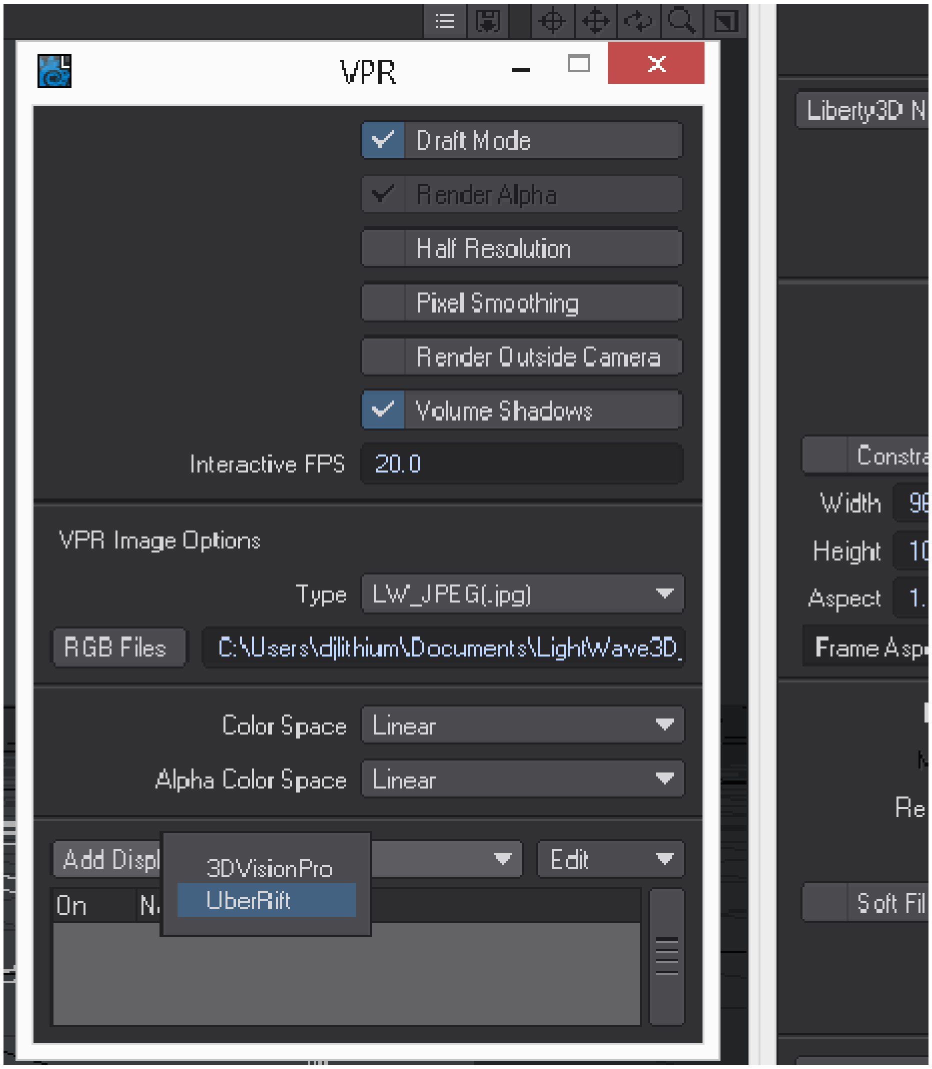
While we are on the subject of stereoscopic, we should point out that Liberty3d.com was instrumental in advancing LightWave's stereoscopic capabilities during the 11.6.3 cycle. It is because of us that you have more than just anaglyph stereoscopic rendering available to you in Layout. Today in 11.6.3 and above you now get additional options found in the stereo tab under the camera panel including convergence point (taken from our Stereoscopic camera) the stereo tracked eye option and different stereoscopic OpenGL modes. Under the Stereo OpenGL dropdown box where it currently says Anaglyph you can set this to be "3D Glasses". If you choose this option, that ugly anaglyph openGL representation goes away, but under the hood, Layout is still making the OpenGL camera view port data available in stereo to quadbuffered stereoscopic enabled video cards such as AMD's FirePro line or nVidia's Quadro line of products. *IF* you don't have one of these types of cards, don't panic. We take advantage of this through our UberRift viewer so you can see the stereoscopic effect in that window which can then be sent to the Oculus Rift.

