

Deep compositing in VFX

Creating a framework for deciding when to render deep images or traditional renders.

Jonas Kylén

**Computer Graphic Arts, bachelor's level
2019**

Luleå University of Technology
Department of Arts, Communication and Education

Sammanfattning

Den moderna VFX-pipelinan balanserar idag artisters produktivitet mot kreativitet och möjlighet att göra drastiska visuella förändringar sent i produktionssteget. Ett verktyg för att tillåta dessa förändringar sent i produktionen är deep compositing.

Denna rapport syftar till att etablera ett ramverk för artister och supervisors att avgöra när renderade element och scener bör renderas med deep data eller inte för att ge flexibilitet i compositing-stadiet. För- och nackdelar kommer att presenteras och deep-metodiken kommer jämföras med mer traditionella metoder.

Abstract

The modern VFX pipeline is all about balancing the productivity of the artist against creativity and allowing for drastic visual changes late in the production pipeline. A tool to allow for these changes late in the production is deep compositing.

This paper means to establish a framework for artists and supervisors, to help decide when rendered elements and scenes should be rendered with deep data or not for increasing flexibility in compositing. Pros and cons will be presented and the deep methodology will be compared to more traditional methods.

Content

1 Introduction	3
1.1 Background	4
1.2 Problem question	4
1.3 Limitations	4
2 Theory	4
2.1 Compositing	4
2.2 Pixels	4
2.3 Rendering	4
2.4 RGBA	4
2.5 Holdout matte	5
2.6 Depth data	5
2.7 Bit depth	5
2.8 OpenEXR	5
2.9 Motion blur	5
2.10 Scanline render	5
2.11 XYZ	5
2.12 Point cloud	5
2.13 Aliasing and Anti-Aliasing	5
3 Method	6
3.1 Method introduction	6
3.2 Preparations	6
3.3 ZMerge vs Deep merge	7
3.4 ZSlice vs Deep crop	7
3.5 Adding cards	7
3.6 File sizes	7
3.7 Method critique	7
4 Results	8
4.1 ZMerge VS Deep merge	8
4.2 ZSlice vs DeepCrop	10
4.3 Adding cards	11
4.4 File sizes	14
5 Discussion	15
6 Conclusion	16

6.1 Recommendation	16
7 References	17
7.1 Publications	17
7.2 Electronic sources	17

1 Introduction

1.1 Background

In compositing, it's very useful to know the location of objects in z-depth in order to place elements at the correct depth with correct occlusion.

Traditionally this is done by a combination of rendering a separate z-depth pass, where depth is represented by a normalized black and white image, using this image to isolate certain depths, and by cutting out a holdout matte for the objects in the render of the element itself. (fxguide.com, 2014)

Let's say there is a rendered, semi-transparent, smoke element. To be able to place an object in the midst of that smoke in comp it needs to be cut out from the smoke render to get the correct occlusion. Some smoke might be in front and some in behind the element. If there is any modelling or animation change later on the smoke needs to be completely re-rendered using the new asset as the new cutout.

Deep compositing is a completely different way of working with rendered elements where every pixel samples an array of values that define the change in depth (fxguide.com, 2014). The same smoke element from earlier would store a multitude of depths per pixel. The element and smoke could be rendered separately and combined at the correct depth with the correct occlusion.

Deep images and data is a fairly recent technology, only introduced by Pixar in the 2000 Siggraph paper "Deep Shadow maps" where shadow maps were created that could store a representation of the fractional visibility through a pixel at all possible depths. (graphics.pixar.com, 2000)

The VFX facility WETA Digital created a set of deep compositing tools for Nuke that were later implemented in Nuke 6.3. Since the deep data is so different from regular 2D images these deep tools can not be used in the same context as regular nodes. (fxguide.com, 2014)

1.2 Problem question

The purpose of this thesis is to test the limitations of the deep workflow and contrast it with other solutions to the same problems.

- When should you use deep rendering in your pipeline?
- What are the limitations and advantages?

1.3 Limitations

This paper will not discuss the many optimization methods used to make it easier to work with deep files. Whether optimizing sampling when rendering or cropping in comp. Depth of field using deep and ZDefocus will not be tested either.

