Chapter 10. Rat Cowboy – Rendering, Compositing, and Editing

Welcome to the last chapter of the module. In this chapter, you will learn advance material creation in Cycles, such as how to use a skin shader or how to create a realistic fur. Next, you will learn about passes and how to do a raw render with the different passes. Then, you will receive an introduction to the nodal compositing so that you can enhance your shots. Lastly, we will talk about the Video Sequence Editor in order to edit the final sequence. Let's start!

In this chapter, we will cover the following topics:

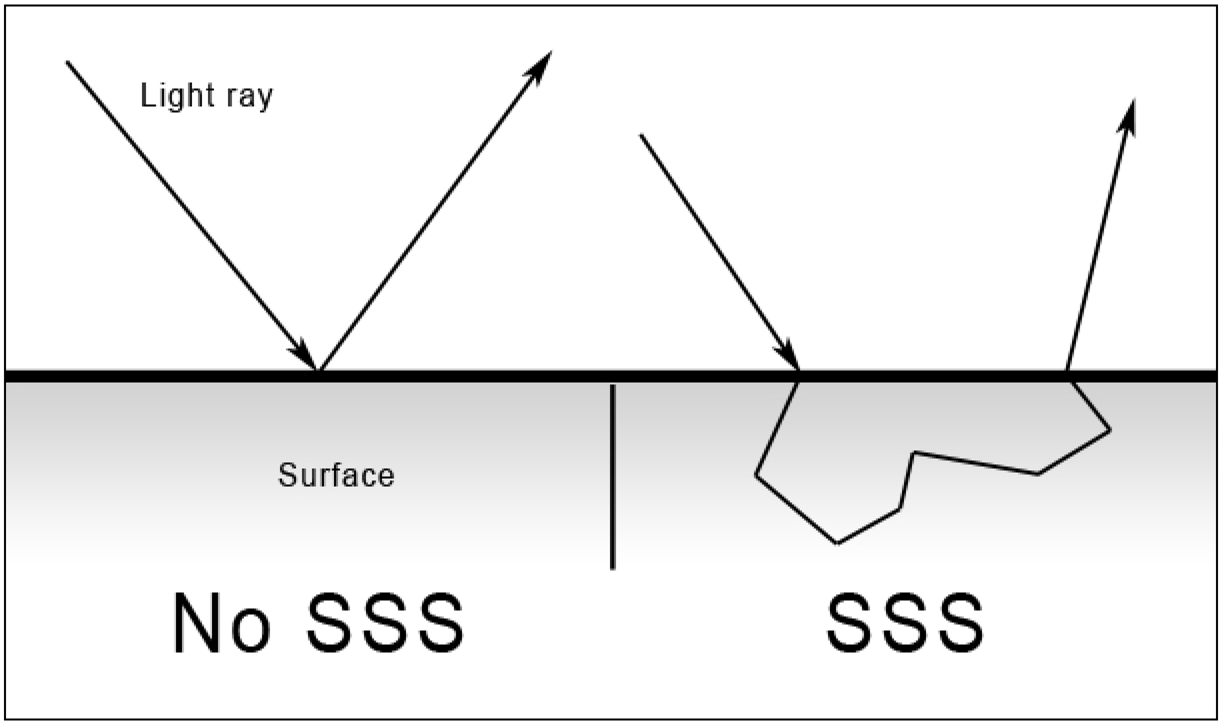
- Creating advance material
- Creating fur particle systems
- Setting up Cycles for an animated scene
- Using passes
- Introducing nodal compositing
- Editing a sequence

Creating advanced materials in Cycles

We already covered material creation with Cycles in the Haunted House project, but now we are going to go further by creating a skin material using subsurface scattering, a complete fur, and an eye material. Let's start!

Skin material with Subsurface Scattering

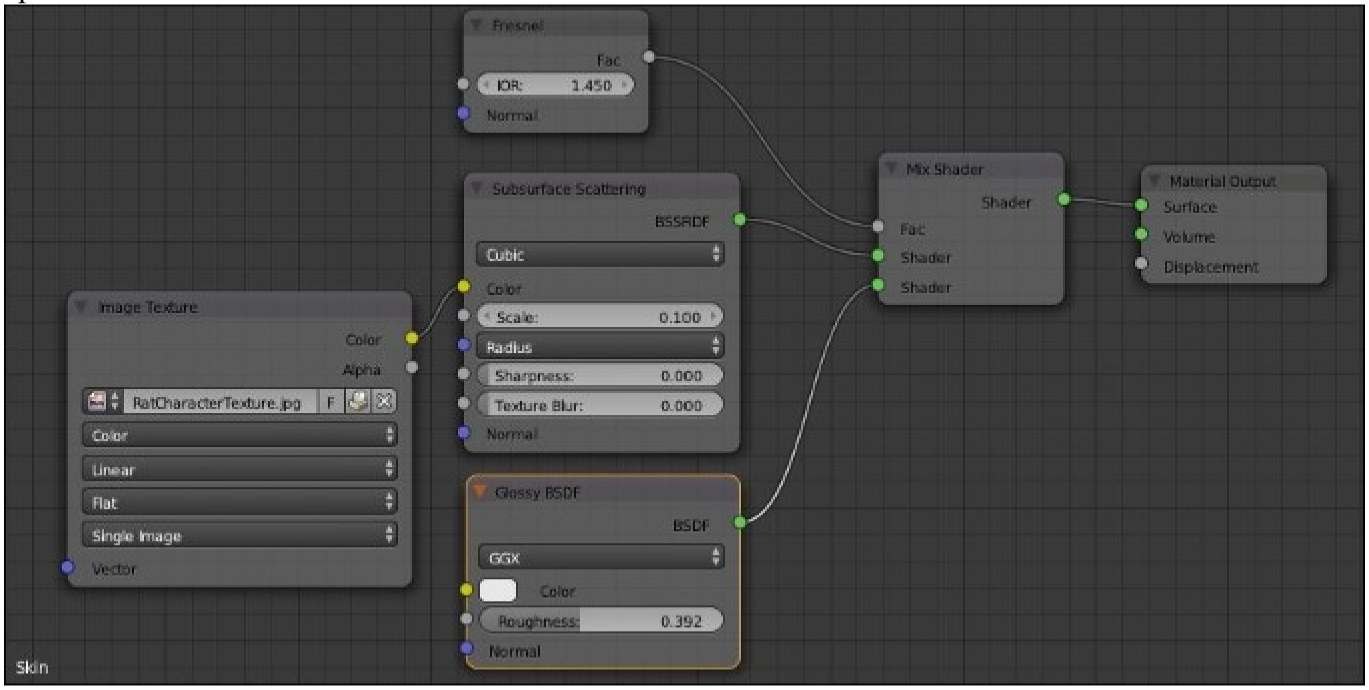
The skin has a very translucent aspect. We can truly see this effect when we pass our hand in front of a lamp or in the thin part of the ear (the helix). So, when creating a skin material, we get this phenomena with a Subsurface Scattering node (usually abbreviated SSS). It is called this because the light rays are scattered through the geometry when intersecting the mesh. This is not the case with a diffuse shader, for instance, as the light rays are simply blocked. SSS often gives a reddish tint to the thin parts where light rays scatter a lot. So let's create the skin material of the rat.



