2016

Designing Accessible Pipelines for Visual Effects and Virtual Reality

Turner Jordan Sinopoli
Designing Accessible Pipelines for Visual Effects and Virtual Reality

by

Turner Sinopoli

A Thesis submitted to the College of Motion Picture Arts
Honours in the Major
Degree Awarded:
Spring 2016
The members of the defense committee approve of
Turner Sinopoli’s Thesis

Defended April 15th, 2016

Ron Honn
College of Motion Picture Arts

Andrew Syder
College of Motion Picture Arts

Keith Roberson
College of Fine Arts
Acknowledgements

The VR Research team at FSU’s College of Motion Picture Arts for their massive brains, hearts, and talent

Ron Honn
Jonathan Stone
Geoff Stegmeyer
Austin Baur
Frank Patterson

The Cast and Crew of the first official FSU VR Test Shoot

Victoria Cragg
Jamie Cernich
Daniel Smith
Victor Ingles
Gabe Souza
Jon Lora
Geri Ingall
Cole Goslee
Amelia Hazlitt
Evan Barber

My Honours in the Major Committee Members

Andrew Syder
Keith Roberson
Ron Honn

Amanda Nearbin and Alexander Russ for proofreading more pages than it is decent to ask for

Mari Kyle for inspiration, proofreading, and formatting

My family for telling me what I do is cool
Table of Contents

Abstract.................................................................................................................4

Deep Compositing.................................................................................................6
  History
  Deep in the Industry
  Advantages
  Disadvantages
  Examples

Deep Production Process.....................................................................................12

VR Production Pipeline.........................................................................................23
  Pre Production
  Production
  Post Production
  Visual Effects
  Distribution

Bibliography..........................................................................................................43
Abstract

The film industry is a constantly changing beast. As with anything tied to technology, trends rise and fall as new tools and techniques are developed and then improved upon. In this paper I have researched and designed processes revolving around two emerging trends in the film industry which have previously been inaccessible to artists at the collegiate or amateur level. I present two workflows that are designed around tools that are highly accessible, adaptable to many different levels of resources, and created with an aim towards adapting existing workflows that may be implemented at smaller universities and colleges.

The first technology I discuss is Deep Compositing (Deep). This is a type of data being utilized by modern Visual Effects (VFX) facilities that uses Deep Data to make changes to rendered images utilizing integrated depth information. This allows for physically accurate manipulation of volumetric effects (e.g. fog) and simulation of real world photographic effects (e.g. depth of field). It also decreases render times by making RGB and baked holdout maps obsolete and allowing for scene-specific changes to be rendered separate from the entire shot. I discuss the history of Deep, the advantages and disadvantages that come with it, its uses in the industry today, practical examples for when to utilize Deep, and conclude with a tutorial that will walk through the complete Deep Compositing process utilizing Autodesk Maya, Pixar’s Renderman, and The Foundry’s Nuke. All of these softwares are available for free in noncommercial versions.

The second process I discuss is the emerging field of Virtual Reality Cinema. Often is the word “revolutionary” thrown around, but rarely does it hold its meaning. However, we are today finding ourselves at a precipice that will change the future of the industry forever. This revolution is centered around the technology of Virtual Reality (VR), which has disrupted industry workflows and has inspired an entirely new form of art: Cinematic VR. This is to say, storytelling in a completely immersive world. Before recent changes, VR equipment was at too high a price point to be accessible for the masses. Thanks to efforts by companies such as Google, Oculus, the University of Southern California, and Samsung we are now at a point where almost anyone can create their own VR content. In this paper I present a guide for live action VR content creation, with a specific focus on post-production and visual effects workflows. I cover everything from pre-production to distribution, while also calling attention to where the VR
process deviates from standard motion picture production. I also pose questions that are valuable for filmmakers to consider when creating projects in this exciting new medium for storytelling. I avoid discussing processes or softwares that are prohibitively expensive or inaccessible to new students. I have also integrated this pipeline with tools that are standard across university Film School pipelines with the ultimate goal of accessibility.

Deep Compositing
What is Deep Compositing?

Deep Compositing is the process of integrating *deep image data* from CGI renders to create a more flexible and physically accurate method of manipulating existing image data. Deep Compositing won a technical achievement award at the SciTech Oscars in 2014. It is an industry standard methodology and is inseparably integrated into the VFX pipelines at almost every major studio in the world.

A Brief History of Deep Compositing

In 2000, Pixar researchers Tom Lokovic and Eric Veach published a SIGGRAPH technical paper titled “Deep Shadow Maps” which describes a method of casting shadows for objects such as hair and clouds which, at the time, was an extremely costly process. Their new process rendered these shadows with greater detail and a smaller memory footprint than ever before. Traditionally, each pixel in a rendered image is sampled one time. With Lokovic and Veach’s method, each pixel is sampled multiple times and is granted an XYZ coordinate for its actual location in the scene. Each pixel is also assigned a *visibility function* which records the semi-transparent objects or effects that might be occluding a certain area of a scene. See the image below for a visual example of this process.

This new method of Deep Shadows was then implemented into Pixar’s rendering engine, Renderman, and used as the primary method of shadow creation. It outputs these *deep shadow maps* into a proprietary file format known as a .dtex.

This paper existed for years without significant impact until it fell into the hands of an artist at New Zealand VFX studio WETA Digital named Colin Doncaster, who at the time was working on the remake of the film *The Day the Earth Stood Still*. Colin was working on a shot where a cloud dissolves a moving semi-truck when he thought of an implementation for this Deep Shadow Technique that Pixar had developed that would allow him to have more control of his hold out maps and prevent him from having to rerender his whole scene every time a change was made to a single element. His research project was approved and he successfully built an implementation of Deep Data that could be used to aid live-action compositing. This was so successful that WETA began integrating Deep into their main production pipeline and thus into their future projects, including the wildly successful
film *Avatar* by James Cameron which was a huge step towards the industry’s eventual adoption of Deep.