2 Theory

2.1 Compositing

The art of combining two, or more images in order to make another. The goal is usually to integrate them and match the color to each other. (Wright, 2001, p. 1). In visual effects (VFX) this is usually done digitally on the computer, but it used to be done photographically.

2.2 Pixels

The smallest addressable element in an image. An 1080p full HD resolution image consists of 1920 x 1080 pixels which equals about 2million pixels.

2.3 Rendering

Sending out rays of light from the camera into the scene, rasterizing a 3D scene into an image consisting of pixels using samples to correctly calculate the color values of each pixel. Increased sample values increases the probability of a correct pixel value.

2.4 RGBA

Red, Green, Blue and Alpha channels. Most common channels to render.

2.5 Holdout matte

An alpha matte that is used to cut out something from another render for correct occlusion.

2.6 Depth data

Each pixel has one, or more, unique values per depth.

2.7 Bit depth

Specification for the bits of data used for each color component. Can be defined as either bits per pixel, bit per channel or bits per color. EXR files are often rendered in 16bit or 32bit to maintain color information.

2.8 OpenEXR

An image format developed by Industrial Light & Magic, capable of storing multiple channels of potentially different pixel sizes and bitrates.

2.9 Motion blur

Camera artifacts of motion made by the mismatch of the movement of the element being filmed and the shutter speed of the sensor of the camera.

2.10 Scanline render

A rendering algorithm, used in one of Nukes 3D renderers, that renders on a line by line basis from top to bottom.

2.11 XYZ

Directional axis used to describe the 3D world.

2.12 Point cloud

A visual representation of the pixels place in depth using points in 3D space with RGB values. Useful for placing objects at the correct depth.

2.13 Aliasing and Anti-Aliasing

Aliasing refers to the incorrect value sampling from edges of an element, usually as a result of too few pixels or sample count. Anti-aliasing are methods used to combat these jagged edges, by for example supersampling the edges, blurring them and much more. (Wright, S, 2001, p.455)

3 Method

3.1 Method introduction

The methodology will consist of a visual analysis of artifacts resulting from combining, and separating, different rendered elements with Z-depth and deep compositing as well as a workflow analysis, assessing the ease of use of each method and other production considerations such as file size.

The rendered elements will exhibit difficult properties for occlusion such as heavy motion blur, semi-transparent edges, occluding cloud as well as high frequency detail. This is done to challenge z-depth and deep workflows and see which is more effective.

These findings will result in a recommendation which can be used by a compositor or supervisor to establish whether it is worth rendering, and working, with the deep workflow. The recommendation will be divided into three categories; “Dont render deep”, “Consider rendering deep” and “Render deep”

3.2 Preparations

A 3D scene was created in Maya, one sphere with Xgen hair, one sphere with motion blur, a platonic surface with an opaque surface and a volumetric cloud.

Each element was rendered in 1280x720p with deep data and as regular Open EXRs. It was also rendered all together as a reference, henceforth referred to as “beauty render” in the report.

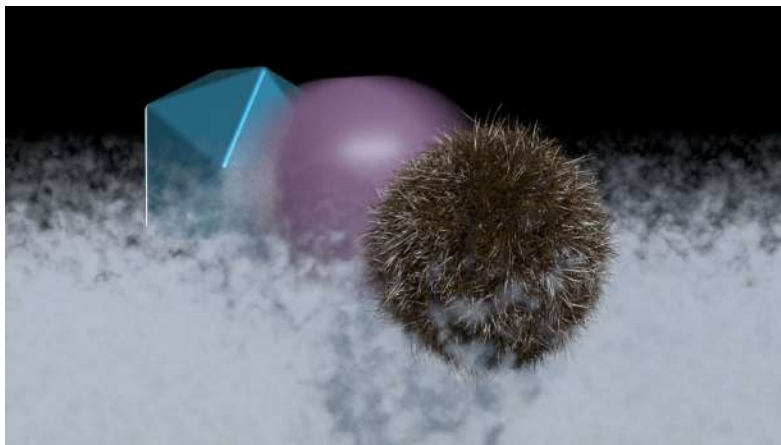


Figure 1 - Beauty rendered including cloud element

3.3 ZMerge vs Deep merge

Each element of the scene will be merged using ZMerge and Deep merge, a visual analysis of the edge artifacts and workflow will be made.