To do this, we need to access the UberRift viewer window. You can find it by clicking on the little set of 3 lines which happens to be VPR's options panel access button.



Access the VPR options window buy clicking that little button.

From here you will be presented with the VPR options panel window. No we are not going to mess with VPR here, instead we are going to add a our UberRift Viewer to the "Add Display" drop down menu.



☑ Draft Mode
☑ Render Alpha
□ Half Resolution
□ Pixel Smoothing
□ Render Outside Camera
☑ Volume Shadows
Interactive FPS 20.0

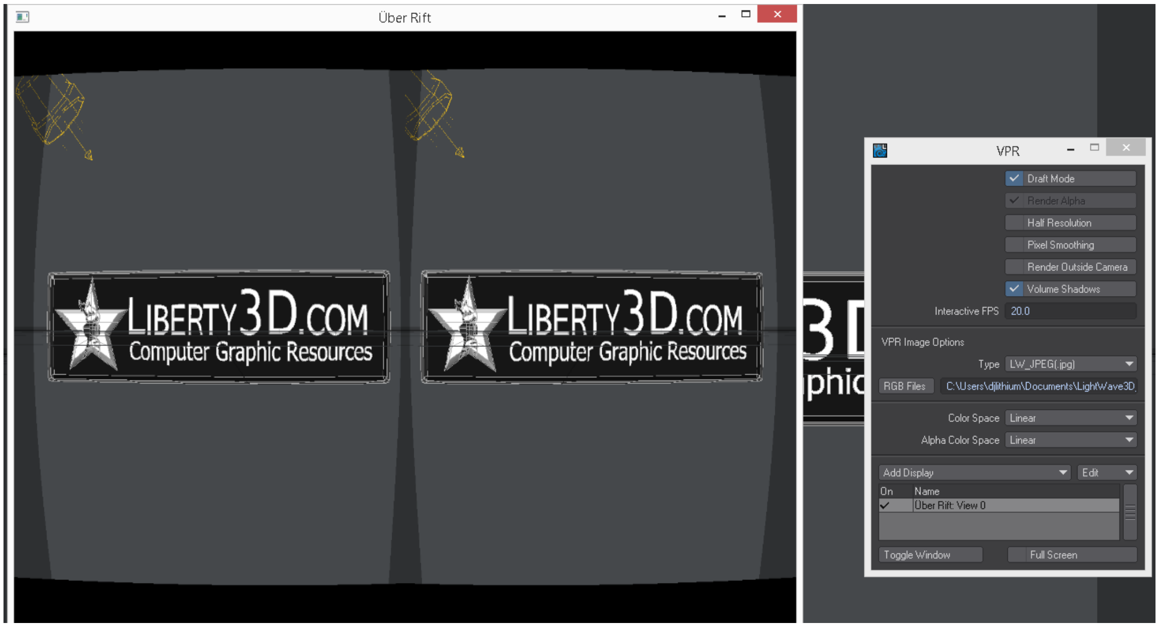
VPR Image Options
Type LW_JPEG(.jpg)
RGB Files C:\Users\djlithium\Documents\LightWave3D_

Color Space Linear
Alpha Color Space Linear

Width 98
Height 10
Aspect 1.

Click "Add Display" and add "UberRift".

Once you have added the plug-in to this list, double click it so that the "Toggle Window" and "Full Screen" buttons appear as shown below.



