Realism in Vue

www.**RealismInVue**.com

e-on software, inc.
6107 SW Murray Blvd, Suite 264
Beaverton, Oregon 97008-4467 - USA

www.e-onsoftware.com www.quadspinner.com

To my father…

who introduced me to the key things that shaped my life:
art, computers, and a love of adventure.

Contents

Acknowledgements

This book holds the name of a single author, but that author could not have completed it without the culminated help, support, and encouragement of a collective of people. No matter the size of their contribution, it was all vital.

My profound gratitude to Cynthia Najim, my muse and partner in QuadSpinner, for all our memorable expeditions, for your ever unique perspectives on life, for those extra nudges to get the job done, for helping me finish this book, for excellent driving that saved our lives on several occasions, and for "sourcing the finest locations…"

Conrad Allan provided invaluable (and maddening) feedback while I was writing the book. He helped make sure the book made sense in more than one aspect. He coined the 'burn the box' metaphor for the Function Editor.

Parvez Ansari, my right hand at Nukeation Studios, single-handedly held back chaos so I could finish this book in peace. Without him, this book simply would have never been finished at all.

My parents – Mahim and Bharti Pandhi – for planting my roots in art since I was born.

Drea Horvath and Michel Rondberg provided testing, constant encouragement, and useful feedback, helping keep the book on track. My cohorts at deviantArt: Matthew Attard, Jonathan Pearce, Kerem Gogus, and the Aussie (and Kiwi) Band of Merry Renderers who kept me focused on finding the right issues to look at and talk about. Marek Mihok came in late into the project, but his help was indispensible.

The masters of Vue: Eran Dinur and Chipp Walters for their inspiration and insights.

Thanks to everyone at e-on for creating Vue. Natural 3D never found a better home!

Finally, special thanks to Nicholas Phelps, father of Vue, for making this book a reality.

About the author

Photo © Cynthia Najim.

"I would love to create a world in Vue and live there."

Dax Pandhi is both artist and technician. His scientific mind prods him to probe, challenge and invent. The results of these trials are refined by his masterful eye for composition and story. Dax's innovations in CGI are testament to his deeper understanding of nature's processes and his effective translation of that knowledge onto the digital platform.

Dax is also a gifted teacher. Beyond this book, he has gained popularity through his training videos that share his unique methods in a simple format. "Learning all of this doesn't mean anything if I can't share it with others."

Dax's artistic roots are in traditional painting and photography - each serving as a feedback loop for the other. In 2007, he released his first photography book, Visit to Earth. He is currently working on two other photo books.

Outside of his background in art, Dax has built a successful career in technology, ranging from CGI to software programming. Based on his innovative design work and service under his company Nukeation Studios, Dax was awarded Microsoft's MVP for 3 consecutive years. He served as journalist for Associated Press covering the aftermath of the devastating 2001 earthquake in Bhuj. He is proficient in 8 computer languages and 4 human languages.

Dax lives in the desert in Bhuj, India.

To view more of his works, visit **http://www.daxpandhi.com**

For information on his Vue products and training series, go to **QuadSpinner.com**.

Foreword

Over the past couple of decades, 3D technology has developed and expanded exponentially. The popularity and rapid fire growth of this new medium are at the core of a new wave of artistic expression. Gamers of all ages are captivated by challenges presented with elaborate imagery. Movie goers are visually seduced by the imperceptible blend of the real and non-real, as well as by the enormously creative feats of pure animation. Advertisers, architects and designers have also embraced the breadth of possibilities that 3D technology brings to their businesses. And students are equally eager to harness these tools to cultivate lucrative careers.

This irreversible trend is fueled in large part by the accessibility of 3D technology. A normal home computer, when coupled with intelligent software, now has the power to produce Hollywood quality CGI. Putting these resources in the hands of inspired users will continue to raise the bar of performance. Dax Pandhi exemplifies these leaps in what is possible and shares some of his knowledge with you in Realism in Vue.

The purpose of Vue is to provide the evolving tools to digitally create authentic natural environments. When working with nature, the defining characteristics are the subtlety of the details and their contrasts – veiled textures, bold strength, naked fragility, and the climaxing ballet of light. Vue was designed to work as nature does in real life. Replicating this in 3D requires a special way of thinking and unusual applications.

Dax has emerged as one of Vue’s most prolific pioneers, artists and trainers. This book takes a grassroots approach to achieve dramatic realism by helping you understand nature and translating it into comprehensible language. These methods uniquely bridge the complex machinations of programming and the delicate nuances of art and the natural world. By experimenting with the software and daring to push the limits, you will be guided to new frontiers with tools that will help you create your own version of magic in Vue.

Seeing what enthusiastic talents like Dax have produced in our software has been deeply gratifying for me personally. Remembering Vue’s simple beginnings and witnessing its extraordinary evolution into one of the leading tools of the CG industry is both rewarding and humbling. I look forward to the continuing journey. For you beginners, welcome. For our loyal friends, my sincere gratitude.

Nicholas Phelps
Founder & President
e-on software

The mistakes the Universe inspires are perfect.

The exquisite beauty found in nature is made of imperfections.

In CGI, the most challenging aspect is to create or recreate such “mistakes”. In nature, these imperfections are produced and ultimately harmonized over eons, whereas an artist is required to achieve this triumph in a miniscule fraction of that time.

The true beauty of the natural world often combines seemingly random series of events.

In CGI, there are no set rules on how to authentically create our natural environment. The 3D artist – much like a lab scientist – must manually and consciously compose that which nature often unsystematically brings into being. Through copious hours of trials, I have discovered and, in some cases, literally stumbled upon techniques (mistakes!) that yield outrageous results – terrains, scenes, atmospheres, and materials that simulate nature.

The tools paramount for success: an understanding of nature’s processes, your intuition, and dedication. To aid you in your development, this book steps away from the setting-by-setting and step-by-step approach. We will first explore the why’s and then delve into the how’s. In doing so, you will be well equipped to deliver dramatic realism in 3D. Enjoy the ride.

Dax Pandhi
Bhuj, India - 2011

Introduction

Chapter 1

A different approach

Realism is a never ending quest for any artist working in the 3D graphics space. Over the past decade, 3D CGI has come very far in matters of realism. One of the first areas to achieve photorealistic results was interiors and manmade structures, followed by humans and fantastical creatures. You can now create a scene in almost any major 3D application of the interior of a house with a person sitting on a sofa and very few people would be able to tell the difference between the render and an actual photo.

However, take the same character and replace the interior with a large dramatic landscape. Now you have reached the frontier of 3D graphics. Creating digital landscapes is not a new concept. It has existed for over a decade. But creating a photorealistic natural scene is still a difficult process.

Vue takes a very different look at natural 3D scenes when compared to normal 3D applications. Everything becomes procedural and infinite in scale. Instead of creating a piece of a world like you would in any other application, in Vue you enter into a world itself and then reshape it. The results of this are fantastic and near-perfect when compared to a photograph.

Realism in Vue takes on a different approach than with most other 3D applications. Instead of setting up a “fake” simulation of objects around a camera, you work within a physical world. Think of it like this: working in a typical 3D application is like working on a studio set with lots of lights, a stage, props, and so on. Working in Vue is like taking a camera outdoors and shooting pictures from atop real locations.

As you can imagine, when you go out of the studio, the rules change completely. You do not create the perfect setup – you work with what nature gives you. Thankfully, in our case, Vue is “digital nature” so you can create a near perfect atmosphere to suit your needs; but at the same time there are “physics” of the digital world you are working in that keep you close to how things are in the real world.

At that point, we reach the topic of this book: realism in Vue. It takes on a very different form than normal 3D, so we will take a different look at achieving realism in your natural 3D scenes.

The Techniques

It should be noted that all the techniques presented throughout this book are based on over 9000 hours of experimentation in Vue. There are often different ways to achieve a single effect. In some things even the Vue manual may say something different than a technique presented in this book, but it is my belief that there is no “right” or “wrong” as long as the end result is achieved.

So I would encourage you to experiment and create your own techniques – whether they are built from the ground up on your own, or based on my techniques. The key to excellence in any technical field – and especially in an artistic form such as natural 3D – is that there is no greater teacher than experimentation and experience.

In furthering that concept, instead of the usual “step-by-step” method most books use, we will be going through a little more of theory and higher concepts rather than tiny, detailed steps. It may sound boring and counter-productive, but the other way would only help you clone the scenes you will see in this book rather than learn how they were made, and why the settings used were like they are. Gaining understanding of the reasons behind each setting will help you become stronger in Vue than anything else.

Throughout the course of this book, you will see new settings with descriptions saying “To achieve this type of effect you have to use a value of X”. Those settings are not the only way to achieve that goal, but rather a starting point. In your scene, the light may be different or the material may use a different color and whatever setting that was recommended may not work perfectly. Experiment around those values by using it as a starting point for further exploration.

The Foundations of Realism

In Vue, realism is based on 5 basic factors or foundations: atmosphere, rock formations, materials, clouds, and perception.

Let's create a simple scene and see how each of these foundations contributes to the final result.

Atmosphere

Atmosphere is the key element that binds all the other factors. It is the key to lighting and the key to the most visible aspect of your digital world. In 3D, lighting is almost everything. Proper lighting can make even the most ridiculous and fake looking object look real enough to make you want to touch it.

However, Atmosphere goes beyond just lighting and the visible sky. The atmosphere also controls how all the elements of the world are shown to the camera, with fog, haze, and even the particle density in the atmosphere that creates the illusion of depth.

Atmosphere is one of the first things that need to be set up when creating a new scene. And as such, in our first scene we see a blank scene with a sunny atmosphere setup.

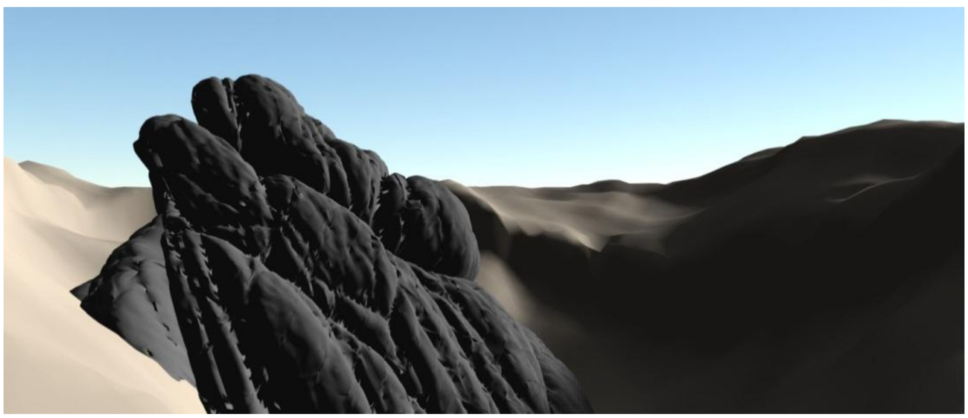
Vue's default opening position

Rock Formations

A natural scene is nothing without a landscape. The landscape creates the anchor of the final image and decides how the rest of the world is shown to the camera.

For the purposes of this book, we will talk about the normal terrains and procedural terrains that are the hallmark of 3D landscapes in Vue, but only briefly. We will focus more on HyperTerrains and other ways of modeling rock formations for adding extra realism. HyperTerrains allow you to break the typical topographical-only terrain model and create alcoves, caves, jutting rocks, strong vertical cliffs, and just about any natural rock formation.

As we take our scene to the next step, we add a typical rolling hill terrain (sand colored) along with a rock outcrop HyperTerrain (gray) added on top of it.



Camera Viewport

Rendered Image

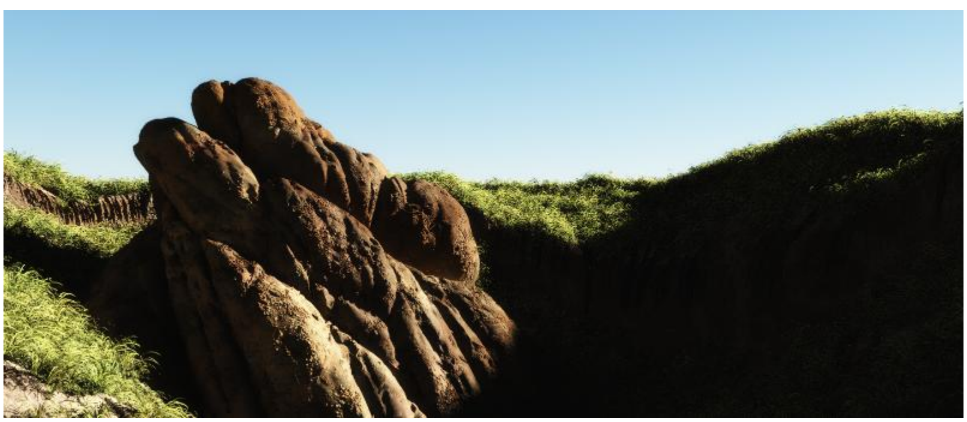
Materials

If lighting can make a fake material look real enough to touch (three dimensional) then it is the material that gives it the form of a real world substance. In Vue, the normal bitmap approach to creating a material co-exists with the more powerful procedural method where the material is made up of scientifically generated “infinite” textures and properties.

This procedural method has literally infinite applications. Instead of scouring for photographic references and stock images to create a texture for your material, you can use fractals and other mathematical functions to create any kind of texture you want. And beyond the normal texturing, the procedural aspect of the material is also functional. It will interact with aspects of the world like height, slope, altitude, transparency, and more. This functional aspect is the core of Vue and adds a powerful layer that we can use to achieve more realism.

And finally, one of the most useful features of the material in Vue – the EcoSystems – an infinite array of unique plants and trees that can be added to any object or terrain with the help of the Vue EcoSystem Material. This adds the most unique layer of Vue to your 3D scene.

Once a scene is set up, making the materials is the next key phase. In our scene, we now add a grassy EcoSystem on a sand-like material, and make our rock outcropping look more realistic with a stone material - all procedurally created.



Clouds

One of the toughest things to create in 3D is realistic cloudscapes. Vue utilizes the new Spectral2 atmosphere system in version 7 and 7.5 and Spectral3 in version 8 and 8.5 to create realistic clouds in all shapes and sizes. There are two main types of clouds: cloud layers that encompass the entire world, and MetaClouds – cloud objects that can be manipulated by hand.

In Vue, you can create clouds to be realistic, fantastical, or both. With the same internal Material Editor Engine and the SmartGraph Function Editor working inside the Cloud Editor you can use fractal functions to create just about any type of clouds imaginable. With the new Planetary Cloud Density feature you can create an overall bitmap that controls the cloud density and distribution over the entire scene, but still maintains the fractal infinite detail of the actual clouds.

Clouds are an inseparable part of a natural scene, and as we move forward in our scene, we add a very Summer-like cumulus cloud layer.

Perception

Perception is more powerful than reality. We are not talking philosophy. In 3D, and especially in Vue where scale becomes a relative concept, it is the perception of the viewer that makes your render look real or not.

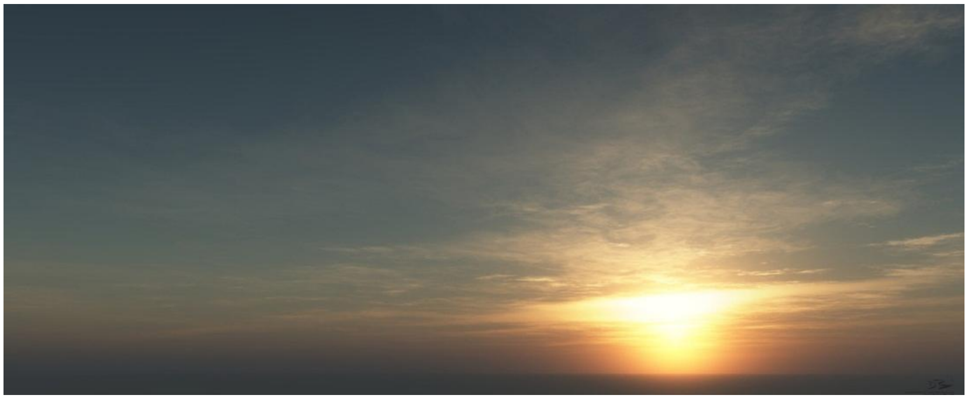
Through millennia of evolution, a very powerful and deeply ingrained feature of the human mind is that whether we consciously decide to do so or not, if something is wrong in a render – it will be the first thing that we see! In the wild, this meant the difference between life and death. Animals that are dangerous to humans have camouflage that helps them hide in the environment and we evolved to be able to recognize such hiding tactics. So whenever we see something trying to mimic nature, we always find out. Often we do not even know *why* it looks wrong, but we know it is wrong. Coming out of the wild and back to our renders, this could mean that your endless days of labor over the render can go to waste if you miss little things. And that is perception.

Unlike the other foundations, perception is not part of Vue itself, but is rather an added layer on all the other foundations and contributes that final touch of realism. While we will discuss each of the other foundations in chapters of their own, perception is a matter that is scattered throughout this book and we will discuss it in relation to each of the other foundations.

Specifically, perception is about making your scene look believable. It can be something as simple as a camera angle, but it can also be about matching the atmospheric haze with the overall lighting of the scene. It can be about making 20 meter tall terrain look 5000 feet tall. It can be about creating a never ending chain of mountains without having to create it all yourself.

In our scene, we add the final touch of perception by adding a light haze and a little ambient lighting, as well as increasing the Aerial Perspective to create a sense of grandness. And we're done.

The Big Secret



One thing that you will find stressed again, and again, and again, and then a few more times is the importance of scale. It may not sound like much, but this is – from a technical point of view – the single most important thing to consider when doing anything in Vue.

The gravity of this concept hit me at its maximum when I heard Nicholas Phelps (the creator of Vue) explain the way light and atmosphere are processed under the hood. The sentence was something like: "*We trace the ray of light as it passes through the atmosphere and calculate the oxygen and nitrogen in the air…*"

This statement made me realize consciously a concept which is so easy to forget: Vue is a complex tool. The ease by which we can manipulate an entire "world" makes us forget that a certain order and meticulousness is required for achieving realism in 3D.

Vue is deliberately designed to mimic the real world – after all, the whole point is realism. So the best thing to do is keep in mind that if you treat every square meter of your Vue world as a real square meter of the real world, for example, by not coercing a 20 square meter mountain to hold 20,000 trees by forcing down the instance size in an EcoSystem, then you will always have a much better shot at achieving realism.

The scaling controls in any aspect of Vue, whether atmosphere, materials, terrains, etc. are for fine tuning. More often than not, to make something bigger or smaller, you should actually resize the object itself to proper real world sizing than to force an effect by changing the scale.

So one last time (in this chapter, at least): do not abuse the power of Scale.

Vue and Compatibility

Over a year ago, as of the writing of this chapter, Vue has been put on a bi-annual release cycle. This means that a new major or sub-major (.5) version is released approximately every six months.

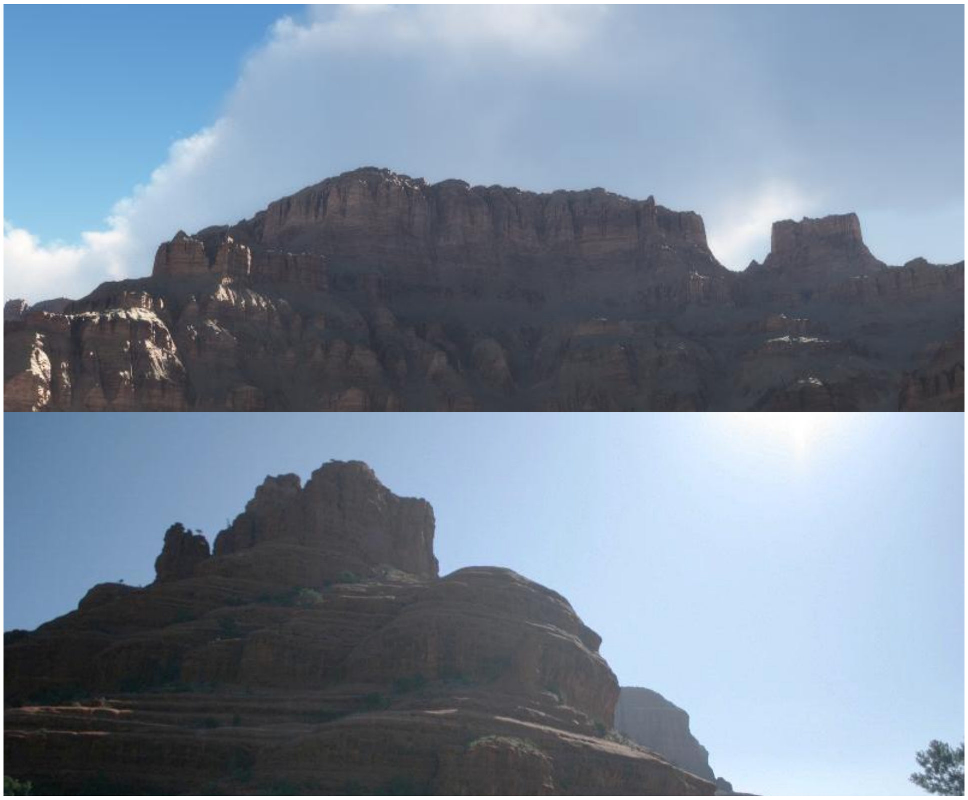
As a result of this, the book was written across 3 major releases of Vue. You will see screenshots from Vue 7, 8, and 9 illustrating examples of techniques and features throughout the book. This is a positive thing, because Vue's core remains the same even if new features are added. What's more is that if you haven't upgraded, or have certain assets that cannot be upgraded due to project restrictions, you will still be able to make good use of the knowledge in this book.

Great care was taken to maximize the compatibility of techniques presented in this book. The majority of the content here will apply to most versions of Vue, while Vue 8 and higher will be completely compatible.

Natural Atmospheres

Chapter 2

The Anchor of Environments



Vue render (top) compared to a photo of Sedona, Arizona (bottom)

In creating a natural 3D environment, the atmosphere plays a vital role. The way we perceive nature everyday has to be mimicked in 3D to make your scene look real. In this chapter, we will explore several different atmospheric concepts, compare them to real world references, and then simulate them in Vue. We will also go over the Atmosphere Editor using the Spectral atmosphere model and understand how each setting affects our atmosphere.

Vue also has other atmosphere models. However, all of them are somewhat obsolete when compared to the Spectral model. The Spectral model is a very complex model first introduced in Vue 6, and now in Vue 8/8.5 we have the third generation of the model. In this model all light is processed as it would be in the real world. When the sunlight passes through the atmosphere, it is exposed to various factors such as the dust particles in the air (haze), water particles and condensation (fog), and many others as we will explore shortly.

As Spectral is the most realistic method for creating natural atmospheres, for the purposes of this book we will use only the Spectral model.

Using only Sunlight

In most other 3D software, it is a very common practice to use multiple light sources to light a scene. Many Vue artists who switch over from other software tend to use this technique in Vue as well. While there is nothing exactly wrong with this method, it is prudent to use a single light source – the Sun – when lighting an outdoor (or even indoor) scene.

On a typical day in the outdoors, there is only one light source - the Sun. So it is logical to simulate the outdoors in 3D using only one light source. It may seem like a difficult thing to do, but unlike omni or spot lights, the directional light source that is used for the main Sunlight light source in Vue is very atmosphere-sensitive and will light your scene very close to how the real sunlight affects the world.

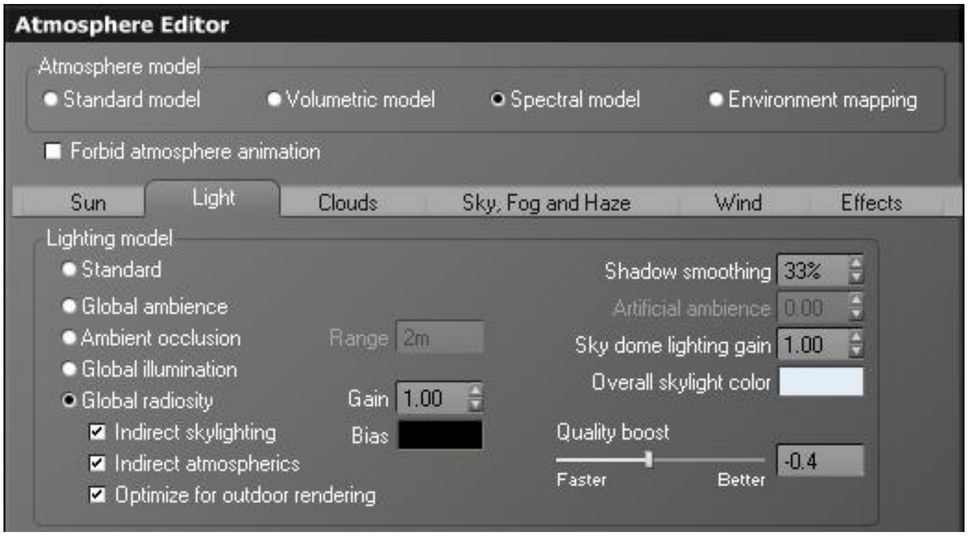
When you use only one light source, one of the biggest benefits is that it lowers the potential for lighting mistakes. You will never have to worry that one light is pointed this way and the other light in some other direction. The overall scene consistency in terms of lighting also increases since a single light source affects the entire world.

In other 3D software, the main reason for using multiple lights is to simulate reflected light. Throughout this chapter we will explore how to create realistic lighting as we see in the real world, and one of the main factors is reflected light. We will simulate this using Global Radiosity, the most powerful lighting model available in Vue.

Before we explore this concept further, let’s explore Global Radiosity first.

Global Radiosity

In Vue, we have several lighting models: Standard, Global Ambience (GA), Ambient Occlusion (AO), Global Illumination (GI), and Global Radiosity (GR). The last one is the focus of all lighting we will explore throughout this book.



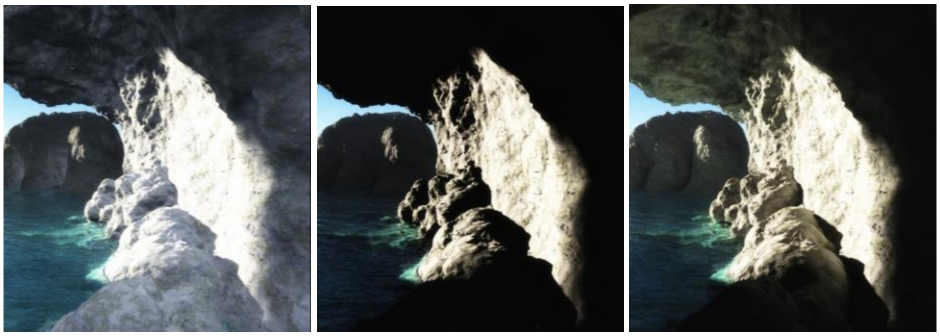
Atmosphere Editor window showing the Global Radiosity options.

The first three lighting models, Standard, GA, and AO are more or less irrelevant to this book, so we will not explore them. You can read about them in the Vue manual. However, to explain Global Radiosity, let us compare it with the second most powerful lighting model in Vue – Global Illumination.

Global Illumination, unlike other lighting models, calculates how the light coming from the sky itself affects the scene. It creates soft shadows based on reflected light around the objects. In the image below you can see the difference between a Global Ambience and Global Illumination render. Notice the softness of the shadows in the Global Illumination comparison.

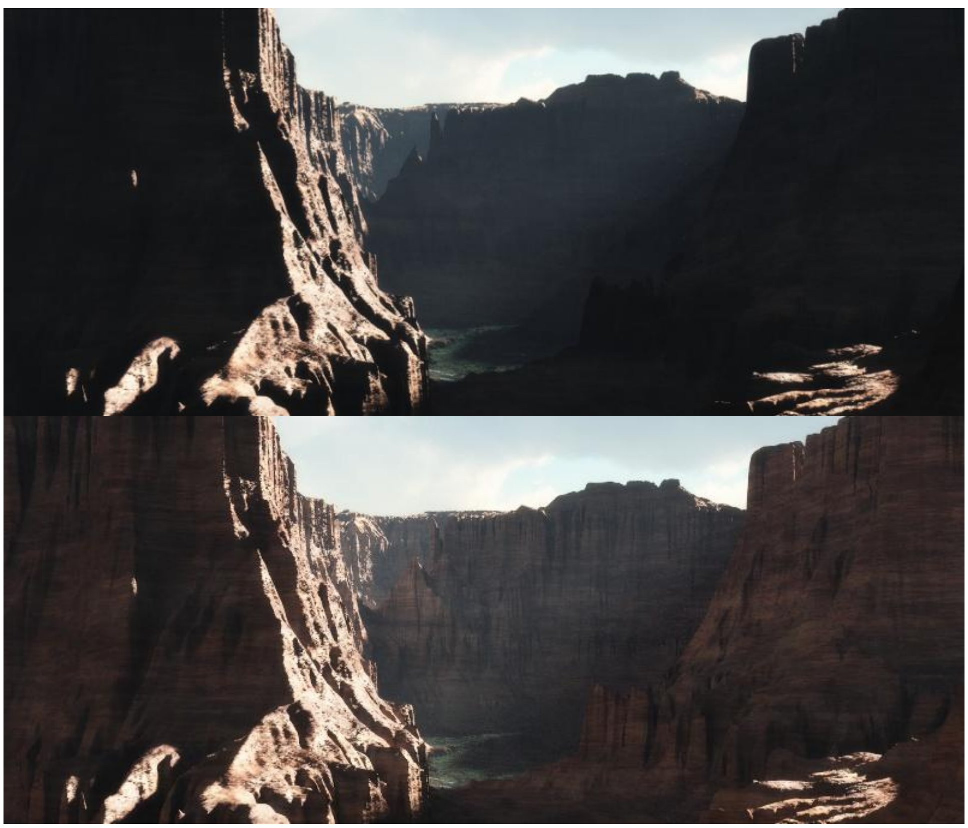
The Global Illumination lighting model is able to simulate sky lighting very nicely, but it is limited in the way it affects the shadows. Put simply, it only scatters light for shadows. The Global Radiosity model imbues color into the ray of light when it reflects it off an object. While Global Illumination scatters only shadows, Global Radiosity scatters all aspects of light.

Here are three images comparing **Global Ambience**, **Global Illumination with Artificial Ambiance**, and **Global Radiosity with Indirect Skylighting** respectively.

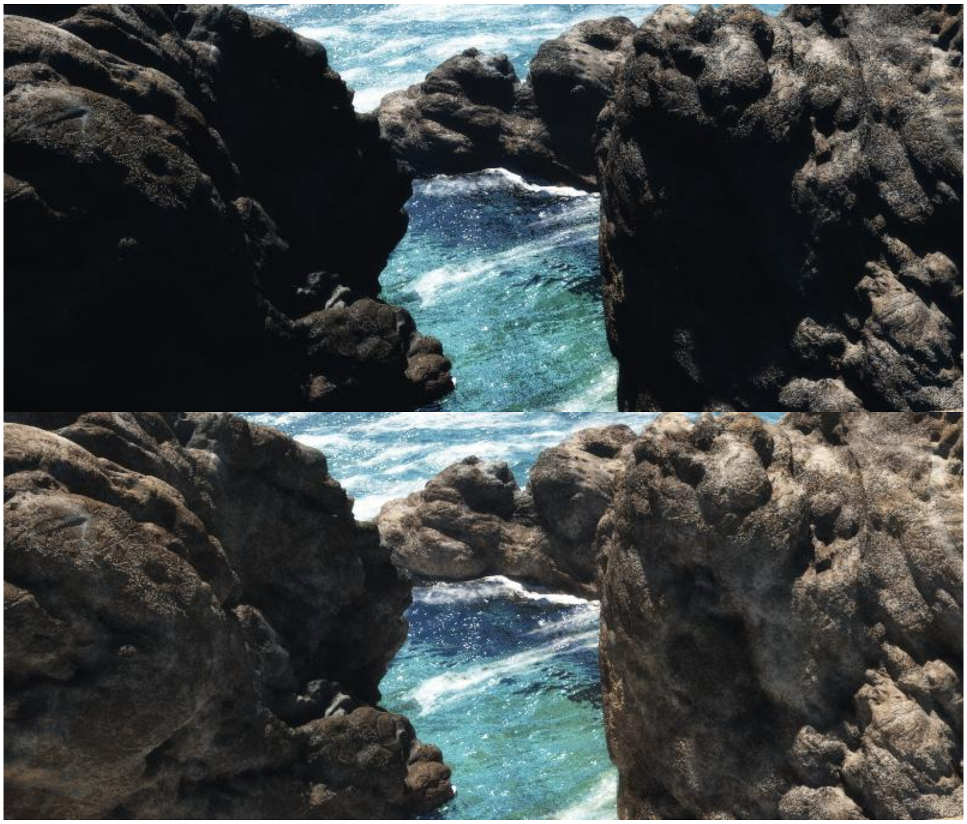


GA, GI, and GR comparison renders.

Let's take a look at how simply switching from Global Illumination to Global Radiosity with Indirect Skylighting can affect scenes. Each of the following scenes illustrate different environments and have two versions, the left version using GI and the right using GR. Notice how the light-bled color affects the entire scene.



The strong light creates red reflected light as it bounces off the red sandstone.



Tropical lighting paradoxically requires strong shadows as well as strong ambient light.

In a tropical setting such as this, the atmosphere is clear and the sunlight strong creating a varying ranged color reflected light – brown-white from the stone and soft blue-green from the ocean. Global Radiosity changes the reflected light color based on the object it bounces off from.

Inside a closed rock formation, such as this waterlogged cavern on the next page, sunlight reflecting on water creates an interesting colored light that spreads throughout the interior. A strong Global Radiosity Gain in this next image helps create the enhanced green reflected light from the water coming from below as well as the soft muted tones of brown and gray as light reflects on the top parts of the cavern. The latter is more noticeable on the top of the arch.

Interior scenes can be enhanced by using the Global Radiosity Lighting Model.

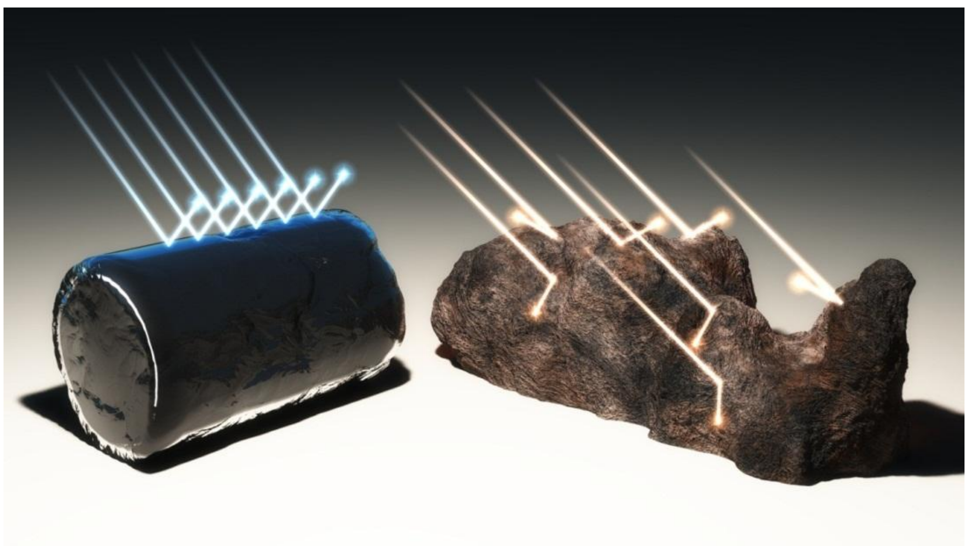
When creating semi-interior shots such as this open cave, reflected light becomes of utmost importance as that is the only way to realistically light the inner physical elements. The Global Radiosity lighting with a little Gain for intensity penetrates the shadows of most nooks, niches, and crevices while still leaving enough shadows to show the depth of the caverns. A small Gain value such as 0.75 is enough for this effect. Larger values may remove the shadows completely.

Reflected Light

We have seen how Global Radiosity can create reflected light. But let's explore *why* we need reflected light. Simply put, because it happens in the real world. To perfectly simulate the real world in 3D, we must do as much as we can to duplicate the effects seen in the real world.

Without going into detailed physics, we tend to think that only mirrors, glass, or other such objects are reflective but actually everything that we see is reflective. But reflection does not mean we will see our own reflection like we would in a mirror or a highly polished piece of marble. That would be "uniform reflection". Most objects that are not shiny or smooth but still reflect have "inconsistent reflection".

Let's understand with this figure. Keep in mind that the surface we are looking at could be at a very close or even microscopic level.



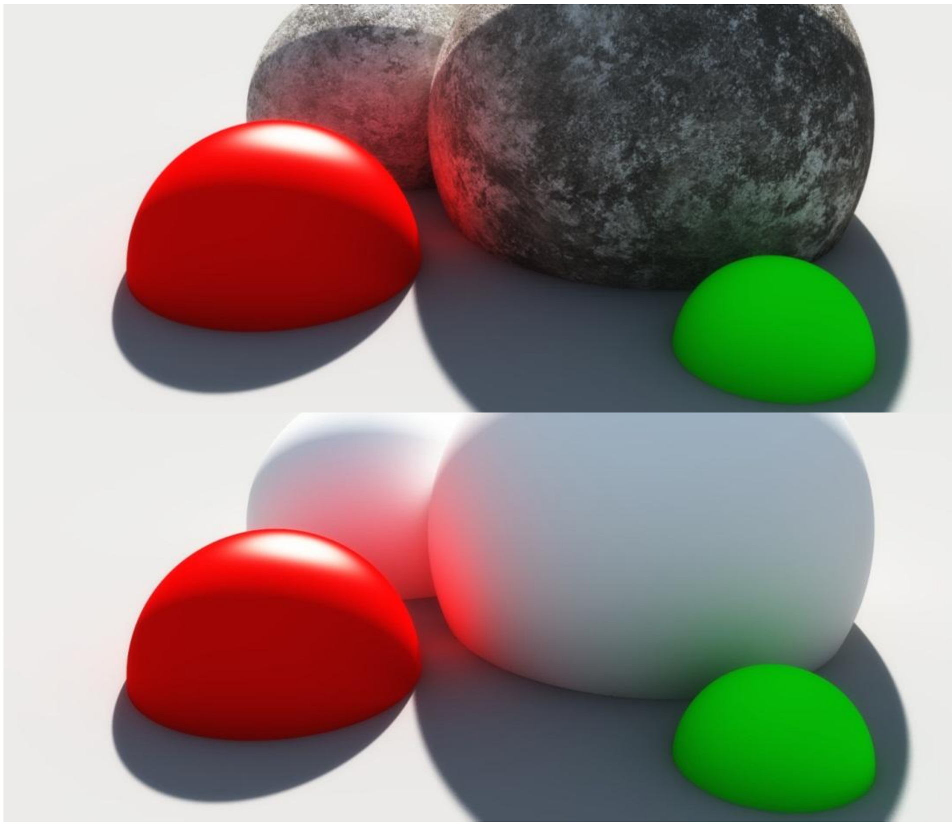
Light rays bouncing off even and uneven surfaces.

The image shows a smooth surface (left) like a highly polished piece of metal. A smooth surface uniformly bounces the rays of light into the same direction. The distribution of color per ray of light (or photon) is equal, and so we see a vivid reflection in which we can see entire scene or ourselves being reflected back towards us.

An uneven surface (right), like a rock or raw unpolished wood, will bounce the incoming rays of light but they wouldn't be reflected in the same direction. Like an uneven object being put under a strong flow of water, the rays of light would be reflected into various uneven directions based on the part of the surface they are being reflected on.

So it is because of this "photon bouncing" that Global Radiosity is able to make 3D objects look very believable as the reflected light takes on the colors or even entire textures of the object off which it is being reflected.

The renders below show how different types of materials reflect light and bleed color in the reflected light when using Global Radiosity.



Global Radiosity effect on a textured and untextured pair of spheres.

Notice how the red overpowers reflected light more easily than the green. The red is receiving direct sunlight which enables more reflected light, while the green is lit by only the Sky Dome Lighting Gain and Global Radiosity reflected light, hence providing only moderate amount of color bled reflected light.

To understand how light reflects we will explore some real world examples. In the images below, the orange beams represent direct light, and the blue beams represent indirect illumination – both the light reflected off the surroundings and the skylight.

Manmade bridge in the jungles of in Costa Rica

Under a bridge, the clean morning sunlight is reflected all around providing soft lighting for the underside of the bridge. As the light fades in the deeper corners, it is worth noting that it still does not become completely dark.

The truly important balance of light to observe in an image such as this is that the reflected light, represented by blue, is a combination of both the Blue of the Sky (Sky Dome Lighting Gain, in Vue terms, as we will soon examine later in this chapter) as well as the pure direct sunlight reflecting off the surrounding area.

Real light, in day to day landscapes, exist as a balance between blue and orange (Sky color and Decay color). Finding that right balance for your scene is what will make the sky and overall lighting look realistic and believable. If either goes out of balance, you may compromise the quality of your render.

Zion National Park, Utah, United States

In this sandstone and limestone canyon in Zion, Utah, the evening light is still strong enough to cause red tinged reflected light combining with the blue overhead light off the sky itself to light almost all parts of this massive rock formation.

When you have selected Global Radiosity, Vue will calculate the reflected light color based on the Ambient Light Color specified in the Atmosphere Editor's Light. Unfortunately, it does not create a realistic enough effect especially when dealing with a scene containing a wide range of colors. To overcome this limitation, we have to turn on Indirect Skylighting.