3.4 ZSlice vs Deep crop

The elements of the beauty rendered will be isolated using the Zslice and Deep crop nodes. Any visual errors and workflow considerations will be discussed.

3.5 Adding cards

A fire element will be placed in the scenes using the deep workflow and Z-depth method. The visual and technical aspects will be discussed.

3.6 File sizes

The file sizes of the deep elements and their Open EXR 2D counterparts will be put on display.

3.7 Method critique

The results will be dependant on the efficiency of the available plugins that deal with z-depth and deep data, not the underlying data itself. There could be better plugins or other solutions out there, or up-and-coming, that would improve the results. Further file- and scene optimization could be utilized to change the file sizes. The visual and workflow analysis will be based on one artist, not a big enough group to have statistical reliability. The resulting guide will be a subjective recommendation based on the findings of this rapport.

4 Results

4.1 ZMerge VS Deep merge

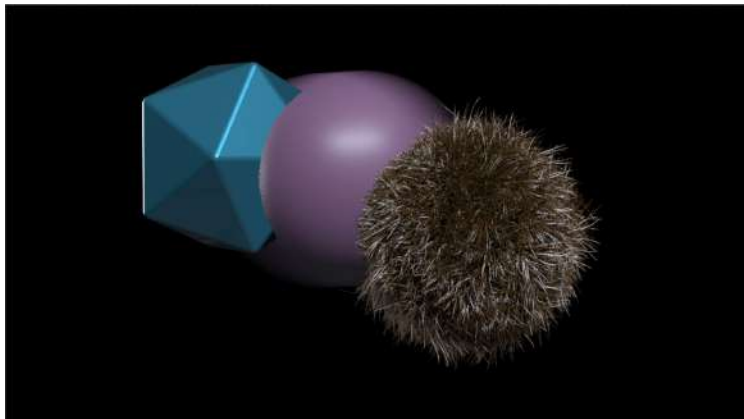


Figure 2 - ZMerged elements



Figure 3a - ZMerge

Figure 3b - Deep merge



Figure 3c - Beauty

Since the semi-transparent edges of the motion blurred sphere are grey in the z-depth, it's considered at a different depth than the rest of the sphere which results in incorrect occlusion, see Fig. 3a. The same goes for the hairy sphere.

Volumetrics

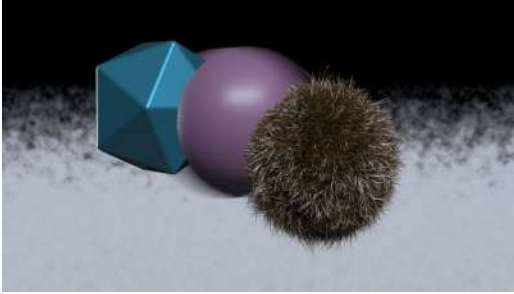


Figure 4a - Z-merge

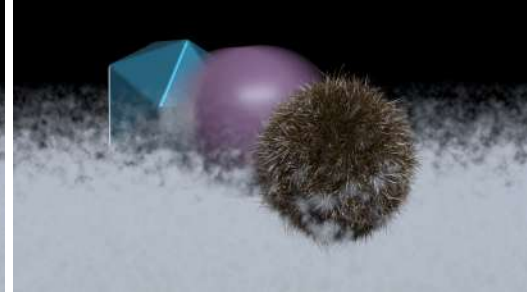


Figure 4b - Deep merge

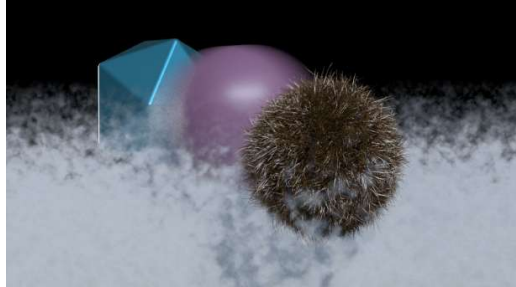


Figure 4c - Beauty render

Z-merge doesn't occlude properly, see Fig 4a. It only knows one depth value per pixel. The deep merge in 4b does it perfectly. The only difference with the beauty, 4c, is contact shadows and secondary light bounces which is to be expected since they were rendered separately.