At the same time this was happening at WETA, an artist named Johannes Sam was making similar developments utilizing Deep for compositing at the Australian animation studio Animal Logic. Johannes was compositing shots in a software named Nuke when he decided that he would design a way to implement this deep data into the industry standard compositing platform. Johannes designed the first deep nodes for Nuke and it was used to great effect in the film *Legend of the Guardians: The Owls of Ga’Hoole*, an animated feature film that was praised for its stunning visuals. The work of integrating Deep Data into such a widely used software was instrumental in making Deep Compositing available to more studios around the world. Johannes then began publishing videos online of Deep Compositing and the advantages that it could bring to the visual effects pipeline, creating even more buzz for this technology.

Not to be forgotten, American artists at Dreamworks Animation were struggling with integrating their characters into volumetric clouds of dust and dirt for their film *Monsters Vs. Aliens*. Artists were dissatisfied with the levels of photorealism that they were able to achieve with traditional compositing techniques. Chris Cooper then began utilizing Deep Data to design a pipeline that allowed geometry to be rendered on its own to later be integrated separately from the volumetric renders in a more physically accurate way.

These men, Colin, Johannes, and Chris, along with Daniel Heckenberg (another artist at Animal Logic) realized they were all developing the same technology and decided that they should be collaborating together to inform the world about this technique that was revolutionizing the VFX pipeline at their respective studios. They wrote a paper to submit to SIGGRAPH describing the process of using Deep Data for compositing. It was promptly rejected on the grounds that it was too similar to the paper Pixar published in 2000 and was deemed not an important development.

Not to be deterred by this, the men continued to develop their tools and techniques all the while releasing videos and maintaining a website with the slogan “No more edge artifacts!” detailing the exciting possibilities that Deep Compositing provides.

WETA continued their development of Deep and in 2011 released a video detailing how they implemented it in the film *Rise of the Planet of the Apes*. This video went viral in the VFX community and then The Foundry officially implemented Deep nodes in their release of Nuke 6.3. This decision
finally cemented the role that Deep Data could play, but it was still contained to Pixar’s proprietary .dtx format which was restricting the access that developers had to the technology.

Deep lost this final barrier when it was announced that OpenEXR, the industry standard image file format, would integrate support of Deep Data in their release of OpenEXR 2.0. This meant that all renderers could now export Deep Data in a format that was not specific to any company and could be freely manipulated by anyone.

The impact of Deep Compositing was finally acknowledged in 2014 when Colin Doncaster, Johannes Sam, Areito Echevarria, Janne Kontkanen, and Chris Collins won a Technical Achievement Award for their early development work on Deep Compositing, now an industry standard practice, at the Academy Scientific and Technical Awards.

**How Does Deep Work?**

Deep Data is generated by casting a single ray from each light source in a scene. This ray is used to calculate a *visibility function* for each pixel while the standard camera rays sample the depth values for each pixel being rendered. Every surface or volume sample making up the final pixel is stored with its colour and opacity as well as its distance from the camera. When putting these together you have *deep image data* which can be used to effectively recreate a 3D scene after it has been fully rendered.

Before Deep Data, scene depth was captured in a “Z-depth Pass” which took the physical depth of a scene and translated it into a 2D black-and-white image gradient, where white represented close to camera and black represented far from camera. This was devised as a method of capturing scene depth while keeping a low memory footprint. It was a bit of a hack to get this image to work and it contributed to many instances of edge artifacting and artistic interpretation of real world phenomena.
Advantages

Deep Data provides a method of reproducing physically accurate optical effects while maintaining control over every pixel of a render after it has been created. Artists can now create effects such as lens depth of field and bokeh without taking artistic liberties or dealing with the issue of edge artifacting.

When creating volumetric effects such as dust or fog, artists can now render their characters and objects separately from the volumetrics and then place those renders inside whatever phenomenon they make. For example, if an artist wants to place a character inside a meadow that is engulfed in smoke they can render the character and the smoke on separate passes that can then be combined inside of Nuke. If the artist then wants to make a change and push the character deeper into the smoke, all they have to do is utilize the deep position data to translate the character on its Z-Axis. Nuke will read the position data of the character and then use the visibility function generated from the fog to accurately occlude the character and make it harder to see as it is pushed backwards.

In the past, if an artist wanted to make a change like this it would require them to rerender the entire scene, which could take hours or even
days. With Deep Data they can now make the change and see the results in seconds. This reduces render times and costs, but also allows for a greater degree of control as well as creativity and iteration.

Utilizing this data, artists can also take a 2D image render and convert it to a stereoscopic image without having to render a separate image for left and right eyes, effectively cutting the render time for a 3D scene in half.

And, as it was originally designed, Deep Data also allows for much faster processing of shadows on objects such as clouds and hair.

The most important impact of this technique is that these optical effects are physically accurate. A compositor’s job is to recreate reality and now that they have access to the tools Deep Data provides they can create images to levels of visual fidelity that could never be achieved before.