Rock formation near Panguitch Lake in Utah, United Stats

In a bright open environment such as this, direct Sunlight plays the biggest role in illuminating the scene, however, important details such as the shadowed side of the rock in this image is illuminated by the strong reflected light from the surroundings.

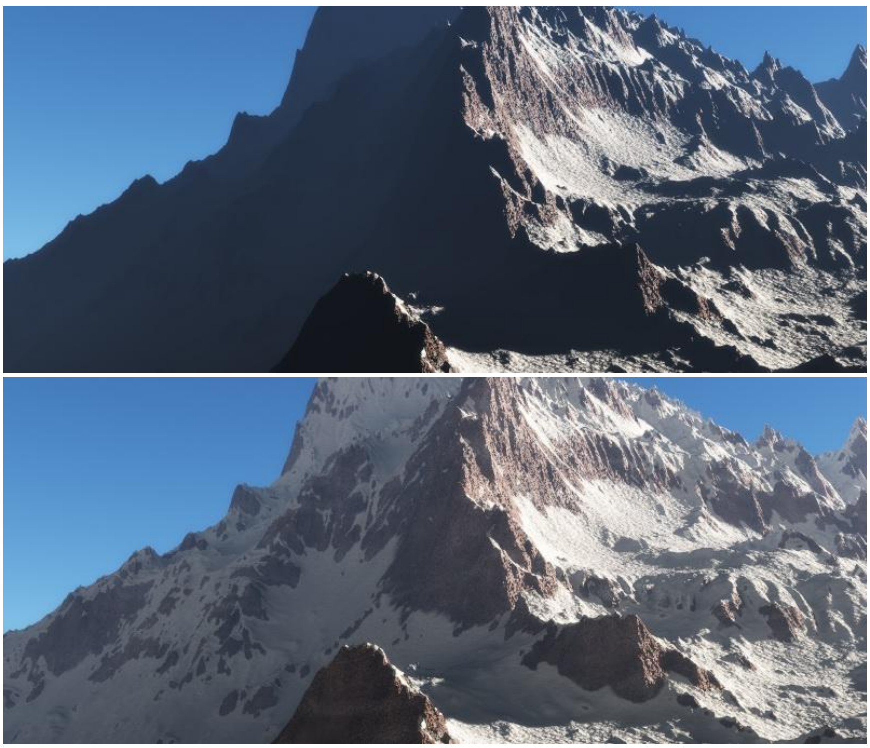
While the effect of a little ambient light coming from the sky may seem extremely subtle, you can see the difference in shadows where objects get a mix of both bled color from nearby objects as well the Sky Dome Lighting Gain color. The shadows are tinged with the color of those objects.

Often it may seem such a small effect may not be worth the render time required for it, but that is one of the major facets of achieving realism in 3D: a massive amount of small, almost unnoticeable touches added by such effects is the reason why the final image looks more realistic.

Sunlit Shadows

As we know, in the real world, there are no absolutely dark shadows in outdoor environments. Even in semi-enclosed or completely indoor environments, there is at least a small amount of reflected ambient light. One of the most common mistakes a 3D artist can make is to have all shadows be very dark. While in some situations it does create a striking effect and may be useful, as far as realism goes, it will work against you.

To illustrate the point, let's look at the following examples. These two renders are exactly the same except that the first one does not have reflected light, while the second one has small amounts of reflected and ambient light. As a result, the second image looks much more realistic than the first one.



No reflected light (top) and realistic reflected light (bottom).

In most situations, just turning on Indirect Skylighting under Global Radiosity can automatically give you realistic reflected light. If for some reason you do not get enough reflected and ambient light this way, the next section discusses several lighting techniques you can use to boost Global Radiosity settings to provide better lighting solutions.

Lighting Scenes

There are literally infinite types of scenes you can create in a natural setting, and the lighting solutions for them are just as many. Thankfully, if you can learn the essential basics of lighting - especially the "why" of the important settings - then you can light any situation intuitively.

To achieve that objective, we will explore some of the most useful lighting situations and create the perfect atmosphere for it. The general categories for lighting conditions are "outside" and "inside". For outside, we will cover large open vistas, high altitude mountain ranges, above the clouds, and vast landscapes. For inside, we will look at being inside a cave, being just outside a cave and looking in, inside a deep jungle, and inside a water-filled grotto looking at the ocean outside.

As we have seen before in this chapter, understanding how light works in the real world is very important; the most critical part being the balance between direct light and reflected light. The second step is the color of those lights, which does not mean colored light solutions, but rather the atmospheric settings which infuse colors into the light in different situations. And lastly, the shadows - perhaps the most important part of CGI.

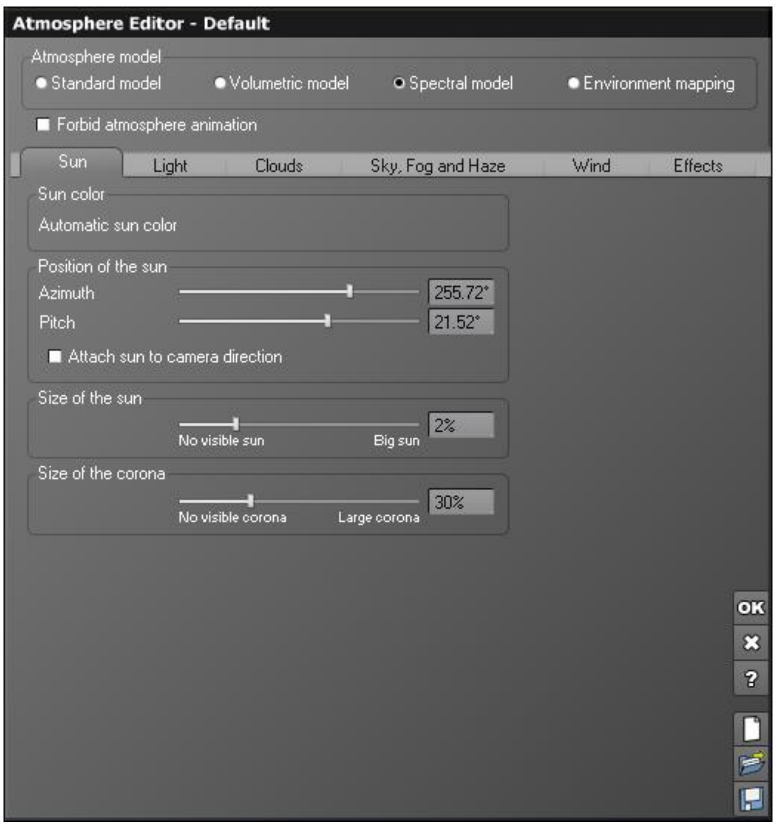
To understand different lighting concepts, we need to make sure we understand the Atmosphere Editor.

The Atmosphere Editor

The Atmosphere Editor is the heart of realism in Vue. It offers several modes as we discussed earlier in the chapter, but since realism is best with the Spectral model, we will focus solely on using the Spectral Atmosphere Model and Global Radiosity. Please note that some settings may behave slightly different in other lighting models and atmosphere models.

In this section we will touch upon the most important settings in the Atmosphere Editor as relevant to our goal of achieving realism in Vue.

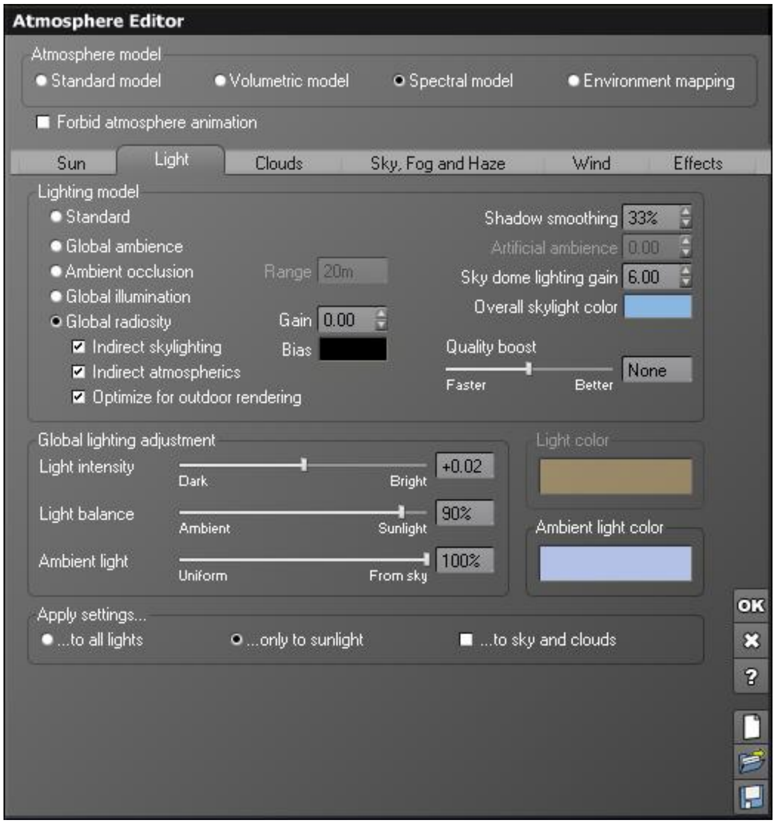
Sun



The first tab, Sun, helps control the position and size of the sun, and the size of the corona or "glow" given off by the sun. These controls are very self-explanatory, and for most of our exercises we will keep everything except the position of the sun at default values.

The Size of the Corona and Size of the Sun setting are loosely, but directly, connected to the Glow Intensity setting we will examine in the Sky, Fog, and Haze tab a little later.

Light



The second tab, Light, allows us to make the high level decisions. Global Radiosity with both Indirect Skylighting and Optimize for Outdoor Rendering should be turned on for all our scenes, except the indoor scenes or any other where specified that Optimize for Outdoor Rendering should be turned off.

Global Radiosity uses photon calculations based on how light works in real life. A photon needs to be bounced back for a proper calculation. But natural scenes are infinite in nature compared to normal 3D. So photons won't bounce back in outdoor scenes, but when you turn this setting on, Vue intelligently decides how far a photon should go instead of infinite and adjusts the rendering engine accordingly to create a realistic rendering solution without creating crazy calculations which in many other 3D applications cause errors or warnings to appear.

Global Lighting Adjustment

Light Intensity controls how dark or bright a scene is. You can use this setting on top of everything else to control the light exposure of the scene. It is perfect for fine tuning the intensity of the scene without changing other settings.

Light Balance controls whether the main sunlight is ambient with soft or no shadows or direct with dark shadows. Settings such as Sky Dome Lighting Gain are dependent on this setting and will be ignored if the Light Balance is at 100% Sunlight.

Light Balance at 0%

Light Balance at 50%

Light Balance at 100%

Ambient Light controls whether the ambient light is distributed equally throughout the scene or is coming from the sky. For the purposes of all the examples and exercises in this book, we will keep it at 100% or completely from the sky, unless otherwise specified.

Light Color and **Ambient Light Color** do exactly as the name says: create the color of the direct sunlight and the ambient light. In the Spectral model the Light Color is disabled as all the atmosphere settings contribute to create a realistic non-linear color of the sunlight. The Ambient Light Color is best left at the default light blue shade as all Earth lighting is close to that color.

Lighting Model Extra Settings

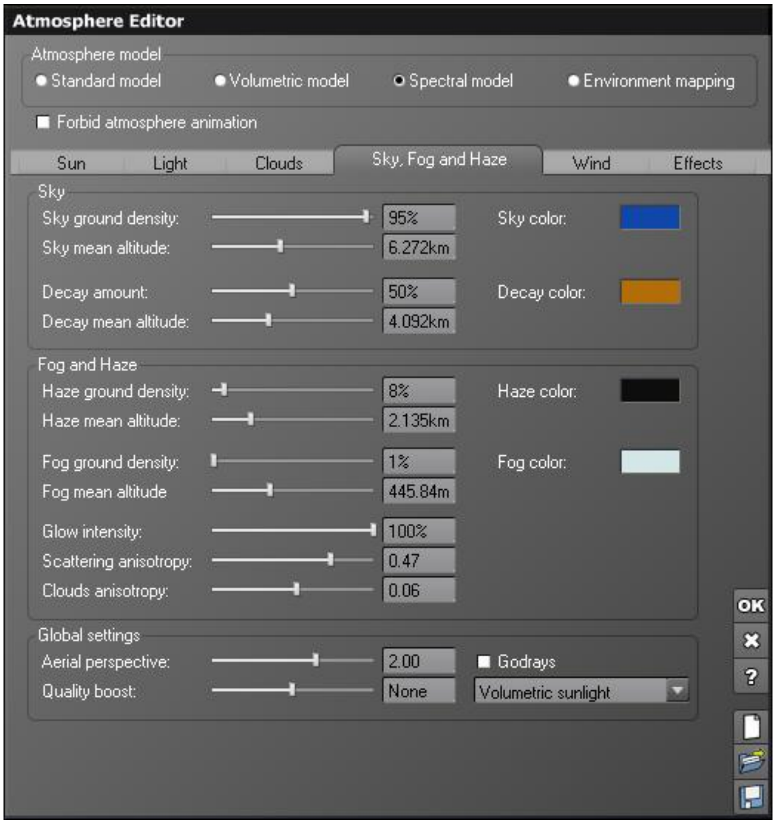
On the right side of the Lighting Model control group where we selected Global Radiosity. We will explore these controls in the next section.

Clouds

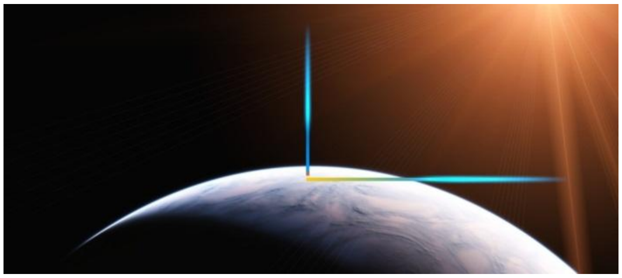
The Clouds tab allows us to create cloud layers in our scene. We will explore this portion of the Atmosphere in the Advanced Atmospherics chapter.

Sky, Fog and Haze

The **Sky** control group allows us to define the color, intensity, and altitude of the sky. In realistic terms, it lets us decide how blue and how "high" the sky should be. The **Decay** controls decide how much the light color is decayed.



Decay is essentially the color decay of the sunlight as it passes through greater distance in the atmosphere near sunrise or sunset as opposed to late morning or afternoon light which passes through less distance in the atmosphere, as shown in this diagram. The orange hue we see during twilight is caused by this effect as light passes through more airborne particles and eventually ends up dissipating part of the light spectrum.



In Vue terms, when you are creating a twilight sky, Decay lets you paint the sky and the light in orange tones. This will happen when the Sun is below the Decay Mean Altitude. Objects below that altitude are all affected even when the Sun is above that level.

Refer to the ‘Extreme Decay’ in Chapter 9 for examples of usage of Decay.


Sky Density at 0%
Sky Mean Altitude at 0km


Sky Density at 100%
Sky Mean Altitude at 3km


Sky Density at 100%
Sky Mean Altitude at 10km


Decay Density at 0%
Decay Mean Altitude at 4km


Decay Density at 50%
Decay Mean Altitude at 4km


Decay Density at 100%
Decay Mean Altitude at 4km

Starting with Vue 8.5 the atmosphere altitude has been extended to 100km. Refer to Appendix B for information on how to take advantage of these changes.

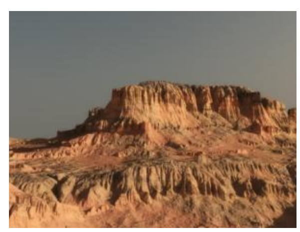
Do note that even though you can extend the Sky, Decay, or Clouds to 100km, it does not invalidate any <10km scenarios we will explore in this chapter. In fact, for best *visually correct* (if not technically correct) results it is recommended that a below 10km atmosphere be used.

Haze Ground Density, Haze Mean Altitude, and Haze Color

Haze as interpreted from the real world into the Vue world is a complex mix of water, dust, and other airborne particles that fill the lower atmosphere.


Haze Ground Density at 0%


Haze Ground Density at 20%


Haze Ground Density at 100%

The **Haze Ground Density** is very much like the Sky and Decay densities. It controls the density of the color from the **Haze Color** that is added to the atmosphere. One of the unique techniques we will explore in this chapter is using “Black Haze”, an extremely dark gray – in fact, almost black – for the Haze Color, and only 5% to 12% for the haze.

The **Haze Mean Altitude** decides how far the haze goes up into the atmosphere. In most scenarios, a value of 800m to 1.2km is more than enough.

Fog Ground Density, Fog Mean Altitude, and Fog Color

The Fog controls work exactly like the Haze controls however both are independent of each other allowing you to add fog in your scenes.

Fog can be used to add an extra layer of 'particles' in the air aside from the Haze. In real world terms, if haze is caused by dust and other particles, fog is caused by intense water particle accumulation. When Fog and Haze overlap, their colors interact with each other, so in practical terms, you can use Fog to change the color of the haze or to simply show fog in your scene.

This method can be seen in practical use in the Godrays section later in this chapter.

While this is a useful thing in some cases, my own techniques use a different method for Fog. Therefore in just about all of my scenes, the Fog is set to a low Density setting of 3%, the Mean Altitude about 500m, and the Color a soft white-blue. These settings offer an almost negligible amount of fog to the scene, just enough to make the horizon look a little whiter for more open scenes but not enough to affect the color of the atmosphere in any major way.

Glow Intensity

The Glow Intensity controls how bright the glow of the Sunlight is in the atmosphere. This value is added on top of the Light Intensity in the Light tab. While the Light Intensity is comparable to the Exposure of a real world camera and controls the brightness of the scene uniformly, the Glow is more directional coming from the Sun.

Glow at 0%

Glow at 10%

Glow at 100%

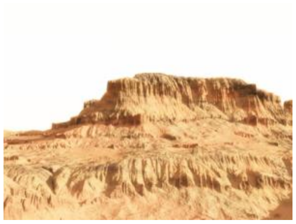
It is very useful to use the Glow Intensity in conjunction with appropriate values for Scattering Anisotropy and Cloud Anisotropy.

The minimum and maximum amounts on the Glow Intensity slider are 0% and 100% respectively, but the text entry field will accept any values. In certain situations, values of 200% or higher can be helpful.

Scattering Anisotropy

Scattering Anisotropy is one of the most important aspects for making the sky and the atmosphere look realistic with sunlight glow. This setting takes the Glow Intensity and disperses it across the atmosphere. A minimum value of -1 makes the glow come closest to the camera, while the maximum value of 1.0 keeps the glow restricted absolutely close to the sun.

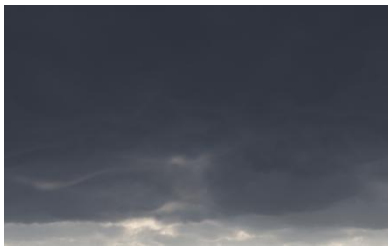
The Scattering Anisotropy value helps create Godrays. In most situations, keeping the value between 0.45 and 0.65 is helpful as it adds a realistic glow in the "middle range" of the atmosphere.


Scattering Anisotropy at 0.0


Scattering Anisotropy at 0.75


Scattering Anisotropy at 1.0

Cloud Anisotropy


Cloud Anisotropy at 0.22


Cloud Anisotropy at 0.90

Cloud Anisotropy works just like Scattering Anisotropy, but is restricted only to the clouds in the scene. This setting decides how light is dispersed inside, especially on the edges of all axes, so a cloud bulge anywhere would get proper highlighting if the light hits the back of that formation.

The minimum value -1.0 means the light from the back is dispersed, or goes through the clouds, as little as possible, while the maximum value +1.0 makes the sunlight punch through the clouds no matter how dense they are. It is worth noting that even at maximum value, only the sun's disc and corona will show through, not the entire sky.

Aerial Perspective

One of the most important settings in your atmosphere is the Aerial Perspective. This setting helps you define the scale of your scene. The default setting of 1 makes your scene real scale – one square meter of your scene is processed as one square meter.

When you enter a higher value, it acts as a multiplier of various sorts. All the internal calculations of the atmosphere: clouds, lighting, haze, dust particles, and so on are multiplied and the effects of such phenomenon are enhanced making your scene look bigger than it is.

For example, a 100 square meter terrain can look several kilometers larger just by turning up the Aerial Perspective value.


Aerial Perspective at 1


Aerial Perspective at 200


Aerial Perspective at 500

Of course, all of this comes with a reminder of the warning “watch the scale” as we discussed in the first chapter. That is always the secret of getting your scene right. But let’s explore *why* it is so important.

The Importance of Scale

One of the easiest aspects of realism to ignore while working in Vue is scale. Sizing settings such as Aerial Perspective, Material Scale, Function Scale, EcoSystem Instance Scale, etc. are all so useful in controlling the size of individual aspects of the scene to make it “fit”. But for enhanced realism, such settings should correspond to each other.

The easiest way to do it is to use 1:1 scale. That is, if you are creating a large mountain terrain, make sure it is sized in hundreds of meters just like a real terrain would be. The Aerial Perspective should be kept at 1.00 as well.

Never be afraid to create a terrain or other large scale object that goes outside of the circular scene perimeter seen in the Top View when you zoom out.

Aerial Perspective Comparison: 4x4 km terrain with values of 1.00 and 5.00

For more information, refer to Chapter 9, Perception.

Light Reflection Techniques

Sometimes light reflection even through Global Radiosity may fall short of your requirements. Perhaps you may need some more artistic effect or a specific look that Global Radiosity default settings may not have. In such a situation there are a few lighting techniques you can use to boost the Global Radiosity settings.

Below is our initial render. The scene is a tropical shallow water scene set next to a rock formation partially submerged in water. In this scene we will explore the three main aspects of reflected light: Global Radiosity Gain, Bias, and Sky Dome Lighting Gain.

Note that you need to have the Sunlight-Ambient Light Balance slider set towards Sunlight (70% or more) to have the pronounced effect of Global Radiosity Gain. If your scene contains too much ambient light, Global Radiosity Gain will not provide the desired effect and may muddle the ambient light.

Basic Global Radiosity enabled scene.

The ambient, reflected light in this image is very basic and unpronounced as defined by the default settings we are working with. Our goal is to have it be dramatic so we will be changing the atmosphere for that goal while keeping the scene realistic, and believable.

Global Radiosity Gain

First of all, we need more reflected light. So let's change the Global Radiosity Gain to 0.5. We do not want to overdo this as too much reflected light makes a scene look much smaller than it should be, like a fake model.

Global Radiosity Gain increased.

With a 0.5 Gain, we now see more reflected light. You will notice the reflected light most in the far left side of the image.

Remember, Gain represents the intensity of the reflected *direct sunlight*. If you use extreme settings, you may have more reflected light than required and may cause the lighting of your scene to appear imbalanced and fake.

Sky Dome Lighting Gain

Sky Dome Lighting Gain added.

Now that we have more reflected light from the objects themselves, we need to balance the image by adding light being reflected from the sky itself. In the real world, even though there is no direct light coming on an object, if the sky is clear and blue the entire scene would be tinged with blue-tinted reflected light. Try looking at a white colored house or wall that on a clear day is well lit but not directly lit by the sun. Think of the sky as a big solid ceiling.

To apply this effect to our scene, we change the Sky Dome Lighting Gain to 6.0.

Now our image has more balanced ambient light, with a Sky Dome Lighting Gain of 4. You can note the effect of Sky Dome Lighting Gain in the top right corner of the image in the nook of the rocks.

Global Radiosity Bias

Lastly, for a more pronounced reflected water effect we want our reflected light to be teal/blue colored since the color of the tropical water should color the rocks in that tone. To do this, we will bias the Global Radiosity reflected light color. We will change it to a teal-like value of RGB 36, 60, 50.

Completed scene with all aspects of reflected lighting tweaked.

With a Bias color, our reflected light is finally where we want it.

Force Reflected Light

The techniques we have seen so far work globally - across the entire scene - but sometimes you may need to reflect light on only a selected portion of your scene.

In the real world, photographers use reflectors to do this. A large piece of reflective colored board is angled to reflect the light on the subject. So if Vue's Global Radiosity engine mimics real life, then we can use the same method in our scene.

Let's explore this in this scene. The face of our character is lit from the right so the left side does not have too much light on it. We can use Global Radiosity Gain or Sky Dome Lighting Gain to raise the ambience but that would change the look of the entire scene. We only want to add more reflected light on our character.


Base scene with character without any reflectors in place.

To do this we will add a reflector object. Any basic primitive object would do the trick. The cube is by far the most convenient. So let's add a cube to our scene and resize it to approximately our character's profile. We now place this cube next to the character but out of the camera's field of view. Finally, we will change the material of the cube to a bright white.

This new render clearly shows how the reflected light adds more lighting to the character's shadowed half. Such a technique can be very handy when you are creating "hero" images or movie poster style images.


Base scene with character lit using white reflector cube.

Now just like with a white reflector, you can use colors to add more dramatic effects. For example, let's say we want our character to look as if she is walking close to a flowing river of lava. But we do not want to change the layout of the scene to show the lava. We work with the perception by insinuating that there is a lava flow nearby. One instinct a 3D artist might have to do this would be to add a low-intensity orange colored light off camera. While that would be acceptable, a spot or omni light would not be appropriate for doing this as we need a more broad scale lava glow/light coming over to the visible parts of the scene to imply that there is lava off camera. To get this uniform large scale glow that would not be possible with lights, we create another cube - this time more flat and low - and place it in front of the character. We add a red-orange material to the reflector cube.

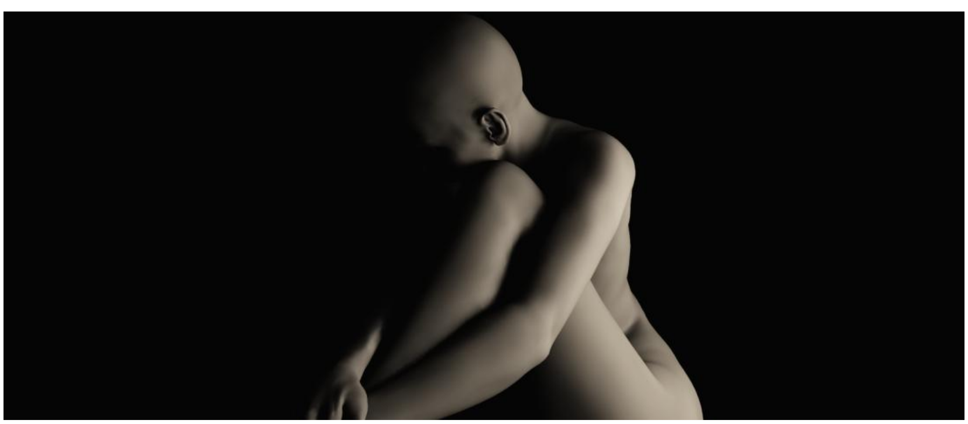
Now our image looks like this. Notice how the cube placement throws red-orange light on the character but not on the boulders behind her.

Reflected 'lava light' on character.

To add more realism, a colored fractal or other texture can be applied to the reflector. With Global Radiosity it will reflect textured light. For more intensity, the Luminous factor of the material can be used at low values as well.

Advanced Force Reflected Light

Taking our reflector technique, we can use an artificial light such as a Spotlight instead of the direct sunlight for use in interior or Sun-less scenes. Instead of pointing the light directly toward your subject, place a reflector near the subject and point the light on that reflector. The image below was created using a white reflector cube with a 500 power spotlight pointed at it.



Completely indirect lighting

Exploring Lighting Scenarios

Now that we know the exact details of how the Atmosphere Editor works, we will explore several different lighting scenarios and look at what specific settings in the Atmosphere Editor help create the core of these atmospheres.

Key Observation Points

As you go through the examples in this section, these are the main points you should try to observe visually in detail. The numbers corresponding to these points are written below each image, but the impact they have on the visuals because of their interaction with the rest of the settings is important and can only be observed visually.

The scale – whether 1:1 or otherwise – is what sets the stage for each scene. A 4 square kilometer terrain can look very different if the Aerial Perspective is set beyond 1.00. In the same way, different amounts of haze will purport different observable scale.

At the end of the day, the numbers are there just to give your bearings. The actual scale is achieved in the visual result regardless of what numbers were used.

The best non-lens depth of field is created by haze. Good haze values, depending on the scale of the scene, will help you differentiate the layers in your scene without washing them out.

Haze, Aerial Perspective, Fog, and other atmospheric settings are brought to life by Glow Intensity. This will allow sunlight to become a more integrated part of the scene rather than just a backdrop feature.

With the right Haze settings, Glow Intensity allows you to create remarkable sunlight effects, including subtle Godrays.

The final item to look for is reflected light. Reflected light is the most realistic, natural method of creating ambient light and controlling shadow darkness.

For this massive 4 x 4 km mountain made from a Standard Terrain of a detailed 4096x4096 resolution, there were three goals: show the immense size of the mountain, create drama in the atmosphere, and hide the tiny imperfections in the rock material under the snow in the terrain. For many artists, there is a struggle between maintaining both speed and quality on a commercial project – which was the case for this project.

For the first goal of showing the immense size, first the Aerial Perspective was set 1.00 – which is real size, so every inch of the 4x4km terrain would be processed in realistic scale. This would make handling the atmosphere slightly more difficult than normal, but the end result would be worth the pain. The EcoSystem of conifers populating the terrain was set to the normal 1.0 scale for Instance Size. Remember, when you go with Aerial Perspective 1.00, everything should be scaled to normal or 1.0 size. Because of such a large and dense EcoSystem (Dynamic EcoSystems come in so handy here!) the size of the mountain was shown quite nicely.

For the second goal of dramatic atmosphere, the haze (more Black Haze) was set to a Density of 38%. Very high for the normal settings I prefer, but I compensated by keeping the Haze Mean Altitude to only 650m. That meant that only the lower half of the terrain would be covered in haze while the top of the mountain would be pretty much haze-less and receive brighter sunlight enriching the perception of the massive size. Volumetric Sunlight was turned on so the early morning shadows on the terrain would become volumetric and create yet more grandness. Global Radiosity Gain was set to -0.15 to hide reflected light which should not appear at this time of day on such a large terrain. The intense morning light was created by pushing the Glow Intensity to 500% and moving the Scattering Anisotropy to 0.40 which helped create the volumetric sunlight. Fog was completely removed, letting only the Haze and the Aerial Perspective create the air density effects.

The Sky itself was the default blue, but only 65% of Density and a Mean Altitude of 4.85km. The Decay on the other hand was at 80% Density and with a Mean Altitude at 2km. This resulted in a crisp orange-tinted light with clear blue skies.

For the final goal of hiding some unwanted non-realism in the materials, the dark shadows caused by the Sunlight Light Balance of 100% helped! The Sky Dome Lighting Gain of 15 with a Skylight Color of RGB 36, 68, 145 created a darker sky with still enough ambient light to show the shape of the terrain in the shadow in the "west" but keep everything dark enough. The contrast between the dark shadows and bright highlights hide the fact that the underlying rock material has too much tiling.

This desert scene needed stark shadows with soft skylight. All objects were scaled to actual size, spanning 4 x 4 kilometers for the main terrain, so an Aerial Perspective of 1.0 was used. 88% Light Balance biased towards Sunlit-shadows was used.

A Sky Dome Lighting Gain value of 7.00 with a soft blue tone and 8.8km Sky Mean Altitude with 90% Sky Intensity create uniform ambient light coming in from the sky, adding light to all the dark shadowed areas without removing their starkness.

In this scene, an Aerial Perspective of 2.00 is used to enlarge the effect of the atmosphere, especially for the distant areas. By pulling in the Scattering Anisotropy to 0.33 and Glow to 65%, the sunlight comes in a soft white wave. The sunlight glow is enhanced by a dark gray haze of 18% intensity, climbing up to 900m.

The desert atmosphere from the previous page is enhanced by using 2.0 Global Radiosity Gain to enhance the reflected light three times more than default.

Such high intensity reflected light increases the illusion of scorching heat rolling off the terrain in the desert. Notice how the shadows are infused by the color bleeding of Global Radiosity Gain.

This ground level shot requires lighting that would normally be considered overexposed for a visible-sky shot. A 3.00 value for the Aerial Perspective enhances the lighting by multiplying the atmosphere calculation by three.

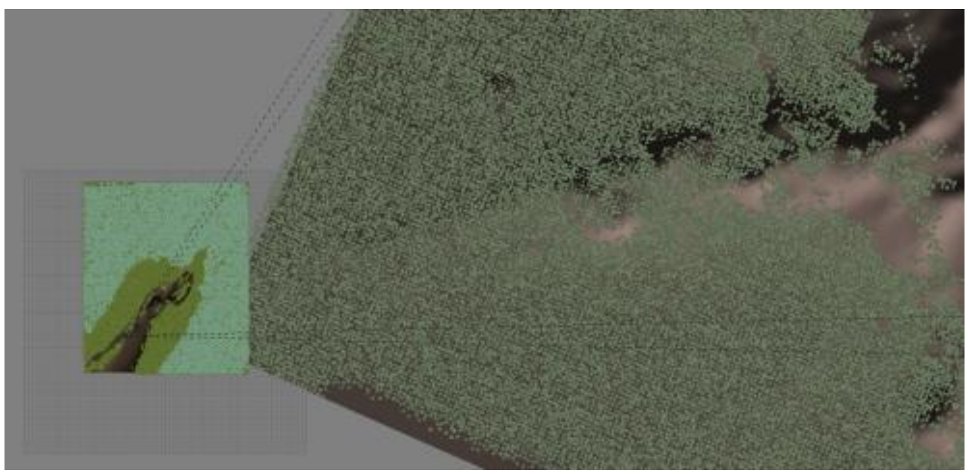
100% Glow Intensity coupled with 0.29 Scattering Anisotropy bring the 100%/5.67km Decay into the foreground and light up the grassy EcoSystem. The overexposed Haze is created by 200% Haze with a Mean Altitude of 400m. This is recommended only for enclosed places such as this. Although this place is semi-indoors, it still uses Outdoor type Global Radiosity optimization.

In this scene, the surroundings play just an important part as the atmosphere when it comes to lighting. This moss covered log is surrounded by trees in such a way that the far background trees are appropriately shadowed by the trees in front of them. This creates the illusion of a larger forest environment without having to actually create one.

The dark shadows are intended to portray a morning dank enough to encourage moss to flourish yet with enough strong sunlight to create a hot and humid feeling.

Sky Dome Light Gain is kept low at 1.5 with a Light Balance of 89% to create the darker shadows while creating enough reflected light under the log to show off details from the displacement. Global Radiosity Gain was kept at a neutral 0.0 to avoid excess color or light bleeding.

This scene is set high up in a snowy mountain region with coniferous forests surrounding the primary mountain peak. The scale of the objects in the scene becomes very important for the haze required to show proper depth. The mountain is 4 km^2 and approximately 1.3km away from the Camera. Using too much ambient light can break the foreground, but with a high mountain size and distance from camera the mountain automatically receives a lot of ambient lighting. For the ambient light we use a Sky Dome Lighting Gain value of 6. An Aerial Perspective of 10 is used to enhance the scale of the large terrain.



Scale difference between the foreground terrain and the background mountain

The sun is positioned to the "west" or right of the scene and a little over 40 degrees. With a low Decay Amount 66% but high Decay Mean Altitude 4.0km the overall scene retains a pleasant blue tone with just a hint of orange decay distributed evenly across the scene.

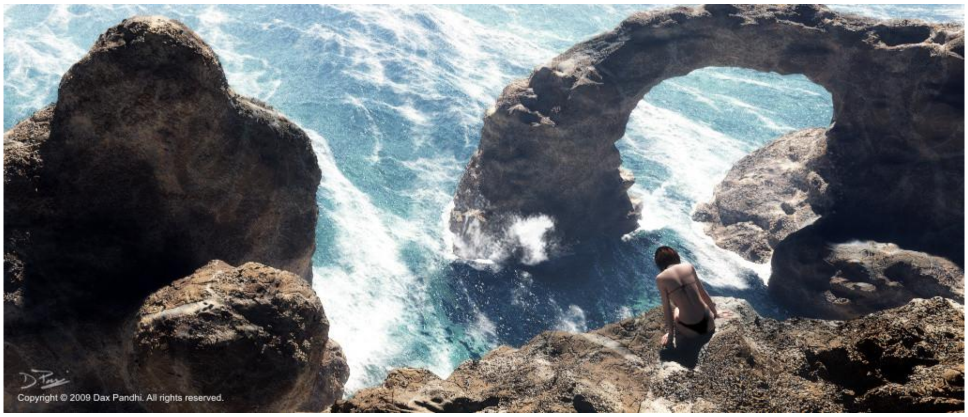
The Light Balance value is 94% which makes sure there is enough ambient light but retains dark shadows for our foreground. Note the scale of both the foreground and background in the top view. The scale is paramount in a scene like this.

For this tropical scene which needed to portray a murky sense, no sky is visible. But the Sky settings still greatly influence this scene. Hot tropical weather is often best shown with an overhead sunlight. In this scene, having the sunlight at a 12'O clock position further helped put highlights in the water.

Because of such a high sun the shadows needed to be dark, so the Light Balance slider was set to 96% (towards Sunlight) and the Sky Dome Lighting Gain was set to 3.0 which amplified the ambient light coming in from above. With these settings the shadows remained dark but there was just enough ambient light to properly show the underside of all the rock formations.

The vegetation in a scene like this requires consideration in terms of lighting since very dark shadows will never be able to show the density of the vegetation on such a large scale. Therefore having Sky Dome Lighting Gain and Global Radiosity to provide the soft ambient and reflected light on the vegetation helps portray the depth and density of this tropical undergrowth – especially on the foreground objects.

Finally, another important scenario where Global Radiosity reflected light is needed is when you have arched terrains like the ones in our foreground here. If the underside goes unlit, it may look too deep and dark making it look somewhat unreal. Besides, the ambient lighting lets you show off the rock formation's underside properly which is an excellent place to create some detail.



For this bright, sunny seaside scene, an intense sunlight and a feeling of grandness were required. The Aerial Perspective is set to 20 to enhance the feeling of depth, while the Light Balance slider was kept at 90% sunlight leaving very little ambient light. To enhance the brightness of the sunlight, a 0.80 Global Radiosity Gain value was used with the Gain Bias color set to a very dark blue (RGB 5, 5, 18) .

For tropical scenes with such bright light, dark shadows are important but to convey depth in the different layers of the rock formations, the farther a layer of rock is from the camera the hazier it should appear. A very common mistake would be to just increase the Haze Density. For a much clearer and accurate result, the rock formations should be taken farther and made larger.

While a good Sky Dome Lighting Gain value is useful to show the bright blue of the sky dome being reflected on the ground, a Global Radiosity Gain value of 0.5 or higher (0.8 in this case) is recommended as well. Because of such bright sunlight, there is bound to be a large scale light reflection off of all these rock formations and the color of those rocks will bleed into the reflected light as well (as seen in the beginning of this chapter).

Simplicity is the key to this image. All the settings for the Atmosphere are shown in the screenshot below. The main principle behind the lighting in this scene is how the Glow Intensity and Black Haze intercept each other. An almost excessive Aerial Perspective of 20 makes the small terrain seem grander than it is.

When you keep the Decay strong but restricted to less than 4km, it paints the scene objects in an evening-like orange hue but prevent the blue sky from turning too orange. The resulting effect is more believable than a full out orange sky of a sunset. Another aspect to consider in this scene is that the sun is not visible. As in photography, 3D renders will look better if the Sun is kept out of the frame.

This stark desert scene uses an Aerial Perspective of 1.0 with a strong blue haze of 8% and height 2km. The scene itself uses terrains that are 16 kilometers large, creating this deep sense of grandness.

A contrasted Light Balance of 98% with Global Radiosity Gain of 1.55, mixed with Sky Dome Lighting Gain of 8.0 create the believable dark, desert shadows that still show some detail through reflected light.

You can learn more about using massive terrains to create a sense of majesty in your scenes in Chapter 9, Perception – Massive Terrains.

For this cold spring morning scene (supposed to be set somewhere in the Rockies) a bright Glow Intensity of 80% with a high Scattering Anisotropy of 0.65 is used. The Sky Mean Altitude and Density are maximized at 10km and 100% respectively, while the Decay Density is set to just 40% and the Mean Altitude at 3.66km, giving just a hint of orange in the light while keeping the skies electric blue.

The shadows are kept dark by using a low Sky Dome Lighting Gain of 1.00 and Global Radiosity Gain of 0.0. The Light Balance is kept at 94% to provide just enough ambient light while keeping the shadows dark.

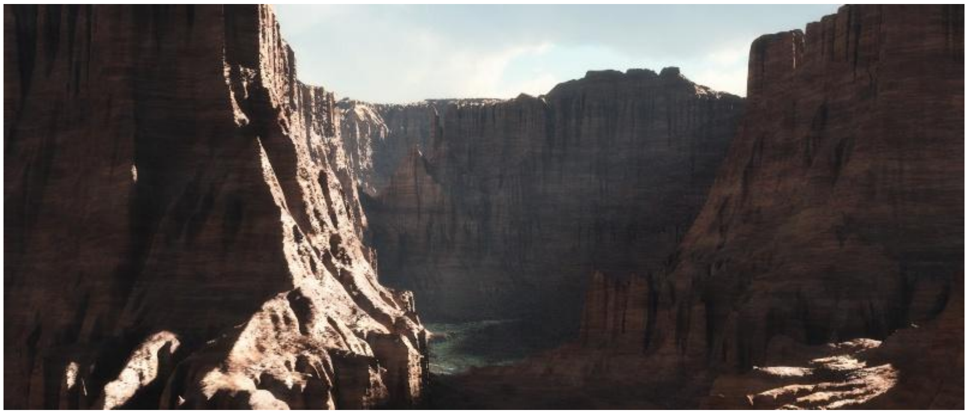
A Haze of 8% with an Aerial Perspective of 3 is used.