The way light rays react on SSS surfaces

1. We will open the `RatCharacter.blend` file and split our interface so that we have a second 3D view for the real-time renderer and a node editor. Note that the real-time renderer for the SSS shader will only work in the CPU mode or with the GPU in the experimental mode.
2. We will add a new slot in the material tab of the Properties editor and a new material that we name as **Skin**. We will press the **Use node** button in order to work in the node editor.
3. In the node editor, we will remove the default **Diffuse** shader by selecting it and pressing **X**. Then, we can add a **Subsurface Scattering** shader by pressing *Shift + A*.
4. We have some options to tweak for this shader. The first one is the **Scale option,** which corresponds to the amount of SSS that we want. In the case of the rat, we set it to **0.1,** but in order to have the correct value, you will need to perform a test. It's just a matter of placing a light in the back of the character and looking at the thin parts, such as the ears.
5. The next very important setting to tweak is the radius that corresponds to the predominant color that will result to the SSS effect. In many cases, we will let more red than green and blue because of the color of the blood that's under the skin. This is a set of three values that corresponds to R, G, and B, in our case, we set them to **1.0**, **0.7,** and **0.5** respectively.
6. Now, we will plug our texture in the **Color** input. Finally, we can connect the shader to the output.
7. We will now add a reflection to our skin by mixing our SSS shader with a glossy shader. To do so, we append a **Mix Shader** on top of the SSS to the Material output wire.
8. In the second shader input, we can bring a **Glossy BSDF** shader and change the **Roughness** value to **0.392** so that the reflection is less sharp.
9. For the **Fac** input of the **Mix Shader**, we will add a **Fresnel** node. This skin shader will be sufficient for our needs, but note that we can go much deeper in the subject by creating a shader

with multiple maps that corresponds to the different skin parts, such as sub-dermal and epidermal.



Fresnel
Fac
IOR: 1.450
Normal

Subsurface Scattering
BSSRDF
Cubic
Color
Scale: 0.100
Radius
Sharpness: 0.000
Texture Blur: 0.000
Normal

Image Texture
Color
Alpha
RatCharacterTexture.jpg
Color
Linear
Flat
Single Image
Vector

Glossy BSDF
BSDF
GGX
Color
Roughness: 0.392
Normal

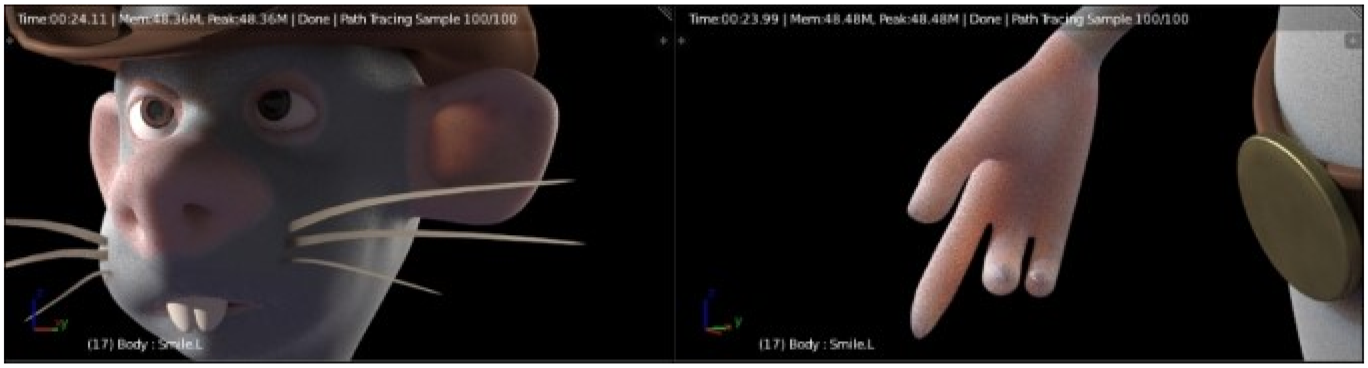
Mix Shader
Shader
Fac
Shader
Shader

Material Output
Surface
Volume
Displacement

Skin

The skin material nodes

The SSS effect on the ears to the right and on the hand to the left in viewport rendered mode will look like the following:



```
Time:00:24.11 | Mem:48.36M, Peak:48.36M | Done | Path Tracing Sample 100/100
(17) Body : Smile.L
```

```
Time:00:23.99 | Mem:48.48M, Peak:48.48M | Done | Path Tracing Sample 100/100
(17) Body : Smile.L
```

Eye material

We are now going to create a less difficult material, that is, the one of the eye corneas. This will be like a glass material, but we will optimize it a little bit because the default glass shader is so physically accurate that it also casts shadows of the glass itself. These take a long time to render and are rarely visible. In order to apply our material, we will model a simple cornea in the eye mesh itself. The front part of the cornea is extruded a little bit. This allows us to catch the reflections rays better:

1. We will first need to add another material slot and a material that we rename as **Cornea**. As you can see, we already have three slots that correspond to the white part and the pupil of the eyes. Feel free to replace it with one material with a texture.
2. Next, we can select the cornea piece of the mesh in the **Edit Mode** with the **L** key and press the **Apply** button in order to apply the cornea material on this part of the mesh.
3. In the node editor, we will replace the default Diffuse shader with a **Glass BSDF** shader.
4. We will now mix the **Glass BSDF** shader with a **Transparent BSDF** shader. A transparent shader simply lets every ray to pass through.
5. Now, in the **Fac** input of the **Mix** Shader, we will plug a **Light Path** node (press *Shift* + *A* and select **Input**) with the **Is Shadow Ray** output. This will tell the render engine to use the transparent shader for the incoming shadow rays and the glass shader for the others. At this point, we should have a nice reflecting eye.

The fur of the rat

Now, let's dive into a complex section about fur creation. In order to create a convincing fur, we will have to create a complex material, have a perfect hair particle combing, and correct lighting. If one of these three parameters is sloppy, it won't look great. Let's start with the particle systems:

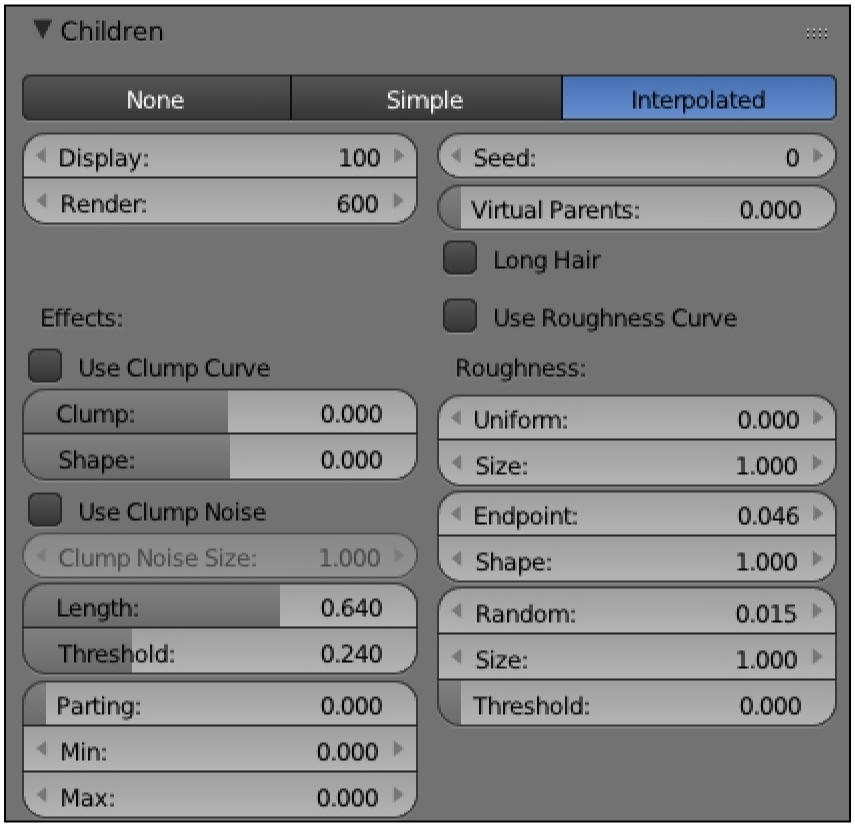
1. In order to add more realism, we are going to create three particle systems. Let's first select the character in the **Edit Mode** and add the main particle system in the Particle tab of the Properties editor. We will name both the system and its settings **Basic_FUR**.
2. We will change the system's type from **Emitter** to **Hair**. In the **Emission** subpanel, we will change the **Number** to **500**, which correspond to the guiding hairs. We can also change the hair length to **0.140**.
3. In the **Children** section, we will choose the **Interpolated** method. The number of children that will follow the guiding hairs is too low. We will change the **Render** option to **600**, so each guide will have 600 children. We can also change the display option to **100** to preview the result in the viewport.
4. In the **Children** subpanel, we can change the **Length** setting to **0.640** and the **Threshold** to **0.240**. This will add some randomness to the length of the children.
5. Then, in the **Roughness** section, we can change the **Endpoint** value to **0.046** and the **Random** to **0.015**. **Endpoint** will spread the tips of each hair strand.
6. Now, we will create a Vertex group that will determine where the fur will be located on the rat. In the **Object data** tab of the Properties editor, in the **Vertex Groups** section, we will start by locking all our skinning groups by selecting the black arrow button and choosing the **Lock All** option. This will ensure that we don't change them inadvertently. We can now add a new group with the + button and name it **Fur**. In the **Edit Mode**, we can select the hands, nose, ears, tail, mouth, and eye contour. We invert the selection with *Ctrl* + *I* and press the **Assign** button with a weight of **1.0**.
7. Back in **Particle** tab, in the **Vertex Groups** section, we can set the **Density** field to our new vertex group. We should now only have hairs on the needed parts.
8. In order to improve our system, we will create a new vertex group named **Fur_Length** that will affect the length of the hair on certain parts. To create the group, we can duplicate the previous **Fur** group with the corresponding option in the black arrow drop-down menu. We can then use

the weight paint tools in order to subtract weight from different parts. In our case, the head is green and the arms and the legs are orange and blue under the belt.

9. In the **Vertex Groups** section of the **Particle Settings** tab, we can change the **Length** field to this new group.

Now, we need to comb our particles as follows:

1. To do so, we can use the **Particle Mode** (located in the same drop-down menu of the **Object Mode** or **Edit Mode**). By pressing *T*, we open the **Brush** panel where we can use the **Comb** brush to comb the character's hair.
2. We will use the same shortcuts as the sculpt mode for the brush settings. The **Add** brush is nice in order to add new strands when you find some gaps. When using this brush, it's best to check the **Interpolate** option so that it smoothly blends with the others.
3. In the header of the 3D view, you have three new buttons (to the right of the layers) that allow you to select the particle in different ways. With the **Path** mode (the first one), you can only control them with brushes, while with the **Point** mode (the middle one), you have access to each point of the hair, and with the **Tip** mode (the last one), you only control the tip. With the last two modes, you can grab and rotate your character's hair with **G** and **R**. In the left panel, we can also activate the **Children** option in order to see the children.
4. Now let's add another particle system and name it **Random_FUR**. We can then copy the settings of the first one by choosing it in the **Settings** field and pressing the 2 button to make a unique copy of it. Now, we can safely change the setting without affecting the other system. We can click on the **Advanced** option.
5. We will start by changing the amount of guiding hairs to 50 and their length to **0.1**.
6. In the **Emit from** section, we will choose **Verts** and check the **Random** option.
7. In the **Physics** subpanel, we can change the **Brownian** value to **0.090** to add a little bit of randomness.
8. In the children section, we will change the **Render** and **Display** sliders to **50**.
9. We will then change the **Length** to **0.288** and the **Threshold** to **0.28**.
10. We will then ensure that the **Density** field still contains the **Fur** group, but we will remove the **Length** field.



