Compositing

Digital compositing, as we are going to be discussing it, deals with the process of integrating images from multiple sources into a single, seamless whole. While many of these techniques apply to still images, we will be looking at tools and methods that are useful and reasonable for large **sequences** of images as well. Before we go any further, let ' s come up with a specifi c defi nition for what this book is all about.

Digital Compositing: The digitally manipulated combination of at least two source images to produce an integrated result.

By far the most diffi cult part of this digital comp ositing process is producing the *integrated* result—an image that doesn ' t betray that its creation was owed to multiple source elements. In particular, we are usually attempting to produce (sequences of) images that could have been believably photographed without the use of any postprocessing. Colloquially, they should look " real. " Even if the elements in the scene are obviously *not* real (huge talking insects standing atop a giant peach, for example), one must be able to believe that everything in the scene was photographed at the same time, by the same camera.

Although so far we ' ve mentioned only a few big-budget examples of digital compositing being put to use, in reality you ' ll fi nd digital compositing at work just about anywhere you look in today ' s world of media and advertising. Pull a magazine off the shelf and in all likelihood you will fi nd that most of the art and graphics have been put together using some sort of computerized paint or image-editing program. Television commercials are more likely than not to have been composited together from multiple sources. Yet whether the digital compositing process is used for a business presentation or for feature-fi lm special effects, the techniques and tools employed all follow the same basic principles.

This book is intended to be an overview that will be useful to anyone who uses digital compositing. However, you will probably notice that many topics, descriptions, and examples seem to approach the subject from the perspective of someone working to produce visual effects for feature fi lms. This emphasis is not only due to the author ' s experience in this fi eld, but also because feature-fi lm work tends to push the limits of the process in terms of techniques, technology, and budgets. Consequently, it is an ideal framework to use in providing an overview of the subject. Additionally, it allows for the use of examples and sample images that are already familiar to most readers.

The title of this book, *The Art and Science of Digital Compositing* , was coined to stress the fact that true mastery of digital compositing includes both technical and artistic skills. As with any art form, the artist must certainly have a good amount of technical profi ciency with the tools that will be used. These tools could potentially include any number of different hardware and software compositing systems. But one should also become knowledgeable about the science of the entire compositing process, not just specifi c tools. This would include everything from an understanding of the way that visual data is represented in a digital format to knowledge of how a camera reacts to light and color. Please remember, though, that all these technical considerations are simply factors to weigh when confronted with the question of " Does it look right? " The answer will ultimately be a subjective judgment, and a good compositor who is able to consistently make the decisions that produce quality images will always be in high demand.

The combination of multiple sources to produce a new image is certainly nothing new, and was being done long before computers entered the picture (pardon the pun). Although this book is about digital compositing, let ' s spend a moment looking at the foundations upon which digital compositing is built.

Projection In Nuke (Converting 2D Image Into 3D)