VPR

☑ Draft Mode
☑ Render Alpha
□ Half Resolution
□ Pixel Smoothing
□ Render Outside Camera
☑ Volume Shadows

Interactive FPS 20.0

VPR Image Options

Type LW_JPEG(.jpg)

RGB Files C:\Users\djlithium\Documents\LightWave3D

Color Space Linear

Alpha Color Space Linear

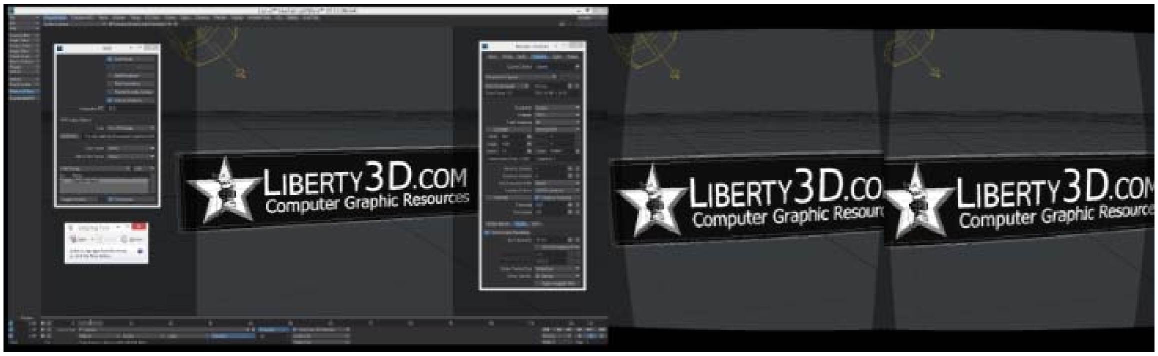
Add Display | Edit

On	Name
✓	Über Rift: View 0

Toggle Window | Full Screen

Here is a screen grab that show the UberRift viewer open and the options available to you in the VPR options panel window.

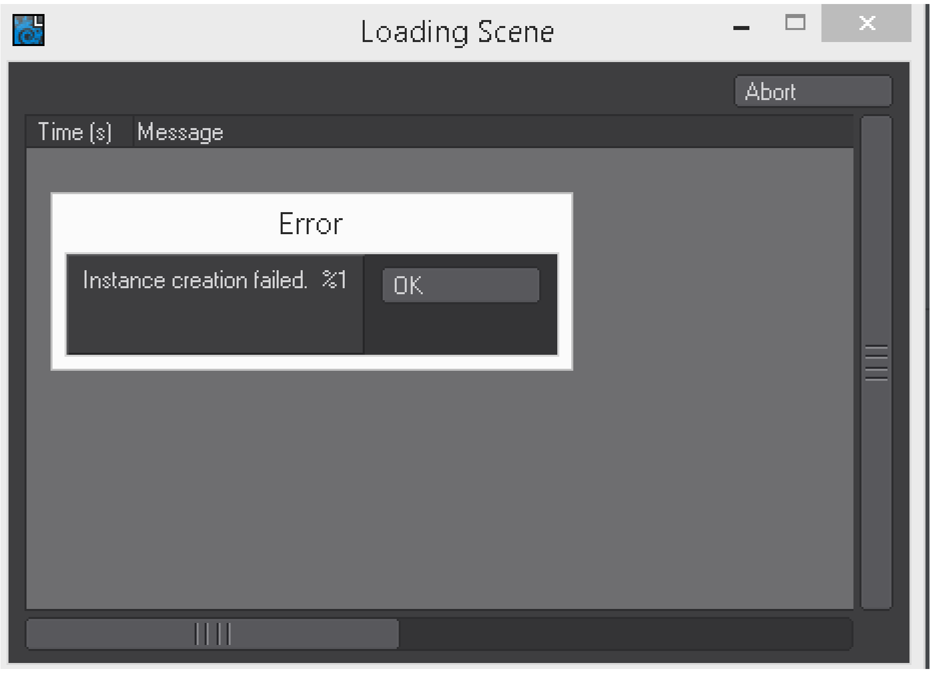
The UberRift viewer will appear and you can now drag it off to the side display where you have your Oculus Rift connected. You don't have to get it all the way onto the screen, just more than half. From here you can hit "Full Screen" on the VPR options window and it will maximize itself to fit the display, in this case the Oculus Rift.
If we could see both displays at once, they would look something like this in the image below, with Layout on the left and the Oculus Rift display being feed the layout camera viewport, in stereoscopic no less with the proper barrel distortion applied to each of the eye views, on the right. Notice how both views lack that ugly anaglyph filter but the Oculus Rift display view is in stereo? Cool huh? You betcha it's cool!