**Disadvantages**

Deep Compositing is still a growing field and is not perfect in its design or its implementation. Chief among the issues of Deep Compositing is that deep image data is very data-heavy and requires much storage space. Utilizing Deep Data increases the memory footprint of a rendered image by 50% in most cases. If a final render of a 4K shot is 80 Gb, then you will also have to manage the 40 Gb of Deep Data that goes along with it. And because this data is generated early in the process, that data must continue to be lugged down the rest of the pipeline until final delivery. Most smaller studios are unable to manage the I/O bandwidth required for Deep Data and thus have been slow to adopt the process.

Because of these file sizes, when working with Deep Data it slows down the entire production process while working on a shot. Everytime Nuke taps into the disk to read an image file it also has to read the huge amount of Deep Data that goes with it before it can be processed. The standard industry process is to get the deep data into Nuke, make any changes needed, cache the changes, and then get it out of Deep as soon as possible.

Deep is also not fully supported by all of the major software applications used by studios. Most notably, Adobe After Effects cannot be used with Deep as it does not provide full 3D capabilities to the user.

This goes hand in hand with the fact that Deep Compositing is still an emerging technology. While it has been adopted by most studios, it is still being researched and the full implications of this technology is not fully known. Working with Deep comes with the caveat that the technology and methods can change any time as it is developed.
Uses in the Industry

As mentioned previously, Nuke has been adopted by almost every major VFX studio, but the best examples of the advantages that Deep provides are being created by the studios that helped spawn the technology in the first place: Animal Logic and WETA Digital.

Animal Logic’s most recent work was on the wildly successful film, *The LEGO Movie*. This film was entirely digitally created, yet was rendered with such skill that many people thought the film was actually stop-motion LEGO bricks. Much of this visual fidelity can be directly attributed to Animal Logic’s heavy use of Deep Data. For example, the film is made to look as if it was shot on anamorphic macro lenses, which have a very distinct depth of field and bokeh shape. Before Deep, these effects would have had to be created by hand and would vary in consistency and quality throughout the film. Using Deep Data, artists could plug in the real world lens data (e.g. focal length, aperture, glass shape) and receive a physically accurate representation of how that lens would distort and defocus if it were in that world.

WETA Digital has integrated Deep so far into their proprietary rendering software that in an interview with FXGuide in 2012, software developer Dr. Peter Hillman at WETA said, “I don’t know how to turn off Deep Data in the pipeline. I am not sure we have a button to turn it off. You can throw away the data if you don't want it, but it is just so useful.” [FXGuide, 2014]

When working on *Abraham Lincoln: Vampire Hunter*, WETA did not have to do any volumetric lighting inside of their renderer. If they wanted “God-Rays” or shafts of light through the dust kicked up by a stampede of horses they could add directional lighting into their volume of Deep Data inside Nuke and achieve accurate volumetric lighting in compositing. This offers immeasurable time saved during the rendering and iteration process.

Studios have also begun building out their digital asset libraries to include Deep Data. Now if an artist wants to add a puff of smoke into their composite they are no longer limited to the options of over or under; they have complete control to place their assets anywhere they want into and around objects in 3D space.
Deep Compositing Tutorial

I will now walk through the process of creating deep images and using them in production. This process uses only tools that are freely available to any person that seeks access and is a method that can be implemented easily into existing pipelines.

This tutorial assumes basic knowledge of 3D tools and rendering setups.

Software Used:
Autodesk Maya
Pixar’s Renderman
The Foundry’s Nuke

Creating our Scene

For this tutorial we will be creating a very simple scene that will showcase how Deep Data can be used to avoid rerenders and control your renders after they are made.

We will start by creating a plane and then populating that plane with a forest of cylinders of various sizes. In the middle of this forest create a sphere that is not touching any of the cylinders.

Here is our current scene:
Setting up Render Passes

If you recall from our history lesson, Renderman produces Deep data by default in order to accelerate the process of rendering shadows. What we need to do is force Renderman to keep that data and then output it into a format that is usable.

So when you go into the output options you must create a new custom channel named “color Ci”, right-click it and select “Create new Output from pass”.

Now Renderman will output color Ci onto its own image sequence, but we need to tell it this is where to keep the Deep Data. So we must switch the “Image Format” to DeepImage, which is a .dtex file. Inside of this output we also need to add in the “Optional Attribute” Deep Image Subimage.

At this point Renderman now knows that it must create a separate .dtex image sequence. It also knows that it should keep the Deep Data and output it into that .dtex image sequence and because we added the Deep Image Subimage it also knows to record the opacity and colour that is associated with the depth samples it is also recording.
Now you can hit batch render and see your results!

**Reading Deep Data Into Nuke**

Because Renderman was the first rendering engine to support Deep Data, Nuke relies on it to do the heavy lifting and interpolate the deep data before it can be used for compositing.

By default Nuke has the “prmanRender” node, but it needs to be setup by changing Nuke’s Environment Variable to direct the Operating System(OS) to the Renderman libraries. This process varies based on OS but the instructions can be found on the Foundry’s website and inside the Nuke Documentation.

After the prmanRender node is set up and working, you can now read in Deep Data with Nuke!

Step one is to import all of your standard RGBA passes and then you can create a “DeepRead” node and direct it to the .d tex files that we created. Viewing this node should appear to be a black and white version of your beauty pass.

*Note: Nuke designs its nodes to be distinct shapes based on the function they serve. All Deep nodes are deep blue and have a rounded right side and a pointed left side so that they are easily recognizable.*

To add the colour back into the render we need to create a “DeepRecolour” node and plug the Deep pipe into our DeepRead and the color pipe into our Beauty pass. This will give you a darker, but in-color representation of our renders. The last step is to go into the DeepRecolour settings and check on the “Target Input Alpha” box. Now we should see an image that looks exactly like our normal beauty pass!