▼ Children

None | Simple | Interpolated

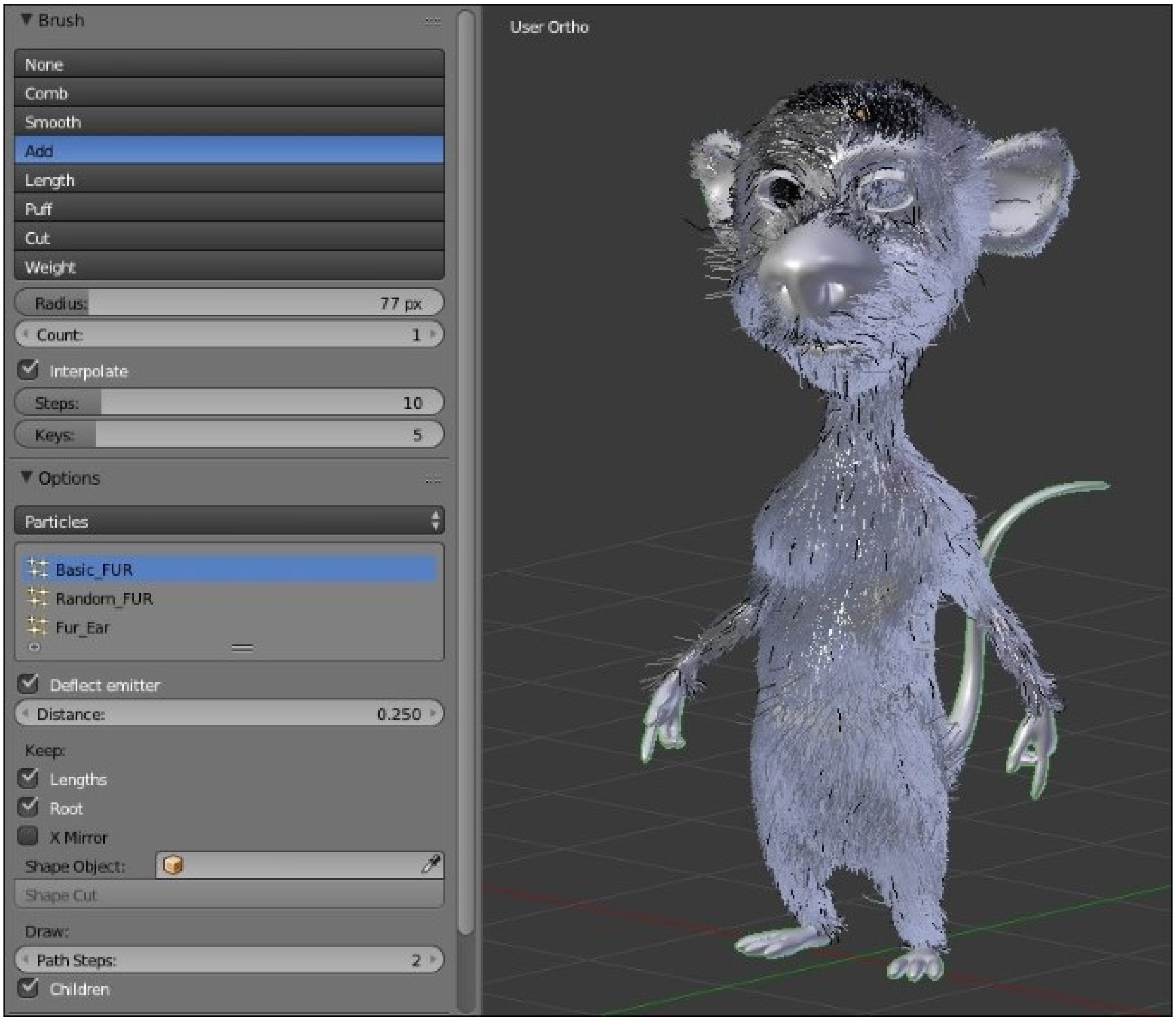
Display: 100
Render: 600
Seed: 0
Virtual Parents: 0.000
Long Hair
Use Roughness Curve

Effects:
Use Clump Curve
Clump: 0.000
Shape: 0.000
Use Clump Noise
Clump Noise Size: 1.000
Length: 0.640
Threshold: 0.240
Parting: 0.000
Min: 0.000
Max: 0.000

Roughness:
Uniform: 0.000
Size: 1.000
Endpoint: 0.046
Shape: 1.000
Random: 0.015
Size: 1.000
Threshold: 0.000

The children settings of the Basic_Fur system.

11. Just as we did for the first two systems, we will add a new system for the fur of the ears. This is very subtle. It has a short hair length and a vertex group for the **Density** field. It also has only 20 children.



Radius: 77 px

Count: 1

✓ Interpolate

Steps: 10

Keys: 5

✓ Deflect emitter

Distance: 0.250

Keep:

✓ Lengths

✓ Root

X Mirror

Shape Object:

Draw:

Path Steps: 2

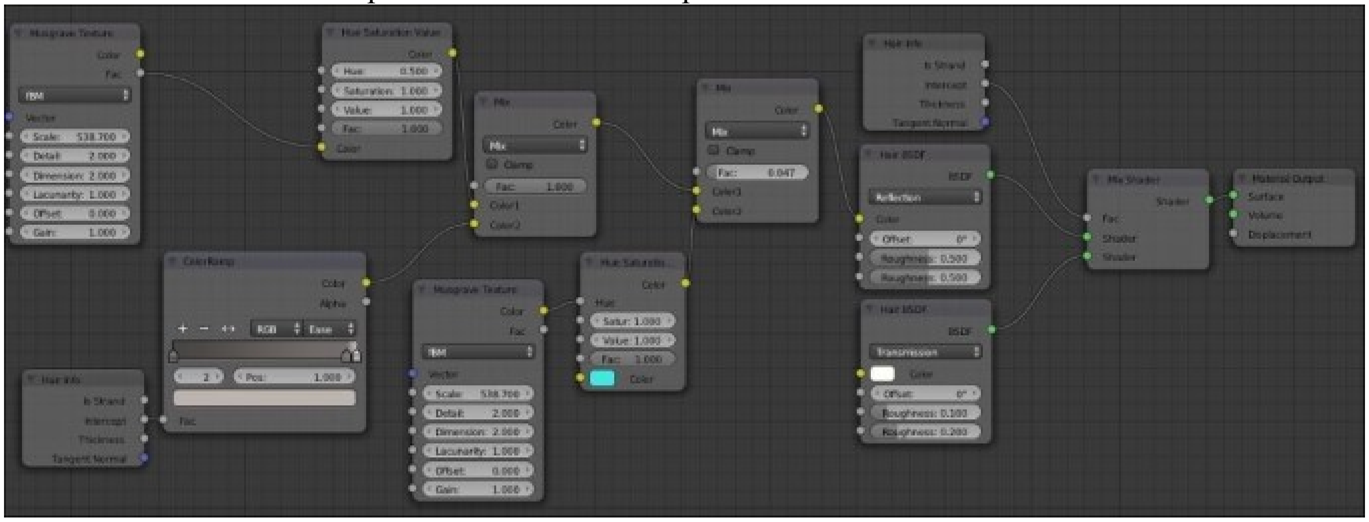
✓ Children

The Particle Edit Mode

You are now done with the particle systems. It's now time to create the materials and change the strand thickness and shape. This will be done as follows:

1. Let's add a new slot in the material tab and a new material named **Fur**.
2. In the node editor, we will delete the default **Diffuse** shader and add two **Hair BSDF** shaders. The first one will be of the **Reflection** type with **RoughnessU** and **RoughnessV** set to **0.500**, and the second will be of the **Transmission** type with **RoughnessU** to **0.1** and **RoughnessV** to **0.2**. We will mix them with **Mix Shader**. The **Fac** input will be plugged with the **intercept** output of a **Hair info** node.

3. We will then add **Musgrave Texture** node with a scale of **538**. We will add a **HueSaturationValue** node with the **Fac** output of the texture connected to the **color** input. We will then mix the result with a **MixRGB** node and **ColorRamp**. The **Fac** input of the color ramp is fed with the **intercept** output of a Hair info node. This will add a gradient map on each strand.
4. We will then plug the output of the **MixRGB** node into another **MixRGB** node. The second input of this node will be fed with a **HueSaturationValue** node. The **Hue** input will receive the color output of a **Musgrave** texture node. This will add color variety to the strand. But in order to lessen the coloring effect, we will change the **Fac** of the **MixRGB** node to **0.047**.
5. Finally, we can plug the result of the last **MixRGB** node in the **color** input of the first **Hair BSDF** node. Our hair material is now completed!
6. We now have to change the **Render** settings of each particle system. In the **Render** subpanel of the particle settings of each system, we will choose our fur material and activate the **B-Spline** option with a value of **4**. This will smoothen the render of the hairs.
7. Then, for the **Basic_Fur** system in the **Cycles Hair Settings** subpanel, we will change the root to **0.5** with a scaling of **0.01** and a shape of **-0.09**. We will use the same settings for the **Random_Fur,** except for the root that we set to **0.15**. Again, we will use the same settings for the **Ear_Fur** system, except for the root that we set to **0.05**.
8. We can now add temporary lights in our `RatCharacter.blend` file in order to do test renders. We will set our samples to 300 and render a preview of our rat.