This scene uses a maxed out Sky Density and Mean Altitude with very low Decay. The Glow Intensity is very high at 80% and the Scattering Anisotropy is 0.45.

The resulting vibrant sky reflection in the water and the bright sunlight on the object in the water create the “fresh blue morning” feeling. The sun is positioned so as to throw just a little bit of glow and highlights in the water in the lower right corner.

The additional glow of highlights on the water – both white 'hotspots' and the blue of the sky reflected in the water – is thanks to high Lens Glare values in the Post Render options.

We will delve deeper in the concepts for such water scenes in the Waterscapes chapter.



For this desert canyon scene, Godrays became an important factor. A low 8% Black Haze (RGB 13, 13, 13) covers this terrain at an Altitude of 2.1km. The Aerial Perspective is 22.00 on this 1.5km x 1.5km terrain. A low Scattering Anisotropy of 0.12 and a Glow Intensity of 100% create the intense Godrays in the Haze.

In this high altitude scene, the light intensity plays an important role as the higher you go the brighter the sunlight becomes of the thinning atmosphere. Not only are we using a very bright sky, with both the Sky Mean Altitude set to 10km and the Sky Density to 100%, but there is very little Decay.

At high altitudes the sky becomes very blue and there is very little atmospheric particles (especially during the day) to make the light become orange. To further accentuate the high altitude of the scene, we use a strong Sunlight-Ambient Balance value of 90% with only 10% ambient light. However, the ambient light seen here is another strong value of 6.0 for the Sky Dome Lighting Gain which adds the blue ambient light to the strong shadows.

The Sunlight uses 2° soft shadows to show the scale of the scene by blurring the shadows being thrown. Further, the MetaClouds use a low Opacity of 70% to create the illusion of very intense sunlight. We will explore that in the Clouds section.

In this high altitude scene, a strong Haze helps create the “way up there” feeling and creates Godrays in the top-right portion.

The clouds are highlighted by an extremely strong Glow Intensity of 200%. The Scattering Anisotropy is kept at 0.50 to avoid brightening the scene too much.

The scene uses an Infinite Procedural Terrain, so to keep everything in proportion, a 1.0 value is given for Aerial Perspective.

A high Glow Intensity of 100% with a very low Scattering Anisotropy of 0.21 causes the 100% Density / 10km Mean Altitude Sky to go completely white and the whole scene ends up looking extremely bright. Sky Dome Lighting Gain of 3.00 with a Global Radiosity Gain of 2.0 causes a lot of reflected light. Low MetaClouds with extremely small values of density (0.5%) create mists in the scene. A dark blue Skylight color helps create the blue reflected light in the dark shadows of the terrain.

In this sunset scene, a high Glow Intensity of 55% is used with a Scattering Anisotropy of 0.97 to intensify the glow of the Sun. The mountain creates perfect shadows while the sunlight is positioned to properly highlight the foreground. The haze is set to **100%** Density and 1.19km Mean Altitude for the perfect glow to be visible in the sky. The Haze Color is default. The Aerial Perspective is kept at 2.00. Sky Density is 78% at 4.8km Mean Altitude and Decay is 77% at 6.8km Mean Altitude.

This surreal image was created by using a steel blue Haze (RGB 43, 49, 63) with 38% Density and a Mean Altitude of 620m. The 21% Glow Intensity merging with a Scattering Anisotropy value of 0.12 along with a Light Balance of 93% creates these long shafts of Godrays on this large stretched terrain. The key to Godrays lies in a strong haze, lots of shadow darkness, and a relatively low Aerial Perspective (3.0 in this case). The Scattering Anisotropy brings the light from the sun visibly forward, and once your camera is in a shadowy area, the strong Godrays will appear in the Haze.

This deep jungle image was made using only a single procedural terrain with a dense jungle EcoSystem loaded on it. The terrain was stretched high enough to create this valley. It also blocks the sun for most of the area visible to the camera. A dull blue Haze (RGB 84, 97, 109) with a Mean Altitude of 1.4KM and Density of 6% encompasses the landscape. A full 100% Glow Intensity and a Scattering Anisotropy of 0.12 bring the bright sunlight closer, but a Light Balance of 94% keeps the shadows deep and murky. MetaCloud fog enhances the misty effect of the scene.

The thick jungle atmosphere is achieved by using a large value for the Aerial Perspective (10.0) while keeping the terrain smaller than the scale should normally be. This increases the "blueness" of the Haze. Sharp overhead lighting increases the damp and humid feel of the rainforest. Global Radiosity Gain of 1.5 and a Sky Dome Lighting Gain of 2.0 were used for this atmosphere. The Decay Amount was kept low to avoid the lighting from getting too orange tinged.

Two scenes were rendered – one with +0.35 Exposure in the Post-Render Effects, and one with 0.00 Exposure. They were blended together in a paint program to have the center be a little darker and less blue than the rest of the scene.

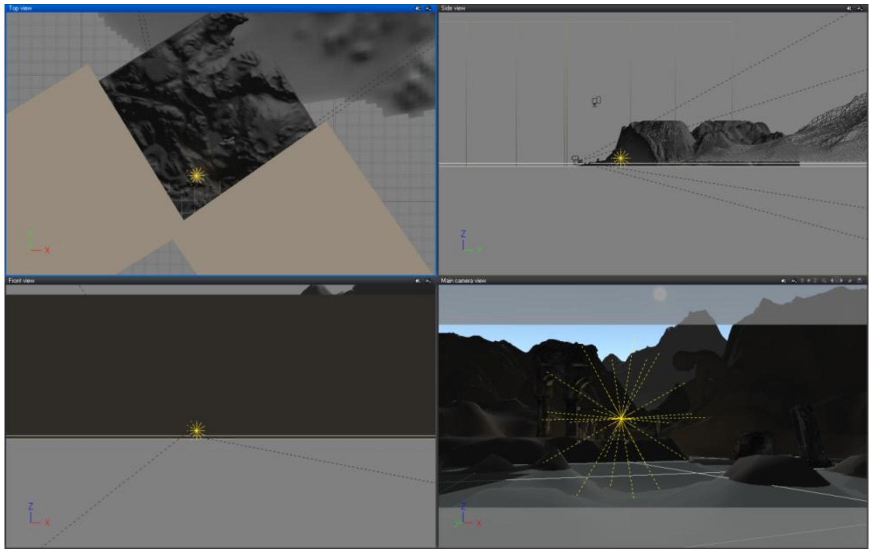
For this forest interior style shot, the Sun is positioned slightly to the left and away (towards the far horizon) from the Camera but still high enough to give us highlights on the top of the trees and the rock obelisk. A small blue (RGB 180, 210, 255) Omnidirectional Light is used to create fake ambient light. Two simple terrains are used as "light blockers" so the trees are not lit too much and the "holes" between the trees where they are not dense enough do not show the empty backdrop.

In this forest scene the same lighting as in the previous scene is used, however the Decay value is increased and the Sun put closer to the horizon to create the twilight effect. The EcoSystem of HD Trees in the back is used for creating the interesting shadows on the foreground HyperTerrains. The Sun's Shadow Softness is set to 2º to create the soft evening shadows.

For lighting the interior of the cave, a strong Global Radiosity Gain of 3.0 is used. The sun is positioned so that only the rim of the cave is highlighted. The Sunlight-Ambient slider is set to 95% so only 5% ambient light is available. The excess Sunlight helps boost the intensity of the Global Radiosity Gain which illuminates the inside of the cave.

With the help of strong Global Radiosity Gain (1.5) and Sky Dome Lighting Gain (3.5), and the Sun positioned overhead but slightly tilted towards the camera, the HyperTerrain rocks in this scene are brightly lit with reflected and ambient light spreading in the shadowed regions. The bright highlights on the rocks also create the white reflections in the water. The Glow Intensity is set to 125% and Scattering Anisotropy is set to 0.45 to intensify the sunlight.

For this twilight-lit grotto, Global Radiosity plays a dominant role. Almost 2/3 of the scene is lit only by indirect light. The direct light is used only to highlight the large mountain and the statue in front of it. The main subject area of the four explorers is lit by Global Radiosity using a Gain of 2.5 and a small 120 power Omnidirectional Light on the “torch”. A Light Balance of 90% creates the deep, dark shadows. A large cube is set off-screen on both sides of the camera to block the Sunlight on the rest of the scene and create the illusion that the camera is inside a large valley.



Light blocking cubes placed outside of the camera's field of view.

Such large blocks can be a very creative mechanic for light blocking techniques. If you require shaped shadows, these blocks can be replaced by large terrains. They need not be high resolution since they only need enough resolution to create a good silhouette.

In this bright sunlit scene with cloud cover, a low Cloud Shadow Density is used with a large ambient value of 78% in the Sunlight Balance slider, along with a Sky Dome Lighting Gain value of 2.00 to create this bright blue sky. The color of the sky is owed to a custom blue color (RGB 17, 71, 171) with a Density of 70% and Mean Altitude of 10km, along with a low Decay Density of 62% at Mean Altitude 4km.

Extended Skies

The release of Vue 8.5 introduced extended sky limits. You can now take the Sky Mean Altitude to 100km. This new extension is provided to simulate what other planets might be like.

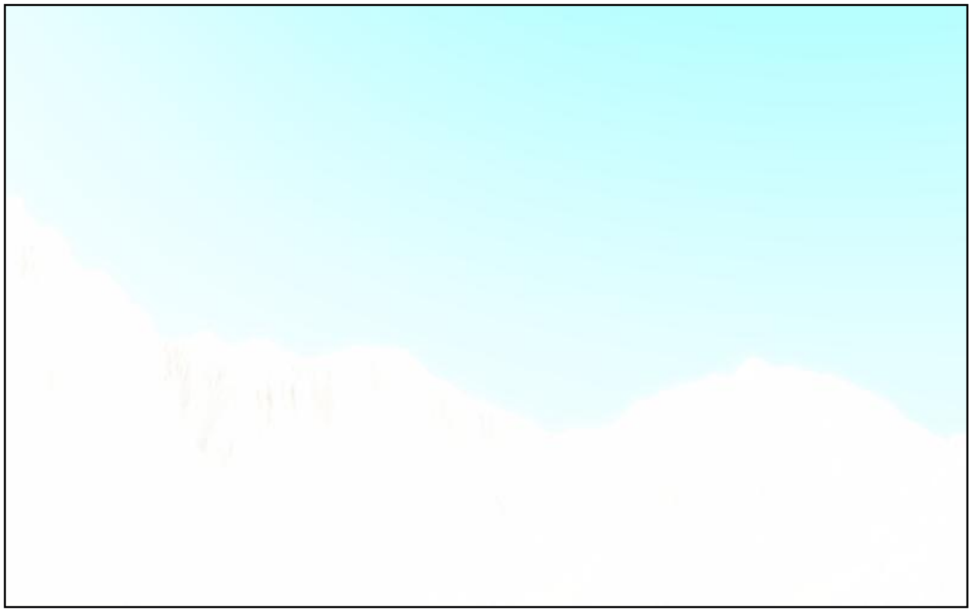
In the real world, on Earth, the Atmosphere does extend to 100km. At 100km, in the Troposphere, we have the Kármán line – the boundary at which the Earth's atmosphere ends and outer space begins. However the atmosphere level as far as visual coloring is concerned (as in, Vue's Sky Mean Altitude) the typical altitude is 8.8km.

Clouds are, generally, limited to the 18km/60,000ft altitude. While Vue does allow you to take them up higher than that, it is recommended that this option be used judiciously as it may make your scene unrealistic.

The main issue we will discuss here is how this affects the lighting of our scenes, as using large and previously unavailable values changes the scene atmosphere dramatically – and in some cases, fatally.

For example, we have this scene below which uses a 10km Mean Altitude and 100% Sky Ground Density: the maximum of the previous versions, and generally realistic values.

If we were to change that to the new maximums, say 100km and 100% respectively for the Mean Altitude and Ground Density, then the resulting image would definitely be labeled 'fatal'.



Whether you look at it on-screen or in print, this image is extremely overexposed. We had to put a black border on the image so it can be viewed somewhat in printed form.

These new levels are extreme and require subtle maneuverings we are not that accustomed to with Vue atmospheres.

By changing the Sky in a new way, we can get our old visuals back by using new settings. If we were to keep the 100km Mean Altitude but change the Ground Density to 20%, we get this image:

The new limits may at first seem harder to control, but they provide a massively improved degree of control over the atmosphere when you are dealing with atmospheric animations or extremely high-altitude renders on non-Earth environments.

You can also use this to fine tune your Earth-bound scenes, but careful calibration is recommended as unrealistic values may create unrealistic renders if you are unsure of your settings and their effects on your Vue world.

Higher density per kilometer of Altitude allows for brighter scenes and stronger lighting when dealing with dark, overcast scenes. Stronger light also means stronger Cloud Anisotropy for edge-lit clouds.

Planetary scenes can have greater atmospheric effects for dramatics.

10km skies (top) and 100km skies (bottom).

Additionally, the 100km skies create some fascinating effects along the terminator line and horizon.

Old 10km limits are still valid, so all your existing assets will keep on working as-is, and you retain the flexibility to move into the new limits too.

Advanced Atmospherics

Chapter 3

Enhancing the atmosphere

Indirect Atmospherics

Indirect atmospherics, or Cloud Radiosity, was introduced in Vue 9. When clouds are large enough, they reflect light and color to the ground. By simply checking this option, you enable a new level of realistic light/color interaction between atmospheric Radiosity and object Radiosity.

Indirect Atmospherics is part of the Global Radiosity lighting model.

A wide cover cloud layer over mountains in the early morning light.

Practically speaking, the effect of this can be big or small depending on the structure of your scene. If you have large, low hanging clouds the effect may be more apparent than small, wispy clouds at high altitude. Indirect atmospherics will also add to the memory overhead of the render and may not be worth it in certain cases – especially long animations. However, at the same time they can be cached along with the indirect illumination for the rest of the scene.

Indirect illumination covered in the Clouds section does the opposite – taking the reflected color of the landscape and bleeding it into the cloud's ambient coloring.

Cloudscapes

One of the most difficult areas to control and even more so to master are the Clouds. They play one of the most important roles in making a render look attractive, yet are the easiest thing to go wrong in.

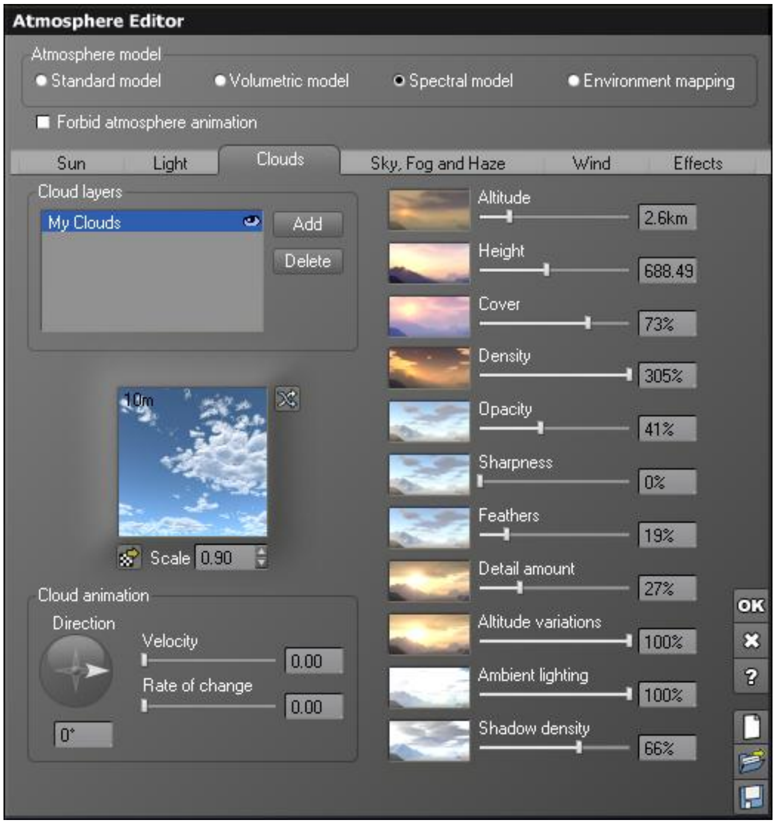
The major difficulty in achieving realism in clouds is to maintain a level of consistency between the lighting of the scene and the lighting of the clouds. Since cloud layer lighting can be changed through the ambient light settings, it is easy to ignore or forget what part the other settings such as Opacity and Density play in the overall lighting of the cloud layer.

In this section, we will explore what makes a cloud, how to push it to the limits by exploring several real examples and taking apart each and every setting for Cloud Layers, Cloud Materials, and MetaClouds.

Cloud Layers

Cloud layers are the best way to create large cloud fields for your scene. While you can change the actual fractal of the cloud layer using the Function Editor, the major changes you need are controlled by the Clouds tab of the Atmosphere Editor.

As we go explore what makes clouds, I would like to tell you of one simple rule to keep in mind whenever you make clouds: clouds are puffs of gas and water. They are not solid objects. If you use extreme settings, which are often seen in novice renders, they will look like solid objects. Avoid this at all costs, otherwise it will be a major setback to the realism of your scene.



If you would like to see the visual changes brought about by any of these settings, load up any default Vue Spectral cloud layer and drag the sliders to different, random settings.

Scale

This is the fractal scale of the Cloud Material. Increasing it changes the overall scale of the shapes of the cloud layer. This is useful if you take your Cloud Layer to high altitudes and it starts looking small or speckled. There is no recommended value for this as you need to experiment with it to find the right value for your scene's needs.

Altitude and Height

This is the altitude at which clouds start to form. The minimum is 100m and the maximum is 10km, however you can enter manual values. The height is the actual vertical thickness of the Cloud Layer, which can be from 100m to 1km, but again you can enter manual values.

Cover

This is the amount of cloud cover you get in your skies. At 100%, and unless you have a fractal in the cloud material that contradicts this setting, the entire sky will be overcast with the cloud layer. At 0%, you have no clouds at all, obviously. Sometimes minor changes to this setting can help get better cloud shapes, so experimentation is highly recommended.

Density

This is one of the key properties of a Cloud Layer. This controls how thick and dense the clouds appear. Normal recommendations are: 60% to 80% for thick, strong clouds; 10% to 40% for high altitude clouds; and 0.1% to 1.0% for extremely thin, wispy clouds. A 100% can be useful sometimes, but is not always recommended as it makes the clouds look like solid objects.

Opacity

This is the sister property to Density. Opacity controls how far light penetrates into the Cloud Layer. If you have a medium or low density cloud layer but want it to appear a little thicker and darker without making it denser, a high 80% to 100% Opacity can help achieve that effect. Or if you have a dense cloud layer where you want to show a lot of light coming through, a 20% to 50% value can help. Experiment with this property in conjunction with Cloud Anisotropy for some very interesting results.

Sharpness

This value controls the sharpness of the edges of the Cloud Layer. For soft, fluffy clouds this should always be 0. With higher values, light is processed more (as in low opacity) near the edges and the areas away from the edges have light processed less (as in high opacity). At the same time, the graininess of the Cloud Layer increases while also giving it a more “solid object” feel. A normal rule of thumb to use is to decrease the Sharpness (if using it at all) when the Density and Opacity go high, and increase it if Density and/or Opacity are low.

This is a sensitive setting so experimentation is the only way to understand its depth in a visual sense.

Feathering

Just as the name suggests, this setting affects how edges of the Cloud Layer ‘feather’ or elongate into small curving wisps. At 0, there is no feathering. At higher values small edge wisps start appearing. At higher values still they interact with other cloud elements (of the same layer) close by and start merging as if it were a MetaBlob.

For most situations, a value between 0 and 25 is recommended. This feature is very useful for creating thin, wispy clouds. However, excessive Feathering can lead to clouds having an “exploding pillow” feel with feathers flying everywhere.

Detail Amount

This is a very important setting. The Detail Amount controls how the “main clouds” are broken into little “cloudlets”. Taking the main Cloud Material fractal into consideration, Vue starts breaking the edges of your clouds into smaller clouds.

Low values under 10% are not recommended for realism. However there aren’t any recommendations for this value. You have to experiment to find the right amount of detail based on your scene. High values of 60% to 90% are useful when creating thin, high altitude clouds.

Altitude Variations

This setting pushes certain random areas of the Cloud Layer to an additional altitude. For example, if the Altitude is set to 200m, a high Altitude Variations value would cause the affected areas of the Cloud Layer to start at 201m, 202m, and so on. This creates an unevenness that adds realism to your Cloud Layer instead of the normal, fake looking even cloud layer. This is very useful for large clouds.

A range of 30% to 70% will create enough altitude variations to make the cloudscape look realistic, however values higher than that may make the clouds look predictable – an aspect you may want to avoid.

Ambient Lighting

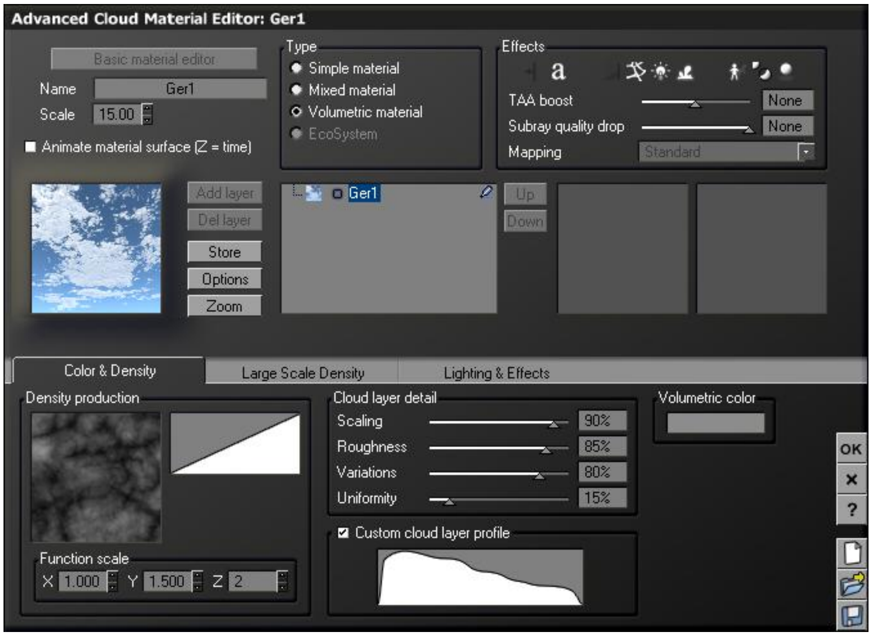
Ambient Lighting controls how much ambient (non-Sunlight) light is applied to the clouds. This is very useful if your scene appears dark because of the clouds. Do note that this applies the light uniformly across the entire Cloud Layer, so it should be used for tweaking. If your clouds appear too dark, it is better to control the Opacity first and then come to this setting. Too high values will make the clouds appear “separate” from your scene, in terms of realism. At the same time, if your clouds appear too dark because of no or extremely low Ambient Lighting, the same effect may happen.

Shadow Density

This setting controls how dark the shadows thrown by the Cloud Layer can be. If you turn on Godrays or Volumetric Lighting, this also affects how strong the Godrays generated by the clouds are. If your cloud cover is high, do not use high values like 80% or more as the shadows on the ground will become too dark. In fact, one rule to keep in mind is to lower the Shadow Density when the Cloud Layer has a high Density and Opacity. If you have to use dark shadows, for example when you require strong Godrays, then using a higher Sky Dome Lighting Gain is recommended.

Cloud Materials

A cloud layer is controlled by the settings we saw in the previous section, however, the actual shape of the clouds is controlled by the Cloud Material.



Scaling

Scaling controls how large the "cloudlets" that form a cloud can be. If you have a large-scaled cloud layer where you see too much small clouds or unnecessary detail in the cloud, you can fix that by changing the Scaling amount of the Cloud Material.

Roughness

This setting controls how the edges of the cloud (along all axes) break away from the central body of the cloud. When you have a very low Roughness value, the edges become smooth and look like soft ice cream rather than clouds. At too high values, they will break into too many small cloudlets and create an edge scattering effect. Depending on the scale of your Cloud Layer, you can experiment with the Roughness value to find the right amount of detail and cloud formation shapes.

Variations

The Variations setting allows you to control how varied the overall cloud formation will be. High values cause the entire cloudscape to have a unique, non-regular look. Extremely high values may cause excessive roughness and breakups in the Cloud Layer, so judicious use is recommended. Keep in mind that this setting works on the entire Cloud Layer and may not create changes that are visible at a smaller scale.

Uniformity

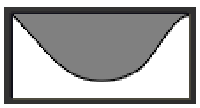
This setting will create a uniform or non-uniform distribution of shapes between the lower and higher altitudes of the Cloud Layer. If you use a high setting, the lower areas of the Cloud Layer (especially the base) will become just as rough as the top side. Lower values cause the underside of the Cloud Layer to become smoother and rounded.

Custom Cloud Profile

This Filter control allows you to change the density of each 'segment' of the altitude of the cloud. You can edit this filter to alter the density of the cloud in specific altitudes.

 This filter, for example, makes the cloud dense at the top, keeps it dense in the middle region and gradually lowers the density in the lower parts of the cloud.

 This default filter keeps the cloud dense at the top and uniformly decreases the density as it goes lower (towards the bottom of the cloud).

 This last filter maintains strong density at both the top and bottom parts of the clouds but thins it out in the middle area.

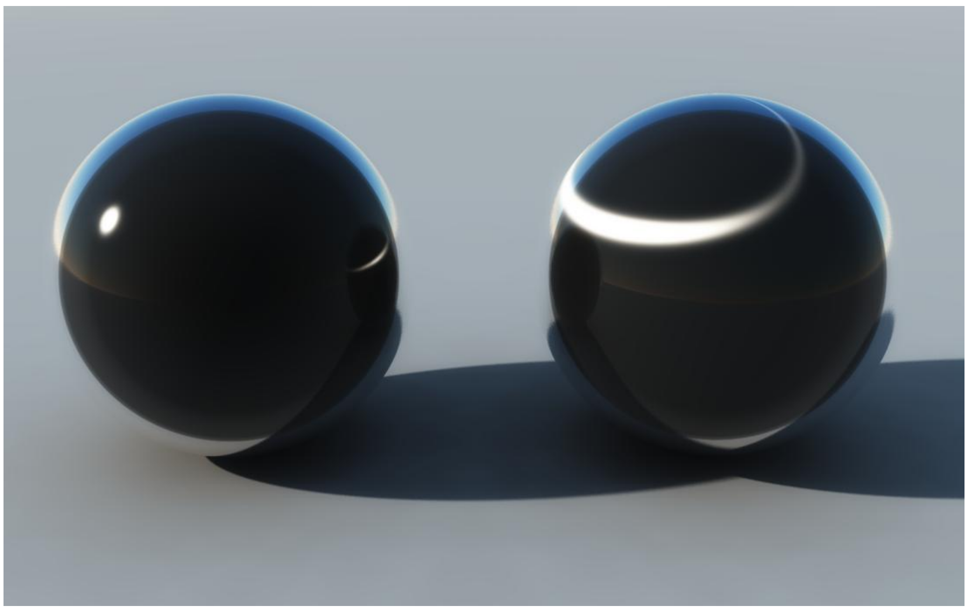
You can find out more about filters, how they work, and how to manipulate them in Chapter 6, Materials.

Cloud Anisotropy

Before we understand Cloud Anisotropy, we should talk about Isotropy and Anisotropy – in how it applies to our work.

In the Advanced Material Editor, you can change the Isotropy/Anisotropy of the shine of the material. If the shine is Isotropic, the angle of shine is uniform or homogenous. If you change it to Anisotropic, then the shine bends along the surface of the object relative to the direction of the incoming light.

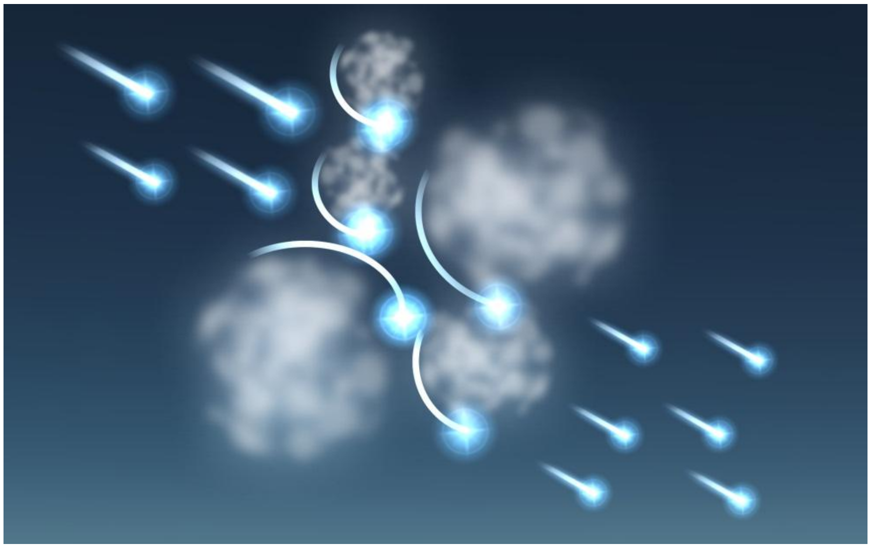
The sample image below shows two metal spheres that use the same material, however the sphere on the left uses 0% Anisotropic Highlights (it is isotropic) and the right sphere uses 99% Anisotropic Highlights (almost completely Anisotropic). You can see how the shine bends in the second sphere.



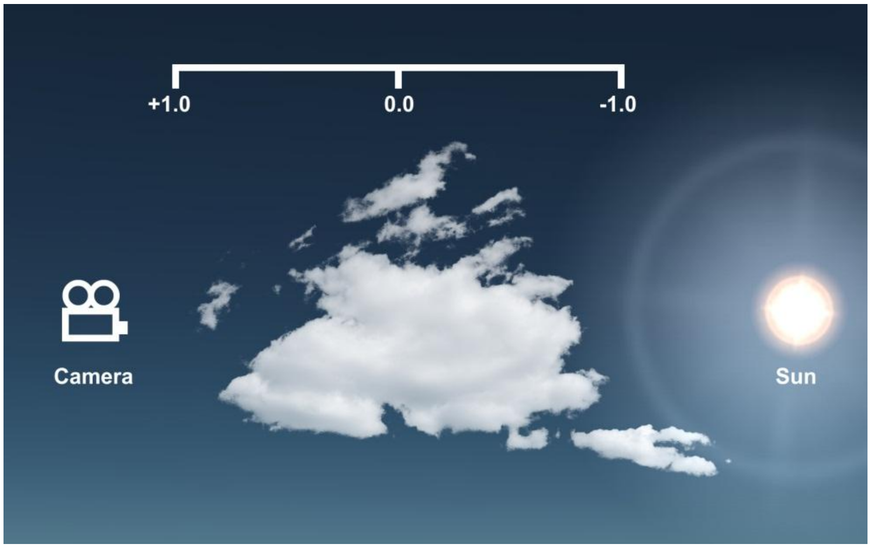
In the same way, with Vue 7.5 and higher, we also have Cloud Anisotropy. This follows the same principle, but works somewhat differently.

Clouds are not solid objects, but a giant collection of small water particles moving in clusters. At a distance the isotropy/anisotropy (in relation to light) of each particle combines to create a larger visual effect – or in other words, the same anisotropy of a single particle becomes the anisotropy of the entire cloud as if it were a single object.

Just like the shine on our metal sphere bending along the surface of the object, if light hits an anisotropic cloud surface in Vue, it penetrates the cloud particle field, bends along the internal surfaces and comes out (lightly or strongly, depending on the amount of Anisotropy chosen) on the other side and towards the camera.



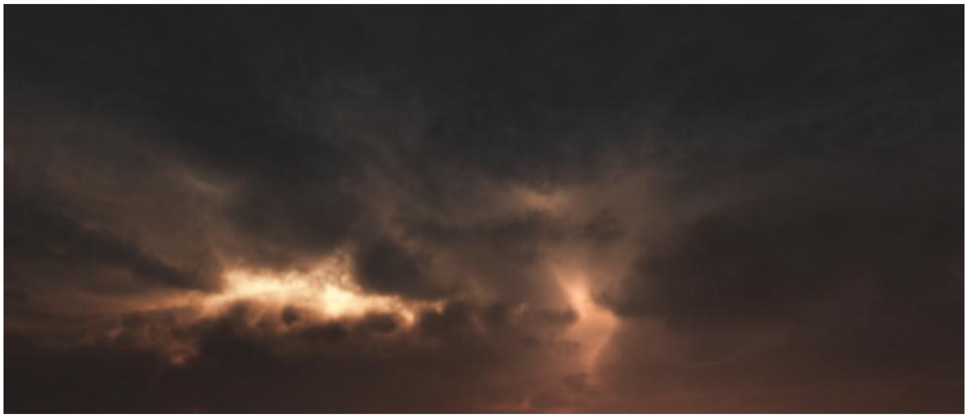
Cloud Anisotropy can create some beautiful effects. Because of the nature of the physical process, Cloud Anisotropy is best experienced with Volumetric Lighting or Godrays enabled, and the Sun is behind the clouds.



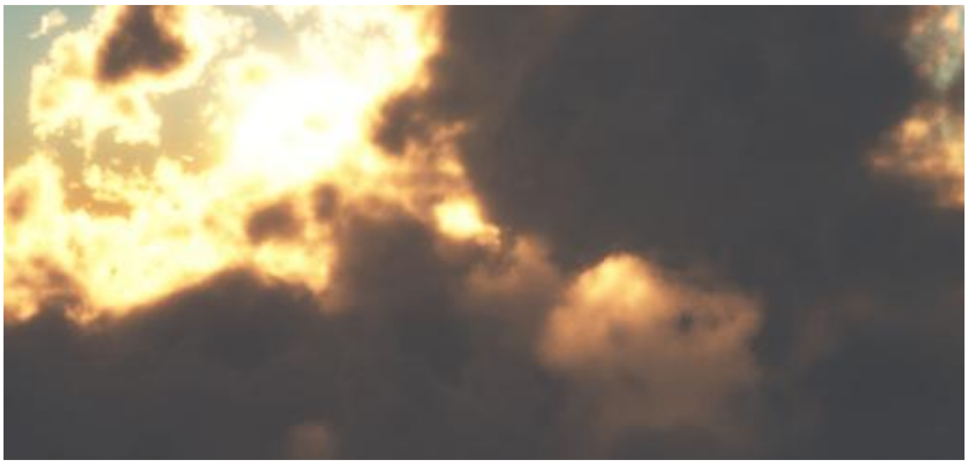
The diagram above shows how the Cloud Anisotropy setting in the Atmosphere Editor works. When your setting is -1.0, the Sunlight does not penetrate the cloud that much. At 0.0 there is neutral or balanced penetration, but still not enough to create large light effects. At +1.0 there is as much light penetration as possible, bringing 100% of the penetrable light across through the cloud and towards the camera.

Do note that your Cloud Layer's Opacity, Density, Ambient Lighting, and Shadow Density also affect how Cloud Anisotropy is rendered visually. Boosting the Atmosphere Quality can also improve results.

In the images below, you can see how Cloud Anisotropy is bending the light along the cloud surface and propelling it in the form of edged ambient light towards the camera.

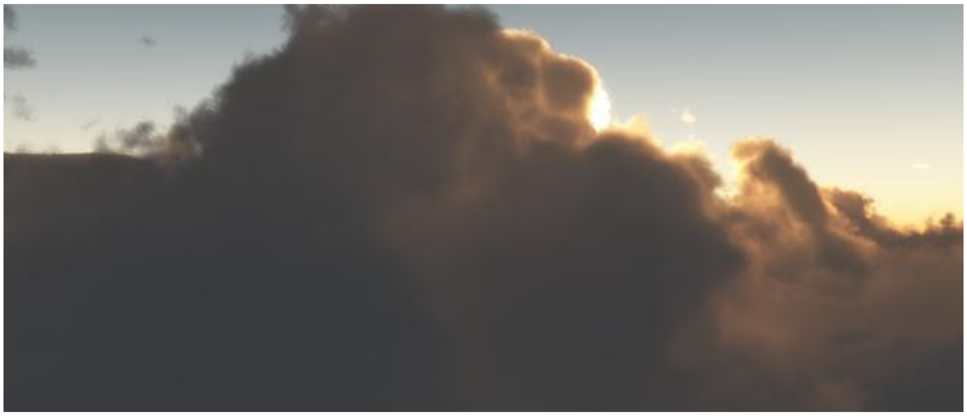


Dark storm clouds.



High altitude sunset among the clouds.

Simulated lightning behind the clouds using Cloud Anisotropy.



Cloud edges flare starkly with a high Cloud Anisotropy.

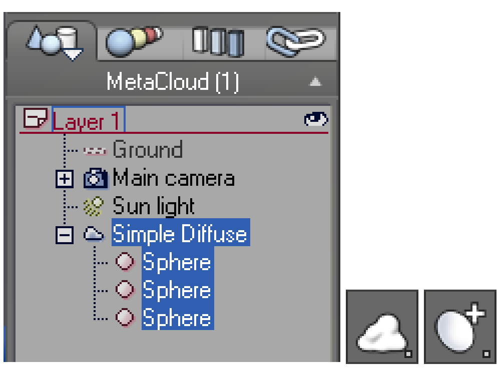
Mammatus clouds with edge details created with Cloud Anisotropy.

Dark sunset.

MetaClouds

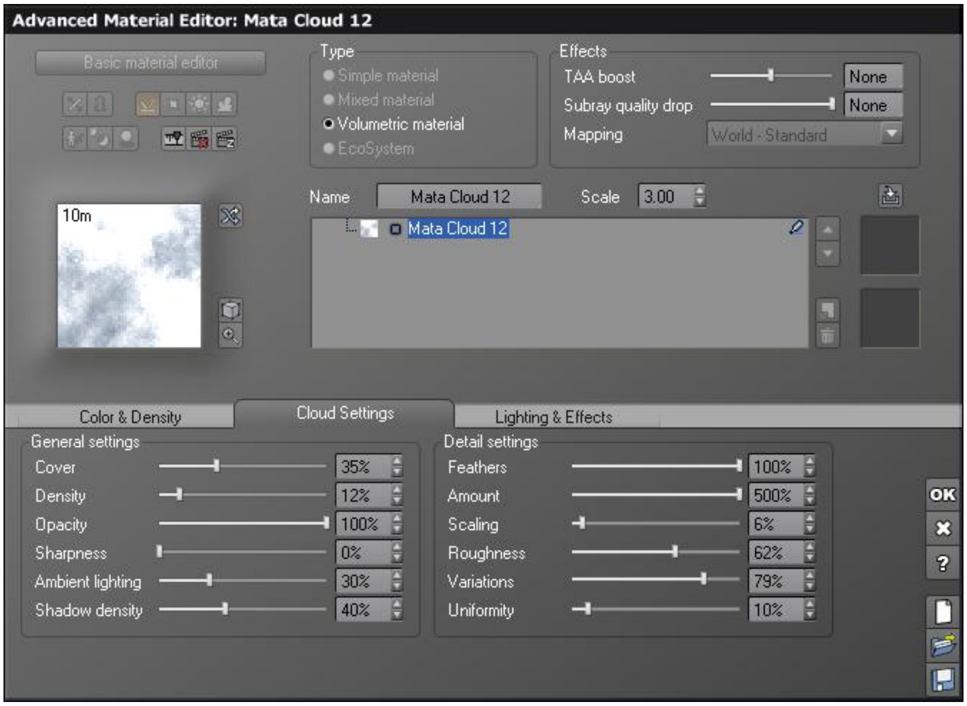
The MetaClouds are similar to normal Cloud Layers except that they are independent from the infinity of the Vue world. MetaClouds are confined to the cloud primitives that create them.

Before we go any further, I would like to remind of the basic cloud rule we talked about in the previous section. Clouds are made of countless small particles. Clouds are gaseous in nature. Clouds are *not* solid objects. Keep them in mind as you explore the settings of the MetaCloud material.



MetaCloud expanded in the Object Browser and MetaCloud/MetaCloud Primitive buttons.

To edit a MetaCloud preset, all you have to do is enter the MetaCloud in the Object Browser by expanding it and selecting any MetaCloud Primitive. At this point the MetaCloud button changes to the MetaCloud Primitive button.



The basic idea behind each setting in the MetaCloud material is the same as the Cloud Layer settings we discussed in the previous section. So in this section instead of repeating what each setting does, we will see how they behave differently when working with MetaClouds.

Cover

As with the Cloud Layer, the Cover setting defines how much of the cloud area (in the case of MetaClouds, the Primitives making up the cloud form) is filled with the cloud. It treats the edges of the primitives as a boundary. Cover based filling of this boundary starts from the densest portion.