To view what our Deep data looks like you can create a DeeptoPoints node which will take the deep data and create a point cloud based on it. You must export your camera from Maya as an FBX and import it into the Camera node so that Nuke knows where the camera should be positioned. Plug the Deep pipe into your DeepRecolour and the Camera pipe into your camera.
Now when viewing this node it should look exactly like the scene in Maya because it is using the same information!

This is what you should see:

Now that we have this information we can create any holdouts we need to when compositing additional elements into our scene.

**Example 1 - Deep Holdouts**

I created a text layer and mapped it onto a 3D card rendered with Nuke’s Scanline renderer. The Scanline render also outputs deep data by default so we can use a DeepMerge node to combine the two scenes inside Nuke.

The following image is the result. Notice how the text is being accurately occluded by the trees in front of it. I can manipulate the 3D position of this text in any way I wish and it will always sit in the scene exactly as it would be if we created the text layer in Maya. The difference is that with Deep I can make any changes I want in seconds instead of hours.

The second image is what the merge looks like inside of Nuke.
The result of a DeepMerge

What the DeepMerge looks like inside of Nuke.
Example 2 - Using Deep for Environmental Effects

A second example using the same renders is created using Nuke’s built in particle system. If we wanted to add synthetic rain to this image without deep data we would have had only two options: to just overlay the rain on top of everything in the scene or to painstakingly mask out the different layers of trees and add rain under/over each layer.

Using Deep we need only create a 3D particle rain and merge it into our Deep pipeline. The rain will now fall in a manner that is more accurate to real life and will be occluded properly by all the trees. Note in the first image that rain is falling not only behind the trees but in between and in front of them as well.

With a system set up like this we can add atmospheric and environmental effects with ease and they will integrate properly with very little effort. We can design and make changes to this effect extremely quickly which will allow the artists to make more creative decisions and not be restricted by technical limitations and render times.

Notice how the rain wraps properly around the trees
With Deep we maintain full access to the particle systems and full interactivity with the renders.

**Example 3 - Using Deep for Atmospheric Effects**

In this example we use the Deep Merge nodes to provide a much more accurate A-B comp of a series of cubes into a volumetric area.

We can see that by utilizing the deep data, we are merging the rendered objects inside of the volume, not just on top of it or behind it.

In the final picture you can see a comparison of the different look achieved by doing a traditional 2D merge and using the DeepMerge. With 2d merging it is simply using a 2D opacity map (alpha channel) to dictate the areas that the underlying image is visible through. With the Deep nodes it is utilizing the visibility function of each pixel in order to accurately depict the visibility of the objects based upon the way the volumetrics build up in space.

You can see very clearly the advantage that deep brings to combining images and effects and the increased levels of visual fidelity.
The Deep composite node structure
Visualizing the Volumetric Deep Data
Here we can see a direct comparison of a standard 2D A over B comp (Image 1) and a Deep Composite (Image 2). Notice in the first image all the detail is lost from the object render when combined with the fog. This is because it is treating the whole image with one opacity map. In the second image you can see that visibility has a falloff with the depth, this is because the physical 3D space is being respected.
Important Considerations for Deep

Deep Compositing affords artists unprecedented control over their images, but this control can serve as sort of a “pandora’s box” especially for newer or less experienced artists.

An important step in the process is the VFX Breakdown where a producer and director or VFX Supervisor must bid time for each task associated with your shots. Especially for students, it is important to make strategic decisions about when it is appropriate and helpful to use Deep data when rendering shots. Artists must be diligent and hyper aware of the amount of data that deep will add to their production. A good rule of thumb is to assume that an extra 50% of data will be added. So if a render is 50 GB, assume there will be an additional 25Gb of just Deep data.

Deep Data is also slow to work with. So the production team must make sure to add additional time to the bid for comp tasks that are using deep data and artists must be smart about when and how they use Deep data. It is important that they make sure to utilize precomps to get the deep data out of the pipeline as quickly and efficiently as possible.

And lastly, it is always a good idea to maintain as much control as you can for as long as you can. But make sure you are not using the control deep offers as a crutch to avoid making actual decisions. Deep should be used to make an image better, not to make the image up front. This is especially important for student directors.
Virtual Reality Short Film Production Pipeline

Immersive Cinema is a form of storytelling that engages the viewer in ways that have never been seen before through the level of immersion it provides. With my research I have helped a team to devise a production pipeline for VR that is built around production tools and softwares that should be common among collegiate Film School pipelines. This is done with the intent of accessibility and to help encourage VR producers and developers with fewer resources to start pursuing projects in this art form.

Until recently, VR production has been too expensive and too complicated to be accessed by most artists at student and otherwise fiscally limited levels. Through the developments made by companies like Oculus, Google, and GoPro; Immersive Cinema is now easier to produce more accessible than ever. The only barrier to entry now is the willingness of the students to learn and design their solutions to problems that even professional teams have yet to hammer out.

Immersive Cinema is a medium that shares many similarities to filmmaking, animation, and videogames; but it also presents very unique issues that are still requiring experimentation to solve.

In this guide I will walk through the production process for a VR short film on a technical and hands-on production level. The tools I use are free for students unless otherwise noted in the guide. I will also highlight questions that directors should be asking themselves while they are planning to shoot their content. The most important thing to remember is that every process is different when it is presented to a view in VR, so you must question every assumption you have previously held when tackling this new project.
Step one: Pre Production

Medium

The first question that you must ask yourself when moving into a production is this: is VR the proper medium to tell your story in? A writer should not craft a screenplay when they should be writing a novel just as a filmmaker should not make a VR short if the story is better suited for traditional styles. It is paramount to really think about what elements will be gained and lost through creating in this medium.