The fur material in the Node editor

The image with a preview render with low render settings is a follows:

Now you have the knowledge to create even more complex materials with Cycles! We are now going to show you how to render the first shot of the sequence, and what's nice with the link is that we will see our fur and materials in each shot file.

The Raw rendering phase

Previously, we have seen how to render an image in Cycles. It is quite different for an animation. It's best to first do a raw render of the shots with the following settings:

1. We will start by adjusting the device. If you have a good graphics card, remember to check the **GPU** device option.
2. Let's now adjust the samples in the **Render** panel (**Properties** | **Render** | **Sampling**). The skin needs enough samples to reduce a noisy effect. 100 or 150 samples are enough to have an idea, but consider setting a higher value for the final render.
3. Still in the **Sampling** tab, we will put **1.00** to **Clamp Direct** and **1.00** to **Clamp Indirect**. It allows us to reduce the noise, but you may lose a little bit of the bright colors.
4. We should remember to check the **Cache BVH** and the **Static BVH** options in the **Performance** tab. It allows us to optimize the render time.
5. You can make a test by just rendering a frame (*F12*). Pay attention to the time it takes to complete the rendering process for only one frame. Thus, you can calculate the time needed for a shot.
6. In the **Passes** tab (**Properties** | **Render Layers** | **Passes**), we will verify whether **Combined** and **Z** passes are checked.

   ### Note

   Passes in Cycles

   Passes are a decomposition of the 3D image rendered in Cycles. We can also render passes with Blender Internal but in a different way. Once the rendering calculation is over, we can combine these passes and combine each pass together to create the final image with directly compositing in Blender or another software. The passes allow us to get a control to considerably improve an image and make changes even after the image is rendered. So, it gives more fine-tuning opportunities and saves us a lot of render time.

   If you want to explore all the passes of Cycles, visit this link:

   http://wiki.blender.org/index.php/Doc:UK/2.6/Manual/Render/Cycles/Passes

7. Now that the image quality parameters are set, we must choose an output format of our animation. In the **Output** tab, we will choose the **OpenEXR MultiLayer** format.

   This format has the advantage of containing all active passes with a lossless compression. The passes are a decomposition of the rendered picture (Diffuse, Shadows, Ambient Occlusion, and so on). In our case, we are going to save some time by only rendering the combined and the Z passes. The combined pass corresponds to the final image with all the different passes already combined, and the Z pass gives us a black and white image corresponding to the depth of the scene.
8. In the **Output** tab, we must choose an **Output** path. We will write the following address: `//Render\01\`. We will press *Enter* to validate the address. The `//` symbols create a file just next to the blend file.

9. It is a raw render, so we uncheck the **Compositing** option in the **Post Processing** tab (**Properties|Processing**).

   ### Note

   OpenEXR

   This is a high dynamic-range (HDR) image file format created and used for special effects in the VFX industry. It is now a standard format supported by most of 3D and compositing softwares. The OpenEXR Multilayer format is a variation. It can hold unlimited layers and passes.

   If you want more information about OpenEXR in Blender, visit this link:

   http://blender.org/manual/data_system/files/image_formats.html#openexr

You are now ready to render the animation. You can press the **Animation** button to start rendering. We must repeat this process for each shot.

Enhance a picture with compositing

Now that we have a raw render, it is time to learn how to improve it using the compositing tools of Blender.

Introduction to nodal compositing

Blender is a complete tool that also allows compositing. This is the ability to edit an image or a sequence after the rendering phase. You probably have already tried compositing, maybe unknowingly. For example, Adobe Photoshop© is a software that allows us to composite a single image. Unlike Adobe Photoshop©, Blender uses a nodal system that provides a great flexibility. We can make changes at any point without the loss of information. Let's try this:

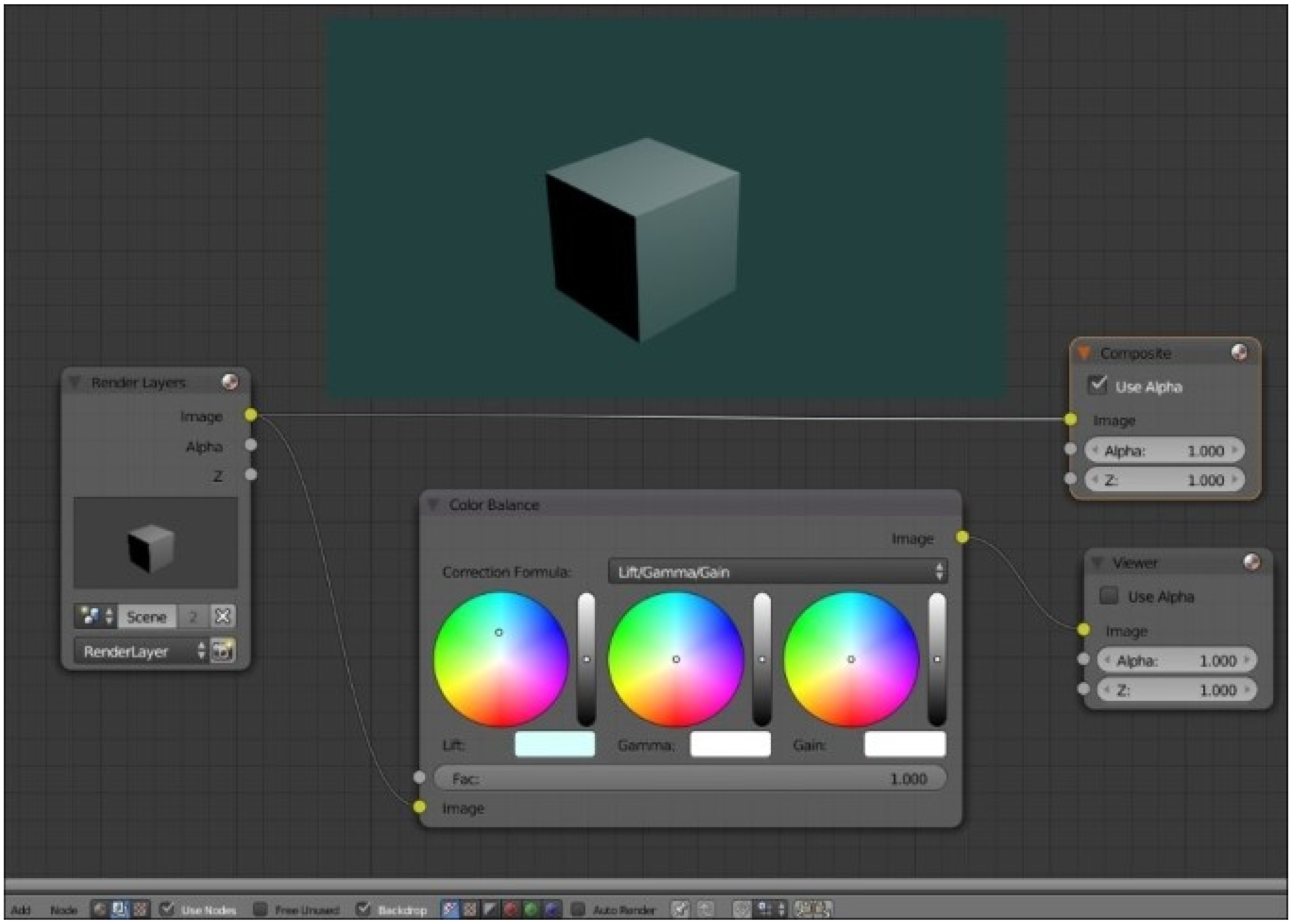
1. For a first approach of nodal compositing, let's open a new Blender scene.
2. In order to access the compositing mode, you must open the **Node Editor**. This is the same as the Node Editor for materials.
3. We must then check the **Compositing** button near the **Shader** button in the **Header** options. It is a small icon button symbolizing an image over another.
4. We must also check the **Use Nodes** button.