4.2 ZSlice vs DeepCrop



Figure 5a - Beauty

Figure 5b - ZSlice

Figure 5c - DeepCrop

The same artifacts present in the ZMerge test is present here in the ZSlice (Fig. 5b) since its using the same aliased z-depth image. We also see through the sphere to the purple motion blurred sphere.

The ease of the workflow is about the same, sampling z-depth to see where to cut it using ZSlice and using DeepSample to see depth in deep comp.

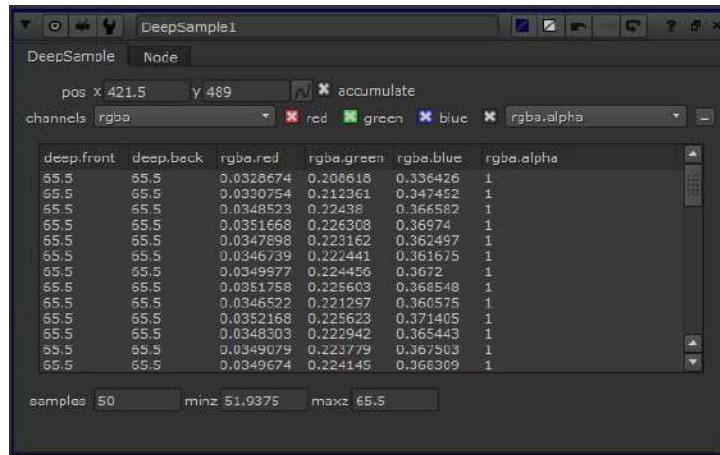


Figure 6 - DeepSample node used to locate depth in deep comp.

4.3 Adding cards

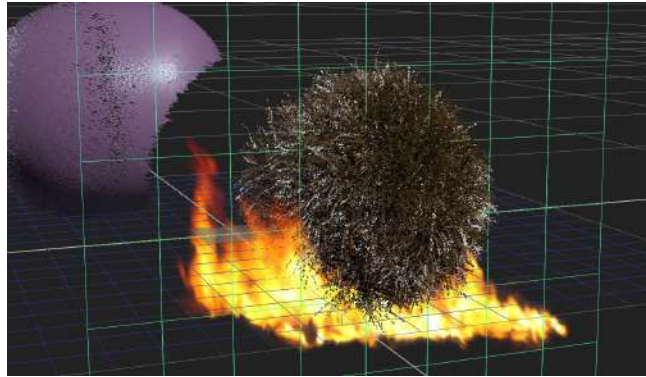


Figure 7 - Card placed inside 3D space of Nuke.



Figure 8 -Deep render

Adding a card at the correct depth is easy using deep since it is perfectly visualized in 3D space using DeepToPoints. A holdout matte is generated and the 2D image can be further manipulated in the 2D context and comped over the beauty, see Fig. 9. Workflow consists of using Deeptopoints to create a point cloud, a card and a Scanline render. Use DeepHoldout to cut out the element from the fire.

Each hair strand has correct visibility.