After that you must consider if VR cinema is the medium that is best for your story. The world of virtual reality extends far beyond live action sets with real people. In many cases a story or theme can be better expressed in a synthetic medium. You can craft an entirely stylized world for your story to live in by taking an animated route; using mediums ranging from Computer Generated Animation to Stop-Motion Animation. These are very easily transferred to VR and can provide an even more immersive and designed world for you to place your viewer into. One step further from this is the idea of telling your story as an interactive video game experience. Through free tools such as Unreal Engine it is very accessible to craft an experience that your viewer can really be a part of and interact with.

As a storyteller, it is your responsibility to utilize the best medium where your story can have as much impact as possible. Do your story justice by creating it in the world that it belongs in. Or mix mediums! It is a wide open world and only you can know the best way to tell your story.

Questions:
- Is VR the correct medium?
- Is live action the correct medium?
- Would my story be better served by a synthetic medium such as Animation or Video Games?

Script

Now that you have settled on your medium it is time to craft the actual story you are going to let the viewer experience. It is important to keep the capabilities VR grants to your audience in mind. For example, they are not
always going to be constrained to one viewpoint and therefore every
direction must be described and fleshed out. A script should detail the world
of a story just as much as the action. Traditionally, scripts are more
action-oriented but VR non-traditional scripts need to put special emphasis
on the world so that the reader can envision what they would see from every
angle.

A writer must also incorporate how the medium feeds into the script.
The way action is staged is completely different now that the viewer can see
in all directions. Sound and effects can also be placed anywhere in an
environment. But it is extremely important to not abuse these new
privileges. Having a conversation take place in an area where the viewer has
to look back and forth to see who is talking can be neat, but after a while it
becomes tedious and annoying. With every placement you are now
requesting that the viewer make an effort to see your content. Will it be
worth the effort or will it just be annoying?

When used as a single viewer experience, VR has a higher than normal
risk of being abandoned midway through. There is now no social pressure of
a movie theatre to stop a person from just taking off their headset and
walking away. Do the viewer justice with your content; respect their time
and effort.

Questions:
- What is the world my story is taking place in?
- How do I take advantage of the freedom of view and field of view?
- Is my design annoying or asking too much of the viewer?

**Previzualization (PreViz)**

Now you have created the story, but how will your viewer experience
it? It is important to be prepared as possible before actually shooting your
content for many reasons. It is important that your story be presented in a
way that is visually engaging and clear to a viewer. Traditionally this would
be handled by drawing out storyboards, but now we have more than just a
2D frame to work with. We also have to account for more than just story
clarity, we have to make sure that our experience will not make people sick
and that their attention is directed to the proper area. It is also required to
test how a scene *feels* as in VR a traditional space can feel claustrophobic or
too empty when viewed through a head-mounted display. This stage is
incredibly important to ensuring a successful VR experience.
There are many ways to approach this problem. The easiest way is to obtain a cheap 360 degree camera and going out to a location to run through different tests of how to stage and shoot your story. A perfect example of this would be a Ricoh Theta 360 camera. Thetas are extremely easy and fast to use, but do not possess the image quality for a final product making them a perfect choice for previz as you can quickly iterate and test how you stage your action and actors. This camera costs about $250 so it can be cost prohibitive in some scenarios. There are alternative cameras as well, so plan accordingly for your budget and production.

For a free solution to previsualization, you can use Unreal engine. Unreal is a fully featured real time video game engine designed for AAA game production. This would require you to create your sets and actions from scratch and requires more work with a higher learning curve. However, Unreal Engine is free for noncommercial use and allows you to output your scenes natively to VR headsets such as the Oculus Rift. This will also require a general knowledge of the 3D process, which is a skill transferrable from programs such as Autodesk’s Maya (also available for free noncommercially). Because everything is digital, restaging and iterating in your scene is extremely fast and easy. Your previz will be output to a headset which will make it easier for you to test for the feel of a scene and catch any instances that might make a viewer uncomfortable or even motion sick. The low cost of this method also means you can create a previz that includes lighting plans and mood sketches so you can get a more accurate plan for your aesthetic.

Questions:
- How does my scene feel?
- Does this design or camera movement make me feel sick?
- How am I blocking my action and characters?
- What do my sets look like?

Camera Selection

Now that you are confident in the story you will tell and how you will tell it, we have to get into the details of camera selection. This is where there are the most costs associated with VR production, however they are very scalable to any sized development team. There are many different designs and prices for cameras ranging from the relatively cheap Samsung 360 camera at $350 to the fully featured Nokia Ozo camera at $60,000.
Each camera has its own drawbacks and functionality, but there are common trends to look for and your requirements will be different depending on your available resources and the type of experience you wish to create.

There are two main types of cameras: mono and stereo. A mono camera will produce a 360 image sphere but will not have any depth. Stereo cameras will produce a 3D 360 image which accounts for left and right eye views. Stereo is more realistic and natural which provides for greater immersion. Mono VR can also contribute to motion sickness. Another consideration is framerate. Standard cinema is shot and projected at 24 frames per second. In VR this is too low. The Oculus Rift requires that experiences be run at 90 fps to avoid motion sickness. For live action cameras a frame rate of at least 60 Frames per second is recommended.

The camera rigs used by most professionals are built around arrays of Gopro cameras. These rigs take 5 - 17 Gopro cameras and arrange them in a sphere facing outwards. These cameras are ideal because they provide a wide field of view and high frame rate at a relatively low price per camera. These rigs are also scalable to any size production, which is why Gopro rigs are my recommendation for VR production. You will need at least 6 cameras for mono 360 and up to 16 for stereo rigs. Most rigs can be 3D printed if you possess that capability or can be bought online directly from Gopro’s website.