We have now two nodes, a Render Layer node and a Composite node.

1. We can split our scene to open the 3D View and make a render of the cube in the middle of the scene (**F12**).
2. We can also add a **Viewer** node (press *Shift* + *A* and select **Output** | **Viewer**). This node will allow us to visualize the compositing result directly in the **Node Editor**. You only need to connect the **Image** output socket of the **Render Layer** node to the **Image** input socket of the **Viewer** node and check the **Backdrop** option of the **Header**.

Now the render image appears behind the nodes. It will be pretty useful to do compositing in full screen. If you want to move the render image, use *Alt* and MMB. Two other interesting short keys are *V* to zoom in and *Alt* + *V* to zoom out.

1. We will add a **Color Balance** (press *Shift* + *A* and select **Color** | **Color Balance**) and connect it between the **Render Layers** node (to the **Image** output socket) and the **Viewer** node (to the **Image** input socket).
2. If we change the lift color, the render image is directly updated.



3. We can also replace the render of the cube by any other picture, by adding an **Image** node (press *Shift* + *A* and select **Input** | **Image**), and connecting the **Image** output socket to the **Image** input socket of the **Color Balance** node.

Note

Looking at a texture node through the Viewer

There is a node that can help you to better visualize what the compositing looks like at a certain point. You can press *Ctrl* + *Shift* and right-click on any node to append a ViewNode and connect to it.

The possibilities of compositing in Blender are enormous. For instance, you can easily use keying techniques that are often needed in the movie industry. It consists of replacing a green or a blue screen behind an actor by a virtual set. Compositing is an art and it take too long to explain everything, but we will see some of its basic concepts so that we can improve the shots of our sequence.

Now, we are going to work on the first shot of the sequence:

1. As with any other image file, we must add an **Image** node (press *Shift* + *A* and select **Input** | **Image**) and connect it to the viewer.

2. We press the **Open** button of the **Image** node, and we take the corresponding OpenEXR Multilayer file. We also check **Auto-refresh**.

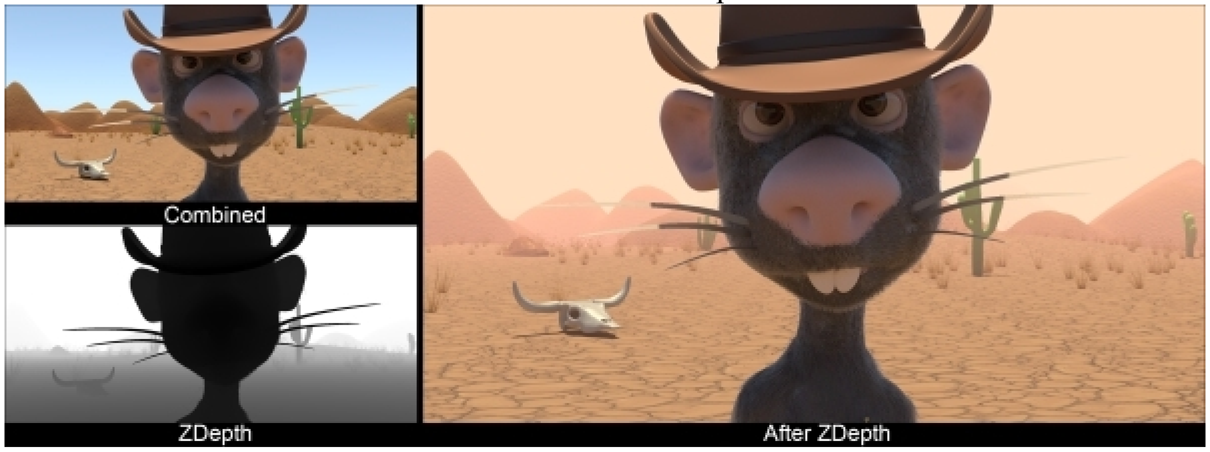
Depth Pass

The following is the combined output socket of the Image node, and there is a Depth output socket. This pass will allow us to simulate an atmospheric depth. It is an effect that can be observed when we look at a distant landscape and a kind of haze is formed. The Depth pass is a visual representation of the Z-Buffer on a grayscale. The objects near the camera will have a gray value close to black, unlike the distant elements, which will have a value close to white. This pass could serve for other things such as masking or blurring the focal depth. It all depends on the context. A controlled atmospheric effect may bring realism to the image.

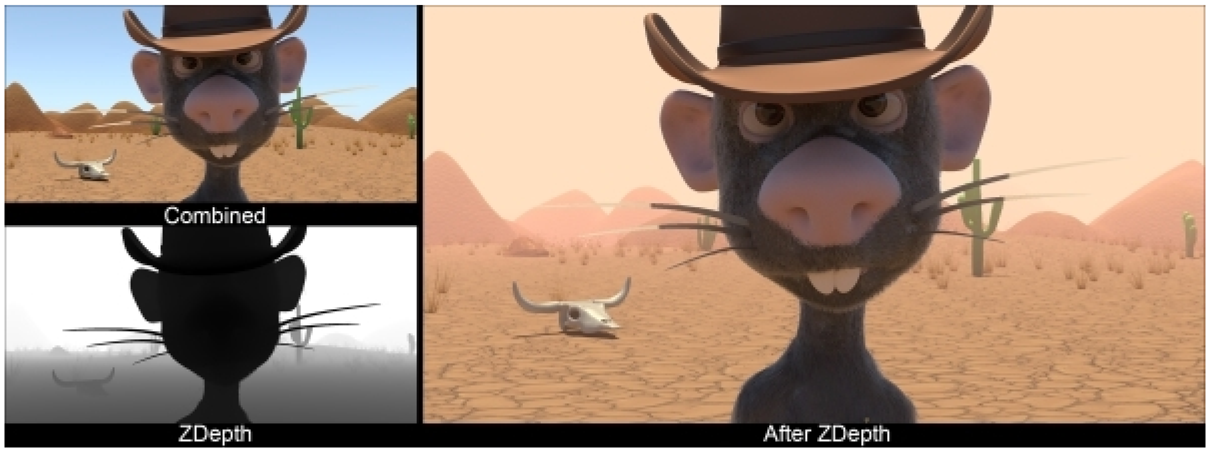
1. We will start by adding a **Normalize** node (press *Shift* + *A* and select **Vector** | **Normalize**). This allows us to clamp all pixel values between 0 and 1. We cannot visualize the Depth pass without a Normalize node.
2. We will add **RGB Curves** (press *Shift* + *A* and select **Color** | **RGB Curves**) to change the contrast of the ZDepth pass and control its strength effect.
3. We will then add a **Mix** node (press *Shift* + *A* and select **Color** | **Mix**) with a **Mix** blend mode.