Yeah. It's cool.

At anytime you can use either the toggle window (for when you want to move around your scene quickly repositioning the camera) or the full screen button so you can move the display around. To disable the viewer entirely you can either remove the plug-in from the drop down list in the VPR options panel window or uncheck it where it says "On" next to the name "UberRift: View 0".

A couple of small notes before we move on. First, yes - it is possible to have more than one UberRift Viewer active at one time. While this would be incredibly tasking on a system, we do allow for it. Even though not many people have more than one Oculus Rift available to them but we didn't want to limit people's options and this was done to make certain things in the future possible. Be advised that only one Rift can be used to drive the selected camera at a time. Second, if you save your scene in with the UberRift viewer loaded in the VPR options panel and come back to it later, you will get a harmless error that pops up on scene load. In order to get the viewer to work again you will have to re-add the plug-in to the VPR options panel like before. This error message isn't very descriptive and simply says "Instance creation failed. %1" with an ok button next to it.



Error

Instance creation failed. %1

A somewhat harmless error when loading a scene that has the UberRift Viewer active when save and then reloaded. Just hit ok and your scene will load.

Normally just hit ok and your scene will load just fine.

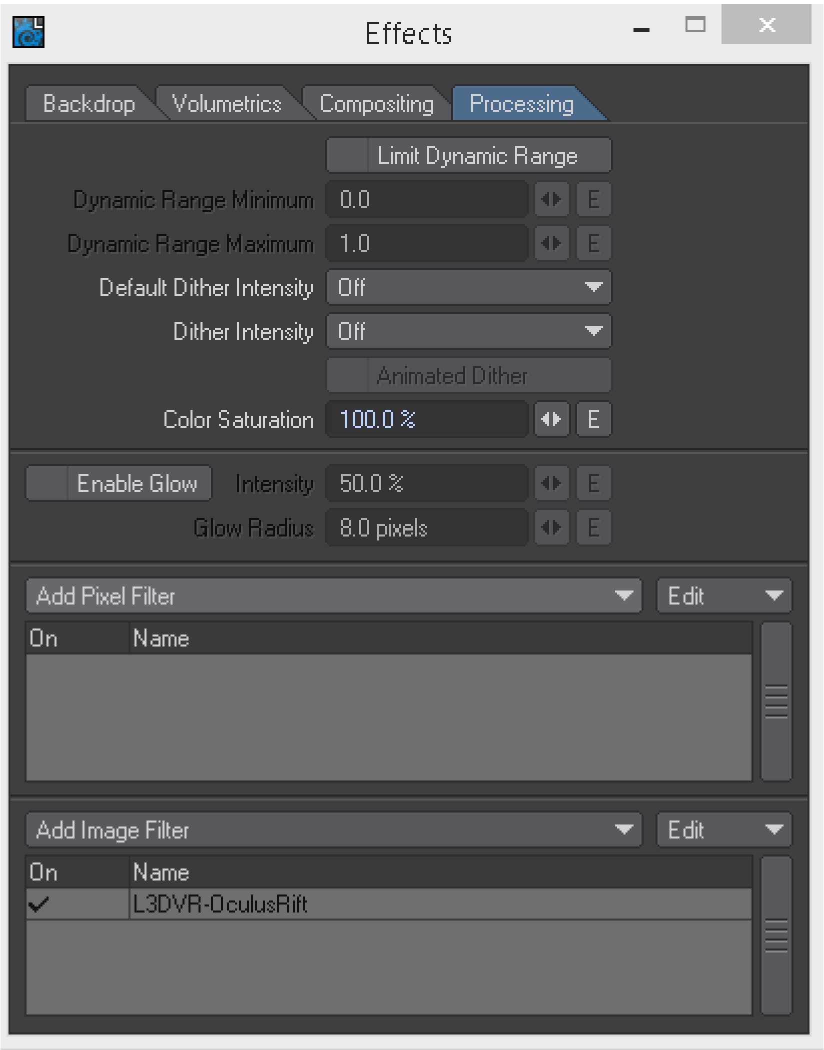
HOWEVER, we recommend that if you ever want to render this scene over the network using LWSN that you remove the UberRift Viewer before you save the scene and send it to your render farm. Otherwise it will stop the rendering process on this error, and it will sit there waiting for someone to hit ok on the button which can't be seen by anyone - especially your render controller software.