In today's production it's necessary to work fast and efficiently and in demand are often fantastic environments which contain originally non present details or are created fully virtually. If it's just a small projection of advertisement on a wall or a full vast environment, it's solvable with projection cameras clearly and efficiently. In this article I am going to describe two possible workflows where the first one will be turning 2D still image into live animated image and the second one will show how to clamp projection into already existing footage. First workflow covers the basic steps and logic of the technique with free camera set, while the second one describes how to track the existing camera and create projection fitting it in Nuke. 1. Projection with Virtual camera As the name indicates this workflow is used when the camera is created virtually. Which means, such camera exists in 3D program such as Maya and is exportable into formats supported by Nuke, or is created directly in the Nuke. This poses quite the advantage as the project remains precise and still modular. On the other hand this workflow is usable only with full CG footages. In this kind of scenario we will be working with matte painting as our background, while our foreground will be detailed geometry with full render ready texturing and shading. If a foreground is present it will be rendered out with the render camera. Our job in Nuke is to fit the background matte painting to foreground. Now our advantage in this case is, that we have the camera file, so we will not need to track our camera. Setting up in Nuke First step after opening Nuke and setting up a new project would be to create a Read node and loading the background of choice. In my case it was a matte painting I found and downloaded online. I prefer to join Read nodes with Reformat and Transform nodes at the beginning. This way I know the range of my work canvas and do not have to worry about excess field of the picture. In Reformat node I set the render resolution of HD 1080 (1920 x 1080). The Transfer node is present because, during reformatting the picture loses its original side relation. So to keep the composition, with point of interest in the middle, it was necessary to move the picture. Also, the Transfer node was used to zoom in the picture needed for purpose of rotation. While a picture is still we see only what is visible from the exact angle. However even the slightest rotation means a change in this angle and revelation of spaces those were originally hidden. To avoid this there has to be a content painted in. However, there will be no need to paint extra spaces on the sides, while we zoom, if some of the picture from the sides is sacrificed. These "hidden" parts will be visible during the turn while camera changes the angle. All further nodes are going to be added between the Read and Reformat nodes. In this step I won't be creating camera, even though it can be helpful for better imagining and knowing which parts of picture will be visible in the end. For next few steps I found it sufficient to simply move around 3D space in Nuke. First layer which is foreground does not need editing so for now this node should stay untouched. The second layer however will need to be edited. This layer is going to cover the area of field where the temple stands, the most dominant mountain in centre, hills to the left and to the right till where the atmospheric fog is present and the forward most wall with waterfalls. The disturbing elements in this case are background, the temple and hills in foreground. So they will have to be painted out. So now comes the step where the "hidden spaces" are going to get painted in. I will

be using Clone tool. I'd recommend to set the view mode to Shift+F3 mode to have the canvas nice and big. I divided the matte painting into four layers. The first layer is foreground, second is midground without temple, third is only the temple and the fourth is background. To divide the layers from one another there will be four Roto paint nodes between the Read and Reformat nodes. pic colour dif layers + tree. Without scaling the sides would reveal black space due to camera turn Layering the picture The problem of revealing spaces that weren't visible before animating camera does not get fully solved by zooming in the background. It solved only the frame edges of picture, but to explain on example of picture that I am using, the hills in foreground will reveal pieces of sky and midground greens during the turn and also the temple in midground will reveal both the sky and mountains in background. Solution to this issue would be dividing the picture into layers. Without doing this step it would cause the projection to project itself repeatedly creating weirdly looking doubles of objects. Projections in 3D work very much as a real life movie projector. One uses a camera to project a 2D image onto geometry in a 3D space. (Learn.foundry.com, n.d.) What makes this method suitable for the task of maintaining quality in relation to speed is that this method not require the model receiving the projection to be UV-mapped, textured or prepared in any other way. Projections also only needs a single frame to be projected throughout the entire sequence, meaning a lot of time will be saved rendering from 3D. Cards Using cards (2-dimensional plane) with projections is a common and effective technique. By placing cards at different distance from the camera one can get a parallax effect when viewed from a moving camera which is desired for a realistic integration. Cards does not allow for multiple levels of depth to the projection, but takes no time to create as it exists as a single node inside Nuke. Example of projection with cards Geometry Compared to projections on cards, projections onto geometry allows the projected image to wrap around the model. This means the projection gets different levels of parallax depending on the geometry's relative angle to the camera as well as the possibilities of giving the projection larger variation in depth. This makes it more suitable for more complex images and images that contains large varieties in depth, but compared to cards it has to be modelled separately for each projection which requires more time. 4 Figure 3 – Example of projections with geometry. Deep format Open EXR is a file format developed and released as open source 2003 by Industrial Light & Magic. The format supports 16 - and 32-bit floating point values for high dynamic range images, and with the release of Open EXR v2 in 2013 "deep-data" became supported. Deep data refers to storing multiple values for different depths for each pixel, making combining elements using deep as simple as merging them together, no masking or rotoscoping needed. Compared to projections it does not need any geometry to be placed correctly in 3D space. The depth data is already stored in each pixel and can be used to precisely extract points in 3D space, making it possibly faster and more exact than what projections would allow for. Model A 3-dimensional shape made up by multiple points, vertices, in 3D space. The vertices are connected by lines and connected lines form faces. All of the faces together form a polygonal mesh. Texture An image mapped to a 3D model for applying surface attributes such as colours, surface detail, depth, smoothness etc. RGBA RGB is short for red, green and blue, the different colour channels in each pixel of an image. A is short for the Alpha channel, representing the level of transparency of the pixel. Lower alpha values make the pixel more transparent and higher less transparent.