Now that we have added the nodes, we are going to connect them as follows:

1. We will need to connect the **Z** output of the **Image** node to the input of the **Normalize** node and the output of the **Normalize** to the **Image** input of the **RGB Curves** node.
2. We will also connect the **Image** output of the **RGB Curves** node to the **Fac** input of the **Mix** node and the **Combined** output socket of the **Image** node to the first **Image** input socket.
3. We must adjust the RGB Curves node by adding another point to the curve. The point is located at **X: 0.36667** and **Y: 0.19375**.
4. We will also need to change the color of the second **Image** input socket of the **Mix** node. The hex code of the color is **D39881**. It will color the white pixels.



A render before and after the ZDepth pass



Color correction of the shot

One of the most important aspects of compositing is color calibration. Fortunately, there are easy-to-use tools in Blender to do that.

1. In order to quickly change the hue, we will add a **Color Balance** node (press *Shift +A* and select **Color** | **Color Balance**). The hue corresponds to the color tint of the image.
2. We must be very careful to slightly change the value of **Lift**, **Gamma**, and **Gain**. They become strong quickly. The RGB values of Lift are **R: 1.000**, **G: 0.981**, and **B: 0.971**. The RGB values of Gamma are **R: 1.067**, **G: 1.08**, and **B: 1.068**. The RGB values of Gain are **R:1.01**, **G: 0.998**, and **B: 0.959**.

Note

There are two correction formulas: **Lift/Gamma/Gain** and **Offset/Power/Slope**. These are the two ways to get the same result. For each one, there are three color wheels and a value controller (**Fac**). You can modify the darker, the mid-tone, and the highlight values separately.

If you want more information about the color balance node in Blender, visit this link :

https://www.blender.org/manual/composite_nodes/types/color/color_balance.html

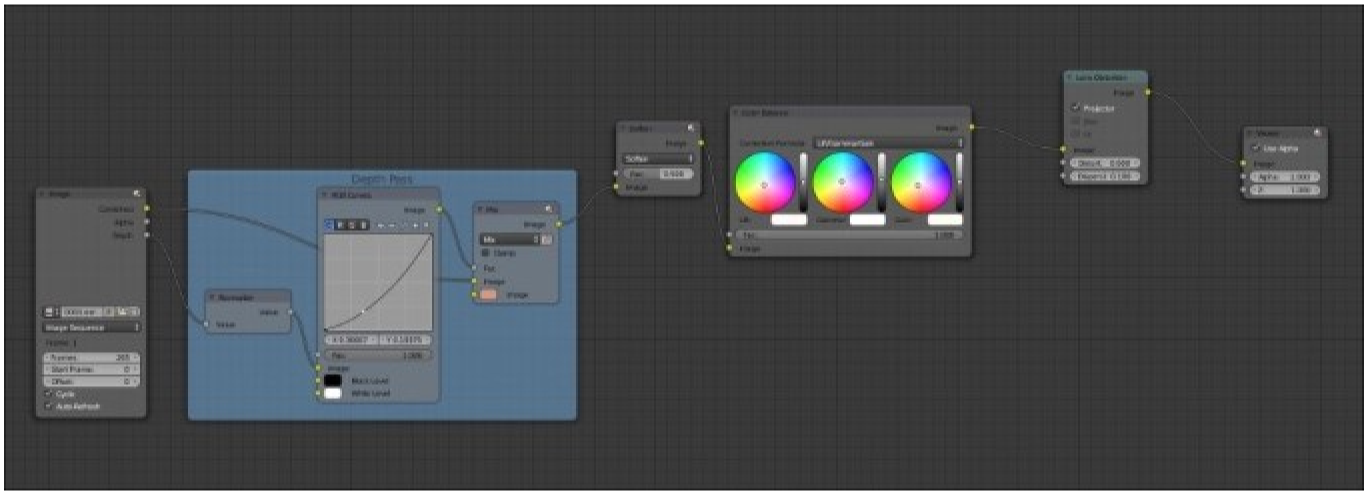
A render image with an adjustment of the **Color Balance** node will look like this:

Before finishing the compositing, let's add a few effects.

Adding effects

We will add a **Filter** node (press *Shift* + *A* and select **Filter** | **Filter**). We must connect the **Image** output socket of the **Mix** node to the **Image** input socket of the **Soften** node, and we will connect the image output socket of the **Soften** node to the **Image** input socket of the **color balance** node**.** We will keep the filter type to **Soften** with **Fac** of **0.500**. This blends the pixels so that the image is less sharp. A photo is never perfectly sharp.

We will add then a **Lens distortion** node (press *Shift* + *A* and select **Distort** | **Lens Distortion**). We must connect the **Image** output socket of the **Color Balance** to the **Image** input socket of the **Lens Distortion** node and the **Image** output socket of the **Lens Distortion** node to the **Image** input socket of the **Viewer** node. We will check the **Projector** button. The **distort** option is at **0.000,** and the **dispersion** option at **0.100**. This node usually allows us to make a distortion effect such as a fish eye, but in our case, it will allow us to make a chromatic aberration. This adds a soft and nice effect.



The nodes of compositing

We have now completed the appearance of the shot.