You have been warned.

Chapter 6: The L3D Oculus Rift Barrel Distortion Image Filter

The L3D Oculus Rift Barrel Distortion Image Filter Plug-in is provided as another simple tool for those who want to make "lean back" stereoscopic content for viewing with the Oculus Rift of to check how something will look in all its fully rendered glory using LightWave3D's rendering engine.

The L3D Oculus Rift Barrel Distortion Image Filter Plug-in can be accessed through the image processing tab or by hitting Cntrl+F8 (or Command+F8 on the Mac) under the "Add Image Filter" dropdown menu box. In the list of plugins in that area the L3D Oculus Rift Barrel Distortion Image Filter Plug-in is called "L3DVR-OculusRift".
Simply select it. It doesn't have any options that you can change so don't worry about that part.



Dynamic Range Minimum 0.0
Dynamic Range Maximum 1.0
Default Dither Intensity Off
Dither Intensity Off
Color Saturation 100.0 %
Intensity 50.0 %
Glow Radius 8.0 pixels

On	Name

On	Name
✓	L3DVR-OculusRift

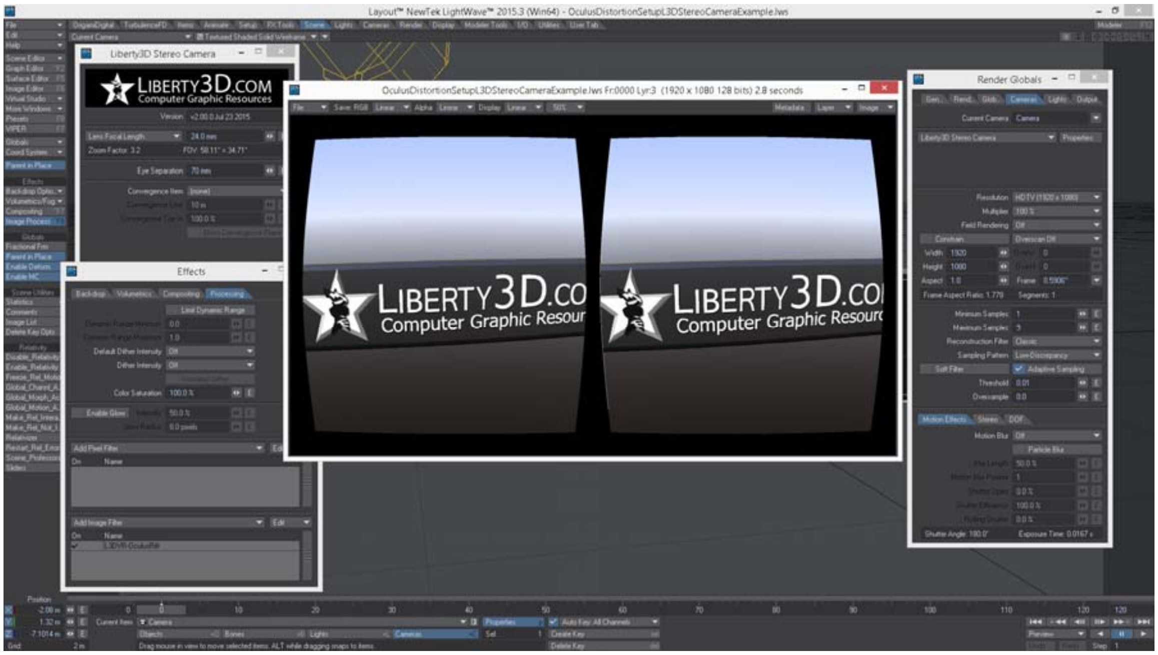
L3D Oculus Rift Barrel Distortion Image Filter Plug-in is called "L3DVR-OculusRift" applied to the Effects processing tab.