Rendering

Rendering is the process of transforming the part of 3D space that is visible through a 3D camera into pixels in 2D, that together make up an image. Scanline render A rendering algorithm for determining visible surfaces in image space on a line by line basis working from top to bottom. Also the node in Nuke for rendering objects from Nukes' 3D environment. Passes Passes is parts of a rendered image that when composited together make up the final image. Base colour, reflection, refraction and shadows is some of the most commonly used passes. It is also possible to save information in passes that is not for combining to make up the final image, but information such as depth, object-id, motion vectors and many more can be stored in an image as passes. Passes allows for more control of specific parts of the rendered image, as well as through passes such as depth and motion vectors being able to create accurate depth and motion blur in comp. Samples and noise When rendering the level of samples represents how many times the colour of each pixel is traced from the camera by shooting out a ray and see what colour it hits. If the samples are set to one, the colour of the pixels will be determined after having been traced by a ray from the camera one time. The problem is low samples results in noise in the image, since the information is not enough to accurately determine the colour of the pixel. The higher the samples, the more times each pixel is traced and can be assigned a more accurate colour. The lower the samples, the less times the pixel is traced and will be assigned a less accurate colour, resulting in noise. Shadow catcher A material that, when rendered, is visible only where shadows hits the model the material is assigned to. A shadow catcher is necessary in the coming experiments as we want to be able to place the shadow on top of the plate without showing the rest of the model the shadow catcher is assigned to. Motion blur An artefact appearing in the camera due to motion during the time the shutter is open. The artefacts make streaks in the image from where the motion starts as the shutter opens to where the motion ends as the shutter closes. The wider the shutter angle (the longer the shutter is open) the more motion blur in the image. To make 3D look realistic it has to be integrated with the plate, meaning we have to match arte facts such as motion blur for a more realistic result.

Camera Tracking

Camera tracking is a method of analysing a video sequence to create an animated 3D camera that follows the motion of the video; a match-moved camera. Point cloud A collection of non-connected points in 3D space. One can extract a point cloud from a plate by using the plate and a previously match-moved camera. The points created will represent the environment of the recorded plate in 3D space. The point cloud can be used as a guide for placing objects at the right position in 3D space. Method Three different methods were used for the experiment, those being projections on cards, projections on geometry and the extracted points in 3D from deep images. Each alternative comp method, as well as the reference, was timed to see how efficient they are relative to each other. The time spent starts with the render in Maya and ended when the final image was rendered in Nuke. The 3D was rendered at a resolution of 2560 by 1440 pixels unless anything else is stated. The results of the compositing methods were compared to a reference render with the "difference" merge operation. The result of the operation showed how much the pixels from input A differ from input B (abs(A-B), absolute value of A-B). When a pixel in Nuke is selected its RGB value is displayed, by selecting the entire image the average RGB values for the entire image is displayed. The average values were recorded every 25 frames for a total of 8