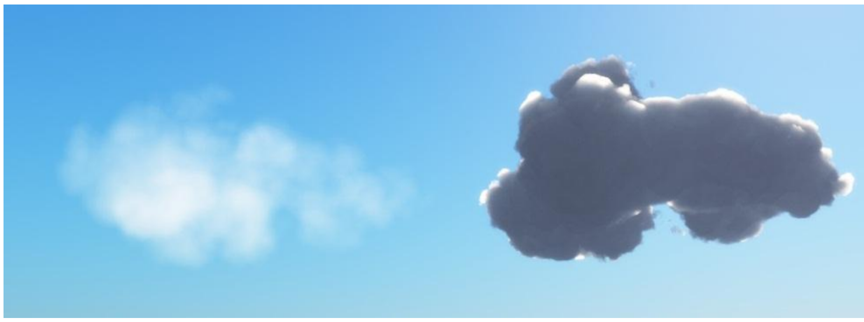
Cover at 50% and 100%. The denser area containing more primitives is filled first.

Density

As the name suggests, Density controls how thick the clouds are. Often when dealing with MetaClouds for "close to ground" situations like a dust storm, or even a gust of wind dragging up some dirt, you may want to use 'micro-densities' or density values between 0.0 and 1.0. You can see examples of that below, and in realistic usage in the examples section following this section.

Density at 50% and 100%. As density decreases, the cloud looks less solid.

Density at 1.0% and 100%. Ideal setting for thin clouds.



Density at 0.25% and 100%. Clouds of dust have such micro-densities.

Opacity

The Opacity setting is responsible for deciding how much light penetrates the cloud. Extremely high values (beyond 80%) may make your clouds look like solid objects.

Opacity at 25% and 50%.

Sharpness

Sharpness defines how sharp the edges of the clouds are. This is a very delicate setting as we discussed previously. Even medium range values may cause your clouds to look as if they were solid. Use with care.

Sharpness at 15% and 100%. Notice how the edges on the right cloud are too sharp.

Ambient Lighting

This is the sister setting to Opacity. Unlike Opacity, Ambient Lighting will provide light from all directions. This setting behaves slightly different on MetaClouds. Since MetaClouds are made of small “cloudlets” but are more tightly packed and visible than in a Cloud Layer, excessive Ambient Lighting may make individual cloudlets visible in ways we may not want them to be.

Ambient Lighting at 100% and 25%. In the left version cloudlets are visible.

Shadow Density

This property controls how dark the shadows thrown off by the MetaClouds are, and how strong the Godrays created by them will be. Unnecessarily high values may create almost solid shadows that may cause the underside (or shadowed side) of the MetaCloud to look unrealistic.

Feathers

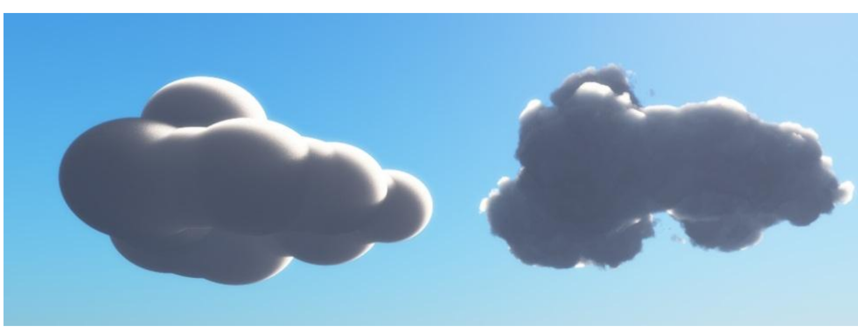
Just like with the Cloud Layers, Feathers shift the cloudlets from “blobs” to “feathers”. In MetaClouds, feathering can often play an important role in changing the shape of the cloud. Experiment in each instance to find unique shapes.

Feathering at 100% and 0%. The feathered cloud has soft wisps growing outward.

Amount

Amount, or more precisely, Detail Amount, controls how much detail is added to the cloudlets inside the MetaCloud boundary. Do not let the 100% maximum of the slider of this setting stop you. You can go beyond the 100% value by manually typing in any value you want. Often with larger clouds or other specific situations, values beyond 100% may give you the right amount of detail.

Use with caution as high values are liable to break up your cloud into too many smaller cloudlets. This setting works in conjunction with the Scaling and Cover settings, as each of them affects the other. With high values resulting in small cloudlets, you may also wish to increase the Density and Opacity settings so as to not have 100% light penetration on the smaller cloudlets.



Amount at 0% and 100%. With a 0 value there is no detail except the primitives’ shapes.

Amount at 350% creating a massively complex and broken cloud.

Scaling

Scaling is a very important factor in shaping MetaClouds. Often with the actual size of the MetaCloud being changed, the resulting effect is completely altered as the Scaling of the MetaCloud material is incompatible with the actual size. When this happens you get either extremely small cloudlets or extremely large cloudlets. Neither of which is ideal. So when you change the size of the MetaCloud object, play with the Scaling setting until you get the right shape.

Also note that Scaling should be used in conjunction with Cover and Amount settings for best results as they interoperate between each other.

Scaling at 6% and 19%.

Scaling at 70% and 19%. Due to the large cloudlet size, only the dense areas are clouded.

Roughness

Roughness breaks the cloudlets as roughly as dictated by its value. This allows for some interesting unique shaping that breaks away from the normal uniform feel of a typical Vue cloud. As with Detail Amount, high values can cause a little too much chaos between the Cloudlets.

High roughness settings can create very interesting effects on large MetaClouds that have strong sunlight and low Opacity values.

Roughness at 68% and 54%.

Roughness at 100% and 54%.

Variations and Uniformity

These settings work in the same way as we discovered in the Cloud Layers section.

Cloud Examples

In the following pages, you will see several examples of both Cloud Layers and MetaClouds. We will also explore the main aspects and strengths of each example, including various Material and Atmosphere settings for the major clouds.

Bright morning clouds with a medium density (40%) and medium opacity (55%)

Low density (20%) and medium opacity (45%) make MetaClouds soft and puffy.

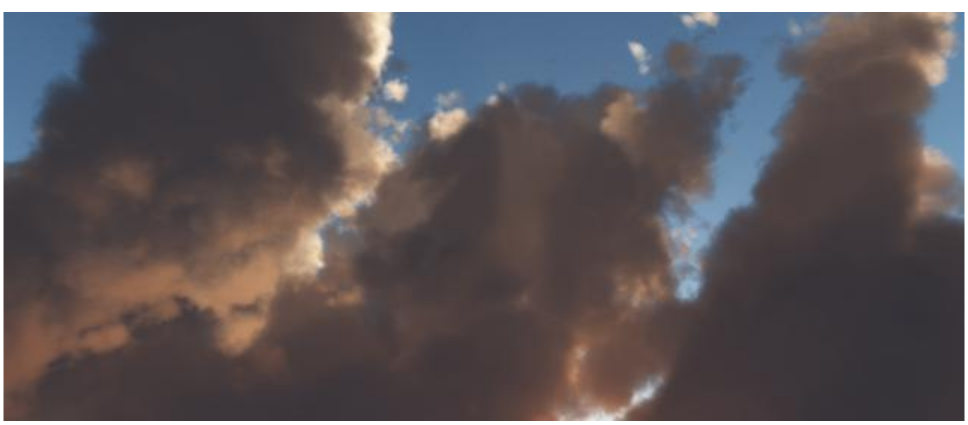
Strong Shadow Intensity mixed with high Ambient Lighting (80%) and low Opacity.

For confined camera angles, MetaClouds can often prove to be more useful than cloud layers. Additionally, they're often easier to sculpt/shape than an entire cloud layer.

Great care must be taken to compensate on both the material scale (overall) and Scaling inside the material for the MetaCloud when resizing the cloud objects drastically.

A seemingly low Shadow Intensity value of 45% creates these strong Godrays.

The same clouds rearranged with lower Shadow Intensity to avoid Godrays.



Again, the same MetaClouds with different altitude and higher Scaling for a rough look.

This cloud layer takes advantage of a very low Sun angle and strong Shadow Density, along with Volumetric Lighting to create this foreboding sky.

Such lighting is time consuming in both render time as well as positioning.

Fractal Cloud Layers are generated mathematically, not artistically so the cloud shape, scale, or position (Origin of Material) often need to be experimented with to get the right cloud shapes or clusters.

Cloud Configuration

Altitude	Height	Cover	Density	Opacity	Sharpness	Feathers
1.2km	1km	77%	100%	100%	0%	63%

Detail Amount	Altitude Variations	Ambient Lighting	Shadow Density
22%	100%	44%	100%

A precisely placed cloud layer interacts with the infinite procedural terrain to create this other-worldly mountain environment.

A large amount of time was spent on experimentation with the *Origin of Material* setting in the Cloud Layer to get the right cloud positioning.

Volumetric Lighting is used to get heightened realism by producing shafts of rays between the clouds and the mountain peaks.

Cloud Configuration

Altitude	Height	Cover	Density	Opacity	Sharpness	Feathers
698m	796m	94%	58%	100%	2%	40%

Detail Amount	Altitude Variations	Ambient Lighting	Shadow Density
40%	100%	0%	100%

MetaClouds form the rising mist of this deep, dark tropical jungle. While the same result can be easily achieved with Cloud Layers, MetaClouds provided a deeper level of precision in the placement of the mist.

Beware that such low densities increase render times exceedingly, often turning hours into days at very high resolutions. To find easier ways to render such scenes, refer to the Appendix for tips on decreasing render times.

Cloud Material

Cover	Density	Opacity	Sharpness	Ambient Lighting	Shadow Density
100%	1%	50%	20%	20%	40%
Feathers	**Amount**	**Scaling**	**Roughness**	**Variations**	**Uniformity**
0%	100%	19%	54%	30%	75%

This menacing sky is made from a dense cloud layer with almost 100% coverage and a low Sunlight angle. Intense Shadow Density is responsible for the dark pockets in the cloud layer.

The Sharpness setting plays an important role here as it provides the pronounced edges for the under-lit clouds.

Cloud Configuration

<table>
<tr><th>Altitude</th><th>Height</th><th>Cover</th><th>Density</th><th>Opacity</th><th>Sharpness</th><th>Feathers</th></tr>
<tr><td>516m</td><td>1km</td><td>95%</td><td>47%</td><td>94%</td><td>10%</td><td>40%</td></tr>
<tr><th colspan="2">Detail Amount</th><th colspan="2">Altitude Variations</th><th colspan="2">Ambient Lighting</th><th>Shadow Density</th></tr>
<tr><td colspan="2">18%</td><td colspan="2">25%</td><td colspan="2">66%</td><td>100%</td></tr>
</table>

A blissful summer scene with low cumulus clouds. The clouds use low density to pick up some color from the blue skies and add to the summertime feeling.

There is also a high level of ambient lighting which bleeds different colors of the surrounding atmosphere into the clouds themselves.

Cloud Configuration

Altitude	Height	Cover	Density	Opacity	Sharpness	Feathers
106m	1km	46%	47%	94%	10%	40%

Detail Amount	Altitude Variations	Ambient Lighting	Shadow Density
18%	25%	100%	Does not cast

Two clouds comprise this cloud field: the low-lying cumulus and the very high altitude wisps of cirrus.

By having multiple cloud layers, even if the secondary (High Altitude) layer is very subtle in appearance, the scene looks more realistic since the real world often has multiple layers of clouds.

Cloud Configuration

Altitude	Height	Cover	Density	Opacity	Sharpness	Feathers
971m	4km	45%	26%	100%	1%	15%

Detail Amount	Altitude Variations	Ambient Lighting	Shadow Density
18%	100%	16%	44%

Giant MetaClouds envelope this craggy mesa to create a massive Sandstorm-front. The accurate positioning of the MetaClouds is key in such a scene as often the puffs of smoke/sand/cloud may not exactly coincide with the visual border of the MetaCloud primitives. A lot of small preview renders were performed to ensure proper cloud positioning before the final full resolution render was performed.

Cloud Material

Cover	Density	Opacity	Sharpness	Ambient Lighting	Shadow Density
100%	43%	61%	0%	20%	40%
Feathers	**Amount**	**Scaling**	**Roughness**	**Variations**	**Uniformity**
0%	100%	19%	71%	30%	75%

A scene inspired by intense summer sunsets in the deserts of India, this multiple cloud layer scene uses small cloud shapes overlapping each other to create a rich cloudfield.

A lot of Ambient Lighting was applied to avoid the clouds from becoming too dark from the intense sunlight.

Cloud Configuration

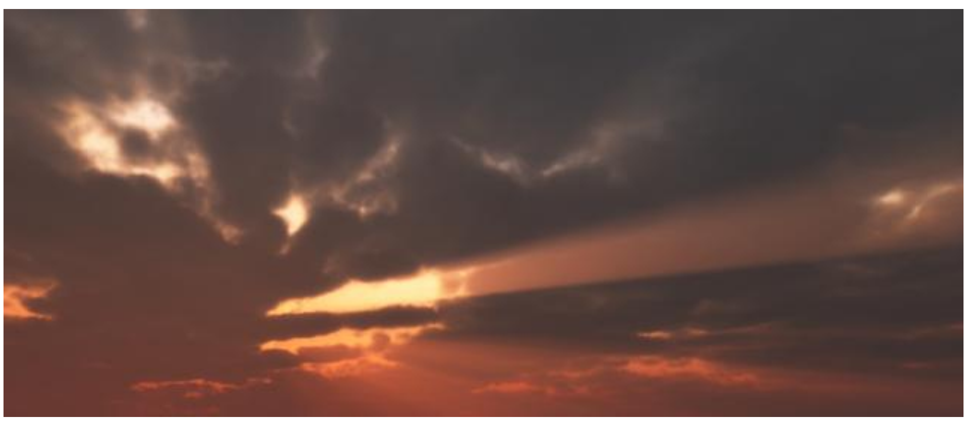
Altitude	Height	Cover	Density	Opacity	Sharpness	Feathers
697m	100m	38%	100%	93%	27%	36%

Detail Amount	Altitude Variations	Ambient Lighting	Shadow Density
169%	94%	80%	100%

Godrays

One of the most sought-after aspects of natural 3D scenes is Godrays. Those almost elusive shafts of light are sometimes difficult to achieve, but when they are created properly the end result gives a massive boost to your scene.

In this section we will explore the basic mechanics of creating Godrays in a Spectral atmosphere. While the new Vue 8 and 8.5 Spectral Atmosphere enhancements create better Godrays, most of the concepts we will talk about apply to Vue 7 as well.



Why Godrays occur

It is often difficult to achieve Godrays in Vue. The reason behind this is that Vue is designed to process atmosphere as it would be in the real world. And as we know from daily life, Godrays are not a constant but rather an occasional occurrence.

To create Godrays with ease, we must first understand why Godrays occur in the real world, and how they occur. Once that is understood, we can create a similar atmosphere in our Vue world and Godrays will be created automatically.

In this photograph, you can see large shafts of sunlight filtering through the trees. Light normally passes through those same trees all the time, but we do not see Godrays every time it does. This is because Godrays are created with very strong shadows, not just strong light.

When there is a dense particle field in the atmosphere, like the dust in our photo, the light bounces off those particles creating the shafts of light and dense shadows among the particles. The result is what we call Godrays.

In essence, there must be a strong haze, fog, or similar natural phenomenon for Godrays to be created. So in turn, we must create strong haze or fog in Vue to create Godrays.

Godrays in Vue

You have to follow three basic rules in Vue to create Godrays. Strong light, dense haze, and light inhibitors – objects that create the shadows.

Taking this render as an example, let's break down the 3 rules into their components.

Strong Light

The Light Balance slider in the Light tab in the Atmosphere Editor is set to 98%, so there is very little ambient light, and subsequently, dark shadows.

The Glow Intensity is set to 100%, with Scattering Anisotropy set to 0.24, so the Sunlight glow is drawn towards the camera.

To avoid overexposing our scene, the Aerial Perspective is set to 1.00. Obviously, Volumetric Sunlight is selected. Godrays is not checked, since we already have Volumetric Sunlight selected.

Dense Haze

A very light color of RGB 220, 205, 218 is assigned to the Haze Color, with an intense value of 250% in the Haze Density setting. The Haze Mean Altitude is a low 800m, which keeps the Haze hanging near the ground rather than all over the sky.

The Fog Color is set to a soft hue with RGB 247, 221, 195 . Fog Density and Mean Altitude at set to 100% and 550m respectively. These settings augment our almost neutral colored Haze with a soft pink/orange hue in the lower areas.

Be prepared to deal with a hot white, overexposes sky. Like with real life, getting Godrays in a 'shot' means sacrificing the neutral exposure of the sky.

Light Inhibitors

Finally, we need objects – in this case, trees – that will block the sunlight partially and create the shadows in the Haze. Large objects are easier to use for blocking sunlight, but smaller objects can work too. Essentially it is important to remember that more light in the frame (not the scene!) has to be blocked than the light that is allowed to pass freely.

Quality

It should be noted that you need a minimum of +3.0 Global Settings > Quality Boost in the Atmosphere Editor to achieve proper Godrays. For better renders, a value of +10 or higher is recommended. This will obviously increase your render times, but the end result quality increases greatly and can be considered worth the sacrifice.

For example, look at the difference between these renders.

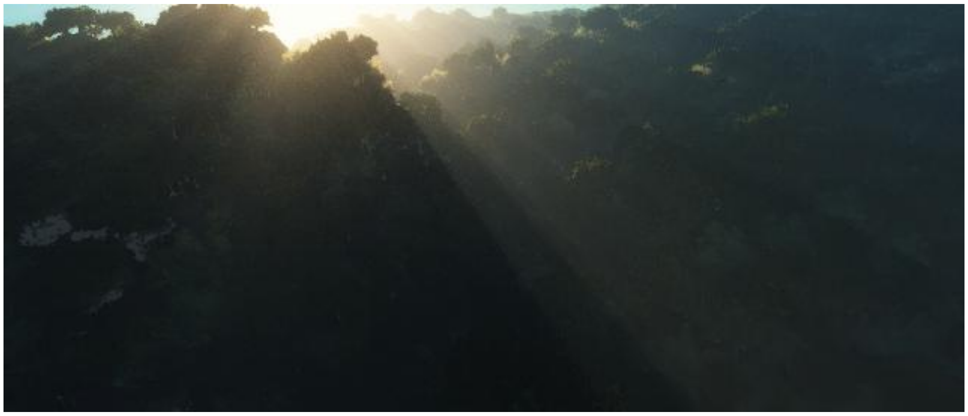
Quality Boost at 0.0

Quality Boost at +4

Quality Boost at +10

Examples of Godrays

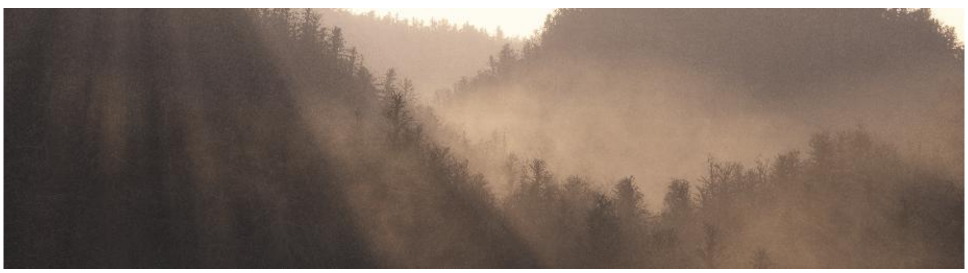
Godrays need not be extreme. Subtle rays can create stunning effects too. In this example, soft Godrays filter in from the top of this enclosed environment creating a musty feeling.



In this dense jungle scene, a large 3.5 x 3.5km terrain with a dense EcoSystem is used as the Light Inhibitor to create our Godrays. With enough Sky Dome Lighting Gain the Sunlight Balance which is set to 96% is overcome with a small infusion of ambient light.

This is a variation of the scene we deconstructed. The Sunlight is position so that the gap in the EcoSystem throws a lot of sunlight onto the 'floor' with a dense population of grass plants. This helps negate the stark darkness of the forest.

This scene uses an off-screen Sun to create out Godrays using the terrain itself as a light inhibitor. The haze isn't too strong therefore the Sky isn't overexposed because of the intense glow that would normally be there. The Aerial Perspective is changed to 7.0 to enable strong Godrays at the expense of some minor realism and scale.

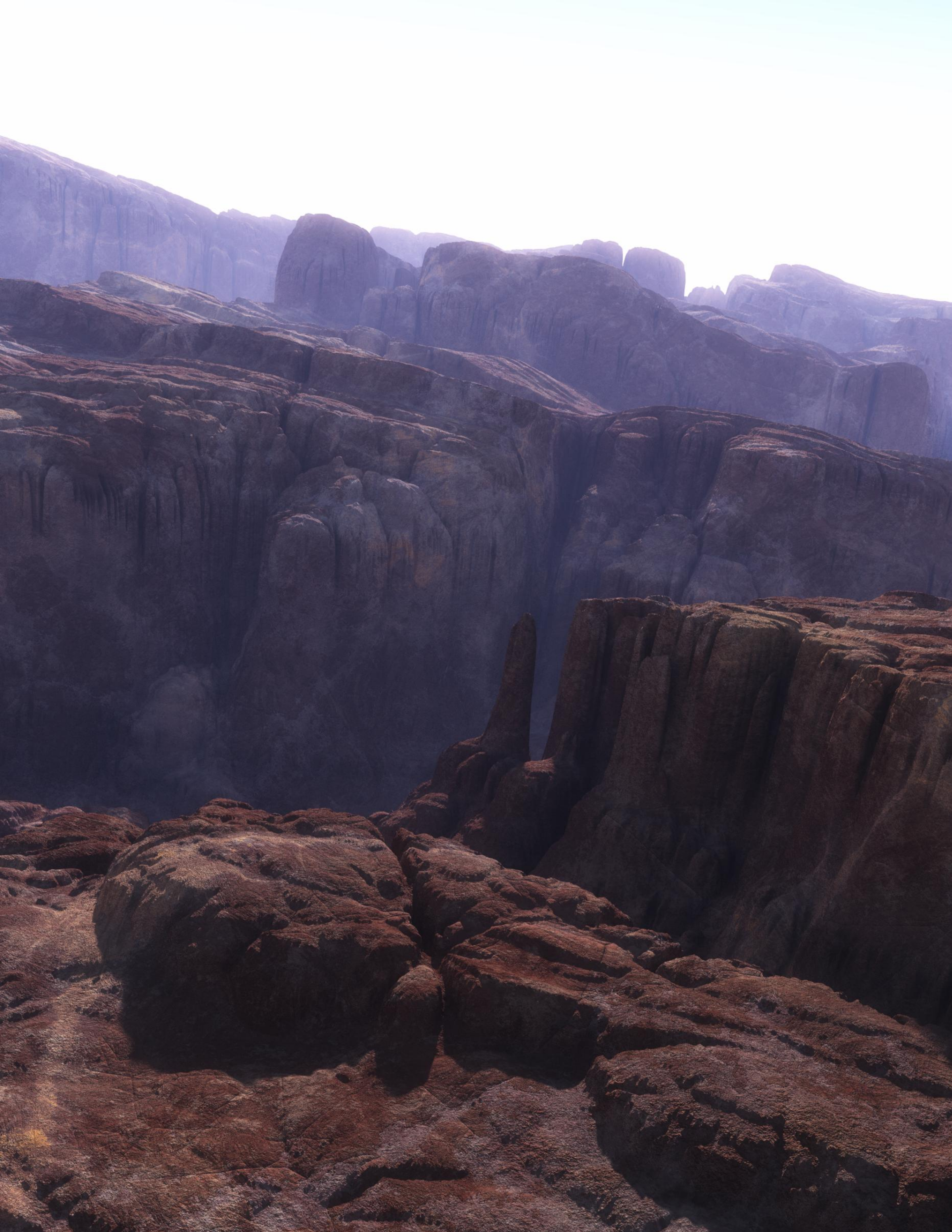


This scene uses extremely low density MetaClouds with 0.32% Density to force localized haze. Strong Godrays are created within these clouds, but come at a heavy price since the micro-density MetaClouds are very cumbersome to render.

Additionally, such delicate Godrays require an Atmosphere Quality Boost of +10.00.

Here is another scene that uses the terrain itself as a light inhibitor. A low, dark haze is used to propagate the shafts of light. A darker haze allows for stronger sunlight without overexposing the scene, ensuring strong Godrays. The Aerial Perspective is 1.0.

This is the ‘Glory’ scene included with Vue. I originally designed this scene to show off the Spectral2 power of Vue 7, but the above render using Vue 8 shows the advanced capabilities of the third generation Spectral atmospheric render engine. The Godrays are created using the clouds as the only light inhibitor. You can find this scene on the Extras disc of Vue and experiment.



Terrains

Chapter 4

Grounding your scenes

In your scene, the terrain will often be the most important physical object. The entire scene depends on it, therefore the more realistic and well-formed your terrains and rock formations are, the more believable your scene appears.

Vue provides three types of terrains, Standard, Procedural, and Infinite Procedural terrains. We will focus mainly on Procedural and Infinite terrains as they provide more realism than the Standard terrains. We will also explore HyperTerrains, a new technique for creating procedurally generated rock formations in Vue.

With a mix of Procedural Terrains creating the larger terrains, and HyperTerrains providing the individual rocks and rock formations, your scene can create a marvelous sense of realism.

Procedural Terrains

Procedural Terrains are now becoming the primary way to create terrain. The main advantage is that Procedural Terrains are infinite in both size (although you can limit them in size) and detail. Because of their procedural nature – grown from a Fractals designed to recreate real world features – the terrain maintains an overall natural quality that is often not found in hand sculpted terrains.

In this section we will cover two major features of Vue that will help you create fantastic terrains – the Terrain Fractal and the Strata Filter.

The Terrain Fractal

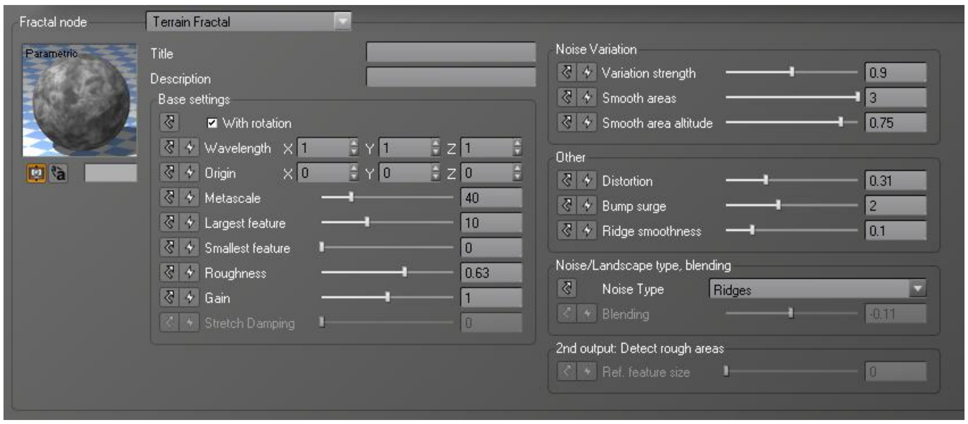
The standard fractals found in Vue such as Perlin or any other prior to version 7.0 were not enough for creating realistic procedural terrains. To address this need, Vue 7.0 introduced the Terrain Fractal. It has its root in existing fractals and behaves in ways we are used to with the Perlin and Granite fractals, however, this new fractal brings out a very powerful shaping system which creates mountains and entire mountain ranges with natural form distortion that other fractals cannot.

A new setting available to fractals is called MetaScale. When used in conjunction with the Largest Feature and the Smallest Feature, it can provide fascinating results. The image above shows a top view – almost satellite image like – of a procedural terrain that uses two terrain fractals. The first fractal creates the massive rock formations and hills, and the other terrain fractal creates the flow lines and water erosion striations at the 'ground' level.

The long striations seen in the ‘ground’ level are caused by a very high Distortion setting. When Distortion is set above 50%, we see long curving shapes in the fractal as if you are dragging your fingers in water. In real life, if you look at satellite images such large scale curving formations can be seen in mountain ranges.

One thing every digital nature artist should keep in mind – especially when creating terrains – is to look at the entire scene from a very high altitude. Such a “Gods Eye View” helps keep your scene in perspective. If your scene looks realistic from high above, no matter what your lower camera angle is, it will always look realistic to the eye at that level too.

Understanding the Terrain Fractal Settings



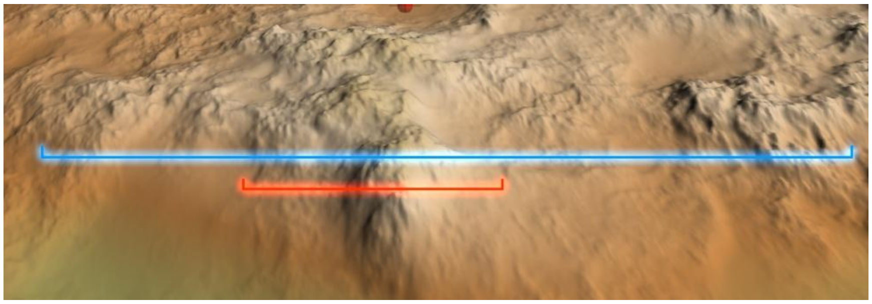
Now, let’s explore the Terrain Fractal and see what each setting can do. We will start with the following settings, and to explore each setting, only that setting will be changed while the rest remain at default.

Wavelength

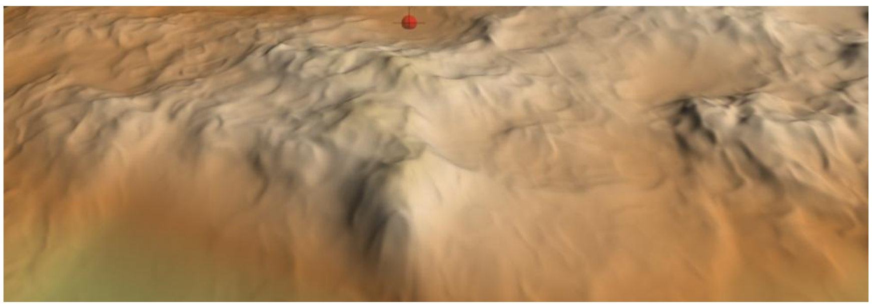
This is the base mathematical wave from which the fractal is generated. You can change the size of the fractal along all three axes – X, Y, and Z – from here. Changes to these settings are often not required, unless you want to use a non-uniform value in one axis to stretch the fractal on that axis.

MetaScale, Largest Feature, and Smallest Feature

The MetaScale is, in essence, the size of any mountain ranges being created (blue line in the image below). The Largest Feature is the size of the largest rock formation or mountains in that range (red line in the image below), while the Smallest Feature is the smallest sized formation in those mountains (image on next page). Usually, best results are found when the MetaScale is larger than the Largest Feature, and the Largest Feature is higher than the Smallest Feature. Of course, using unorthodox values for each can bring unique results.

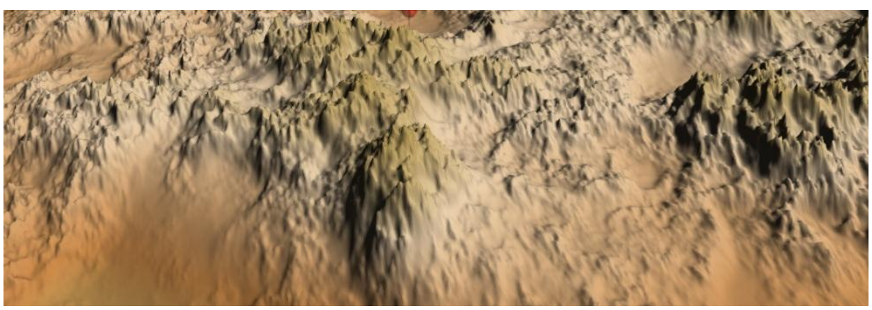


The smallest feature also serves as a base value for Roughness. When you have a high Smallest Feature value like 10, the fractal discounts all minor details and creates only the larger shapes as you can see in the image above.



Roughness

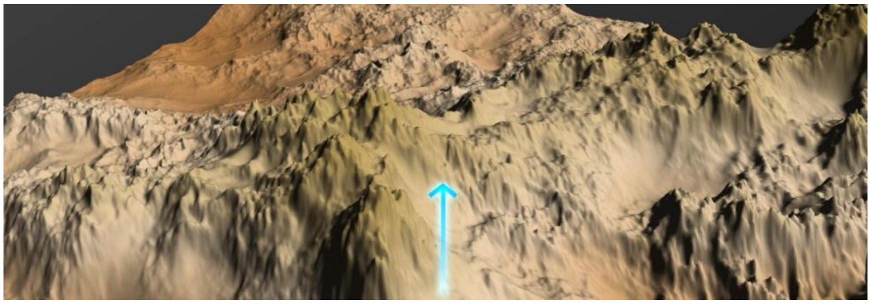
This is the amount of roughness – or strong raggedness – that is applied to the fractal. Between the ranges of 0.35 and 0.65, this can create interesting and useful shapes. Near and above 0.7 values will create sharp spikes across the terrain in most situations.



If you need to use a high Roughness value without creating jagged spikes over the entire terrain, use a medium or high Smallest Feature value. If you need your terrain features to have more height, then Gain may help achieve that. However, do note that with a higher Gain value some of the lower features may decrease in altitude or disappear completely if the Gain overpowers them.

Gain

This is the height boost given to the entire fractal. 1 is the default value, but usually a value of 2 or 3 is more useful. Values of 4 to 10 can create strong mountains but must be used with care. Using this setting in conjunction with Bump Surge is recommended for optimal terrain outputs.



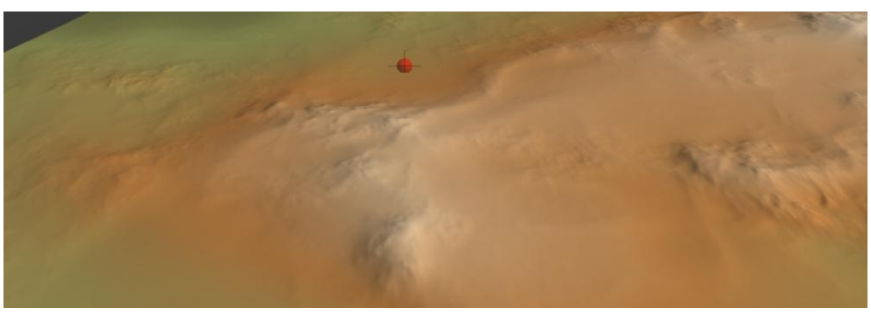
The blue arrow in the image above shows the direction the terrain is intensified by the Gain setting.

Stretch Damping

If any of your axes use a value different from the other two, you may see stretching in the fractal output. If you want to dampen, or soften, the stretching, you can use this slider; unless if the stretching is intentional in which case you may skip the damping.

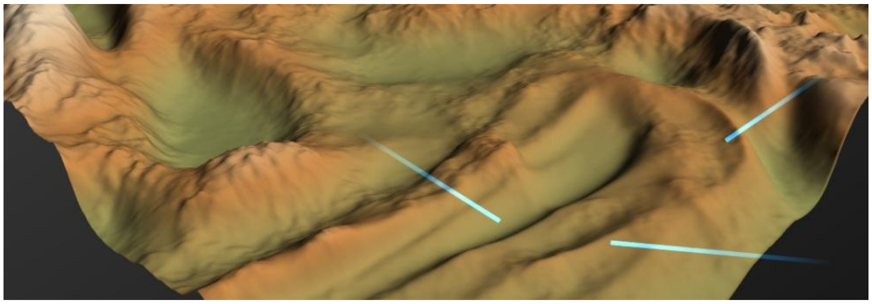
Variation Strength, Smooth Areas, and Smooth Area Altitude

Variation Strength controls the variation of the noise in the fractal that creates medium to large scale variations of noise or shapes in the Fractal to give it a more non-uniform feel. A low value in the Smooth Areas setting will create an intense amount of smoothness in your terrain, while a strong value will create sharper terrains. Smooth Area Altitude controls the altitude level up to which the smoothing from Smooth Area is applied. High values (up to +1) cause less smoothing allowing more altitude variations, while lower values (down to -1) cause massive smoothing creating less altitude variations between the highest and lowest altitudes in the Fractal.



Distortion

This is one of the most important settings in this fractal. It distorts the terrain fractal by smearing the shapes around as nature does in real life. These striations can create some marvelous effects.



The blue lines above mark some of the large positive (high altitude) and negative (low altitude) distortions being introduced into our Terrain Fractal.

Bump Surge

This setting increases the bump, like the Gain setting, but works mostly on the higher altitude areas that are above the 'average' level of the terrain.

Ridge Smoothness

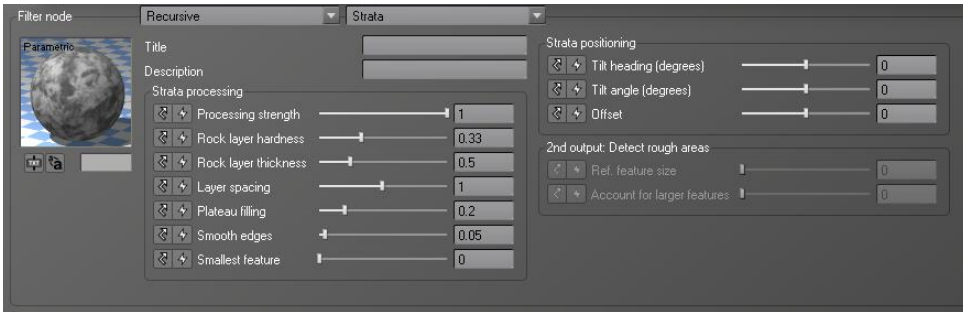
The ridges on the top of the terrain, at the highest altitudes, are softened using this setting. It is very helpful if you see anomalous artifacts or unwanted sharpness on your ridges. Often these artifacts are caused by the Roughness or you may also wish to check the Roughness and Smallest Feature settings.

Noise Type

This list lets you select between a few different basic fractal types that affect the overall shaping of the terrain. There is no 'best' or recommended setting for this as they all create unique effects that are best chosen by experimentation or by the specific needs of your terrain. However, in most situations the default 'Ridges' option works well.

The Strata Filter

One of the most useful new features in Vue 8 is the Strata Filter. With this new feature creating sedimentary terrains with strata of rocks and bedrock becomes so much easier. The Strata Filter comes in two varieties – the normal Strata Filter and the Confined Strata Filters. Both are available in the Recursive category of the Filter Node. The main difference between the two filters is that the Strata filter works on the entire terrain while the Confined Strata filter can be restricted to a specific altitude range.



A Procedural Terrain made using the default fractal

Same Procedural Terrain with a Strata Filter applied

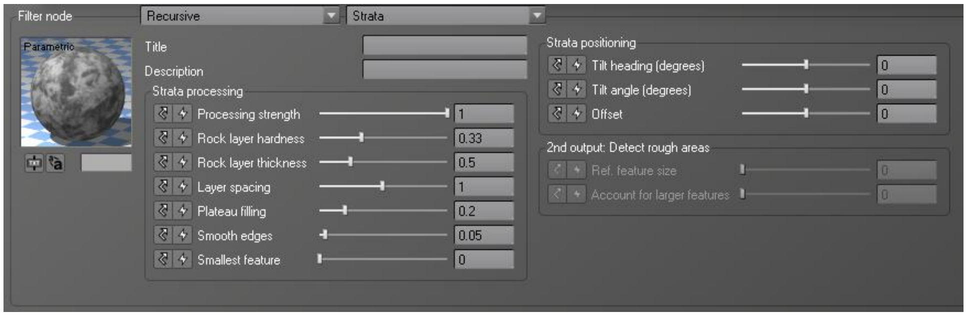
Same Procedural Terrain with different settings for the Strata Filter

Apart from the stratification of the terrain, the Strata Filter provides a fascinating new feature – terrain tilting. The filter tilts the entire terrain at the fractal level, which is different from tilting the terrain by just rotating it along one axis. This will change the shape of the terrain as can be seen above.

The realism factor provided by this specific feature is very important. In the real world, so many terrains – especially desert mesas or large flat bedrock layers – are flat in nature, but over millions of years they become tilted as one side sinks into the earth. The picture below shows a perfect example of how a stratified terrain is often tilted.

Stratified Terrain reference photos taken at the Colorado - New Mexico Border, USA.

Strata Filter Settings

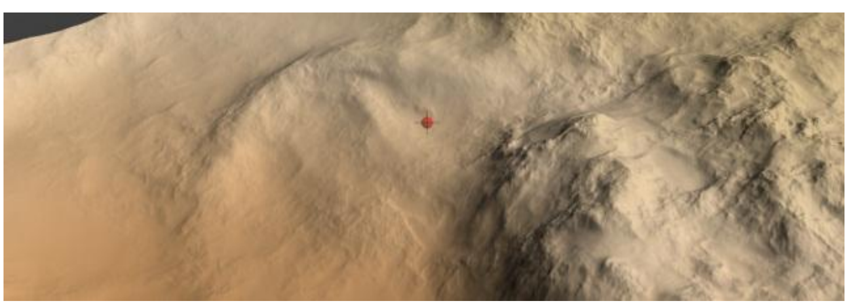


Processing Strength

This setting is the intensity with which the stratification process is applied to the parent fractal. Increasing it makes the terrain more stratified, while keeping the value low makes more of the original fractal stay as it was without stratification. For most purposes, values between 0.6 and 0.9 are best.