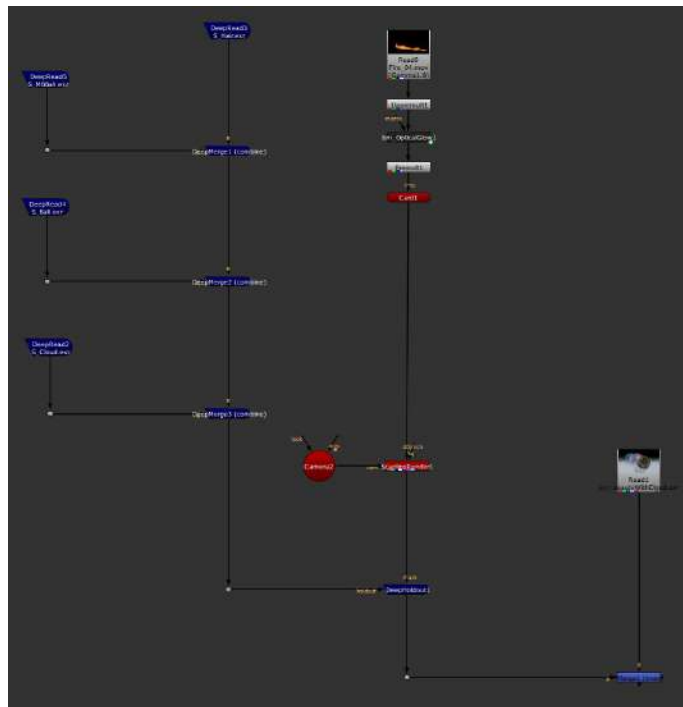


Figure 9 - Nuke script of Deep workflow

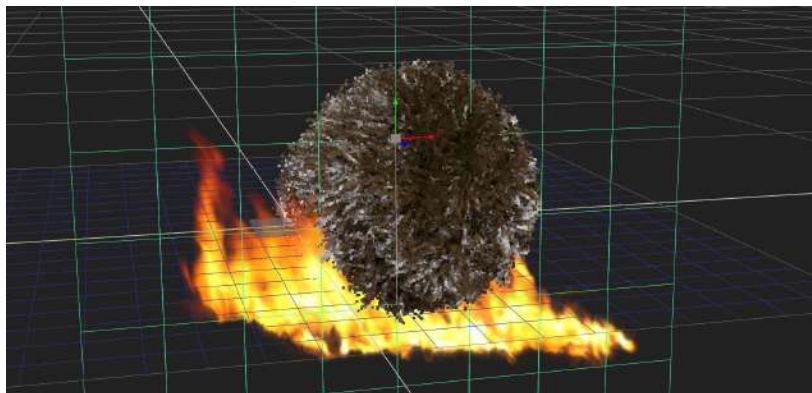


Figure 10 - Card placed inside 3D space of Nuke using point cloud.