A render with final compositing

Compositing rendering phase

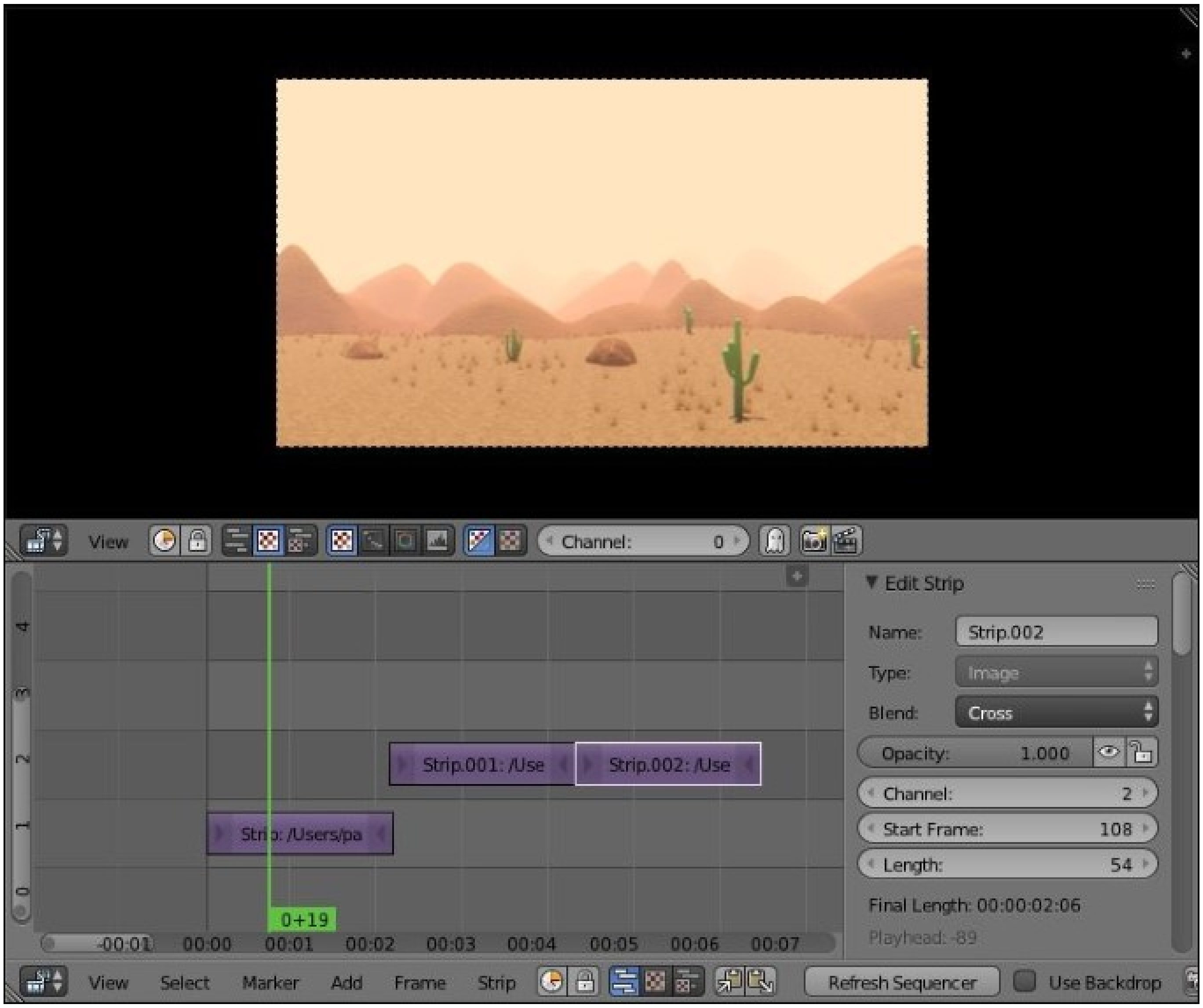
We are now ready to make the render of our compositing:

1. In the **Output** options (**Properties** | **Render** | **Output**), we must change the Output path. We will write the following address: //**Render\01\Compositing\**. We will press *Enter* to validate the address.

2. We will also change the Output format to **TGA**.
3. It is a compositing render, so we will check the **Compositing** option in the **Post Processing** tab (**Properties | Processing**).
4. Now, we are ready to render the scene. We will use the same process for each scene.

Editing the sequence with the VSE

Now that we've rendered and done some compositing on each shot, it's time to bring back the whole sequence in one final place. In this section, we are going to do a basic video editing with the VSE.



Edit Strip
Name: Strip.002
Type: Image
Blend: Cross
Opacity: 1.000
Channel: 2
Start Frame: 108
Length: 54
Final Length: 00:00:02:06
Playhead: -89

Two VSE, one is set to the Image Preview (top), the other to the Sequencer (bottom)

Introduction to the Video Sequence Editor

The **VSE** or Video Sequence Editor is a method of video editing in Blender. It is really simple to use and could be very powerful. The best way to use it is to use the **Video Editing** layout located in the menu bar. We usually don't use the Graph editor here, so we can join it back. We have now an interface with two Video Sequence Editors and the Timeline at the bottom. On the head of the VSE, we have three icons that are used to display **Sequencer**, the place where we edit strips, and **Image Preview,** where we see the result of our editing, or both. In Sequencer, we can add different types of strips with *Shift* + *A*. We are mainly using **Image**, **Movie**, or **Sound**. We can import Image Sequences with the **Image** option. You can select a strip with RMB and move it with *G*. When you have a strip selected, you have access to two buttons to the left and right represented with arrows that define the start and the end of the strip. You can cut a strip by placing the timeline where you want the split to be and pressing *K* (for knife).

We are not going to go deep into every setting of the VSE, but for each selected strip, you have some options in the right panel of the editor (*N*), such as the **opacity** of an image or movie strip or the volume of an **audio** strip. Of course, you can animate each option by right-clicking on them and choosing the **Insert Keyframe** option, or by simply pressing *I* while hovering over them. You can press *Ctrl* to snap a selected strip to another strip. You can also use the *Shift* + *D* shortcut in order to duplicate a selected strip.

Edit and render the final sequence

Let's now create the editing of our sequence with the shots that we've composited and rendered before:

1. In order to edit our sequence, we will open a new fresh file. We will also change the layout of our interface so that we have two **VSEs**, one with **Image Preview** and the other with **Sequencer**.
2. Now, we can add our first shot by pressing *Shift* + *A* in the sequencer and by choosing **Image**. In the file browser, we will go to the **Render** folder and select the **01-compositing** folder in order to select every `.targa` file with *A*. We will now have a new strip that corresponds to the first shot. We will repeat the same process with the other shots.
3. Now, we can move each shot one after the other to create continuity. Be sure that each strip is snapped to the previous one by pressing the *Ctrl* key while moving them.
4. We will also need to readjust the animation start and end in the timeline from the beginning of sequence to its end frame.
5. If you want, you can add **Sound** strips in order to add music or sound effects.
6. We are now ready to render our final edited sequence. To do this, we will change the **Output** path in the **Render** tab of the Properties editor, and we will choose the **H.264** file type. In the **Encoder** section, we will use the **Quicktime** type, and we will, finally, press the render button.

Summary

First, congratulation for arriving at this point of the module; we hope you've learned a lot of things and that you can now realize all the ideas that you ever dreamed about. In this last chapter, we learned how to finalize our sequence by creating advance materials. We also learned how to use the particle system to create a complex fur. Then we set up our render with the OpenEXR MultiLayer format and discovered what passes are. After this, we saw the power of nodal compositing by changing the color balance and adding effects to our shot. As a bonus, we learned how to use the VSE in order to edit our full sequence. Be aware that we didn't explain a lot of things, and you can go deeper into each subject. We recommend that you practice a lot and skim the web and the other books of the PacktPub collection in order to extend your knowledge. Remember, you can learn a tool, but creativity is one of the most important things in this field. We wish you a successful continuation.