Processing Strength = 1.0

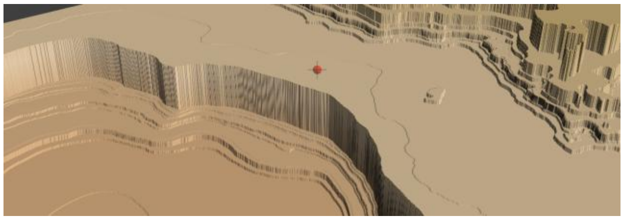
Processing Strength = 0.5



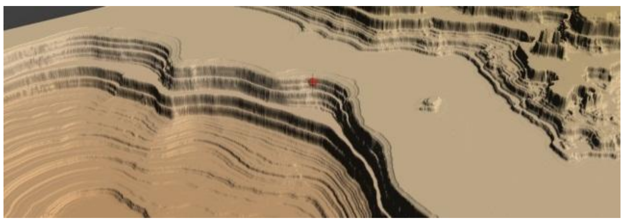
Processing Strength = 0.0

Rock Layer Hardness

In the preview, high values of this setting look as if the fractal image is being posterized and contrasted. The actual effect is that the edge of a stratum is made sharp and it abruptly drops to the next lower level. Recommended values for most purposes are between 0.65 and 0.85. Too much Hardness will create artifacts and unrealistic shapes in your strata.



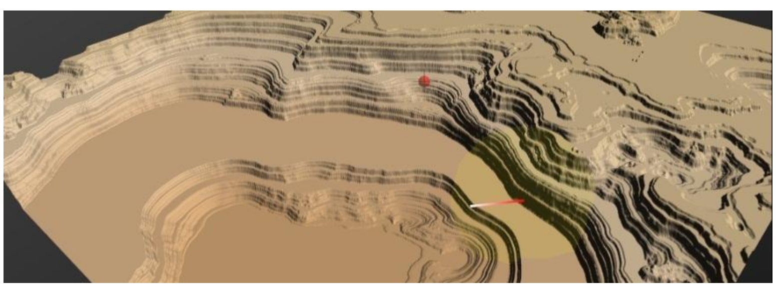
Rock Layer Hardness 1.0



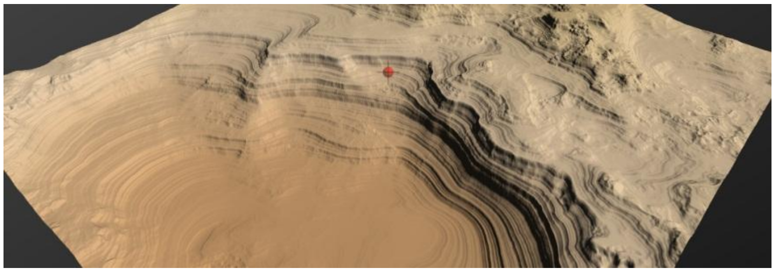
Rock Layer Hardness at 0.33 (Default)

Rock Layer Thickness

This setting controls the thickness of a layer of rock in the strata, creating almost 'sublevel strata' in each stratum. Rock Layer Thickness is responsible for creating small steps or terrace features in strata and between two strata transitions. However, if this value is larger than the Layer Spacing (see below) it may be ignored. A good value for this setting is between 0.2 and 0.6, depending on your Layer Spacing setting. Alternatively if you are using very high Layer Spacing, choose values between 20% and 60% for the Rock Layer Thickness.



Rock Layer Thickness 1 with Layer Spacing 1

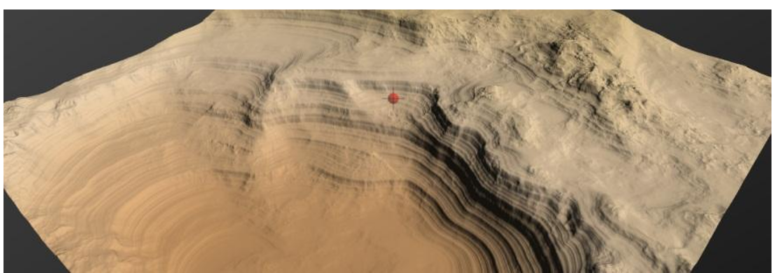


Rock Layer Thickness 0.5 with Layer Spacing 1

Layer Spacing

The height or altitude of a stratum before the subsequent one is set with this value. In other words, it divides up the terrain vertically into strata or 'slices'. High values create fewer strata (because of greater height of each stratum), while lower values create more strata (because of the reduced height of each stratum allowing room for more layers). For realistic results, a midlevel value of 0.6 to 1.5 is recommended, but the actual settings needed may depend on your scene's visual requirements.

Layer Spacing = 1.0 (Default)



Layer Spacing = 2.0

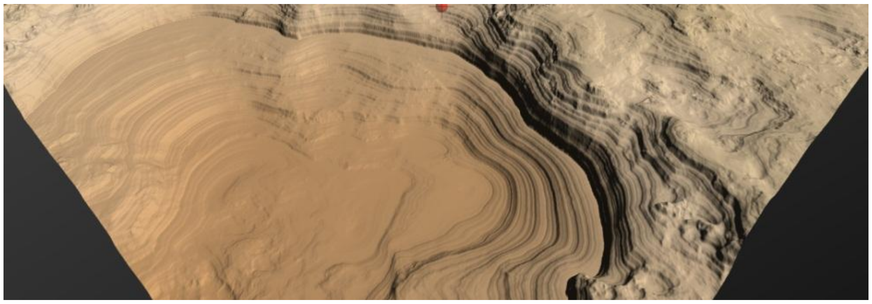
Plateau Filling

The Plateau Filling controls the slope of the strata. Higher values 'fill' the top of strata with sedimentary deposits, therefore controlling the slope by creating a range of landslide style deposits that break the monotony of the sharp terrace style strata. Usage values are completely arbitrary depending on your needs. However, for most situations a value of 1 is generally recommended. Too high values will result in loss of the finer features of stratification.

Plateau Filling at 1.0 cause "forward" moving sedimentary deposits.

Smooth Edges

To avoid too many sharp edges and jagged artifacts in your strata, Smooth Edges allows you to do just what the name says – smooth the edges. Depending on your visual requirements, a good amount of smoothing can add more realism since sharp edges in the real world are eroded by water and/or wind erosion. Values of 0.05 to 0.45 are usually enough.



Smooth Edges at 1.0. Note that all sharpness has been removed from the edges.

Smallest Feature

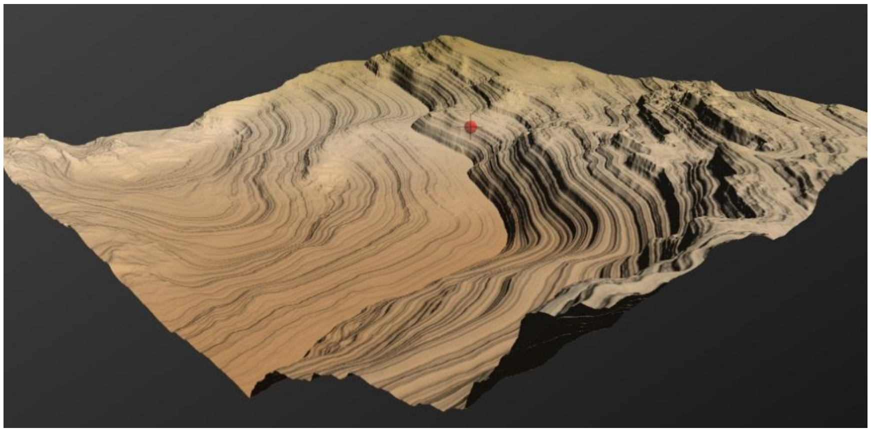
Like with the Terrain Fractal, this controls the smallest amount of detail. The default value of 0 means that the detail is almost infinite – the closer you zoom in, the more the fractal will undergo iteration resulting in more detail. If you do not require too much detail, higher values can be applied – especially useful for very far away terrains.

Tilt Heading

One of the most exciting new features in fractals is the Tilt in the Strata filters. The Tilt Heading sets the direction towards which the strata will tilt the terrain.

Tilt Angle

This setting is the angle at which the strata will tilt. At 0, the tilt is absolutely parallel to the ground (no tilt) and at -89.9 or +89.9, it is almost vertical. Note that as we saw earlier this is not a simple rotation but an actual tilt at the fractal production level which causes the shape of the fractal itself to change according to the tilt.



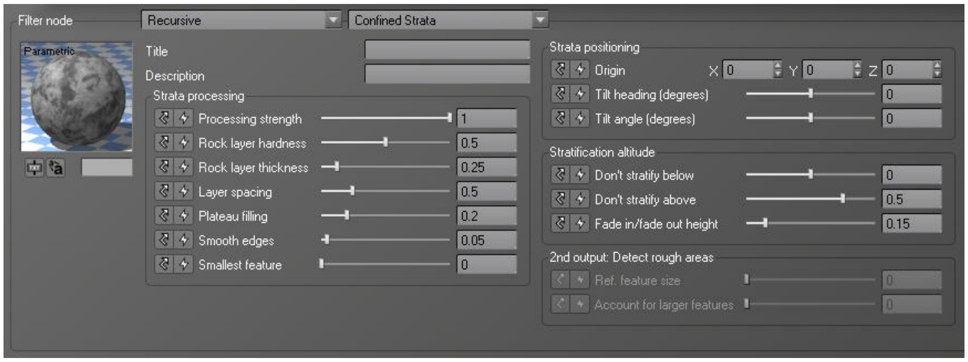
A Tilt Angle of -45 degrees.

Tilt Offset

The Tilt Offset can be used to fine tune the tilted strata in case you see features you may want to avoid. Essentially, it shifts the stratification process “forward” or “backward” along the Tilt Heading. Think of it as a water wave moving forward (or backward, to be a little imaginative). This allows for a certain amount of tweaking of edges by offsetting them to get or remove a certain feature.

This is a very case-specific setting and no ‘recommended values’ are applicable. It is best to experiment with these settings to find the right values for each terrain.

Confined Strata Filter Settings



The Confined Strata Filter has two set of controls, the Stratification Altitude and the Tilt Origin that are different from the normal Strata Filter.

Strata Position > Origin

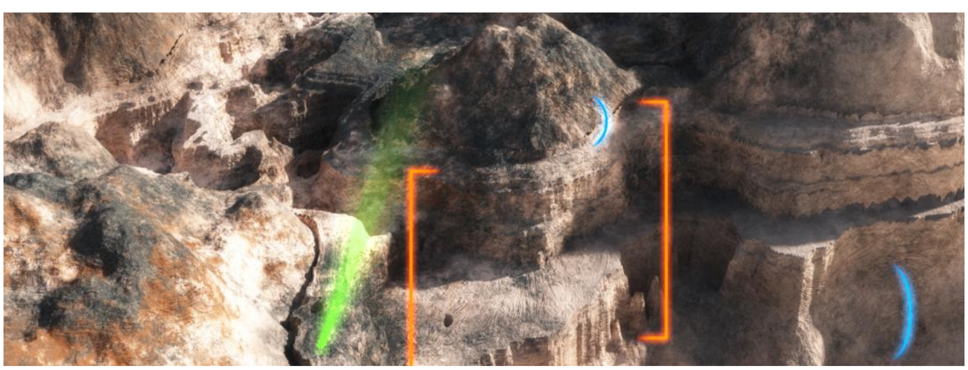
The Origin setting can be used to define the location on your terrain where the tilt will 'start'. This setting is relative to the center or origin of the fractal/terrain. The green line in the image below shows the Origin when set to the 'bottom left' corner of the terrain.

Do Not Stratify Below / Above

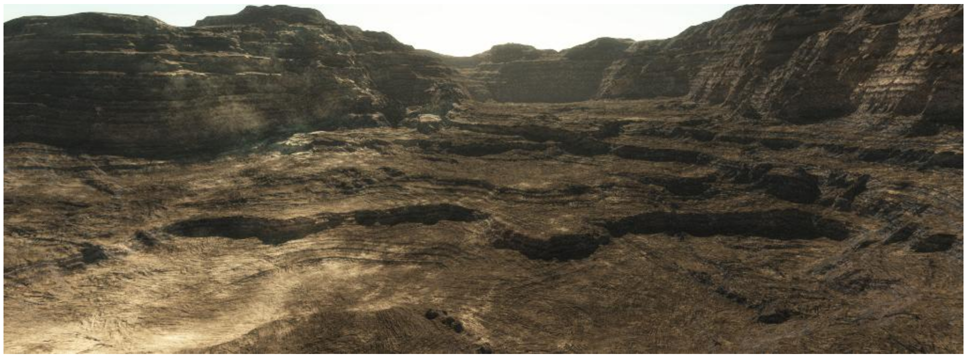
These two settings decide the altitudes between which the stratification will be applied. The orange lines show the range in the example image below. This is the only range of the terrain that is stratified.

Fade In/Fade Out Height

This setting represents the relative height where the fading of the stratified and normal fractal output is merged. The higher you set the value, the smoother the transition will be. However, high values may result in unrealistic results. The fade in/out range is shown with blue lines (both above and below stratification) in the image below.

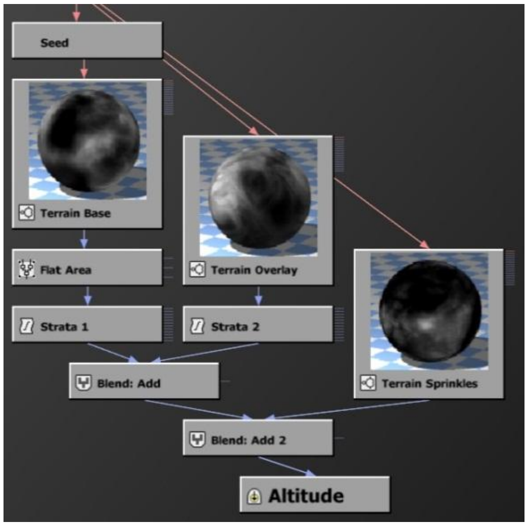


Mixing Fractals



One fractal is often not enough to give the terrain shape you want. By combining more than one fractal in a Procedural Terrain you can easily achieve different effects such as localized erosion, unique formations in only one sector of the terrain, or even lakes and rivers that are created without affecting the rest of the terrain.

Desert Strata Example



In the stratified desert terrain above, a Terrain Fractal is passed through a Strata Filter, and then blended using a Combiner node with another Terrain Fractal with smaller Feature values to create the minor details over the larger strata.

Terrain Base

This Terrain Fractal creates the base shapes of the terrain with medium level distortions and large formations using the Largest Feature. A “Flat Area” MetaNode is used to flatten the central area of the fractal.

Terrain Overlay

This largely distorted and ‘mangled’ looking Terrain Fractal adds almost random shapes. They are called random only because they are different from the Base fractal.

Strata

Both Terrain Fractals are passed through a Strata Filter (both Strata 1 and Strata 2 are identical) to stratify the entire terrain. Large amounts of Plateau Filling are applied to create believable stratifications rather than simple terraces.

Blend: Add

The two Strata Filter results are passed through a Blender Node that uses the Add mode to mix the two fractals using a ratio of 0.15, so most of the Terrain Base fractal is preserved and only a small effect of the Terrain Overlay fractal is added to the mix.

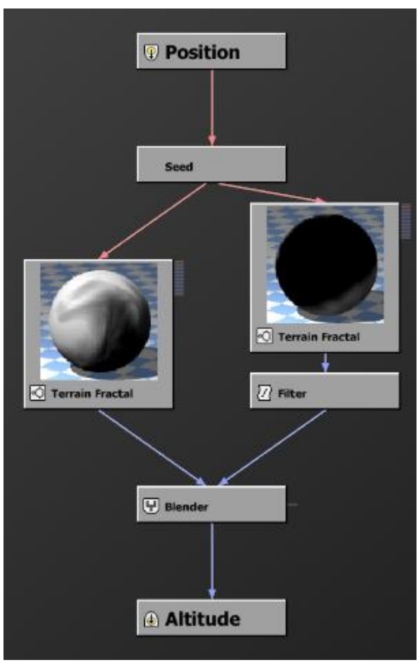
Terrain Sprinkles

This is a very small MetaScale and Largest Feature enabled Terrain Fractal with very strong yet small distortions for adding superficial features to the terrain.

Blend: Add 2

Another Blender node is used to mix the output of the previous blending with the Terrain Sprinkles. This Blender also uses the Add mode, and mixes it with a 0.4 ratio. The Terrain Sprinkles has a low Gain value, so a larger ratio is allowed as opposed to the minor ratio in the previous blending.

Large Scale example



Two large scale fractals combined using a Blender node.

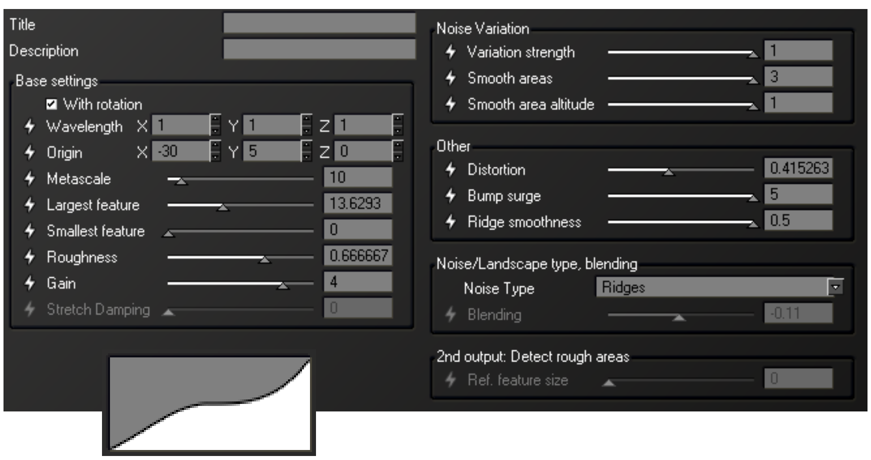
Another example is this large terrain as seen from a very high altitude, that we first saw in the Terrain Fractal section at the start of this chapter.

As mentioned before, the long 'ground' level is created using one fractal and the mountainous areas by another.

The Function here shows exactly how the terrain was made. Over the next pages we will deconstruct this example and see how it was put together, and what each node does.

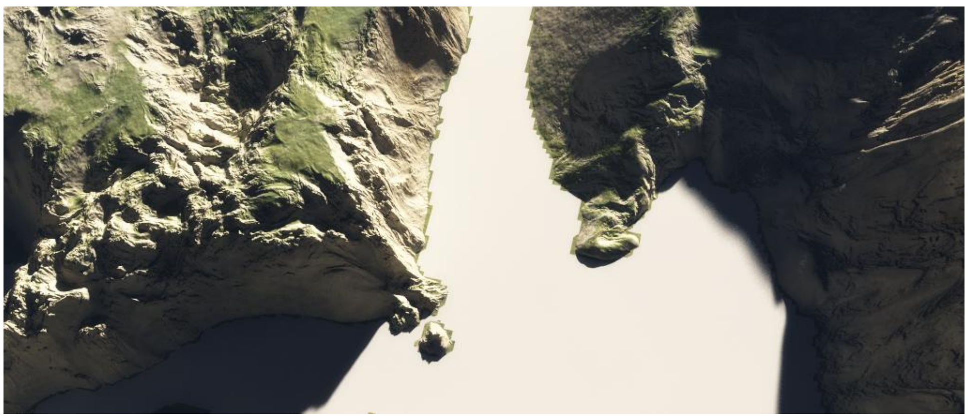
The long striations of the 'ground' fractal are caused by the strong Distortion value. They are exaggerated in the fractal itself so that when it is blended with the 'mountain' fractal, the damped down version of this fractal will maintain the essential features without losing them in the blending.

The solid mountainous formations are enhanced by the filter shown overlaid on the settings. These formations exist as large separated entities as opposed to the joined striations in the other fractal. This separation helps create a better blend.



Both terrain fractals are combined using a Blender node using the Blend mode, and a ratio of 0.7071.

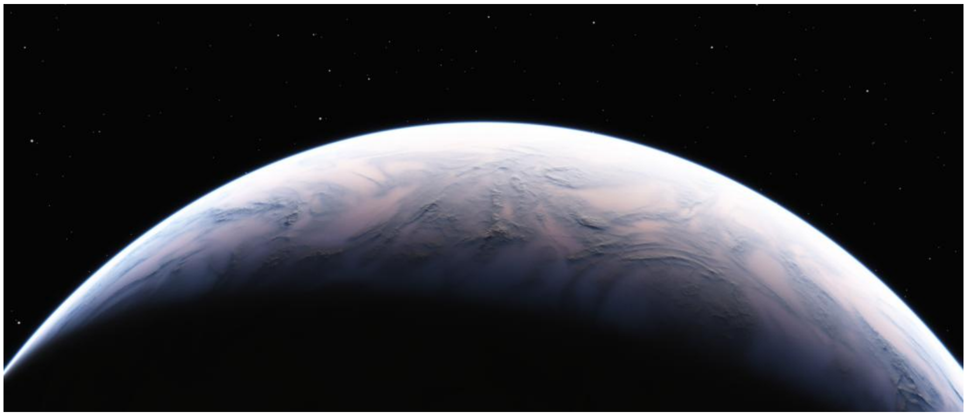

The base or "ground" fractal.


The overlay or "mountain" fractal.

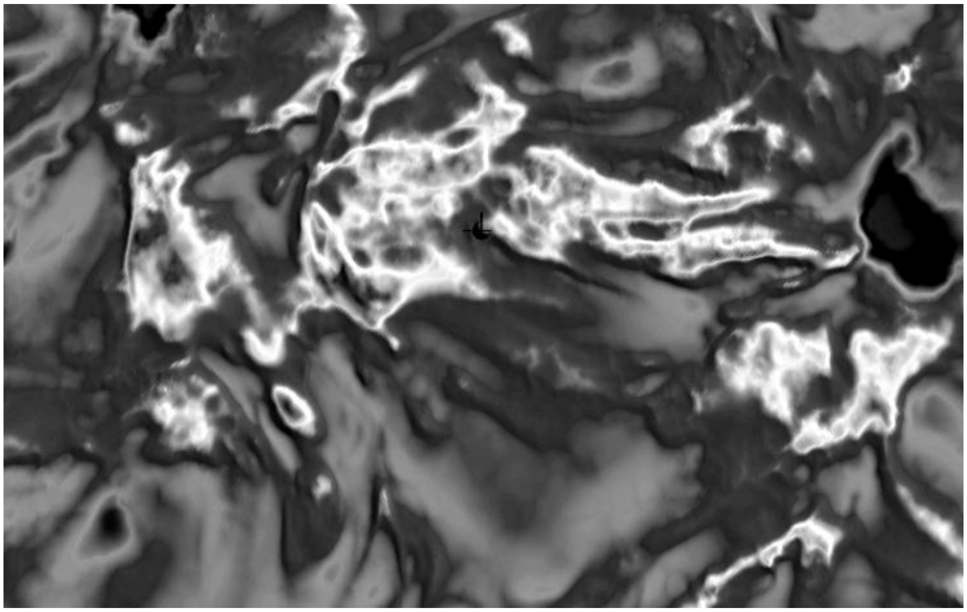

The combined fractals produce an elaborate result.

Planetary Terrains

Creating a planet scene in Vue can seem tough, but once you get a handle on the Terrain Fractal, you can create some intensely creative terrains that encompass an entire world by using just that one fractal.



This planet scene was created using a single Terrain Fractal shown below:



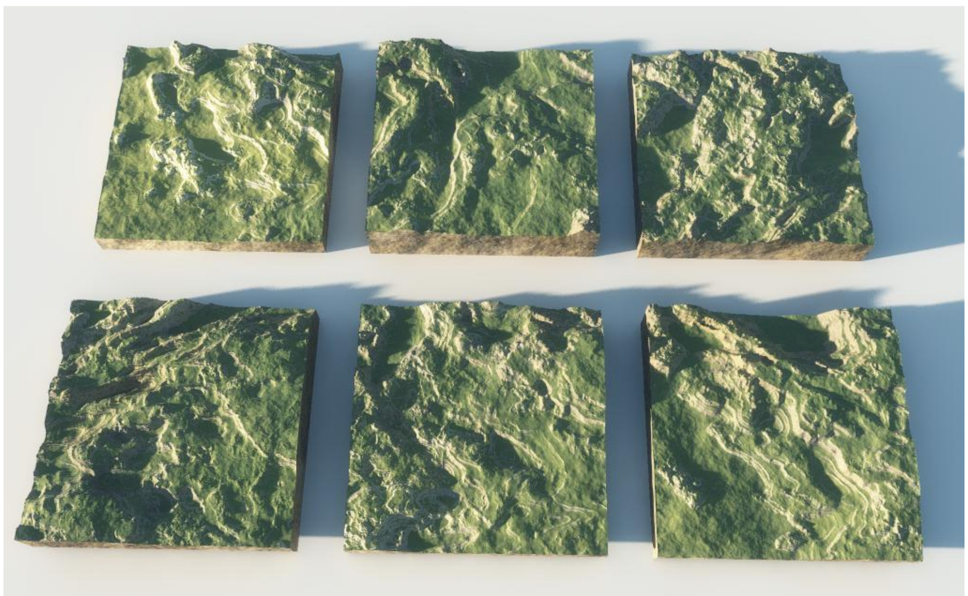
A terrain fractal with a very large MetaScale of 500000, Largest Features set to 150000 for giant mountains, and Roughness of 0.63 for creative shaping. Distortion was set to 0.57315 with a 4.0 value for Bump Surge to make the mountains seriously tall. Finally, a Noise Variation Strength of 0.9 created large variations in the altitudes between different levels of the terrain.

Infinite Variations

One of the best advantages Procedural Terrains give you over standard terrains is the ability to create as many variations based on the same fractal.

To do this, all you have to do is change the Origin value of your fractals that contribute to the terrain shape. If your fractals draw upon a Seed node created by Vue, you can change the origin values of all fractals drawing from that Seed node by changing the Offset values in the Seed node. You can also change the individual fractal Origins to achieve greater control over the shape.

Below are examples of 6 different terrains achieved from one fractal by changing the Origin property.



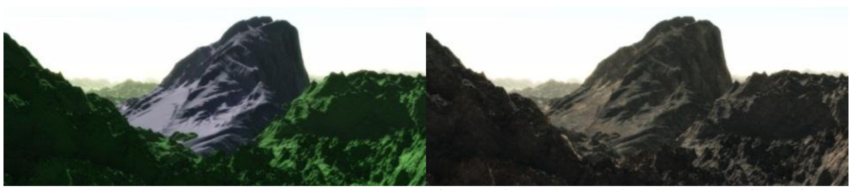
Changing the Origin property can completely reshape a terrain.

Combining with Infinite Terrains

A single mountain or hill can be confining when trying to compose your scene. Having a horizon is helpful in such cases, but you need the general terrain (not the mountain itself) to extend into infinity in the horizon. The obvious solution here is to use an Infinite Procedural Terrain. However, instead of using the same terrain and turning it into an infinite terrain, it is much better and easier to combine it physically, not fractally, with a new infinite terrain.

The key here is to use the same material (or a variation, if needed) on both terrains. If your individual terrain has Zero-Edges turned on it will be better still as it will merge easily with the infinite terrain.

Below are a few examples to illustrate this method:



Screenshot and render of infinite terrain (green) and Standard terrain (blue).

A Procedural Infinite terrain with a Standard terrain inserted.

As clearly visible in the image above, if your materials match the infinite and non-infinite terrain will merge seamlessly.

HyperTerrains and HyperBlobs

Chapter 5

Breaking limits

One of the most restrictive aspects of 3D terrains has been that all terrains are confined to up-down topography only. Think of contour maps of a location. You can see the peaks of a mountain and the flat areas near a lake but you cannot see caves, alcoves, arches, natural bridges, and so on.

How can we create convincing digital images of nature if we cannot replicate such common and beautiful natural phenomenon? Obviously, you could use a sculpting or modeling software to create such objects but there would be certain disadvantages: the detail would be fake-ish because it would be handmade rather than procedural; it would be extremely time consuming; it would be extremely difficult, if not near impossible, to create similar looking variations based on one object. The new terrain sculpting tools are big step forward in this area but it is still limited by many of these same restrictions along with some new ones.

I started experimenting with different techniques until I accidentally used Displacement on a MetaBlob. This suddenly opened up new avenues for creating groundbreaking rock formations. Thus were born HyperTerrains, a fractal modeling technique.

Three years after HyperTerrains were invented I started experimenting with “Mark III” HyperTerrains. Instead of using displacement, a HyperTexture was used to create the surface of the MetaBlobs. The idea was passed on to the creators of Vue and in the next major release – Vue 9 – the new HyperBlob technology was introduced.

HyperTerrains

Creating HyperTerrains

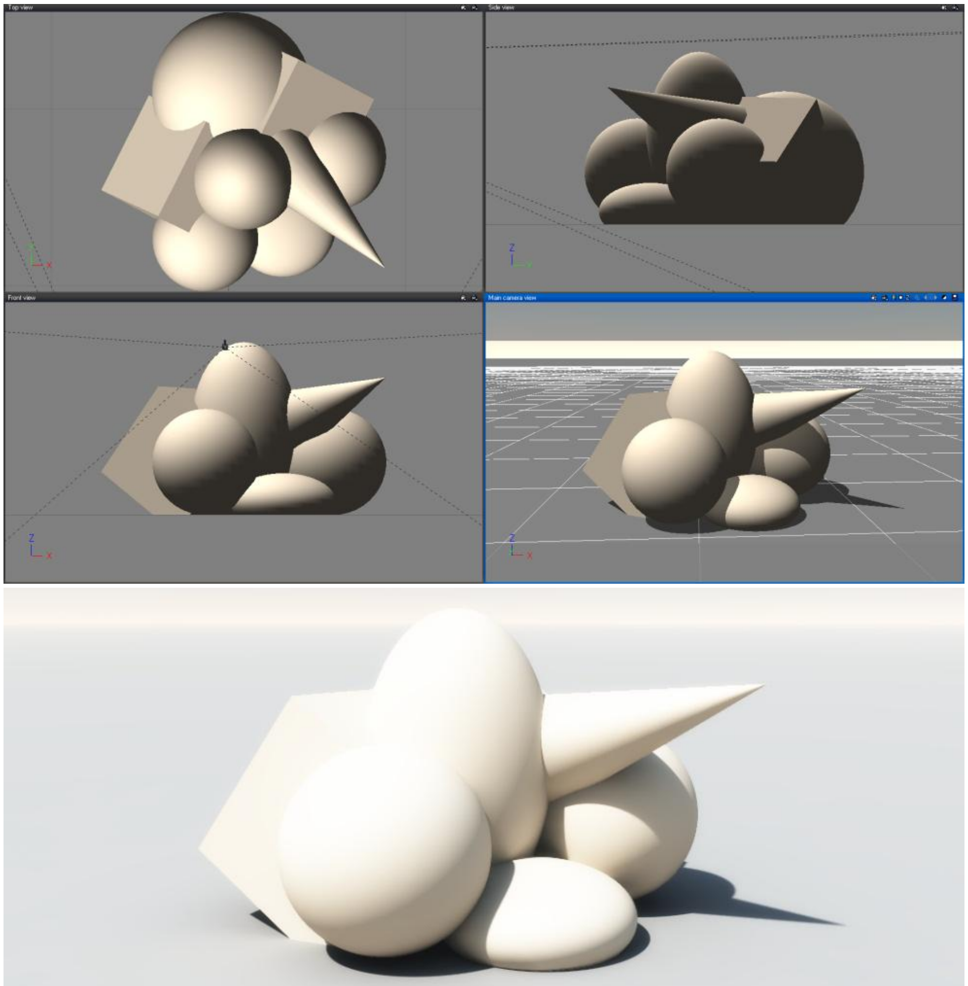
The image below shows the basic HyperTerrain creation process – Primitive Geometry becomes a MetaBlob, then the MetaBlob is displaced into a HyperTerrain using a fractal function defining the sub-primitive shapes.

The basic steps of HyperTerrain creation

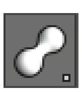
Let's start by making our first HyperTerrain.

Define the MetaBlob

To create a HyperTerrain we start out by defining our vague shape in the form of a MetaBlob. In Vue, MetaBlobs are made from the Primitive Geometry offered in the forms of Cube, Sphere, Cylinder, Cube, Cone, Pyramid, and Torus.



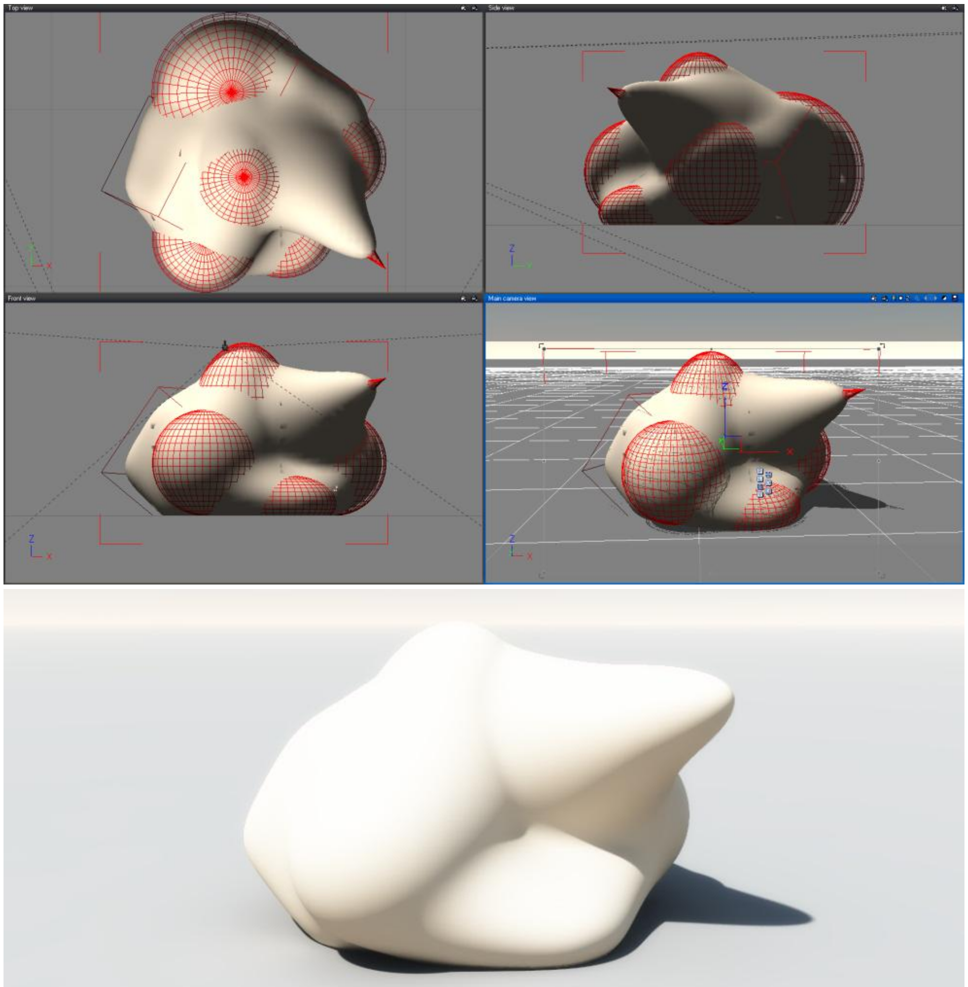
5 spheres, 2 cubes, and 1 cone creating a vague shape



Create MetaBlob button

Now we convert into a MetaBlob using the MetaBlob button on the left toolbar.

The primitives will be used as an inner boundary instead of an exact shape as in the case of a Boolean operation. While this may seem somewhat detrimental to the traditional modeling method, it does serve a very important purpose. As with sculpting – digital or traditional – first a working shape must be sculpted onto which several iterations of details are carved. Similarly, this vague shape



The resulting MetaBlob

This is our basic shape. Think of it like a lump of clay that will be sculpted into a statue.

There is no limit to how many primitives you combine, but the better way to get more detail and shape is to work on the Fractal Displacement rather than the primitive geometry.

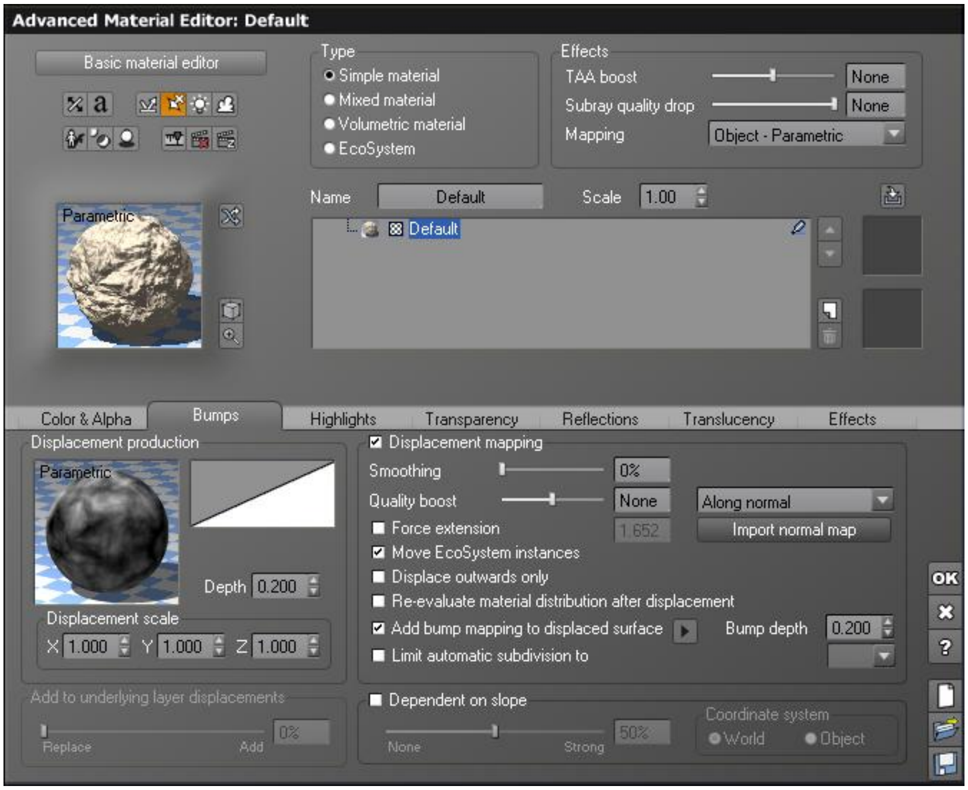
Define the Displacement Fractal

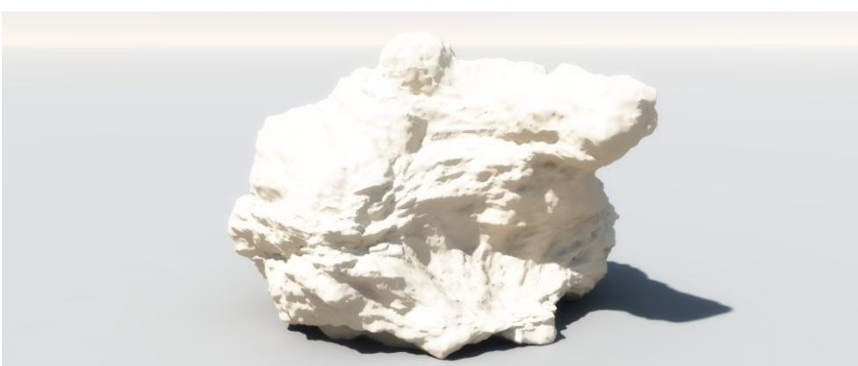
The next step to creating the natural details of the rock formation is to create the Displacement Fractal for our MetaBlob. Using a Fractal for displacement has the great advantage of making our HyperTerrain procedurally generated, giving us an infinite amount of detail and creating an almost natural shape which we normally cannot do by manual modeling.

In the Material Editor, go to the Bump tab and go to the Load Function dialog. To do this, right-click the Bump thumbnail and select Load Function.

Since this is our first HyperTerrain we will use a Vue fractal that ships with the software rather than making our own. Vue ships with some amazing fractals which may seem simple, but provide exquisitely detailed results. For our purposes, we will select the fractal “Transformed Cristals”.

Now that the Fractal is loaded, let’s enhance the effect by setting the Bump Depth to 2.0, and checking on “Displacement Mapping”.

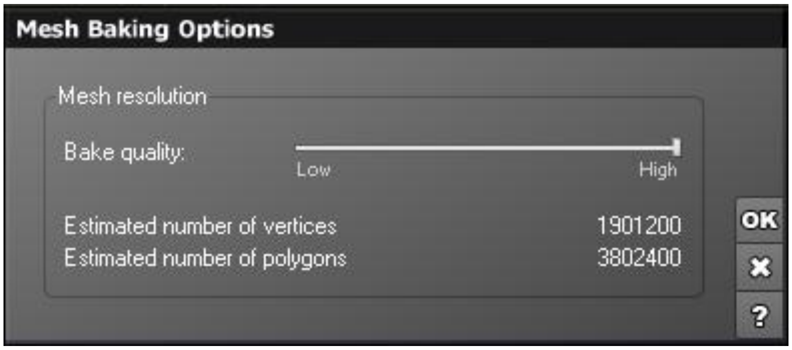




The resulting displaced MetaBlob

Baking the MetaBlob

To preserve the shape of our displaced MetaBlob, let’s bake it by right-clicking the object and selecting Bake to Polygons. In the dialog that shows up, drag the Bake Quality to maximum. In Vue 7.5 and earlier versions you will reach near a million polygons, while in Vue 8 and higher you can go to 4 million polygons. Click OK to bake the MetaBlob into a solid polygonal object – our HyperTerrain.



You can choose to leave the MetaBlob with live Displacement, but we will discuss that technique in the next section.