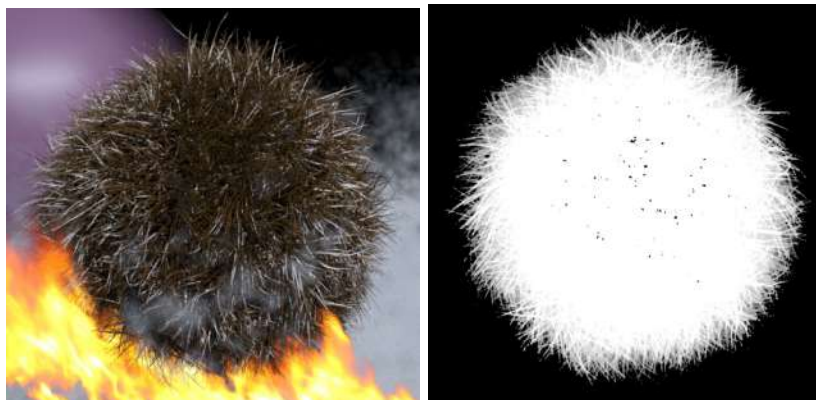


Figure 11 - Comp using graded z-depth Figure 12- Z-depth used as alpha

4.4 File sizes

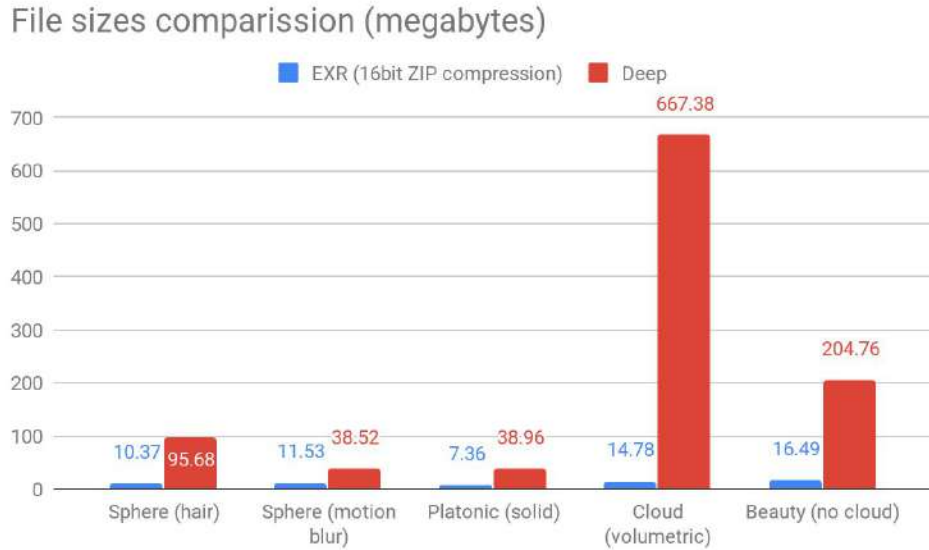


Figure 14 - Data of file sizes

Sphere (hair)	+822%
Sphere (motion blur)	+233%
Platonic (solid)	+428%
Cloud (volumetric)	+4415%
Beauty (no cloud)	+1141%

Figure 15 - File size increase in percent

As we can see in fig. 14 and 15, there is a significant increase in file size resulting from the use of deep with the lowest being 233% increase. The volumetric elements has a much higher percentage due to it sampling more in depth unlike opaque surfaces.

5 Discussion

Deep seems to be most useful when it comes to semi-transparent elements and volumetrics. This was to be expected since the deep workflow rules out the need for holdouts when rendering. But since each pixels has much more depth, these also require a lot more storage. The main question the artist has to ask themselves if they are rendering for volumes/semi-transparent surfaces:

Is the cost of 1000-4500% file size increase worth it?

It should be noted that the file size difference between the objects is not directly comparable since they are differently sized in the render, bigger object require more pixels which require more deep data.

One major advantage of deep is that it allows the artist to render, and combine elements from completely different renderers and maintaining perfect occlusion due to the fact that the deep data is not filtered and interpreted such as with the regular z-depth.

Another major advantage is the deep transform node which allows you to move and scale your element in xyz which allows for reframing and flexibility. Obviously no rotation since that would require more pixel data. Z-depth pass is scaled and normalized which allows for it to cover the whole depth, but therefore does not indicate the actual scale of the scene which is problematic when trying to use it for placing objects in the actual camera space.

There are some misconceptions about deep, what it can and can't do. Deleting an opaque surface will not reveal the surface behind it in a beauty render. It will only leave a hole.

The still frame with motion blur and fur, which had semi-transparent edges were nearly impossible to use with the traditional z-depth operations.

6 Conclusion

Deep compositing is an exciting new prospect of VFX and will only continue to develop as computing gets faster and storage more affordable. The deciding factors discussed in this paper might not be as relevant in the future due to technological improvements.

But here are some guidelines to help decide whether to render using deep or not today based on visual quality and workflow efficiency:

6.1 Recommendation to establish whether to use deep data:

Don't render with deep:

- If you are only dealing with opaque objects with no semi-transparent edges and you are not planning on adding intersecting cards in comp that need perfect edge quality.
- If the 3D won't change and require re-rendering later.
- If the FX element is not partially occluding your objects and/or interacting with the FX. That would require re-rendering anyways.
- If small file sizes are important to you.

Consider rendering with deep

- If the rendered objects have semi-transparent edges such as fur and motion blur. Most comp operations work using z-depth but not to the same quality as deep nodes.
- If you are unsure whether the elements will be changed and require re-rendering..

Render deep

- If you want a flexible comp where both the object and FX can be re-rendered separately.
- If you are adding many cards at different depths, with different occlusions and need good edge quality.
- If you are rendering elements from different renderers that require correct holdouts.

7 References

7.1 Publications

Brinkmann, R. (2008). *The art and science of digital compositing*. 2nd ed. Amsterdam: Morgan Kaufmann Publishers/Elsevier,

Wright, S. (2001). *Digital compositing for film and video*. 3rd ed. Burlington: Focal Press

7.2 Electronic sources

Seymour, M (2014). *The art of deep compositing* [online] Available at: <https://www.fxguide.com/featured/the-art-of-deep-compositing/> [Accessed 16 May 2019]

Hanika, J, Hillman, P, Hill, M and Fascione, L (2012) *Camera Space Volumetric Shadows* [online] Available at: https://jo.dreggn.org/home/2012_camera_space.pdf [Accessed 16 May 2019]

Lokovic, T, Veach, E (2000) *Deep Shadow Maps* [online] Available at: <https://graphics.pixar.com/library/DeepShadows/paper.pdf> [Accessed 16 May 2019]