frames. All of the RGB values were added together and divided by 24 (3 * 8 = 24, 3 being the RGB channels and 8 being the number of frames). The result was an average value of how much each colour channel differs along the full sequence. (The reference render was the perfect goal in this experiment and any differences compared to it were seen as a decrease in quality). The average difference values per sequence together with the time spent would reveal the best method for keeping quality in relation to time. Difference of reference compared to cards. Preparations Before the actual experiment could take place a number of things was prepared. A plate was needed to have something to integrate the CG onto. A Blackmagic Production 4K together with a 24mm lens was used to record a plate, frame rate and shutter angle set to 25 fps and 180 degrees. The files were saved as .MOV, which made the quality lower as it uses lossy compression, though made it faster to work with as the 4K resolution made the file heavy as it was. The plate was 3D tracked to create a match-moved camera for making the CG stick to the plate. Plate Assets for rendering were picked, a rock and a tree, from Quixel Megascans' free asset library. They are well suited for this experiment as the models and textures are highly detailed and thus deviations from the original render would be easier to determine rather than if a simpler, less detailed asset was used. To achieve correct shadows a ground shadow catcher was needed. In order to make the shadows follow the ground in a realistic way a ground mesh was created in Nuke. A point cloud was created by analysing the plate sequence, and from this point cloud a mesh could be generated. The mesh was a lot larger than needed to catch the shadows from the tree and the rock, which would result in longer render times. The mesh was exported to Maya where it was reduced to a smaller size covering the area that would be nearest the models in the 3D scene, reducing the area needed to be rendered and by doing so saving render time. Reference sequence The sequence of 176 frames was rendered in Maya and imported into the Nuke scene. Since the sequence was rendered through a tracked camera there was nothing needed in comp for the render to stick seamlessly to the plate. The CG was rendered in Nuke together with the plate, completing the reference sequence. The render settings in Maya were adjusted and optimized to be as fast as possible and noise free, not to keep the sequence render at an unfair disadvantage as the other methods was going to reach for the highest quality at the fastest speed possible as well. Method 1 The first comp method was projections on cards. First the tree, rock and ground shadows were rendered separately in order for the projections to be placed at different distances from the camera and getting parallax as the camera moves, adding to its realism. A new camera was created in Maya that fit all of the CG in frame, by doing so it was possible to project all of the images with the same camera. For the shadows to stay under the objects correctly the shadow pass was divided for the rock and tree and combined separately. The images were projected onto individual cards, the card with the rock being placed in front of the tree. To place the cards in the correct location in 3D-space the point cloud used to generate the ground mesh was used as a guide. The cards were piped into a scene node and would be rendered with a through the previously tracked camera through a scanline render. Motion blur was added after the scanline render as the reference sequence was rendered with motion blur as well. Lastly the card projections were rendered together with the plate. created with the quad draw tool with the existing rock model as a live surface. Rather than manually drawing faces on the tree one by one, since it was a much more complex model, the tree projection model was done by simply using the existing model for the tree and use Maya's reduce option to get a much lower resolution that Nuke could handle. The tree surface was extruded outwards somewhat, increasing its thickness, to make sure no jagged edges would appear where the projection would go outside of the model. The shadow catcher geometry used in Maya was could be used for the

shadow projections in Nuke as well, thus a new model did not have to be made. It was important for all projection geometry to be placed at the same location in 3D space as the original assets, if they are not the projection and render would not match up. All images were projected onto their corresponding piece of geometry, added to a scene node with the tracked camera and a scanline render, had motion blur applied and was rendered together with the plate.