The cameras that you choose to use will vary especially based on how you intend to distribute and view the footage that you shoot. If an Oculus Rift or Samsung Gear VR is your intended output it makes high frame rates and high resolution essential. However if you only intend to view your projects on a Google Cardboard or you plan on skipping the viewer and going straight to something like FaceBook or YouTube 360 then your requirements will also fall lower in terms of framerates and resolutions.

It is important to also consider the landscape of VR cameras. We are currently at a point where new technologies and cameras are being developed and released constantly. So make sure you heavily research which camera is right for you and why before investing money into a camera that does not suit your needs.

Another extremely important consideration is the I/O pipeline that you possess. Shooting HD footage at high framerates generates a lot of data, so you must possess the storage space and read speeds that will make your footage actually viewable. A rule of thumb is that mono production will run at about 4Gb of footage per minute and stereo will be about 8Gbs.
Questions:
- What capabilities does my production require?
- What are my available resources?
- What is my budget?
- How will I be viewing/distributing this footage?

Summary:

Software:
- Unreal Engine 4
- Autodesk Maya

Headsets:
- Oculus Rift
- Google Cardboard
- Samsung Gear VR

Cameras:
- Samsung 360
- Nokia Ozo
- GoPro Hero

The camera setup we use at FSU is the following:

16 GoPro Hero Blacks
Purple Pill 3D Printed Stereo VR rig

Camera settings:
1440 resolution
60 Frames per second
Step 2: Production

Shooting

Now that you are on set there are many more elements that must be taken into consideration. You should already have a good idea of where the camera is going to be placed based on your previz, but now you must consider the actual elements of your location.

When shooting with multiple cameras you have to be aware of stitch lines, which are the areas that the camera’s field of view overlaps with the rest of the cameras. At this point it is common for there to be a visible line if the images do not sync together 100%. When you are staging your actors or planning any practical effects such as smoke or fire you need to divide your scene into sections based on your camera rig. Make sure the action taking place stays in that section and be very deliberate and heavily plan when an action will move across a stitch line. A good way to ease the transition is to distance the action from your actual camera rig; stitch lines will be more intense and visible the closer the action is to your camera. This is due to the overlap in what the cameras see and the algorithms that stitching programs use to combine the footage. If an object or actor is seen by more than one camera at the same time it will begin to look doubled or “ghosted.” The Purple Pill VR company refers to the area around your camera that causes ghosting as the “Glitch Zone” and generally places it in a circle about 1.5 meters from your camera rig.

Questions:
- Where are you placing your camera?
- Will the action cross between the cameras?
- Is your action within the “glitch zone”

Directing

The process of directing revolves around how you work with your actors to tell your story. But this is problematic in VR as there is nowhere for a director to hide and give notes on the fly or review the footage. Some
solutions to this issue are for the director to hide underneath the camera rig, in an area that might be either pure black or covered by an overlay. Another solution is for the director to actually act or be a part of the scene being shot. As long as they blend in and do not call attention to anything they do there is no issue with a director appearing to be an extra or even a lead actor (if they have the ability).

The next issue that arises for on set production is the ability to review footage as you shoot it. Assuming you are using Gopros, you must stitch the footage together before you can properly review it. The solution we came up with for this is to have an on-set data manager/assitant editor who can utilize software on a laptop to organize and roughly stitch shots on set while the shoot is ongoing. This provides an answer to Data Management and also for review. This goes hand in hand with very accurate script supervising and record keeping. It does not make sense to stitch shots that you know will not be used, so utilizing circle takes and proper notation will avoid wasting time on shots that will not be used.

The idea to keep in mind is that a VR production is much closer to theatre than it is film. The scenes require full runs and long takes in order to maintain the illusion of reality. Rehearsal is essential and a director needs to both trust their actors and know when a scene is working.

Questions:
- How will I direct a scene?
- Where will I sit during a take?
- Can I be in the scene?
- How can I know if a take is usable?

Crew

VR production means that there are no more hidden areas. This means that you need to think through where you stage your crew and your equipment when you are shooting your scene. It is important to strike a balance between distancing your crew and gear and also making sure that the set can still be run efficiently. If your gear is too far away, it will slow down production. If you gear is too close it runs the risk of being in your shot. A solution to this issue comes with VFX, if you shoot a clean plate for every shot it is possible to paint out and remove any unwanted objects from your scene which will expand the possibilities for what you can create.
This transparency permeates everything that your crew does. Cinematographers and gaffers must be aware of how they approach the process of lighting the scene. Most lights will be visible in the shot and must be removed in post. Unnatural light sources or lighting effects can break the realism of the shot. The most common approach is to use either completely natural light or to work with production design to incorporate lights into the scene itself. When thinking about your production design and set dressing you need to consider every direction a viewer can look. How does your production design support both the story and flesh out the world. A barren area or empty set will detach the viewer from your story and cause them to disengage. It will eliminate the heightened sense of realism that VR can provide. The design of the set and the lighting must permeate the whole scene while remaining invisible to the viewer.

Questions:
- How much VFX can I plan for?
- Where will my crew and equipment hide?
- What is visible in the frame?