As we mentioned earlier in this document, the L3D Oculus Rift Barrel Distortion Filter is ONLY to be used with our Liberty3D Stereoscopic camera. Use with other cameras and the native stereoscopic functions in LightWave will produce a crash. If you enable "stereoscopic" under the L3D Stereoscopic camera tab this will also produce a crash. So don't use the native stereoscopic options here.

To set up the Liberty3D Stereoscopic camera, ready for producing a proper image result we will use a normal resolution such as 1920x1080 in the case of the DK2. In general we recommend this resolution anyway even if you are looking at it on a DK1.1 because the display with down res the image to fit.

The L3D Oculus Rift Barrel Distortion Filter works on normal width images in this way because if you think about it 1920x1080 is the value of the right and left eye (960x1080 each) combined side by side.

While it isn't required, you can make use of the convergence point function in ***our*** Stereoscopic Camera options panel. Once you have things set up you can simply hit render and once the render is complete you can save the image out or render a full animation and then play it back on the Rift through various commercial and free players. You can also maximize your image viewer from layout on the rift display.


Layout™ NewTek LightWave™ 2015.3 (Win64) - OculusDistortionSetupL3DStereoCameraExample.lws

Liberty3D Stereo Camera
Version: v2.00.0 Jul 23 2015
Lens Focal Length: 24.0 mm
Zoom Factor: 3.2
FOV: 58.11° × 34.71°
Eye Separation: 70 mm
Convergence Item: (none)
Convergence Unit: 10 m
Convergence Toe In: 100.0 %

OculusDistortionSetupL3DStereoCameraExample.lws

Effects
Dynamic Range Minimum: 0.0
Dynamic Range Maximum: 1.0
Default Dither Intensity: Off
Dither Intensity: Off
Color Saturation: 100.0 %
Intensity: 50.0 %
Glow Radius: 0.0 pixels
Add Image Filter
On ✓ Name: L3DVR-OculusRift

Render Globals
Current Camera: Camera
Liberty3D Stereo Camera
Resolution: HDTV (1920 x 1080)
Multiplier: 100 %
Field Rendering: Off
Overscan Off
Width: 1920
Height: 1080
Aspect: 1.0
Frame: 0.5906"
Frame Aspect Ratio: 1.778
Segments: 1
Minimum Samples: 1
Maximum Samples: 3
Reconstruction Filter: Classic
Sampling Pattern: Low-Discrepancy
☑ Adaptive Sampling
Threshold: 0.01
Oversample: 0.0
Motion Blur: Off
Blur Length: 50.0 %
Motion Blur Passes: 1
Shutter Open: 0.0 %
Shutter Efficiency: 100.0 %
Rolling Shutter: 0.0 %
Shutter Angle: 180.0°
Exposure Time: 0.0167 s

Current Item: Camera
Drag mouse in view to move selected items. ALT while dragging snaps to items.

Here is an example of the settings normally used and the render result.

F9, F10 and LWSN renders using the image filter will produce images as you see above.

Chapter 7 Troubleshooting

The following section covers common questions about Liberty3D UberCam for LightWave. The section also covers a list of known issues. If you have any questions, suggestions or comments about Liberty3D UberCam for LightWave, e-mail kat@liberty3d.com

Common Questions

The following table contains answers to some common questions about Liberty3D UberCam for LightWave.

Question Answer

Are volume purchases available?

Yes, contact kat@liberty3d.com for more information.

"Are any other cameras being planned?"