Assigning a new Material

Now that we have our HyperTerrain, we can load a new material without affecting the displacement map. With a new material, this is what our finished HyperTerrain looks like.

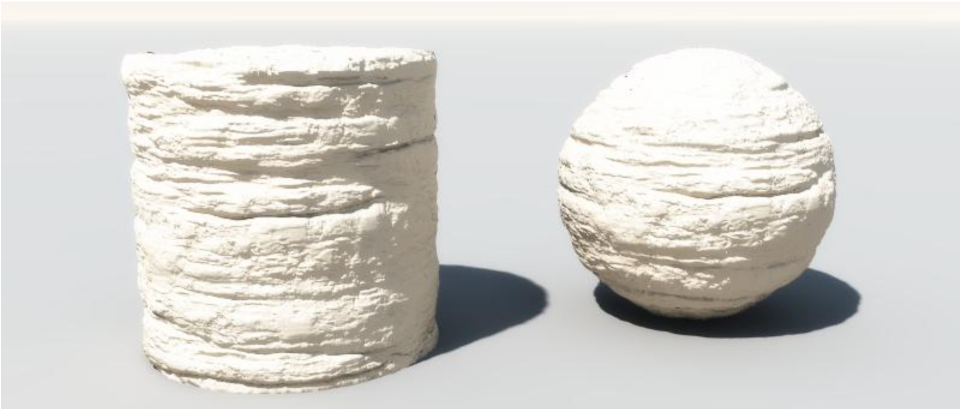
HyperTerrain Displacement

Displacement serves as the key function in the HyperTerrain creation process. This section explores some of the aspects of Displacement you can apply using common fractals found in Vue, as well as controlling the scale to create effective displacement shapes.

Vue comes with several pre-built Fractals that provide excellent displacement for HyperTerrains. In most situations you can use a single fractal from this library to create your HyperTerrain. Other times, you can combine one or more fractals for more detailed shapes, as we explore in the next exercise.

First, let’s explore some of better fractals from the Vue Library.

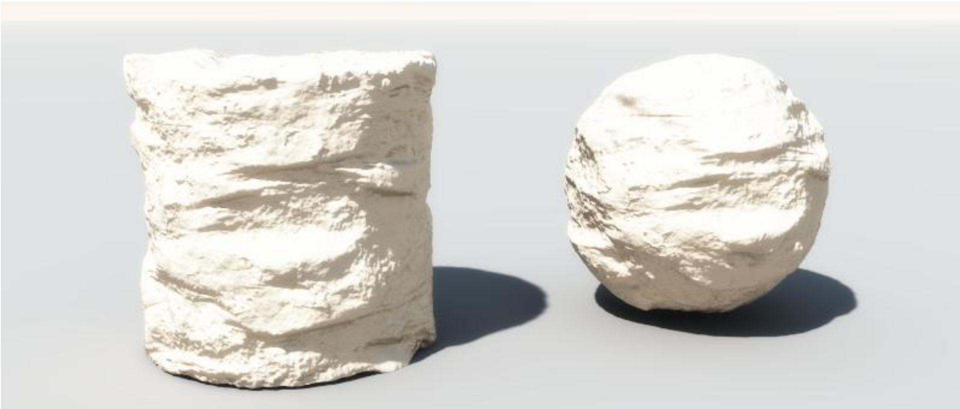
Sedimentary Rock



A very common yet very powerful function that provides a sedimentary look that can be applied to any type of rock formation ranging from rocks in a flowing stream to large cliffs on the shore of an ocean or even large mountains in a jungle.

For tall HyperTerrains (i.e., where the height is far greater than the width/length of the HyperTerrain) you may wish to use a less than equal value for the Z-axis of the Bump/Displacement Function. For example, if your X and Y axes are 1.0, then use something like 0.5 or 0.3 (depending on the height) for the Z axis.

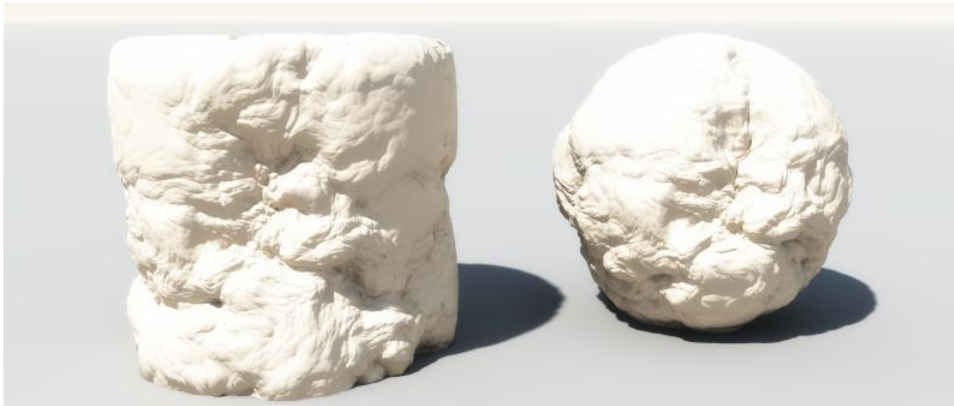
Complex Sedimentary Rock



Similar to the Sedimentary Rock Fractal, the Complex Sedimentary Rock fractal combines several different sedimentary patterns into one giving off a sharper look.

This is ideal for large curving HyperTerrains, but the Scale has to be kept slightly large otherwise the fractal may pinch and/or break the delicate features of the HyperTerrain.

Lava Flow



One of the most versatile fractals is the Lava Flow. It creates soft undulating flow blobs. With strong Bump Depth, this can make for very interesting effects. This is perhaps my most used Fractal.

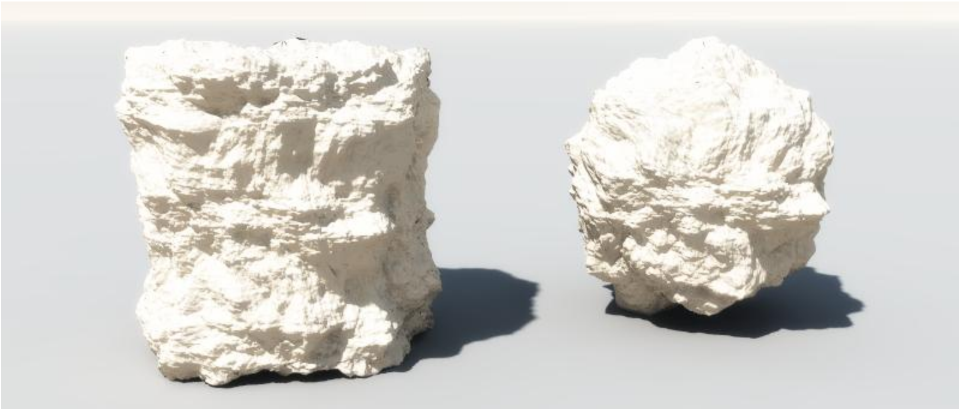
Asteroid

This out-of-this-world fractal is useful for creating cave interiors or jagged rocks. It will feature small craters and hard crevices.

This fractal provides better results on harder surfaces such as cubes, cylinders, and pyramids rather than spheres, torii, or cones.

There is a small warning for using this fractal with large Bump Depth values when baking as some of the sharper curves in this fractal can cause baking artifacts.

Transformed Cristals

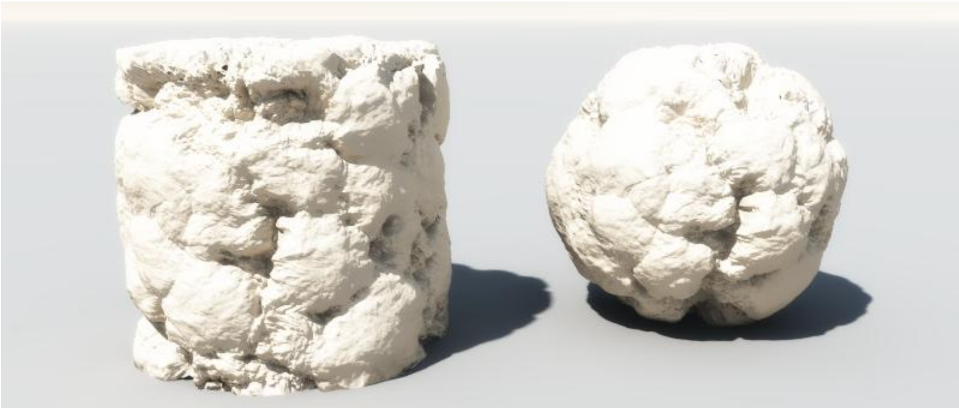


Another versatile fractal, Transformed Cristals provides sharp, jagged edges suited for rock formations or even entire mountains that seem to be jutting out of the earth.

Just like with the Sedimentary Fractal, using a non-uniform value for the Z-axis can produce some interesting effects for tall HyperTerrains. Sometimes even shorter or flatter HyperTerrains can get a better shape with such a non-uniform value.

With a versatile fractal like Transformed Cristals, you can try the same thing on the other axes as well.

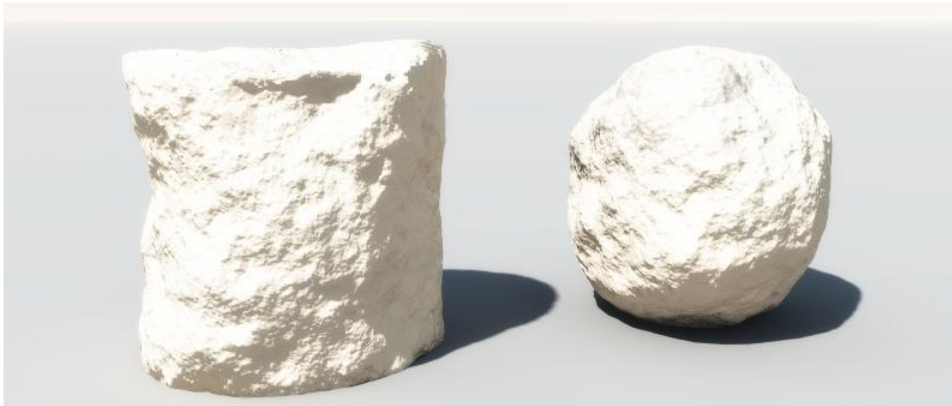
Stones in Dirt



This fractal creates a veritable ecosystem like rock pitted surface on a HyperTerrain. At large scales you can get unbelievable formations in the HyperTerrain.

If you are mixing fractals, consider using a 6 x 6 x 6 scaled version of this fractal to give the overall shape while using the Asteroid or Lava Flow at 1 x 1 x 1 for detail.

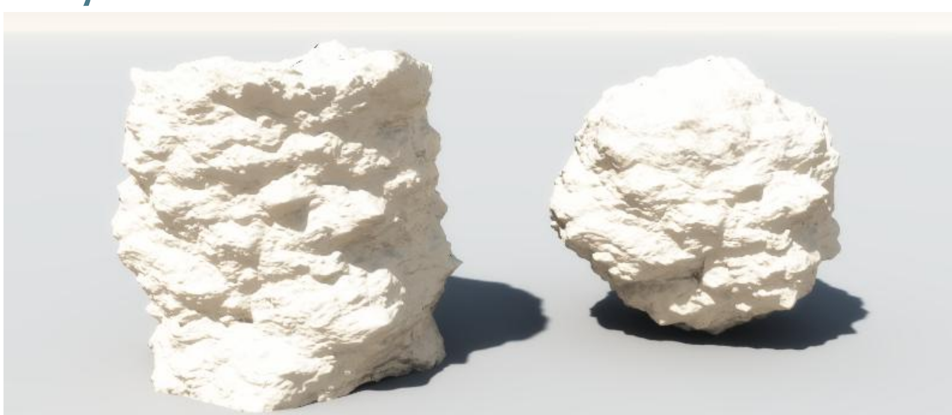
Fast Rock



Just as the name suggests this is a quick rendering fractal useful for distant objects that do not need too much detail.

Just by increasing the Bump Depth value, you can create pronounced shapes with this fractal, which work well both for simple near-by objects as well as large distant objects.

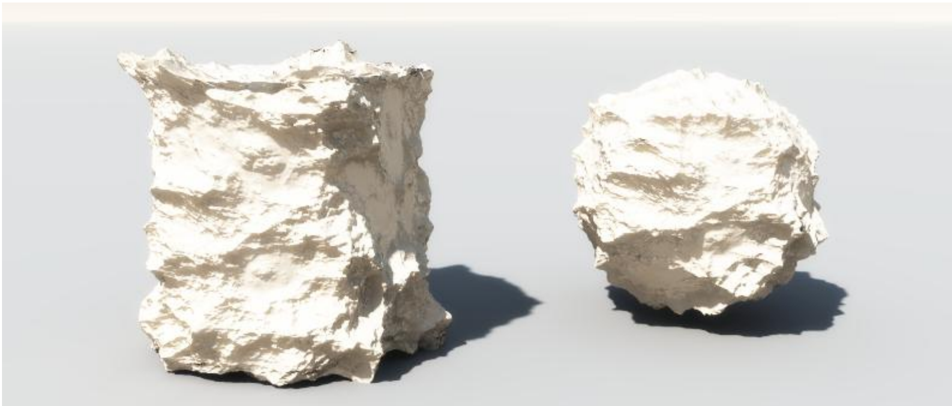
Grainy Blobs



Another fast rendering fractal, Grainy Blobs creates soft detailed rock features. Another fractal you can consider using at a large scale for the shaping of the HyperTerrain.

A great combination is to use this for the base layer and add a new layer that additively blends the Stones in Dirt or Lava Flow fractals with a smaller Function Size.

Ultrasharp



Ultrasharp is an extremely rough and jagged fractal, which when used with care, can create very interesting sharp HyperTerrains and hard cut rock formations.

Smoothing Artifacts

Often, a fractal will create triangulation artifacts or bad spikes in your Displacement process. There are two ways to solve this problem.

The first is to go inside your fractals and raise the Smallest Feature value. This will lessen the smaller shapes being created by the fractal and remove most of the spikes and polygonal artifacts.

The second method is specific to Vue 8 and higher only. In the Bump tab of the Advanced Material Editor, you can use the Smoothing option in the Displacement control group. Usually a smoothing value of 20% is more than enough, as larger values may take away detail from your fractal output.

Balancing the Scale

This is the most critical thing to look into when a HyperTerrain does not displace or bake as it should. For example when you see too many spikes and jagged edges or if the desired fractal shape does not appear at all, this is because of an incorrect Scale setting of the displacement material.

Sometimes you may need to change the scale of the material – either globally (of the entire material itself) or in part (of the Bump function or a Fractal inside it) – to get the shape you want, you must also keep in mind these following caveats.

If the size of your MetaBlob is too small and the scale of the material is bigger (that may happen even at Scale 1.0, relative to the size of the MetaBlob) then the fractal shape applied may be so big that it will not show any particular effect on your MetaBlob. For example, if the object is small than 3 cubic meters, this may happen.

On the other hand, if your MetaBlob is too big and the scale is too small for that MetaBlob, then you will see too sharp edges, spikes, jagged edges, or a completely distorted displacement. For example, if your material Scale is 1.0 and the MetaBlob is over 90 cubic meters.

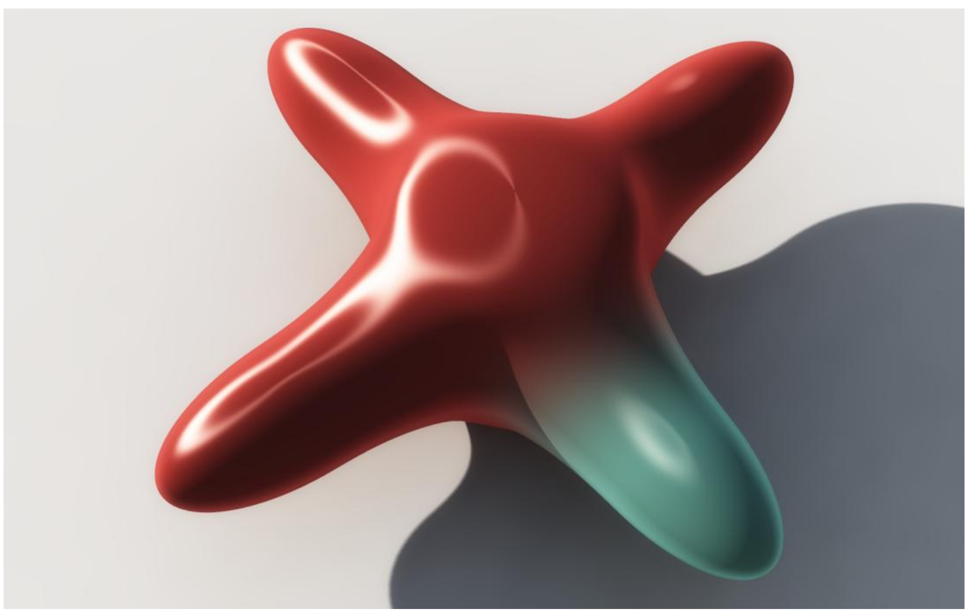
The recommended method is to keep the scale of your MetaBlob primitives near the default size that Vue sets during generation. If not for all primitives then at least the major shape defining ones. This will apply the scale of 1.0 properly. The general recommended range is between 1 to 9 cubic meters.

After you have baked the HyperTerrain, you can make it as large as you want.

In all the HyperTerrains shown in this book, the above technique was used so keep this in mind as a benchmark when you make your own HyperTerrains.

Variable Displacement

Controlling the shape of a HyperTerrain can sometimes be a difficult task since the entire object's shape is being funneled through the rules of a fractal. However, you can still achieve a broader degree of control by changing the material or displacement fractal values of individual fractals.



A primitive with a different material than the other primitives in the MetaBlob.

The primitive that has a different material will blend its own material with the materials of the closest neighbors it blobs with. This does not necessarily mean you should assign a different material. While that is possible and can be desired in some cases, to achieve finer control all you need to do is select the individual primitive or primitives and simply edit their material.

The blending is governed by the primitives blobbing properties such as Intensity and Envelope Distance.

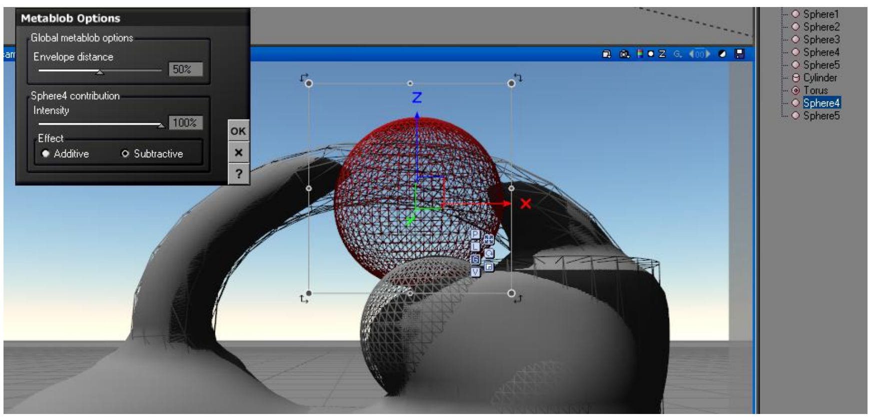
Overcomplicating the MetaBlob by using disharmonious fractals – where the fractal shapes are starkly different – may cause strange artifacts to appear on the geometry. In extreme cases, it will cause Vue to crash or run out of memory.

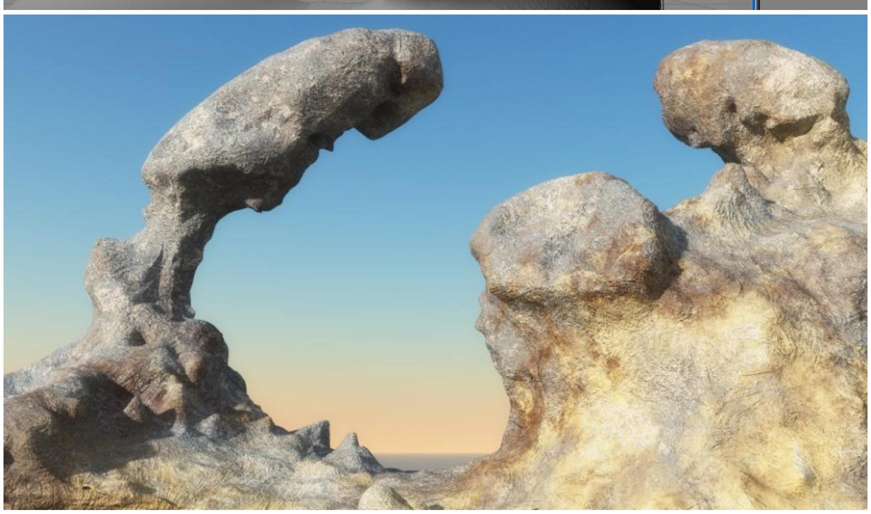
Adding an extra displacement layer on individual primitives in the MetaBlob is a viable practice but must be handled with extreme care as it may cause disharmony in the fractals as well.

Shaping with Subtractive MetaBlobs

One little known feature in creating MetaBlobs, and eventually in creating HyperTerrains, is using the Subtractive effect in a MetaBlob primitive to create 'holes' in your MetaBlob rather than adding a shape.

In the example below, a Sphere is selected and in the MetaBlob Options (Object > Edit Object) window, the Subtractive mode is selected. Now the arch created by the Torus in the MetaBlob is broken in the middle where the Subtractive sphere exists.





More examples are shown later in this section.

Baking vs. Live Displacement

If you want to achieve more detail than normal, you should consider not baking the HyperTerrain but rather keeping the displacement live and then adding a layer or more of materials on top of the base Displacement material.

The differences between a baked HyperTerrain and a HyperTerrain that uses live displacement are shown below.

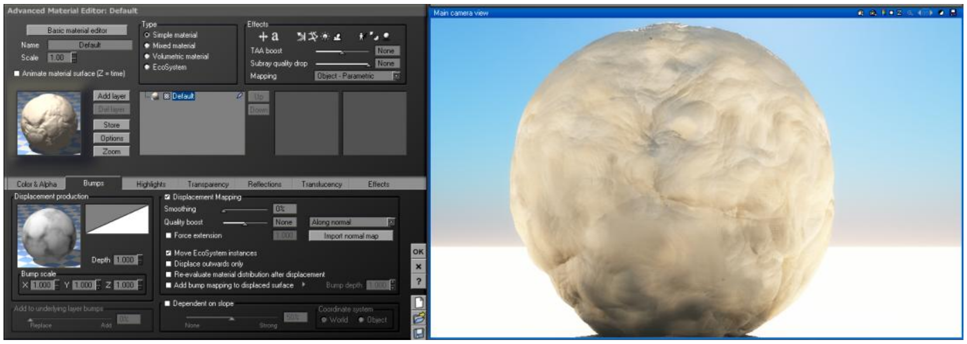
Baking	Live Displacement
4 Million polygon limit	Theoretically unlimited polygons
Faster renders, limited quality	Slower renders, better quality
Easier management of visual material when scaling them. Especially useful for using Mixed Materials.	Layered materials providing the visual material have to be managed in conjunction with the displacement material layer.
Larger file size and slower loading time, very quick render buildup.	Smaller file size and faster loading time, render buildup takes longer.
Baked meshes are permanent. You cannot go back and edit anything.	Live displacement always allows editing any aspect of the mesh.
Same memory usage whether near or far from the camera.	Lower memory usage when distant from the camera.

Live Displacement previews will often look somewhat different from the actual displacement. Use a Final quality render for testing Live Displacement.

Live Displacement HyperTerrains often come in handy when you are doing a close up shot of a HyperTerrain and prefer to have more detail.

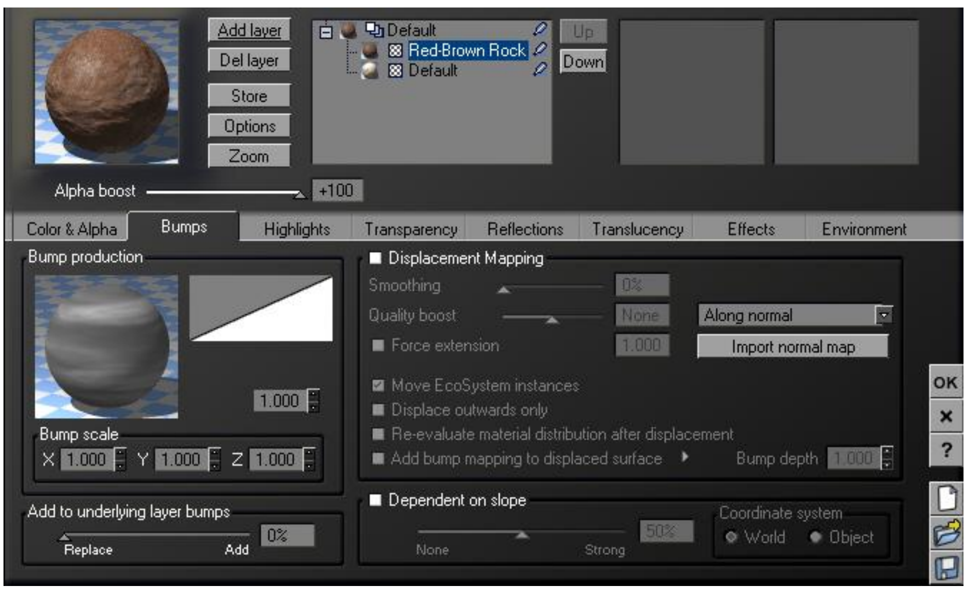
We will now explore the proper procedure for adding a new material layer to our live displacement material (which for the sake of convenience will be referred to as the Displacement Layer) while making sure that all the hard work and effort put into the displacement fractal is not affected by our new material layers.

Creating a Multi-Layer Live Displacement Material



Let’s start with this simple spherical MetaBlob with the Lava Flow displacement fractal loaded. The Advanced Material Editor window and the viewport render show the state of our material.

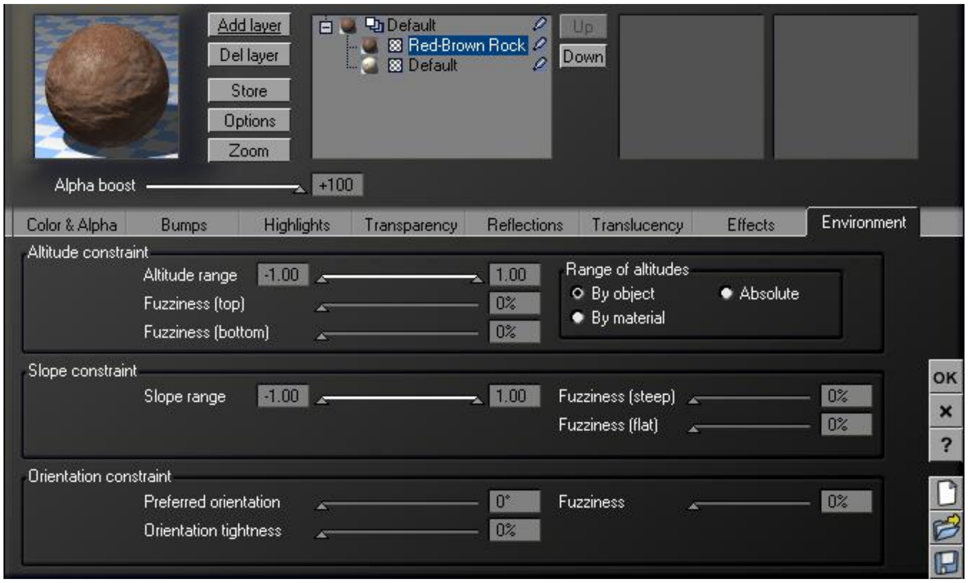
To add our new material, click the Add Layer button and load a new material: for example, the Red-Brown Rock material in the Rocks category.



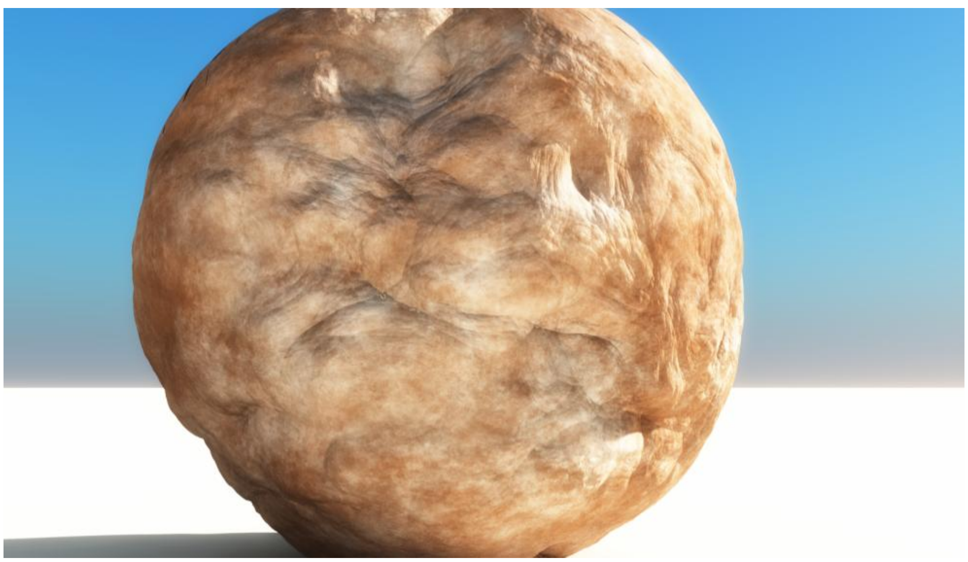
When we load this material, a new layer is added. But some Vue multi-layer material defaults will affect our Displacement Layer, such as Alpha Boost, Highlights, and Bump.

By default, the Alpha Boost will not be at 100%. However, we need it to be at 100% so it covers up our underlying Displacement Layer.

Next, we go to the Bumps tab and move the ‘Add Bumps to Underlying Layer’ slider from Replace to Add (100%). Make sure Displacement is turned off on the new layer.



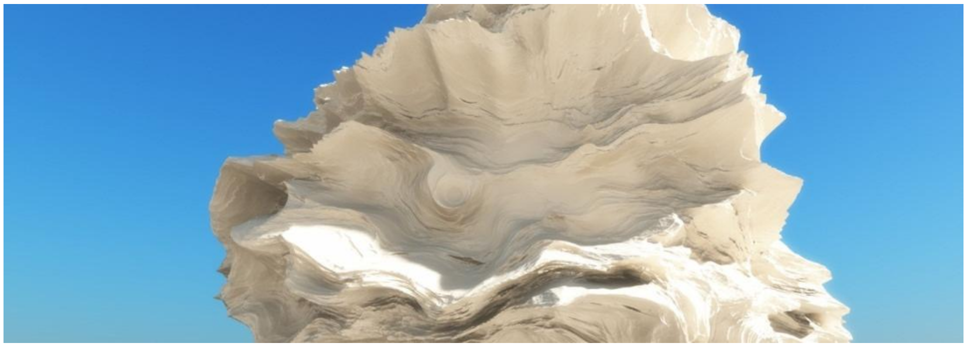
The Environment settings also affect how our new layer is applied on the base layer. So make sure all sliders are at their default values. Mainly, the Fuzziness sliders should all be at 0 and the Slope and Altitude sliders should be at their minimum and maximum values as shown in the screenshot below.



The final render showing a multi-layer material with live Displacement.

If you notice any strange artifacts in your renders, like too much shine, or unwanted transparency, set all the unused sliders on Displacement Layer in the Transparency, Shine, and Reflection tabs to 0 (or the appropriate minimum value).

The Will of the Fractal



Procedural fractal modeling requires something that is not commonly practiced in the 3D world. It is a partnership with the fractal being applied and yourself, by accepting “The Will of the Fractal”.

Because a fractal in Vue is more or less random, you are bound to end up with something slightly different from what you originally intended. After creating over 2000 HyperTerrains of all shapes and sized, one conclusion I have reached is that you can never really draw something on paper and expect to get the exact same from your displaced MetaBlobs. Not without excessive amount of work – and even then it will still not be EXACTLY the same.

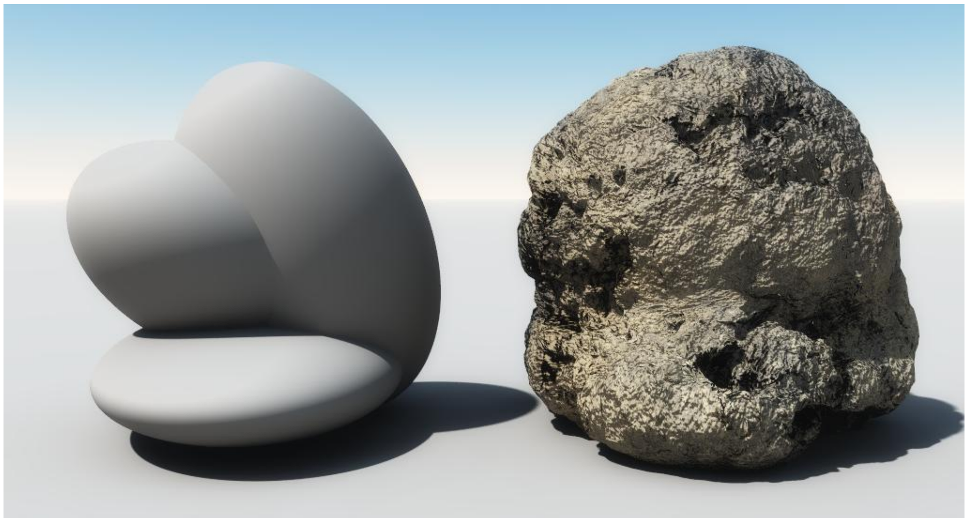
This does not mean you won’t get a shape you are looking for. If your MetaBlobs are perfectly created and placed, and the displacement fractal is well formed, you can get excellent results. But no matter how hard you try, they won’t be truly “modeled” in the traditional 3D sense. The fractal itself will create shapes and details you did not anticipate or want.

To a new HyperTerrain modeler, this may seem like a problem or a restriction. Personally, I’ve never experienced it as such. The perspective that has helped me create truly wonderful HyperTerrains is to approach it with only a little planning and be ready for a “fractal adventure”. By letting the random values in the different fractals surprise me more often than not I get a rock formation that is better than what I was originally hoping to create. It requires a little flexibility to shift your scene’s concept slightly sometimes which may seem like a pain, but the results are always worth it.

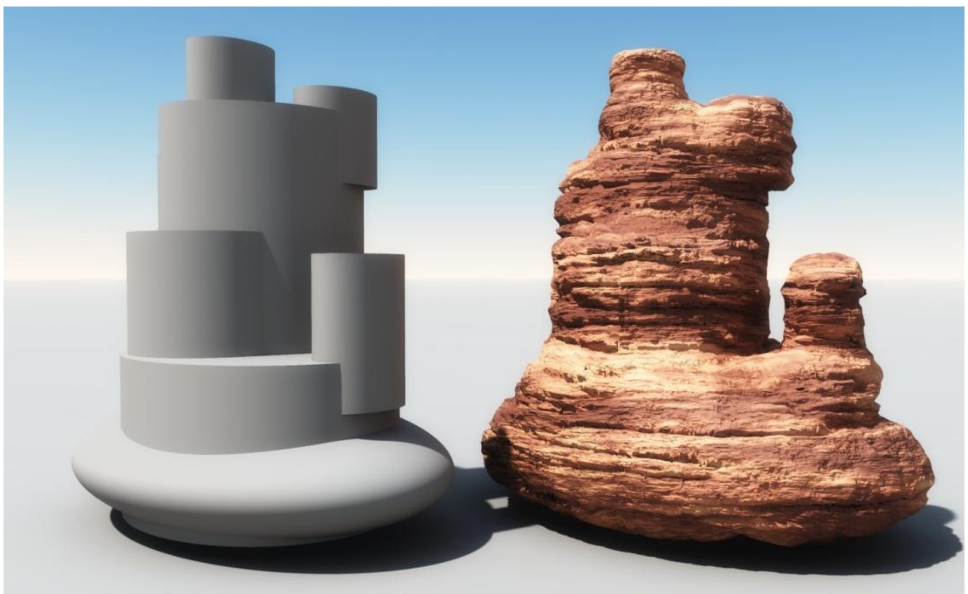
The best advice I can give from my own personal experience is: learn the depths of the fractals and their uniqueness, and that is the best way to achieve a greater degree of control over the HyperTerrains.

And always remember, the MetaBlob only defines the boundaries. The actual HyperTerrain is modeled in the displacement!

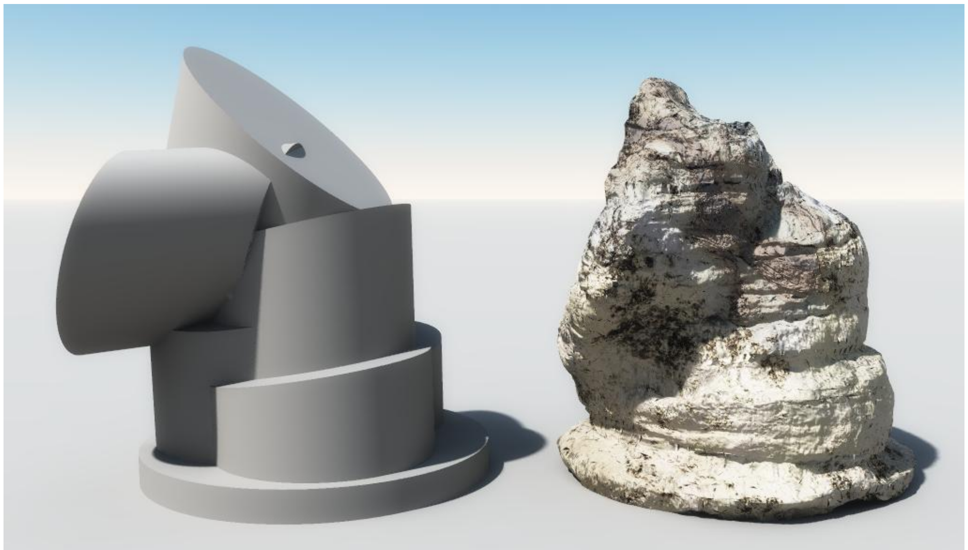
Exploring HyperTerrain Styles



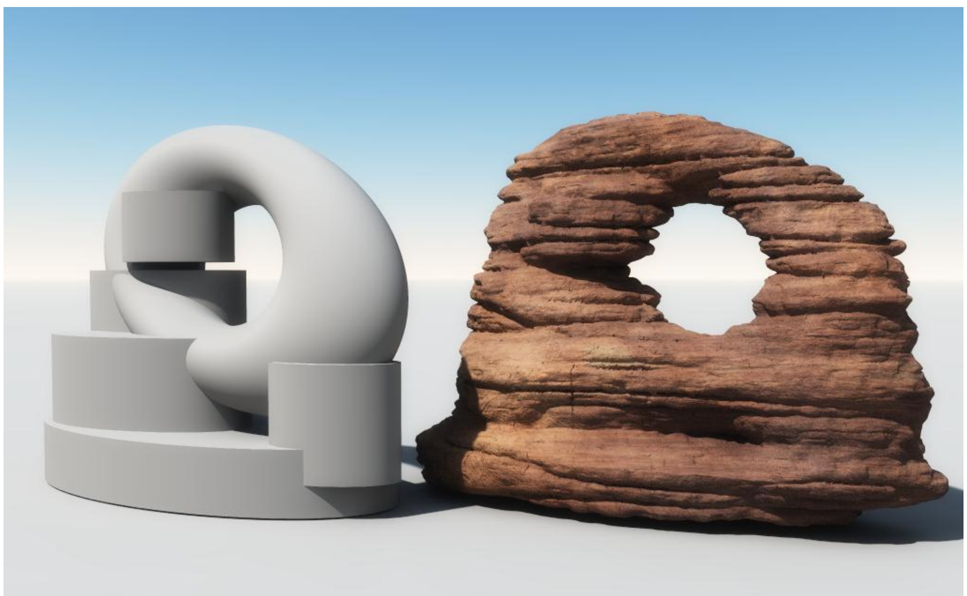
Boulder created with squashed spheres.



Mesa type formation created using a sphere and cylinders.



Limestone cliff rocks created using only squashed and rotated cylinders.



Desert Sandstone Arch created with cylinders and a deformed torus.

HyperTerrain Scenes

Magical Forest

A stack of 9 flattened Spheres with interlocking Cylinders at the base and top create this fantasy rock monument. Spheres have a way of blobbing a little too much, so using a Cylinder can help restraint the MetaBlob shape. However, it has to be used with care as sometimes a bad “join” caused by the position of the Spheres near the Cylinder can tear the mesh apart. In such cases, do some test renders by moving the Cylinder a bit in any direction or by scaling it.

Ruins in a cave

This scene uses only HyperTerrains to create the canyon these ruins are located in. Unlike normal terrains, these are very useful for creating craggy vertical rock faces with various protruding formations. There are three major HyperTerrains: the far cliff wall, the cliff wall right behind and to the right of the ruins and the independent rock formation in the sunlight. All were created using only Spheres with the Lava Flow fractal displacement. QuadSpinner Mineral Infinity materials were used to smooth over and hide jagged edges and broken polygons in these meshes. With a lot of grain in the material the polygons are almost unnoticable.