Croma Removing

Keying is one of those fundamental compositing skills. You can't composite anything until you have mattes pulled for the elements you want to layer together. It's nice to say you could just push a button to complete this task, but as you probably know, one keying operation seldom produces an acceptable matte. Image quality, lighting conditions, subject motion, colours—even camera moves—affect the steps required to get a clean matte for your composite. So how do you get a clean matte in Nuke? The best approach is to understand the strengths of each keying tool and combine them as needed. This tutorial shows how to pull keys in Nuke and how to layer the results with channel operations, merge nodes, and rotoshapes. Keying with Primatte The Primatte keyer includes a quick "Auto-Compute" option that evaluates your image and determines a good baseline key. From there, you can easily tweak the settings and generate an acceptable matte. The two examples in this section show how to pull a key with the Auto-Compute option (method 1), and also how to manually sample a colour from the screen background and build your key from there (method 2). To pull a key with Primatte (method 1): 1. In the project file, locate the node tree labeled "Keying with Primatte," and make sure a viewer is attached to the Reformat1 node. 2. Choose Keyer > Primatte to insert the keyer between the foreground image and the viewer. 3. Drag the bg connector from Primatte1 to the Reformat2 node, which supplies the background image for this example. The fg connector should be attached to Reformat1. 4. Move the time slider to frame 50, and click the Auto-Compute button inside the Primatte1 control panel. That's it. You're done... well, nearly done. We need a "free-floating" goldfish, but the reflections in the aquarium glass clearly indicate "captivity." NUKE 466 A garbage matte will easily remove the reflections, and you'll learn how to do that later in the section on rotoscoping. For now, let's keep working with Primatte. As you've seen, Primatte's auto-compute option can quickly pull keys on certain images. However, you should also know how to pull and tweak keys manually. You might, for example, need more control over the transparency of the fins on the goldfish. To pull a key with Primatte (method 2): 1. Continuing from the previous example, open the Primatte1 control panel. 2. Click the "undo" button at the top of the control panel to step back to the previous state of the Primatte1 node. Or, you can also delete the current Primatte1 node and insert a new one. 3. Scroll down through the Primatte options and set the keying operation to Select BG Colour. 4. The current colour chip should display the eyedropper icon. If it doesn't, click on the colour chip to toggle the eyedropper. 5. Hold down the Ctrl+Shift keys (Mac users, hold down Command+Shift) and drag—or scrub—over a portion of the greenscreen in the image displayed in the viewer. NUKE 467 This returns an average colour-pick of the sampled pixels. If you want a colour pick from a single pixel, press Ctrl or Command and click once over the greenscreen. After you pick, you can clear the red square by Ctrl- or Command-clicking again. 6. Press A over the viewer to toggle to the alpha channel display. Looks like the aquarium is not as clean as we thought. Our colour pick gave us a fairly noisy key, so let's clean it up. Now you'll sample a few areas of the image to "push" selected pixels to one of three areas: the transparent matte, the opaque subject,

or the semi-transparent part of the matte. 7. In the Primatte1 control panel, change the keying operation to Clean BG Noise. 8. Press Ctrl+Shift or Command+Shift and drag a small square over the dark area in the lower-right corner of the image. NUKE 468 This second colour sample cleans the background by “pushing” the selected pixels into the transparent area of the matte. You probably need a few more samples to get a better key. 9. Scrub a few small areas in the background, focusing on the grey pixels until the matte improves. The background doesn’t need to be solid black. We’re just trying to get a good separation between our foreground subject and the greenscreen background. 10. Change the keying operation to Clean FG Noise. This time, sample areas of grey pixels inside the goldfish. One or two small samples should be enough. The colour pick pushes the selected pixels to the opaque part of the matte. NUKE 469 You want to keep the grey pixels inside the fins to retain a semi-transparent matte in these areas. If you go too far, you can always press the undo button in the control panel to step back to the previous action. 11. Press A again over the viewer to toggle to all colour channels. Your image should look similar to the example shown below. You may see some detail dropping out from the fins. 12. Change the keying operation to Restore Detail, and scrub over the fins to bring back some of the edge detail. You may get different results than those shown here, depending on the pixel values you sample from the image. Use Restore Detail to push the selected pixels back toward the opaque part of the matte. Use the Make FG Transparent operation to fine-tune the semi-transparent area. You could go back and forth, between cleaning the background and foreground, but this usually produces a matte with “crunchy” edges. The goal is to find the balance between foreground and background that produces an acceptable matte for your subject. Later in this chapter, you’ll use the rotoscoping tools to clean-up this matte and combine this with the image from the next example.