**Sound**

Traditionally, films are made with stereo sound. This works well when viewing a 2D image on a screen but in VR you need to create a full 360 degree soundscape. This type of sound is known as Ambisonic Audio. It works by utilizing headphones to create the illusion of both distance and direction for sound design. The easiest solution for capturing ambisonic sound is to utilize an ambisonic microphone such as the Core Tetramic, but this specialized gear can be expensive. Another solution is to utilize lavaliere microphones on each actor and then use two traditional stereo mics pointing in opposite directions. Capturing all this data onset will allow you to create an ambisonic mix when you design the audio in post-production.

Questions:
- How will sound help direct the viewer's attention?
- How will I capture the on set sound?
- Where/What is the source of a sound?
Data Wrangling On-Set

When shooting for VR, VFX become almost essential for any production. This means that on set you need to have someone dedicated to gathering as much on set data as possible to make life easier once in post. It is the data wranglers job to get measurements of the height of the camera rig for each shot and the distance of the actors from the lens. They must also ensure that a clean plate is shot for each set up and they also need to take measurements of the set and take as many reference photos as possible, especially of the floor and walls in case anything needs to be recreated in post production.

This job can also be coupled with camera team positions and they can be in charge of ensuring that all the camera settings stay consistent between takes. Synchronizing footage is essential for stitching work so the Data Wrangler must ensure that multiple precautions are taken to make this process easier. Every take should begin with a clap for audio synch, a “woggle” where you shake the camera rig around a little bit, and a flash of light to provide a point for visual synchronization.

While this may seem tedious, it is the most important thing you can do to ensure an end product that is usable and of a high quality. Nothing breaks the illusion of reality like stitch and sync errors.

Questions:
- What level of VFX am I capable of?
- Did I get all my measurements?
- Did I shoot a clean plate?
- What information will I need when I do my VFX?

On-Set Data Management

I will now go into detail about the setup we used for on set data management. When shooting with Gopros you only get about an hour of footage on one card and only about 45 minutes of battery life. Both of these are unacceptable for a full 12 hour film shoot so you must take precautions to conserve battery power and offload footage quickly and efficiently.

Our solution to this was to create a VR data cart. This was a camera cart outfitted with a powerful laptop, external hard drive, car battery with converter, AutoPano software for stitching, and ManyCams for importing and sorting footage.
Between set ups for new scenes, the camera will go straight back to the cart to be plugged in for charging and footage offloading. This will also give the cameras a chance to cool down so they do not run a risk of overheating and shutting down during takes or set ups. This is also an opportunity to check the setup and stats on all the cameras to make sure they are all in sync with each other.

When the cameras are in use for filming, the Data Manager can spend this time doing rough stitches of the takes with AutoPano and sorting the footage into appropriate folders on the external hard drive. When stitches are done they can also begin to create an assembly edit on set.

Having this dedicated position helps to ensure a set runs smoothly and will also decrease time spent in post production sorting through the hundreds of clips and comparing them to the paperwork to make sure they line up, which can lead to inaccuracies and errors in the post production pipeline.
Step 3: Post Production

In this section I have also included “Bids” which are estimates of how long each task should take. This will be in days or in hours for a per-task time estimate. These are only estimates based off of my personal experience and will vary depending on the shot and the complexity of each task.

This section of the guide is written assuming you have used a Gopro based camera rig, but the theory and techniques are applicable to most other rigs and types of footage.

Data Management
Bid: 1 Day

The first step when you get your footage it to organize it all. There exists software to automate this process, such as ManyCams by The Purple Pill VR. But the theory is this: you need to divide your footage into folders based on take. Inside each take folder should be the corresponding clip from each camera. These clips should be named accordingly, for example:

SHOT#_TAKE#_CAM#_PRODUCTION.mp4

This is a very time consuming process so make sure you have allotted enough time in your schedule. Unless you have utilized an on set data manager, then this should already be taken care of.

The amounts of data you will be dealing with are large so disk speed is very important. If you can, solid state storage is ideal so you reduce waiting time when working with the footage. On top of that, if you are working in a facility you should have a dedicated server for the footage that is tied to only the computers that will be accessing the footage. Our server at FSU is a 10 gigabit connection utilizing solid state storage on both the local drives and the server.

Synchronizing Footage
Bid: 10-15 minutes per shot (depending on method)
The first step towards stitching your photosphere is making sure all of your cameras are in sync. Various technical issues can contribute to your cameras being either a few frames off sync or even a few seconds. We will be using Kolor’s Autopano software to execute this task as it allows you to synchronize cameras based off of their motion. This is what the “woggle” is for on-set when you shake your rig around. Autopano knows that when one camera moves all of the others should move at the same time. It will analyze your footage and then offset each clip so that the motion lines up. This provides highly accurate results very quickly.

Alternately you can use your standard video editing software to manually sync each clip just as you would traditionally. Using the clapper board and the audio tracks you can line up the audio peaks and then trim each clip to start at the same time. Then export each trimmed clip back into your stitching software.

**Stitching Footage**

**Bid: 15-20 Minutes per shot**

Now that your cameras are aligned, you can begin the all important process of stitching. The way stitching software works is by analyzing your footage for prominent features, it then take those features and compares them against all of your other cameras to find areas of overlap. After it finds these areas it can line up the features automatically and create your 360 shot. Stitched shots are exported as Equirectangular projections, which you are familiar with if you have ever seen a map of the Earth. This projection takes a 3D sphere and lays it out in a 2D image; you can see that it leads to some issues of distortion especially around the top and bottom areas of the image, known as the Zenith (Top) and the Nadir (bottom).

I will walk through the process using Kolor’s Autopano and then the same process with The Foundry’s Nuke.