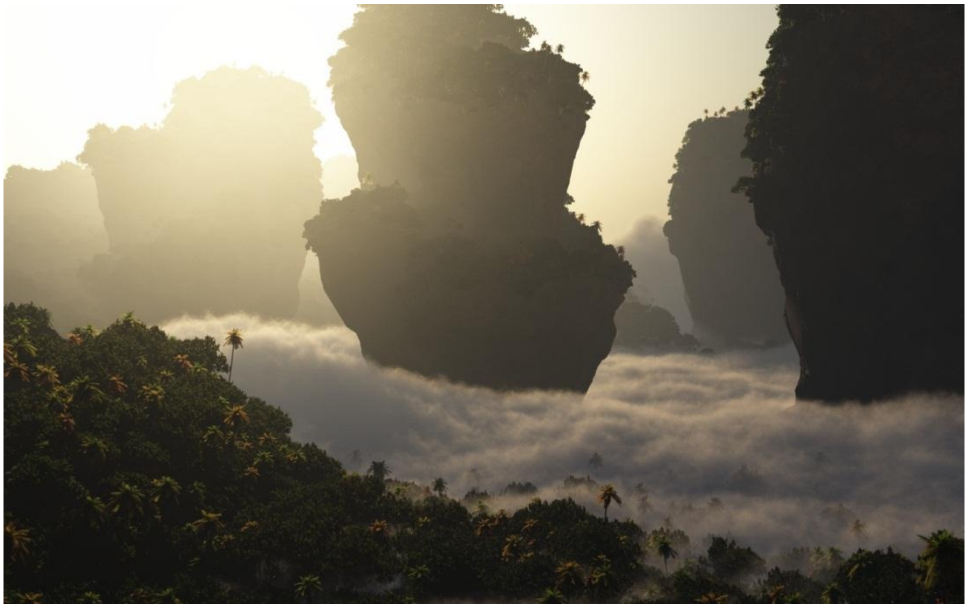
Corner of a Mediterranean garden

This is a simple scene of an old city courtyard. The subject of the render - simple HyperTerrain – is made with 2 Cylinders, half of which is submerged in this scene, a Cone for the central formation, and three rotated and stretched Spheres surrounding the Cone. The Complex Sedimentary Fractal was used to bake this HyperTerrain. Thanks to the blobbing of the MetaBlob spheres of alternating size, you can see some natural rock-like stretched areas in the top half. It is often helpful to rely on the HyperTerrain's MetaBlobbing nature to create distortions. Experiment with various sized spheres by keeping them barely in range so they blob together.

Elder Peaks

This scene uses a complex HyperTerrain as the main subject with normal terrains in the background. The HyperTerrain is a live displacement version with a combination Grainy and Terrain fractal providing the crumbling look and feel. A mixed Snow/Rock material (which we will explore in greater detail in the Materials chapter) takes advantage of the roughness to create a soft sprinkled effect for the snow. The material bump adds to the underlying HyperTerrain displacement. This provides additional microscopic details that add richness to the scene, although at the expense of increased render time.

The base HyperTerrain is made of a single Cylinder, a perfect example of my favorite Vue expression that "the detail should be in the fractal, not the model." Vue's ability to create extremely complex visuals using fractals often helps make objects become more sophisticated than their base mesh may normally allow in any other 3D application.



Tall granite spires coming out the deep tropical jungle.

This scene comes from the 3D World Magazine (Issue #125, available at www.3dworldmag.com) tutorial for creating a rich fantasy jungle. The HyperTerrain pillars are all a single model rotated to look as if they are different.

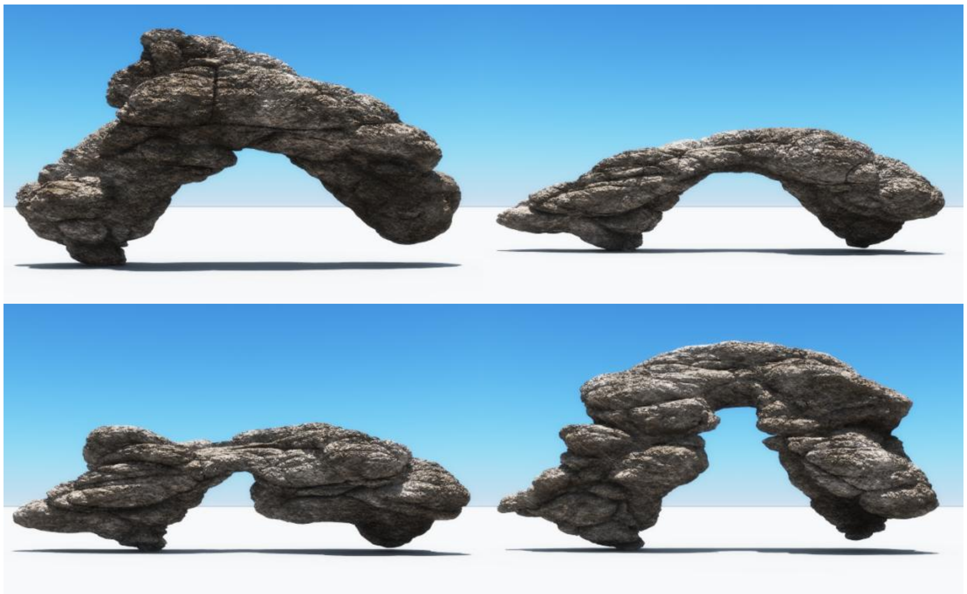
These baked HyperTerrains are made from 6 spheres and 3 Cylinders (for the outcrops/platforms). By rotating the Cylinder on an uneven axis (two axes rather than a single one) and then stretching it vertically allows you to created knife-like shape the normal Cylinder does not provide. This strong edge can be used to create platforms or terraces in your HyperTerrain. A Sphere or two at the base or portion where the Cylinder connects with the main MetaBlob helps ensure a smooth transition rather than a torn mesh.

Vertical cliff face with terraces.

This scene contains two HyperTerrains, one far right in the distance, and one up close with the platform supporting an EcoSystem. Both are the same HyperTerrain.

A 3 Cylinder MetaBlob was subjected to extreme Voronoi fractal based displacement with the Bump/Displacement Value being 32.00. The resulting HyperTerrain has strong vertical cracks that portray a vertical cliff face often found in large canyons.

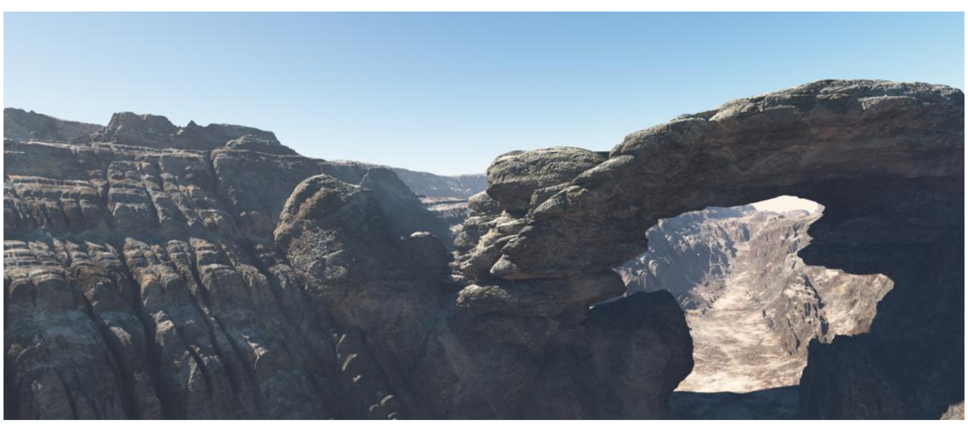
In the baking process, the polygon count (or Mesh Resolution) was kept intentionally low (0.65 million polygons) so the mesh would have square-ish faces. For further refinement, the Decimate function in the mesh was used to lower the mesh by 25,000 more polygons. This can often help create a blocky terrain.



Ancient Arches HyperTerrains

These HyperTerrains were created using a single torus, several spheres and cubes on the sides, and a large cylinder that used Subtraction on the bottom half of the MetaBlob.

Such creative negative blobbing can create very creative shapes.



A large arch rock formation merged with a Standard terrain.

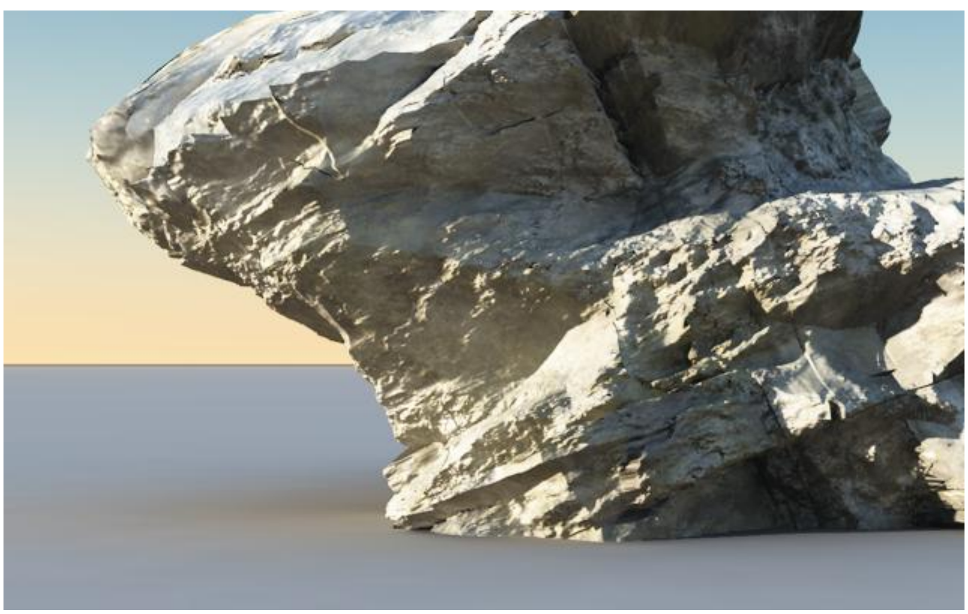
Such an arch can be combined with other HyperTerrains or embedded into a terrain. Refer to the next section for more information on this technique.

Voronoi HyperTerrains in rocky, desert environment.

A powerful Voronoi-based Function is used to create the displacement of the two massive 2-sphere MetaBlob. The first HyperTerrain is the one in the foreground cradling the tree, and the second is a vertical obelisk wedged into the mountain face behind.

These MetaBlobs were baked at 3.5 million polygons for extreme quality. Two level-of-detail versions were created using the Decimate function (in the Edit Object dialog) to bring down the polygon count to 2 million and 500,000 polygons each.

A strict 'Displace Outwards Only' option was chosen for these HyperTerrains, and the Fractal used a strong 15.5 Gain to create the sharp cracks in the rock.



Isolated model of limestone sea cliffs

A single cylinder was displaced with large scaled Square Samples and Crystals format Variable Roughness Fractals, resulting in this strongly eroded limestone rock formation. A compression on the Z-axis by making it less than the other two axes, caused the stratification to occur strongly.

Heat-fused cracks.

Another HyperTerrain made from a strong Voronoi fractal. In this case, not even a MetaBlob was required. A single sphere (slightly squashed horizontally) had the same Function from the previous example applied to it. The gain in this case was lower and the 'Displace Outwards Only' option was disabled allowing for deeper cracks and crevices to be designed.

Such individual rocks are great for monument style rock formations. They can often be duplicated, rotated and placed overlapping each other to create interesting formations.

Extend Terrain Geometry

A HyperTerrain attached to a standard terrain using the same material.

We explored the integration of a standard terrain with an infinite terrain by using a common material. The same is possible for HyperTerrains and HyperBlobs – between themselves or different objects.

The key possibilities are:

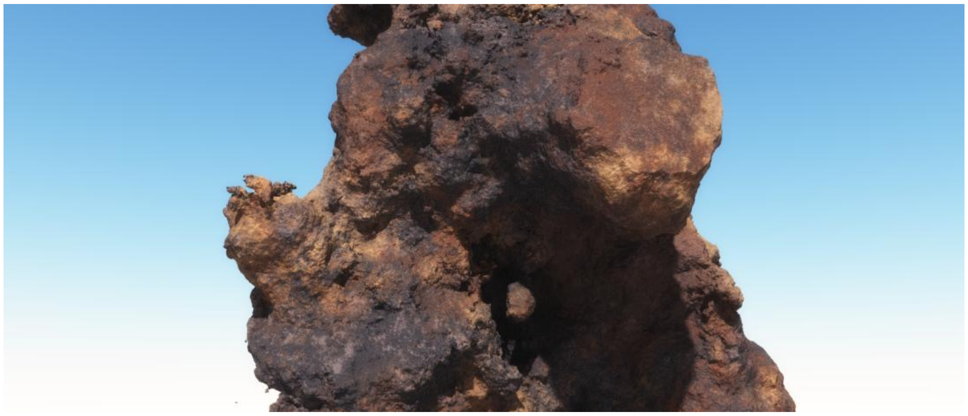
- A HyperTerrain combined with another HyperTerrain
- A HyperBlob combined with another HyperBlob
- A HyperBlob combined with a HyperTerrain
- A HyperTerrain combined with a Standard/Procedural Terrain
- A HyperBlob combined with a Standard/Procedural Terrain

And so on.

The idea behind this is to combine different techniques to get terrain shapes that are not possible by any single method alone. A good example of this is where a procedural terrain with stratification is combined with a HyperTerrain that uses a Confined Strata filter to stratify the bottom third of the HyperTerrain. This allows for a near seamless shape merge when the HyperTerrain is simply position to overlap the procedural terrain. The need to manually model the shape into the terrain is replaced by a more manageable solution.

As with mixing Standard and Infinite terrains we saw in the previous chapter, the materials have to be identical so the general coordinates of the fractal functions that create the bump and color output merge seamlessly with the different objects.

HyperBlobs



A rock formation created using a HyperBlob.

Three years after HyperTerrains were invented, the creators of Vue took an idea I had for extending that process to create even more complex meshes. In the next major release – Vue 9 – the new HyperBlob technology was introduced.

HyperBlobs are very similar to HyperTerrains, yet behave very differently. HyperTerrains are shaped based on a displacement material on top of a MetaBlob and provide detailed control over the shape. HyperBlobs are however made of a MetaBlob with a HyperTexture material and the level of control is sacrificed a little to bring forward a granular method of rock creation that has never been possible before in natural design software.

To understand HyperBlobs, we must first talk about HyperTextures.

Understanding HyperTextures

A HyperTexture is a volumetric material. Vue takes the MetaBlob and treats it as a three dimensional boundary inside which a HyperTexture is "filled". Imagine a blown up balloon being filled with toothpaste.

A single sphere with a HyperTexture material applied.

With HyperTerrains, the MetaBlob serves as a starting point rather than a limiting boundary. When a displacement fractal gets intense enough, it goes beyond the MetaBlob boundary. A HyperTexture will instead respect the boundary and stay with it. Like the toothpaste, it will flatten itself to the shape of the balloon (boundary).

The fractal powering the HyperTexture creates small shapes by micro-displacement. These micro-displacements help create complex shapes for which creating a manual boundary like a HyperTerrain would be near impossible. For example, a typical piece of large sandstone rock has lots of small perturbations on the surface. Modeling it by hand, HyperTerrain fractal displacement, or otherwise is difficult. A HyperTexture can create micro-displacement (as well as large displacement) that a normal displacement material may not be able to.

The core area of a HyperBlob becomes so dense that it is completely solid. The rest of the 3D space between that dense core and the boundary becomes variable in density – changing from one place to the other based on the density fractal used to create the HyperTexture.

While HyperTextures have been possible in Vue for several years, HyperBlobs adds a dynamic new feature that saves a lot of time: baking.

HyperTextures take a long time to render, as well as have free floating pieces inside the boundary where the boundary and density don't harmonize to our needs. HyperBlob technology shaves off these floating "particles" and bakes the HyperBlob into a mesh dynamically during render. Unlike HyperTerrains, you get a real-time preview of the mesh as well.

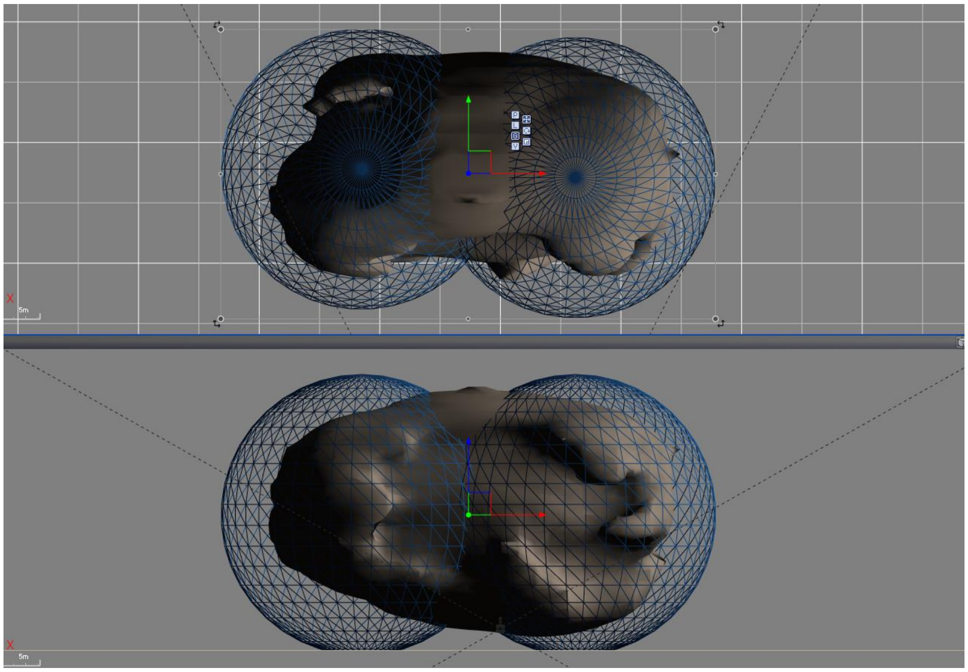
When baked, a HyperBlob renders quite a few times faster than a normal HyperTexture.

Since a HyperBlob is a dynamic baked mesh, unlike a normal HyperTexture object, you can apply a displacement material on it for a far more complex object that possible by any single technique.

Creating HyperBlobs

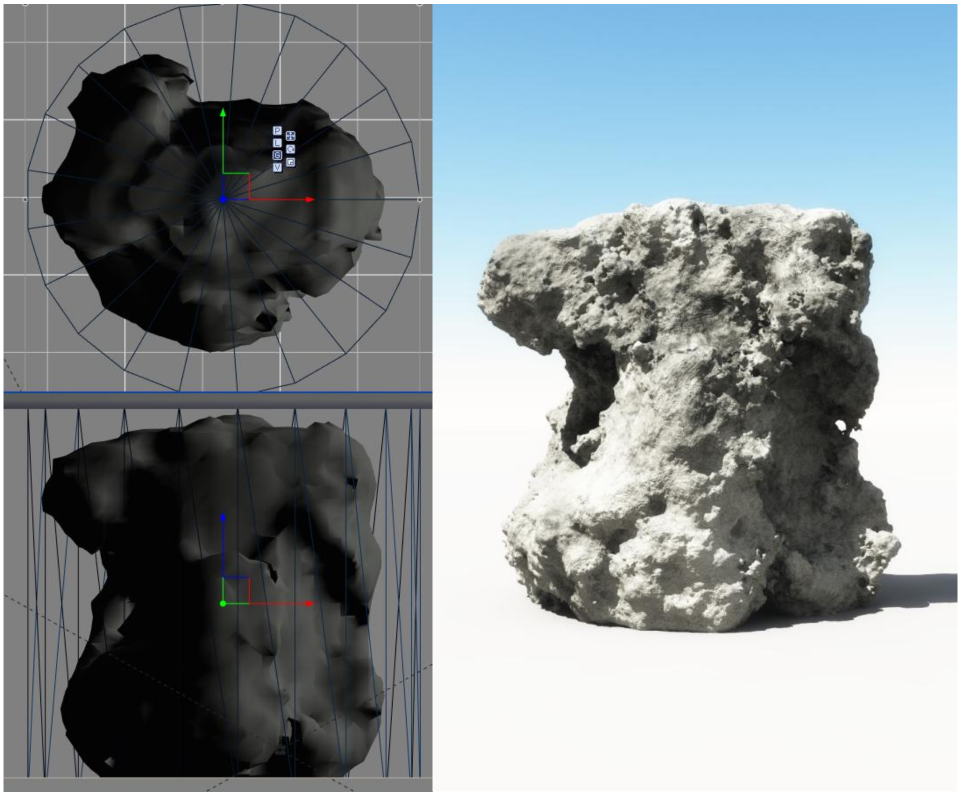
Since HyperBlobs are based on the HyperTerrain technique, they are created in the same manner. First, we create some basic primitive objects that will be converted to MetaBlobs.

When right-clicked, the MetaBlob button alternates as the HyperBlob button.



A HyperBlob, when selected, will show the primitive

A live, viewport preview of the HyperBlob lets you easily manage it in the scene and work with creative composition much faster than with a HyperTerrain.

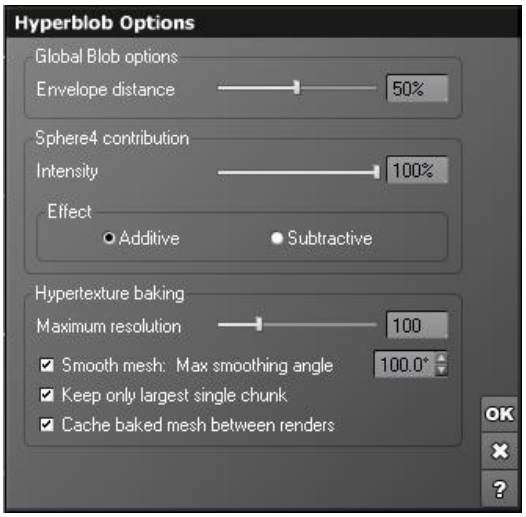


A single cylinder primitive converted to a HyperBlob

Controlling and Baking HyperBlobs

Unlike HyperTerrains, HyperBlobs don't need to be manually baked. Vue intelligently bakes the mesh for you during the render process. Edit the HyperBlob to bring up the HyperBlob options.

Just like a MetaBlob, individual primitives can have an additive or subtractive effect, as well as variable intensities.



HyperTexture Baking controls how the baked HyperBlob is managed.

Maximum Resolution

This is the mesh resolution used to limit the baking density. The higher the resolution, the more quality and detail the HyperBlob will contain. But with complex HyperTextures this can cause slowdowns in your render.

For test renders, the resolution can be temporarily lowered but beware of some shape changes depending on the type of fractals used to create the HyperTexture density.

Smooth Mesh / Max Smoothing Angle

This option allows the mesh to be smoothed. The maximum angle works exactly like the normal object smoothing available in Vue. A higher angle allows for stronger smoothing.

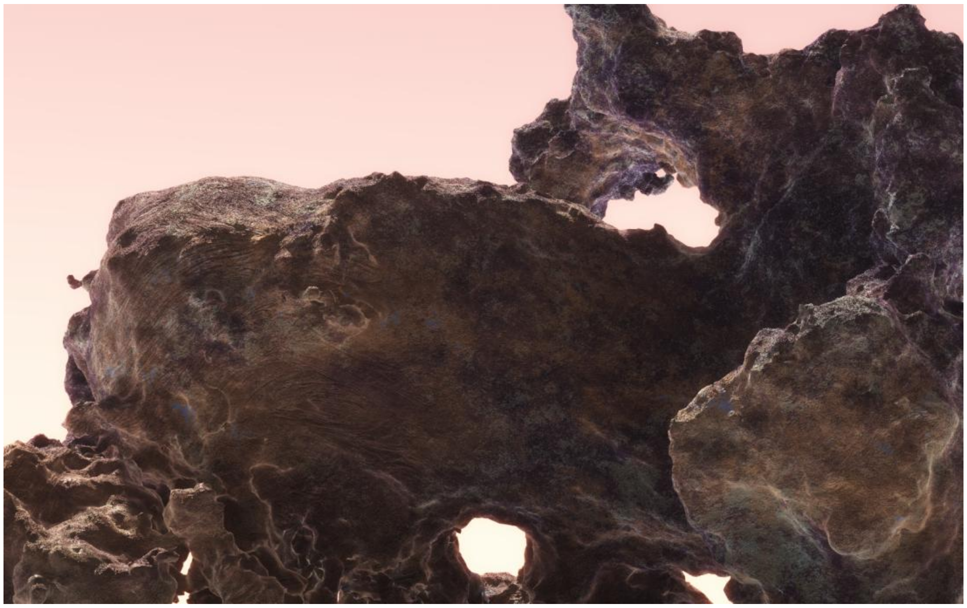
Keep only largest single chunk

Unlike simple HyperTextures, a HyperBlob will shave off 'free floating particles' that are often a side effect of HyperTextures.

Cache baked mesh between renders

A HyperBlob is baked right before rendering. By allowing Vue to cache the baked mesh, the renders start faster. If you face problems updating the mesh, uncheck this option.

Examples of HyperBlobs



Alien surface created with HyperBlobs.

Same HyperBlob with different configuration of the primitives.

HyperBlobs provide a unique procedural modeling method for creating both realistic and fantastic shapes.

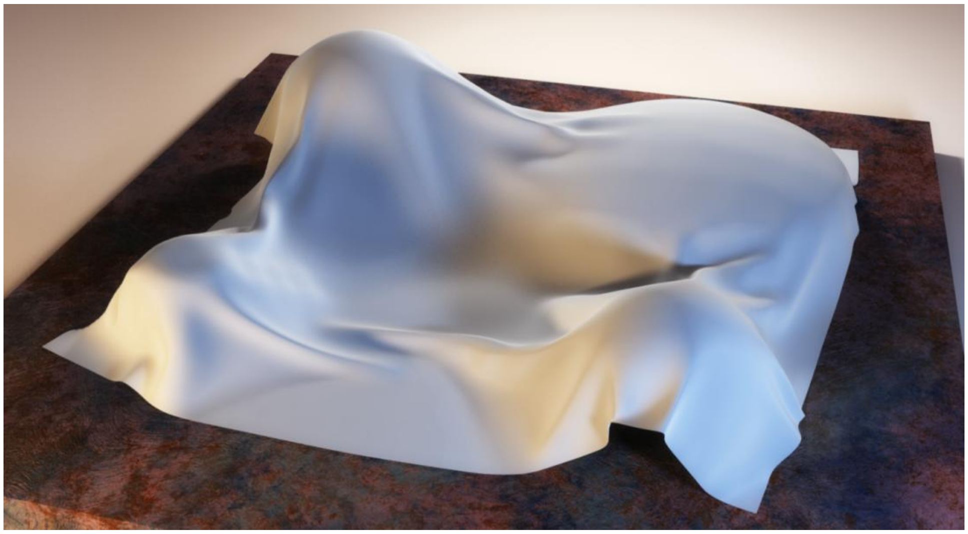
A single cylinder HyperBlob

HyperBlobs have a distinct advantage over HyperTerrains – their detail is much more complex than what simple displacement can offer.

Materials

Chapter 6

The Importance of Materials



Perhaps we shouldn't say 'the importance of materials', we all know materials are important in a 3D scene. Perhaps we should say 'the importance of the *right* materials'. Like bad lighting, a bad material can ruin your scene. The actual shape of any 3D object is altered by the shape *perception* added by the material.

The most important step in a 3D artist's learning is to start creating their own materials from scratch. In other 3D applications, this step comes in early. But for Vue users, who are given a large library of presets, this step often doesn't come in time. This plateaus the skills of the artist and reduces the creativity and uniqueness of the renders.

One reason Vue users often do not create their own materials, choosing to use presets instead, is the fear of the complexity of the Function Editor. The Function Editor is complex, no doubt, but it is not insurmountable. One of the major aspects of this chapter is going to be learning about the SmartGraph Function Editor, how it works, why you should use it, and to experience firsthand how easy it can be.

Creating properly designed, realistic materials adds yet another aspect of realism to your scene. And because these are your own materials, created using your own unique ideas, your scene will always look different from all those others that are using presets.

SmartGraph Paradigm

The SmartGraph Function Editor is the heart of Vue. You simply cannot create sophisticated, serious imagery without touching the Function Editor. However, a lot of Vue users are confused or afraid of the Function Editor, deeming it complex and difficult to master.

In truth, the Function Editor is simple to understand and use once you grasp the basic tenets. The Function Editor depends mainly on a few fundamental elements: color, grayscale values (numbers), filters, input, and output nodes. It is true that the Function Editor is capable of creating extremely complex mathematical functions and processes however our focus is going to be on creating materials and learning about the Function Editor's basics from that exercise.

Contrary to the highly technical outlook the Function Editor might bring out in our minds, it actually depends a lot on imagination rather than technical knowledge. You do need to understand what piece does what but combining those pieces into a comprehensive, usable function requires imagination. You need to be able to visualize something in your head first without working on any fractals.

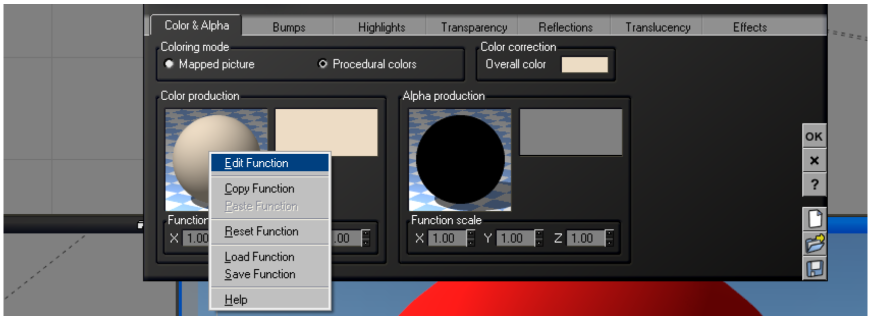
Like any other aspect of Vue, or anything in any field of work, some people exhibit more talent for one specific area than others. So do not let yourself be disheartened if the Function Editor seems too complicated. Going to its deepest levels may require a good grasp of logic and a somewhat warped sense of imagination, but the basics of the Function Editor are approachable, learnable, and quite easily usable for anyone.

In all the years I have used Vue, I have never needed to go beyond the basic to medium range of the Function Editor except for a few instances. That range can still give you massive power to wield in creating your digital worlds.

The primary thing we will do in this chapter is learn about the Function Editor by creating real-world projects like new materials. So let's jump straight into creating our first material.

Creating your first material

Let's start by creating a simple sphere. It will have a default sand-like generic material. Open the Advanced Material Editor. To enter the Function Editor, right-click any spherical thumbnail in the Color, Bump, or Highlight tab and select Edit Function.



Since we are in the default material, you will see this function already laid out in front of you. Select these three nodes – Color, Filter, and Constant - and delete them.

Fractal Node button

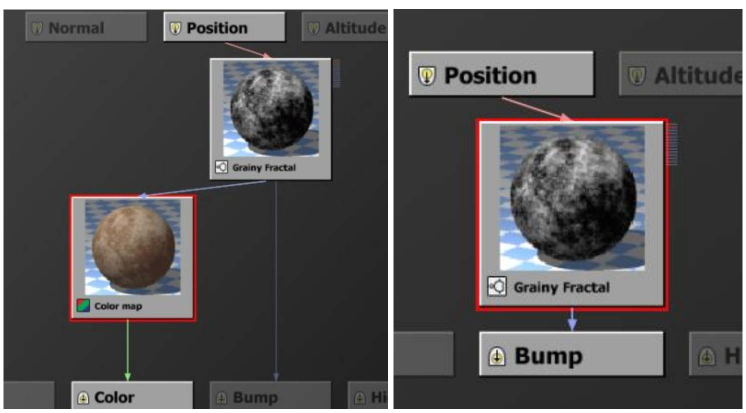
On the left toolbar, you will see the Fractal Node button. Click anywhere in the graph area in the Function Editor and you will see a blank block highlighted with a red border. Click the Add Fractal Node button.

By default this will create a Simple Repeater type fractal. We will explore other fractal types shortly, but to show you how easy it is to create lifelike materials we will select the Grainy Fractal.

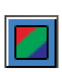
The Grainy Fractal is very closely related to the Terrain Fractal we saw in Chapter 4. Unlike the Terrain Fractal which is built for generated large scale terrain, the Grainy Fractal is designed to create realistic sand/grain/sediment patterns.

Let's change the Origin of our Grainy Fractal to X: 45, Y: 90, Z: 250 to give it a little uniqueness from the default Grainy Fractal. In fact, this is recommended for any fractal.

This should suffice for our basic material. Let's drag the little blue arrow from the Bump node as shown in the image and connect it to the underside the Fractal Node which will be represented by a small blue circle when you drag the line. When you connect the nodes, you will see a popup appear asking which output you want to use. Select Altitude. (We will be looking at the Rough Areas output in a separate section later.)



The shape of the material is set, so now we need to add color.

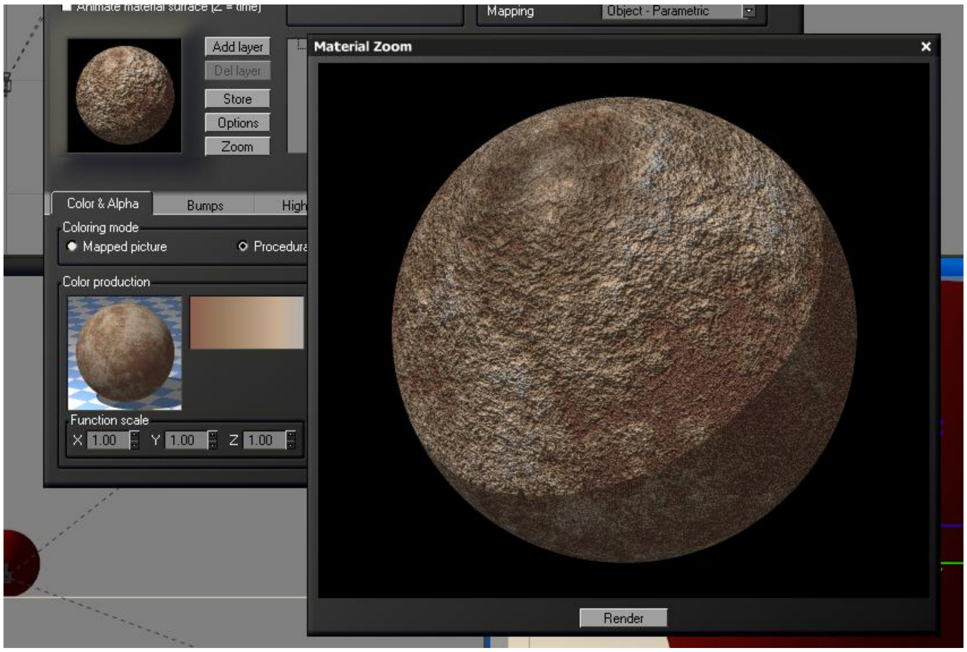


Color Node button

To do this, select a blank space and click the Add Color Node icon. You will now have a new Color Node with a gradient in it. This specific Color Node is called a Color Map. It takes a fractal and distributes the gradient over that fractal pattern. Right-click the gradient and select Load Color Map. Load the Rocks and Grass > Metallic Brown color map preset.

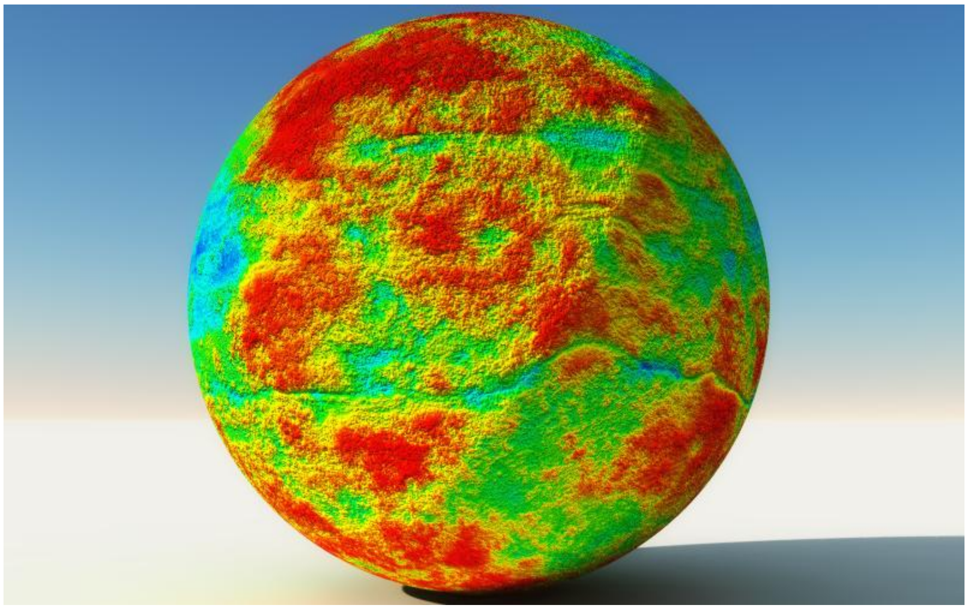
Connect the Output of the Color Map to the Color Output node as shown in the image. When asked to select an option for the output during the connection, select Color Output. Then connect the top input of the Color Map node to the Grainy Fractal. When asked for the Output option, select Altitude.

Let's click OK and go out of the Function Editor. This is what our material looks like. Notice the fine grainy details and curving distortions in the preview.



The Fractal-Color-Bump Interaction

We chose to pass the color map and the bump map through the same fractal. This is because in real life a color change in an object – rock in our case – is often created through a change in the shape (and therefore a change in the intrinsic material). So when we pass the color and the bump through the same fractal our object takes on color changes in synchronicity with the shape changes in the bump.

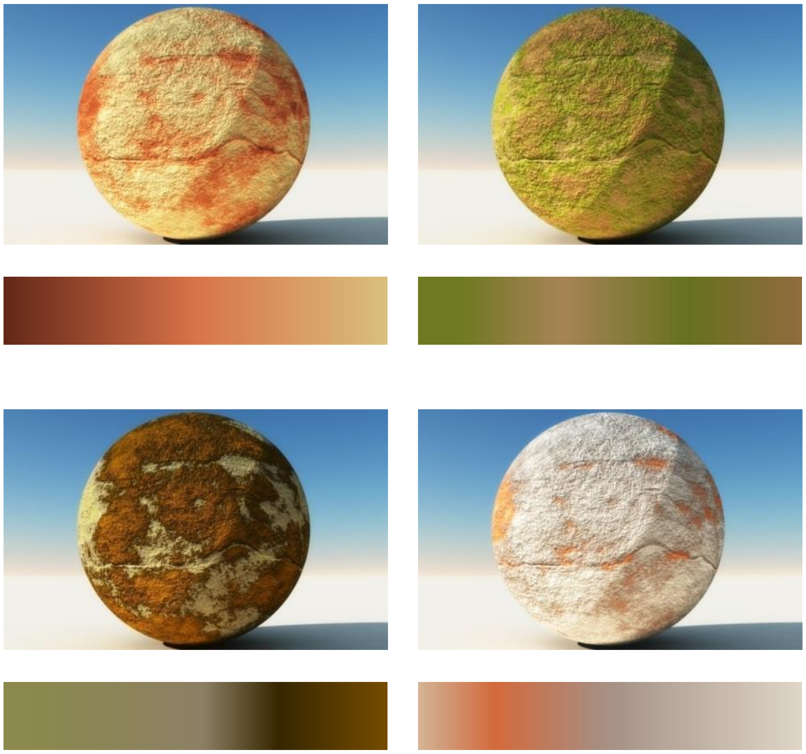


Full spectrum color distribution over a sphere

In technical terms, the grayscale output created by the fractal is made up of numbers – 0.0 representing the black (lowest altitude), and 1.0 representing the white (highest altitude), and the infinite number of grays by the infinite decimals between the two. The gradient map is then divided into bits with the left side representing 0.0 and the right side 1.0, with the colors in between representing the infinite decimals in between. Finally, each color point from the color map is applied to the object using the grayscale fractal output as a guide by painting each altitude decimal level with a corresponding color from the gradient. The image above shows a rainbow gradient with blue representing black and red representing white.

Color Changes

If we were to swap out our color map with other maps, this is the kind of changes we could see. You can see how the color maps are distributed over the fractals.