Yup but we can't tell you which just yet. All suggestions are welcome, and will be evaluated based on viability and usefulness. We have several new cameras that we want to build and are working on them based on our own production needs and priorities but we are always willing to hear your thoughts and comments. Please send them to us!

"Will you be adding support for more VR and AR Headsets and if so, when?"

The answer to this is absolutely yes! The answer to when is a bit more complicated. This is largely due to supply of units from manufacturers and shipping times if and when we get a slot during pre-order phases which seems to be the popular approach to selling headsets these days. It also takes time to build and test this stuff.
We have plans to enhance support for the Oculus Rift immediately now that the CV1 edition is shipping and add support for the HTC Vive as soon as we can get our hands on one. We are at this time in line.

On the AR side we are interested in and are working on a number of devices but we can't say which just yet although you can probably guess. We can't promise when we may provide users of UberCam with support for these devices. So put it out of your mind for now and when we drop the hammer it will be a pleasant surprise.

Known Issues

"Lens Flares do not work in multi-up cameras correctly. Why?"

Lens flares and other effects which are the result of a post-render compositing step will not work correctly in a multi-up camera, such as the Stereo Camera. This is because they assume that a linear camera transform is being used in order to determine the location of the lens flare light effect. The current workaround is to use comparable volumetric light effects in order to obtain the desired visual effect. A more efficient solution is being examined for Liberty3D UberCam in the future.

"Why doesn't my FiberFX set up appear where it is supposed to in the image?"
If you are using the post render image filters or pixel filters for FiberFX, you are running into the same problem as we described with lens flares. You will need to use normal (brute force) rendering options for FiberFX to make it work with any of our non-linear cameras. Remember, this isn't a limitation of our Cameras, it's a limitation of LightWave's post processing image and pixel filters. Hopefully in future updates we wil be able to see the LightWave Group update these areas of LightWave and that will solve these issues once and for all.

"Cameras parented to Other objects or items may not perform correctly with motion blur. Why?"
Some beta testers have reported to us that when certain cameras are parented to other objects or items in the scene, that the camera doesn’t perform exactly as expected. One known instance of this is the stereoscopic convergence plane in the StereoCam Camera. If this camera is parented to an object while the convergence plane is being previewed, it will generate the plane at the world axis point instead of in front of the camera at the appropriate convergence distance set in the properties panel for the StereoCam Camera. This issue and others may take place when parenting UberCam cameras to other items in the scene. We are looking into what causes this and will update the package immediately once a solution is found.

"Overscan camera with MB and jitter shows horizontal tearing on multi-core machines or with Multi-Threading Enabled in the render globals panel. How do I fix this?"
Some beta testers have reported to us that when the overscan camera is used to render a motion blurred scene rendered with a multi-core system using multiple threads, there are horizontal tearing lines in the final image. This can happen when certain motion modifiers or textured motion tools are applied to the jitter envelope itself in the graph editor. Some of these tools such as Noisy Channel are not able to work with the Jitter function without first baking the key frames on the envelope that Noisy Channel creates. Baking the key frames of the envelope will eliminate any issues with shredding of the image during render while using multi-threaded rendering. We recommend you do this anyway if you are going to render your shot over a network as some machines may be different processor types and translate the randomness different that others, giving you undesirable results - however in our testing on mixed machine networks at Declaration Pictures in Vancouver and in use on several productions by our users and beta testers around the world this problem has rarely popped up.

A final note:
Thank you again for purchasing Liberty3D UberCam 2.4 and supporting our development efforts! We appreciate your support and welcome your ideas on new tools and products. Suggest them by visiting our site at www.liberty3d.com and signing up to our forums or send us an email! Make sure to check out our other great tools like QuadPanels and Weighter2 for LightWave Modeler. More tools and video tutorials are always being released and we look forward to seeing your images made with LightWave3D and our products. You can submit them to the gallery area in our forums at Liberty3d.com

Liberty3D UberCam

For NewTek’s LightWave3D

Version 2.4