Autopano is designed to be easy to use and foolproof so it provides a very effective autostitch. When you are on the tab you hit the “stitch” button, and then you can move to the “live Preview” and hit the edit button to make any specific changes you need. The biggest problems to look out for when stitching are ghosting and obvious stitch lines. To check for these you can look for points of high contrast as these are where errors will be most prominent. A white object such as a smoke detector or a street light will look as if there are two of the same sitting next to each other. A large window will give stitch issues along the edge if two cameras detect different
exposure averages. If you see these problems you can go into the manual edit and manipulate the footage so that these doubled objects are perfectly aligned. If you have stitch lines based on exposure then you can solve them in the next step: Colour Correction.

If you are using The Foundry’s NukeX VR tools then the process is much more involved but can achieve greater results with more control over your image later on in the pipeline. Import your shots into Nuke and then create a VR_CameraSolver node. Attach your shots to this node. Inside the node settings you need to tell Nuke how many cameras you are using. If you are using a premade rig, you can select it from the preset menu. Now select “Setup Rig”. Go through the following settings and fill out all the known information about the cameras and lenses you are using. If unknown, you can select the “Optimize per camera” options which generally work well. Set Keyframes for your sync when the cameras are clear and you don’t anticipate issues, E.G. do not select a point when someone’s face is visible in three different cameras. Now you can go to the bottom of the options and hit “Solve” and it will calculate your camera placement and then output a stitch in whatever resolution you select.

The following image is what a stitched equirectangular projection should look like.
Note in this second image the obvious stitch line where the sun is. This is caused by exposure levels not matching between the image sensors and a lens flare that is not being picked up by the other cameras.

A good way to check the accuracy of your camera solve is to select the bottom option and export your cameras. Viewing the output cameras in 3D space should roughly resemble the placement and orientation of your camera rig in real life. This can also be used later on when we discuss the VFX process.

**Colour Correction**

**Bid: 5 minutes per shot**

Now we have created out camera projection, but the lines between camera might look obvious as each sensor is going to see the world a little bit differently.

In Autopano you can switch to the Colour Correction tab and hit analyse. It will match the cameras automatically.

In Nuke you need to add a VR_ColourCorrect node to your pipe immediately after your camera solve. Select a keyframe to analyze (a "Key Frame” is one that is representative of all following shots). In the node
settings you should also switch to "Exposure and Colour" as it will provide a better looking result.

In my experience, AutoPano will provide a more accurate colour match. However, Nuke will retain your access to the original footage and you can add additional grade and Colour Correction nodes to these clips before they go through the VR_CameraSolver node.

**Distortion/Horizon Correction**  
**Bid: 5 Minutes per shot**

Now that your footage is stitched it is highly likely that the horizon line of your sphere is not accurate to the real world. In Autopano you can fix this by pulling areas of the clip in the edit window you access through the Live Preview tab. In Nuke you can adjust this with the VR_RotateLatLong node that you add to your pipe after the colour correction. This process is just done by eye and you can line up the horizon to whatever makes the most sense based on your footage.
Post rotate LatLong and Colour Correction
Notice the Ghosting on the smoke detector resulting from improper camera alignment
The outcome of the preceding techniques should be completely and seamlessly stitched footage.

Once you export the equirectangular renders you can bring them into any video editing software and treat them just like you would normal video clips. Do not start the process of VFX until after you have a locked edit, as this can lead to a lot of work being thrown away if the edit is changed.
**Step 4: Visual Effects**

When you are working with 360 video there are always at least two additional steps added onto the normal VFX process. These are to essentially unwrap the equirectangular image, do your effects as you normally would, and then reproject the image back into the equirectangular projection. Currently there are only two ways of adding VFX to Virtual Reality footage: The Foundry’s Nuke and the Skybox360 plug-in for Adobe After Effects. This section will discuss the process using Nuke, but the theory and practice applies to any VR VFX pipeline.

**Standard VR VFX Pipeline:**

Step 1: Add a VR_Sight node to your Nuke script. Switch the option from spherical to rectilinear and at the bottom of the option box select the button for Paint and Reproject. This will create a backdrop with two nodes, a rotoPaint and a VR_Reproject. Your image in the viewer will also change to look like a punched in version of your Equirectangular image. This is the un-projection.

Step 2: Use the VR_Sight node to navigate the un-projected image over to the area you want to work on. After this area is filling your viewport you can create your modifications just like you would with normal footage. The area between your rotoPaint and the VR Reproject can be changed in any way possible. After making your changes the image will be reprojected to the Equirectangular output and will wrap appropriately when viewed in your headset.

Step 3: If you require effects or paint work in another area of your shot, add in another VR_Sight after your Project. Then create another instance of Paint and Reproject. Repeat as necessary for all VFX Shots.
Painting out your Rig
Bid Time: 30 Minutes - 1 Hour (depending on floor composition)

The most glaring need for VFX when you are working with 360 footage is to paint out the tripod that is causing the down direction, the Nadir, to be a sore spot full of stich issues and tripod legs. The solution to this issue is a standard paint process, but with VR you have the added issue of reprojection.

Removing the tripod using the rotoPaint node after the VR_Sight
**Keying/Matte Extraction**

*Bid: Half Day (4 Hours) - Full Day (8 hours) depending on clarity and interaction with foreground elements*

This process is nearly identical to the rotoPaint process, utilizing the VR_Sight and then Reproject nodes. The actual keying must be done between these nodes.
Bibliography/Citations/References

Works Cited


Companies Referenced:
Oculus
Samsung
Google
Purple Pill VR
Pixar
WETA Digital
Animal Logic
Dreamworks Animation SKG

Software Used:
Unreal Engine 4
Autodesk Maya
The Foundry’s Nuke
Kolor Autopano Pro