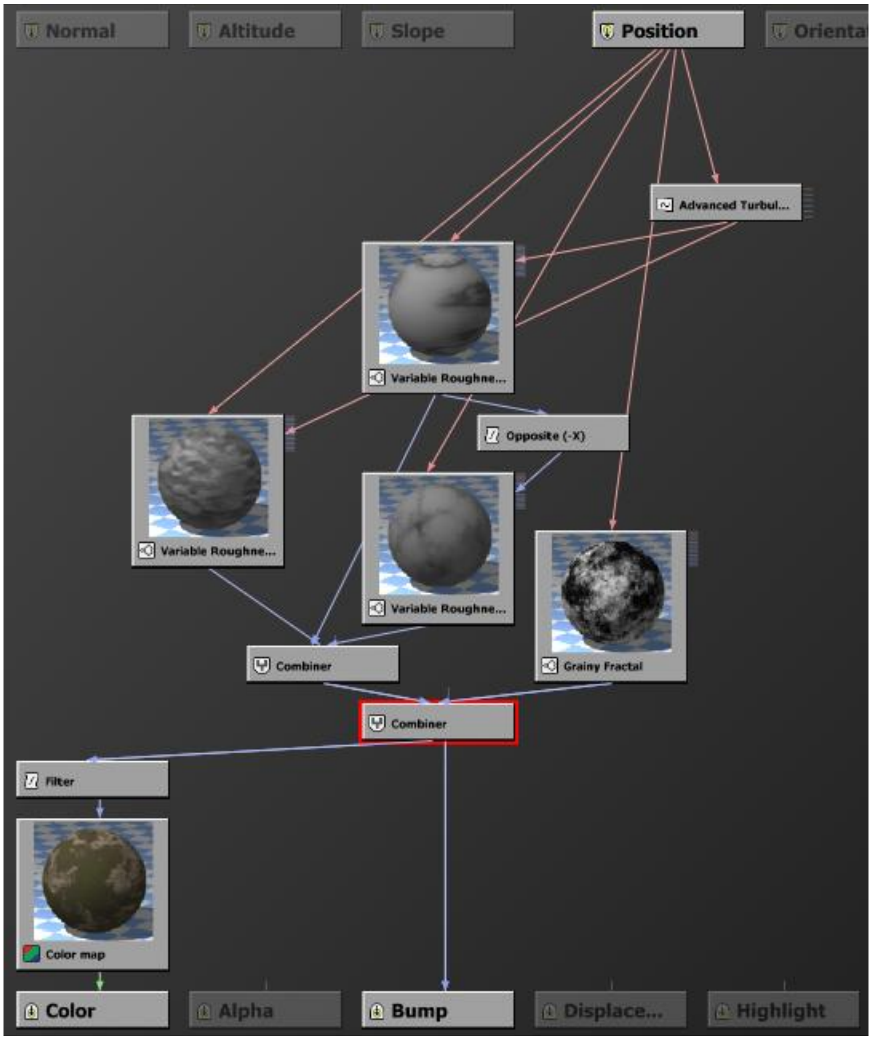


Different color maps creating different shapes on the same fractal.

Working with different color maps is one of the easiest and best ways to create variations of a single material – whether you change the entire map or tweak the colors. It is always a good idea to save your custom color maps as you can often use them

Creating a complex material

Now that you have seen the basics of the Function Editor, let's dive a little deeper by understanding a more complex method of creating materials by using filters, combining different fractals and using variable properties.



In the Function Graph above, you will see five node types: a Color Map, four Fractals, a Turbulence node, two Combiner nodes, and two filters.

Combiners are a simple node that takes two (or more) inputs and combine them using one of several modes such as Blend, Add, Subtract, Max, Min, etc. We saw one use of this in the Terrains and Rock Formations chapter when we combined two different terrain fractals.

Filters are, in their simplest form, an X-Y axis graph that helps define the bias of the input and applies it to the output. Sounds extremely boring and complicated, but in essence it helps fine tune which parts of a color map or fractal gets more emphasis and which parts get less. We will explore these in detail with practical examples later. In their more complex forms, Filters can create massive changes in a color map or fractal.

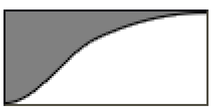
A Turbulence node is made for creating fractal based turbulence. It uses a fractal noise to create a turbulence map that can be used instead of a normal fractal node. However, the detail level is very low so it is not advisable for such usage. We will use it to help modify our fractal nodes.

The purpose of our material is to show a rocky cliff surface with moss-like vegetation growing on it. So let's deconstruct our complex material.

The Color Production

This is perhaps the simplest part of our material. A color map made up of different brown tones (mostly light colors) is mixed with a bit of green. Once the fractal was set up, the colors were readjusted to make the peaking bits of the fractal turn green representing our mossy vegetation. The rest of the colors create the rock.

A filter is added to the input coming from the combined fractals with this curve.



A custom filter curve: Filters are the easiest method to tweak an existing node.

This biases the gradient to use some values more than others.

The material without (left) and with (right) the filter applied.

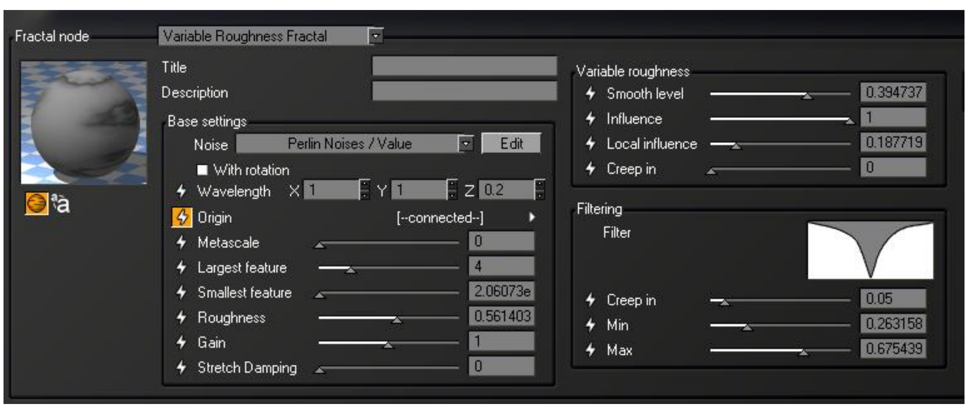
The Fractal Production

This is the heart of our material. Notice in the function graph that the Turbulence node feeds values to the side of the two variable roughness fractals rather than the top or bottom as we saw in the other functions before. This is because many values in fractal (or other) nodes can be variable, that is it can be a multi-value fed from another node rather than just a simple constant value (a number, in most cases).

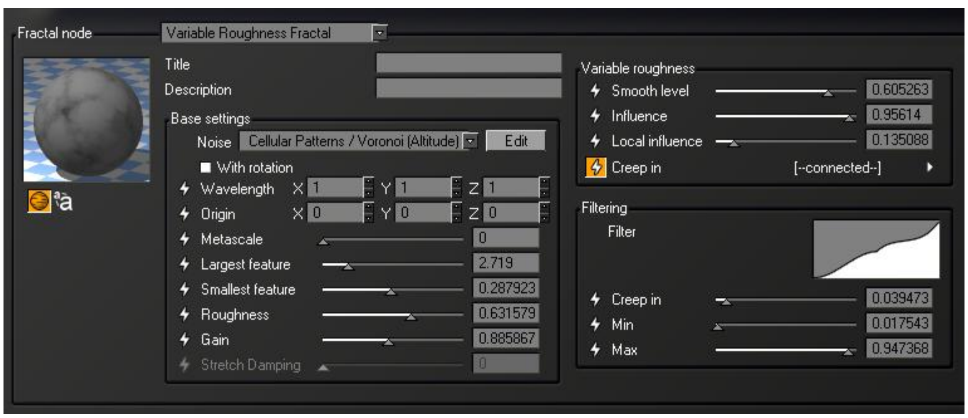
If you try to connect a non-compatible value, the “receiving circle” on the target node becomes gray. This means you cannot connect the two.

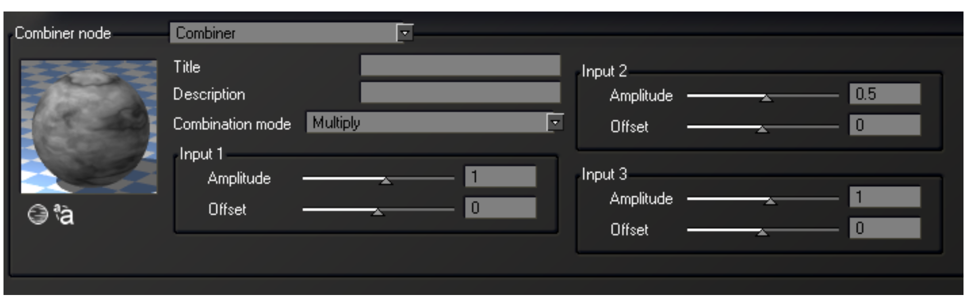
However, for us the Turbulence node creates a basic turbulence which we connect to the Origin of our Variable Roughness Fractals. The effect can be clearly seen in the image below where the left images show the fractal without the Turbulence node connection, and the right image with the connection.

Fractal 1 and 2, through the Advanced Turbulence node provide a horizontal and slightly curving set of lines that create a sedimentary feel.



A “Opposite (-X)” filter is added to the output of Fractal 1 – which acts as an Invert function would on a bitmap – and passed on to the Creep In smoothness value of our Voronoi-based Fractal #3, while a combiner node mixes the first two fractals along with this third one. The reason for passing the inverted fractal output to another fractal and then mixing that fractal with the original is to give it a sort of familiarity in the shape output while still making the third fractal a bit more unique by passing on values from a fractal rather than standard numbers.

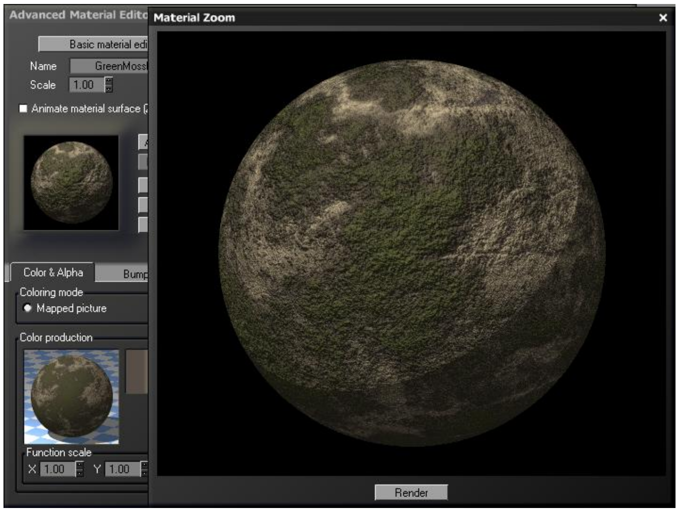




The fourth fractal, a Grainy Fractal, serves as the “natural noise” node. Instead of using a standard noise node which creates a limited shape, this node gives us a more variable and natural looking noise. The output from our other combiner and the Grainy Fractal is combined using a new combiner that uses the “Multiply” mode.

This fractal output is passed on the Bump channel as well as the Color Map.

Now let's take a look at the output of our function graph in the form of the complete material, as well as the material used in a scene.



Natural Noise

Traditionally, noise is just noise – a random sprinkling of dots to add a grainy feel to a material. But nature often employs a different type of noise; a noise where large level changes occur in the consistency of the dots.

Forget the shape of the rock in this picture but “thinking fractally” look at the color changes that would be a noise-producing node in a fractal. It would have a large range of inconsistency in the noise. The lower right, and the far left areas have darker patches of dots which occur because of the change in the physical shape of the rock not the basic material (sandstone) from which it is made.

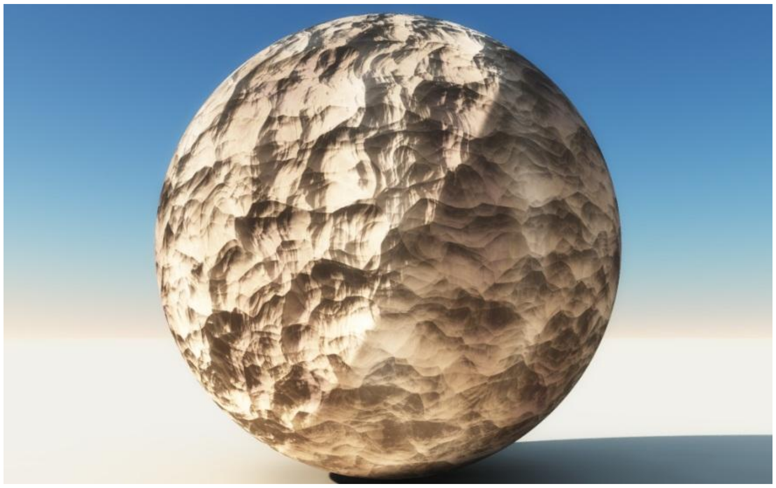
This is why we use a Grainy Fractal node instead of a noise node for creating such visual noise. The noise node can be instead used for providing connected values (like we did for our Origin values in the previous section) since they are less memory hungry than a Grainy Fractal.

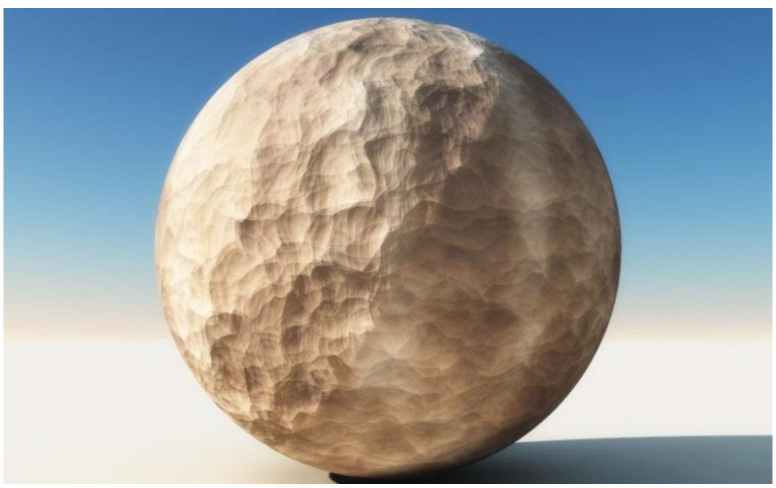
Do not avoid noise

Often when you render your scene using a high-noise material you will feel the bump/noise is overwhelming. Instead of removing or lowering the noise, one thing you should try is rendering the scene at Broadcast or higher quality, or by increasing the Texture Antialiasing in the Render Settings (see Appendix A). The real world has a LOT of noise, and so to keep your renders real, you must have good amounts of noise in your scene too. They can be softened to make your renders look cleaner, but it is important for the sake of realism to keep the noise in the image.

Scaling Bump Issues

Sometimes if you increase the Scale of the entire material for the fractal shapes to grow larger, the bumps start looking bad. Let's define bad first: if your bump is strong enough but the shape of the object (a sphere in this case) does not show that shape on its edges, then the bump is bad.



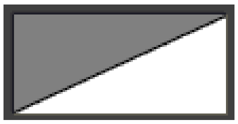


If this starts happening, try lowering your bump value in large scale factors. For example, if you changed the Scale from 1.00 to 20.00, lower your bump from 1.00 to 0.200. For a Scale change from 1.00 to 50.00, use 0.050 for the bump, and so on. There is no fixed formula for this. Try different values until you find the right effect.

Filters

Filters come in a variety of forms. The most basic form we see is the graph-based filter which is available in almost all tabs of the Advanced Material Editor. The other types of filters include Smooth Clip, Gaussian, Bias, etc. We covered the Recursive Strata Filters in Chapter 4.

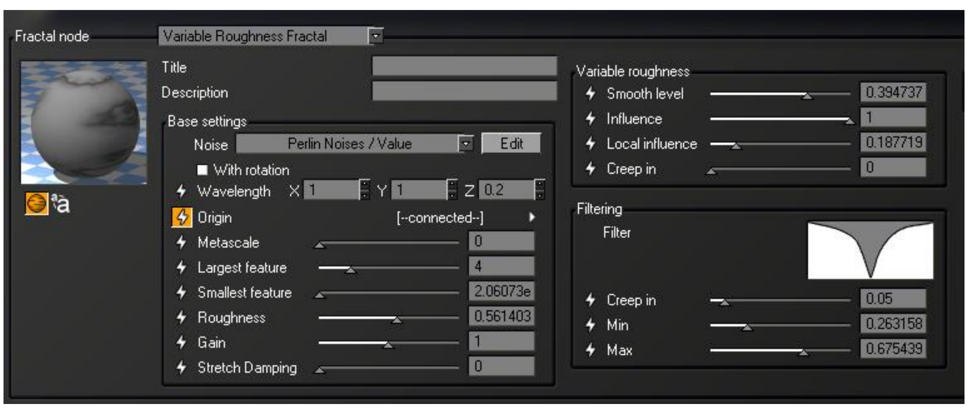
To understand how filters work, look at our Fractal-Color-Bump interaction.



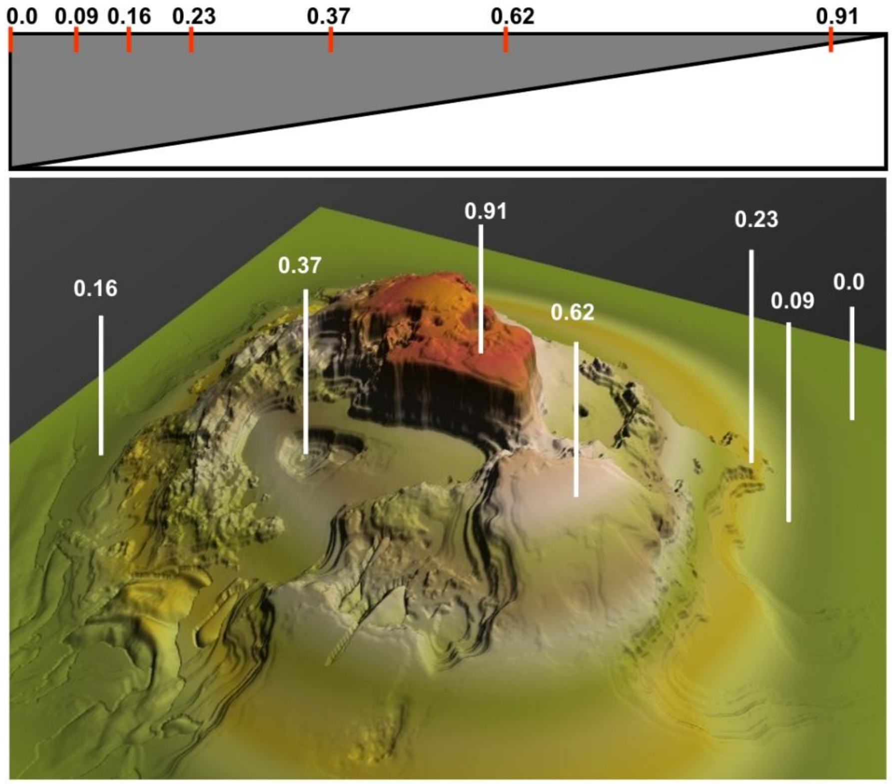

We discussed in that example that a fractal output of altitude is basically a set of values ranging from 0.0 to 1.0, including all the infinite decimals in between. This output is visibly shown as black for 0.0 and white for 1.0, with infinite grays showing the decimals in between.

In the default filter graph shown above (left), the left most side represents 0.0 and the rightmost side represents 1.0. The systematic downward slope represents the decimal values between the two.

If we changed the filter as shown above (right), values between 0.0 and 0.5 would be normal, so would the values between 0.9 and 1.0, but the range between 0.5 and 0.9 would have a downward curve which means the “bright grays” in the altitude become a little darker, resulting in a lower altitude.



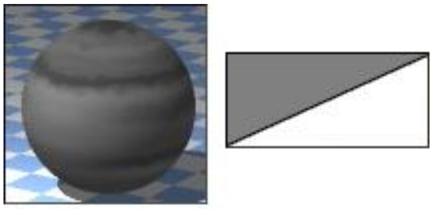
Take a look at this fractal that creates wide Perlin noise. But the preview shows only strong grooves instead of large terrain like features that we normally see in such a fractal. Because the 0.0 to 0.4 and 0.6 to 1.0 ranges are very high, only the central values are processed giving us the sedimentary groove shapes needed for our material.



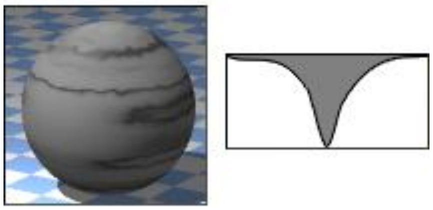
In this image above, you can see which portion of the altitude of the terrain is represented by which part of the filter. The same applies to any fractal or Color map (Refer back to the Fractal-Color-Bump section for information on how a fractal interacts with a Color map) with a 0.0 to 1.0 distribution of values across every type of node.

Filter Examples

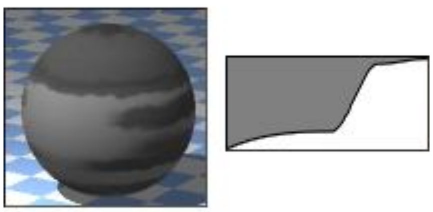
Taking our fractal and filter from the example we just saw, let's see how different filters change our fractal output.



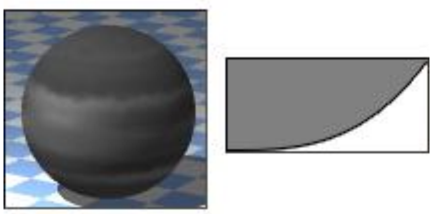
In Filter 1, the default filter, the output is untouched.



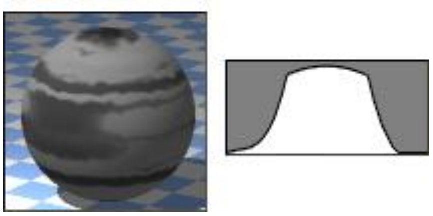
In Filter 2, we create our sedimentary grooves by using only the middle range values.



In Filter 3, we create a mesa effect by flattening the low-range values, creating a sharp jump in the values right after the mid-range, and flattening the high-range values.



In Filter 4, we create a very smooth eased curve where low and middle range values are lowered and the high range values get a gentle boost with still lower than normal values when compared to the default.



In Filter 5, we create a strong middle range mesa with sharp drops in values outside of the middle range that create sharp hardness in the fractal shape output.

Filters are useful because they let you adjust (or literally filter) values provided by a fractal without having to change the fractal itself. For example, in a terrain if the lowest areas have too much noise, then using a filter like the one in example #4 above would smooth them out.

You are never limited to one filter either. You can use multiple filters as we will soon see in our materials in the following section. The more creatively you think, the more powerful your filters' usage can be.

Fractals

The complete list of fractals available in Vue, and their potential uses, is so large that we would require a separate book for that. However, we will touch upon some of the more useful ones. Except for the first two samples, the renders shown use a Variable Roughness fractal.

Grainy Fractal

This is one of the most useful fractals. At its most basic usage, it can create a varied range of noise and grit for your rock materials. At a higher usage level it can create large terrain-like noise without using too much memory.

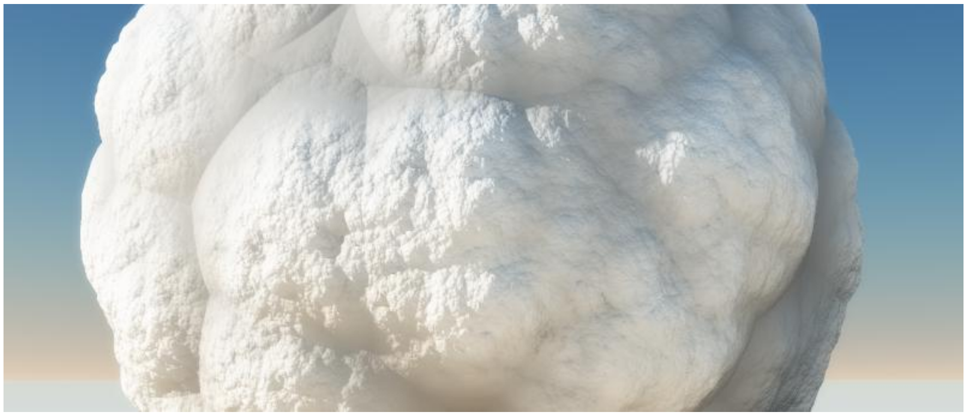
With a higher Smallest Feature value, you can tune out the excess noise and create some very interesting realistic rock-based shapes.

Terrain Fractal

This is the ‘big brother’ of the Grainy Fractal. It takes the rock-like realistic shapes and creates them on a larger scale.

Just like the Grainy fractal, the Terrain Fractal features a distortion effect that can be smoothly controlled to give you large sweeping shapes that are realistic enough to use on terrains and rocks, and yet organic enough when given a higher value that they can be used on other surfaces too.

Voronoi



One of the most commonly visible fractals in nature – especially in rocks – is the Voronoi fractal. In a “hard” form it looks like multi-faceted shapes joined at the sides. With the softness available in this fractal in Vue, they can help you create the perfect rocks and even rock cracks.

This fractal can create small rock fields or large rocks with massive cracked surface features depending on the size and roughness used.

Round Samples

By itself this fractal doesn’t look too impressive, but when combined with a rough fractal like Voronoi or Grainy, it can create very realistic shapes.

It is always a good idea to use filters or high roughness to adjust the shapes of this fractal so they do not look too geometric.

When overlaid on another fractal like Voronoi, it adds different yet permeable shapes that result in a unique, life like pattern.

Clumps

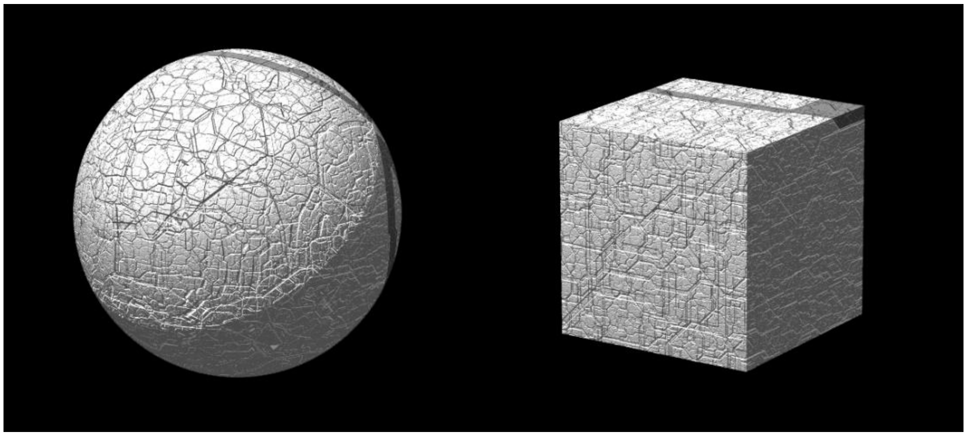
Another fractal that by itself may look too geometrical, but when combined with a rough fractal, it can create realistic wide range cracks and surface variations all along the material.

It is advisable to use a Combiner or Blender node to mix this with another fractal so this fractal appears only in some parts of the entire surface rather than all over the surface as that may look too systematic.

Cracks

A relative of the Clumps fractal, the Cracks fractal has stronger crack-like features and can be used to create strong gaps and lined cracks in stone materials.

Just like the Clumps fractal, it should be used sparingly by combining or blending it with another fractal to avoid geometric patterns.



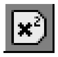
When creating your materials, take the extra time to render it on more than one shape. If you are relying on the Material Editor's preview, try changing the Options to Cube, Cylinder, etc. instead of Sphere. This will help you spot irregularities in the UV mapping the fractal is producing.

UV or UVW mapping is a mathematical coordinate mapping technique. While UV mapping is associated more with other 3D programs than Vue, it does happen in the internal calculations of Vue render engine. Essentially, it is how an "unwrapped" map (whether bitmap or fractal output) is wrapped around an object.

One of the easiest ways to do this is to create an empty scene and add a HyperTerrain to it. Use the HyperTerrain as a testing object for the material. By their very nature, HyperTerrains produce complicated UV output. So if the material looks good on a HyperTerrain you can be sure that it will work well in most other places.

Math Nodes

A large array of simple and complex operations in the Function Editor is done via the Math nodes.

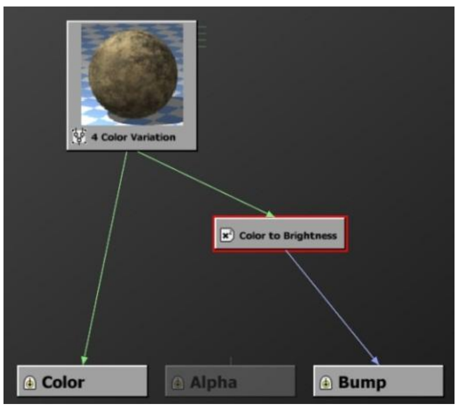


The Math Node icon.

Math Operations

You can use the Math Nodes to do mathematical calculations and operations such as Arc Cosine, Sine, Floor, Multiply, etc. You can also invert values with the Invert node. In practical terms, the Invert Node would have the same effect as an Opposite (-X) Filter.

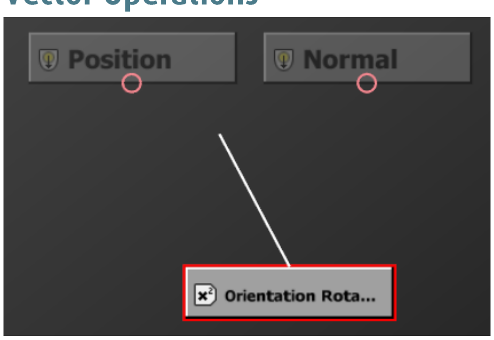
Conversions



One of the most useful Math nodes is the conversion nodes. For example, if you have a 4 Color Variation MetaNode creating color, and you want the same pattern for your bump you would need to use a Color to Brightness conversion math node to convert the color output (green) to altitude or brightness (blue) to pass it on to the Bump channel.

Other conversion operations include RGB and HLS (Hue-Luminosity-Saturation), as well as Vector to Color. The latter is useful for taking input such as Slope, Height, etc. and performing color changing operations based on those values.

Vector Operations

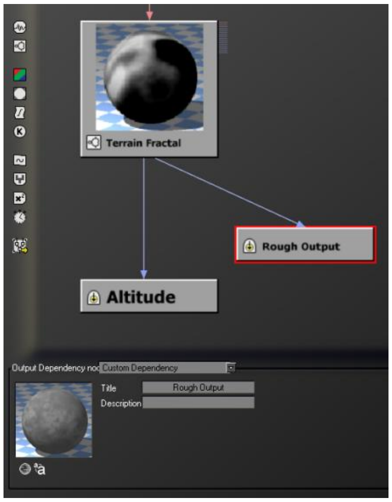


The more seemingly complex category of the Math nodes is Vector Operations. This category contains nodes that allow you to take vector input such as Position or Normal, and perform various operations on it. For example, if you want an additional 10 degrees on a certain axis more than being provided by a certain node or other input from the scene, then you can take the Orientation Rotation math vector node and offset the additional degrees needed on whichever axis you would like to. The output of these nodes is Vector so you can perform multiple operations if needed.

These nodes require an in-depth analysis with detailed examples to be understood. Unfortunately, this falls out of the scope of this particular book.

External Dependencies

One of the advanced features of the Function Editor is to allow the output of a node in one function to be access outside that function by another function that is completely separate from the first function. This is known as an External Dependency.



Graph showing External Dependency from a Terrain Fractal.

Let's follow a simple example to understand how it works.

In this image, you can see the function that creates a Procedural Terrain. The Terrain Fractal that creates the terrain has two types of outputs – "altitude" and "rough areas". We are going to take the second output, Rough Areas, and use it in a mixed material to distributed two materials.

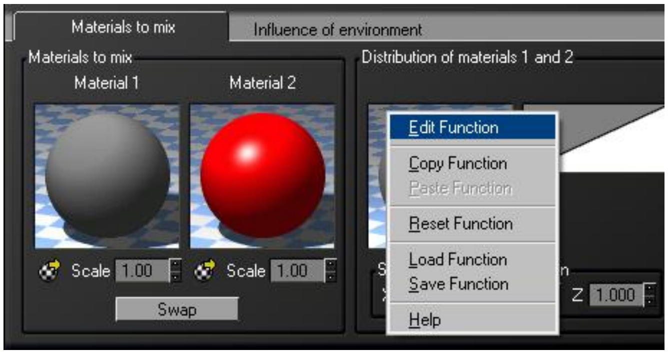
We create an External Dependency node by clicking the Output Node icon.

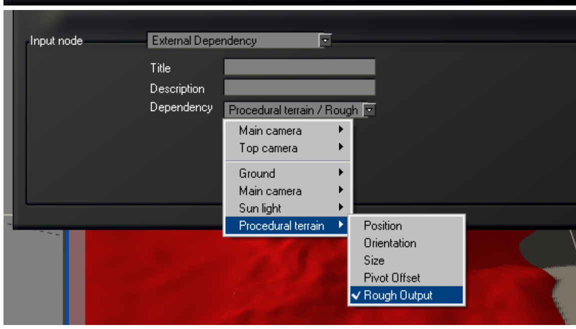
Once you have the node, connect its input to the Terrain Fractal output. Select Rough Areas when asked to select an output.

The External Dependency node can be named for clarity. We will call it Rough Output.

Click OK and close the Function Editor.

We will use a Mixed Material for the Terrain which has a flat gray material mixed with the Basic Red material so we can see the exact areas affected by our Dependency Property.

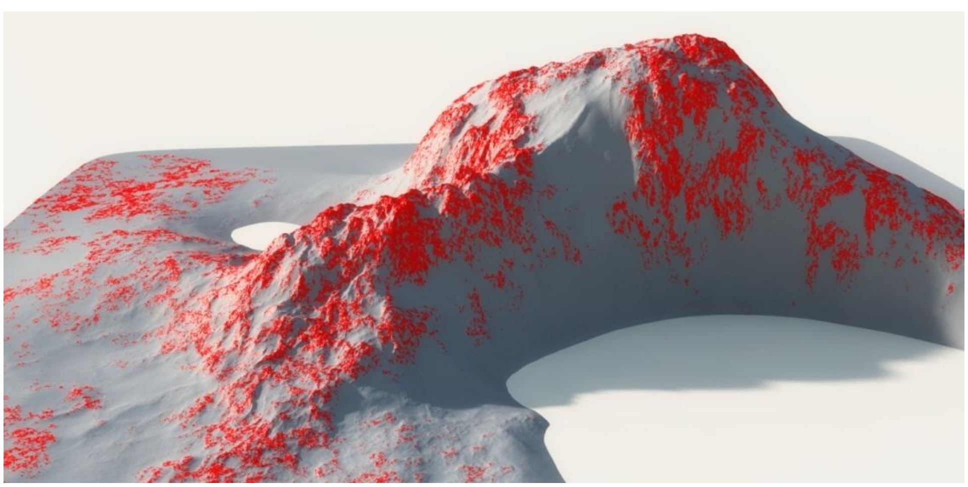




The Input Node icon.

Now, we go into the Distribution Function and create an Input Node called External Dependency. In the node, you can select the External Dependency as shown in the image. We connect the output of our Dependency node to the Distribution input.

The clear effect of the distribution can be seen in this completed render.



Similarly, you can use External Dependencies to use any numeric value, fractal output, or any values provided by a node in the graph in one object and use it in another object's graph.

Variations

Applying the same material to multiple objects – especially similar or duplicated objects – can be a serious detriment to the realism of your scene. To avoid a tiling or patterning effect, try creating material variations.

This is easier than it seems. Like the Origin property of a Fractal node, all Vue materials themselves have an Origin property. It can be found in the Effects tab of the Advanced Material Editor in the form of three numeric inputs for X, Y, and Z axes.

By changing these values, you essentially offset the surface map of the material by these new coordinates. Since Vue fractals create infinitely large maps, you can enter any random (or planned) values in these three input fields and your material will look different on each object.

The image here uses the same material on four different spheres, but unique Origin coordinates are entered to create completely different material features.

Common Material Scenarios

Practical application is perhaps one of the best learning resources. To understand the Function Editor and the process of building both simple and complex materials, we will explore some common material scenarios and explore their finer points in detail.

It may sound like a joke, but excelling in the Function Editor often requires you to warp the way you think. One Vue artist, an editor on this book in fact, commented that “You cannot just think outside-the-box. You have to burn the box and throw it away, and never, ever think inside the box ever again.”

Nothing in the Function Editor is set by any rules. The only rules you have to follow is that you cannot force a vector input into a bitmap, an RGB input into an altitude output, and so on. Other than that you are only limited by your imagination.

In the following examples, you will see some nodes or settings that may make you wonder how they were thought up. Some were created accidentally which led to understanding a new way of doing that particular thing, others were found by experimentation, others still by trial and error.

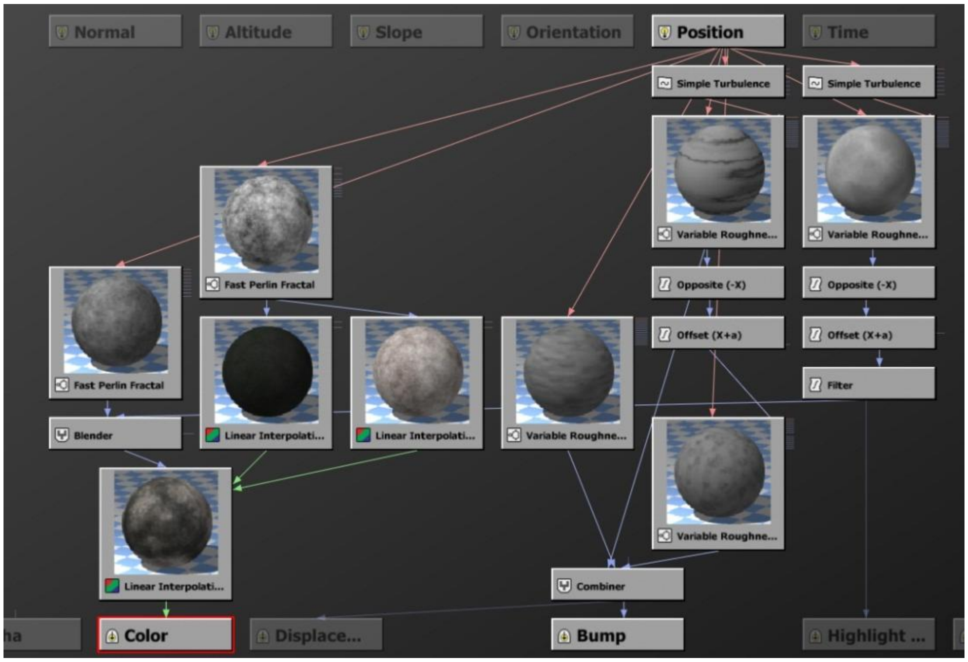
As with everything else about Vue, experimentation is the key to greater knowledge.

Soft Rocks

While seemingly complex, this is a fairly straightforward material.

The color and bump shapes are mostly different except for some slight blending using the Blender node to combine the color producing Fast Perlin Fractal and the Variable Roughness Fractal that creates the overall splotches for the Highlight. The Bump producing Fractal is passed through two filters – the Opposite (-X) filter which inverts the fractal output so it creates grooves or carved areas instead of raised areas, and an Offset (X + a) filter that deepens the intensity of the fractal output. The same process is repeated for the “splotch” fractal for the Highlight and Color, with the addition of a filter that adds contrast to the output. These fractals use a Turbulence node for their Origin to give it uniqueness.

The Bump fractals are combined and passed on to the Bump channel.



A Fast Perlin Fractal powers two Linear Interpolation color nodes providing ranges of dark grays and light browns to the final Linear Interpolation color node as Color 1 and Color 2 inputs. When you can define only a single color in such a Color Node, you can use other Linear Interpolation nodes to combine two colors and pass the range of that color into the final node, giving you a larger array of color variation than the two individual colors you would normally be able to specify.

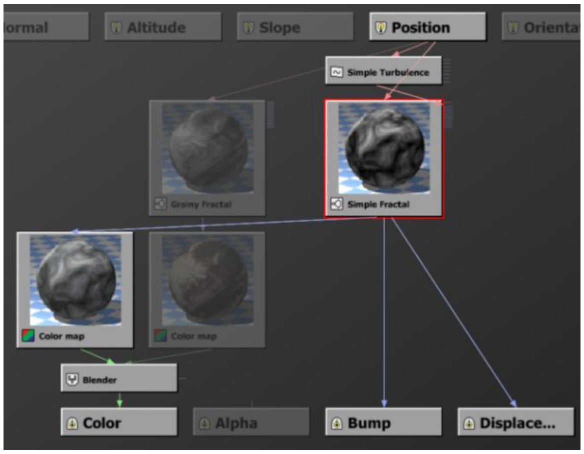
The final color node also uses the blended Fast Perlin and Variable Roughness discussed in the second paragraph above as the fractal input to distribute the interpolated color values over.

The resulting material is a soft, time-eroded rock suitable for cliff faces, sea shore rocks, or even river banks.

Hard Rocks

This is a deceptively simple material. The “Transformed Cristals” fractal is loaded in the Bump channel and two different Color Maps are blended into the Color channel.

One Color Map uses the Transformed Cristals fractal input, and the other uses a highly distorted Grainy Fractal. Displacement is turned on. The combination of these maps is more powerful than using just a single one as it is funneled through a fractal.

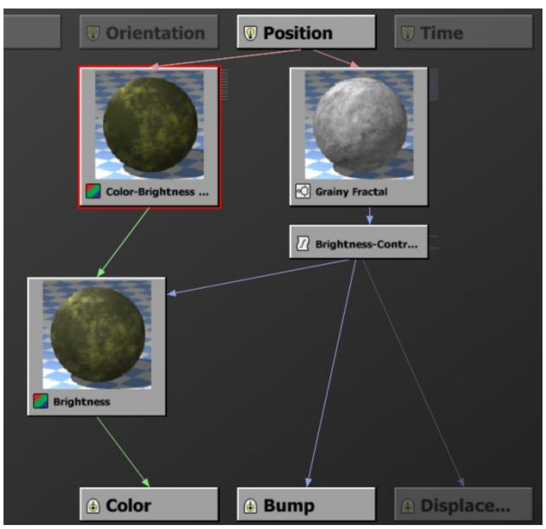


Two different methods of fractal color distribution.

Half of the color is the same pattern as the bump so it provides familiarity in the material shape, yet the other half sweep colors over the bump creating separate patterning which is more lifelike.

Distant Grass

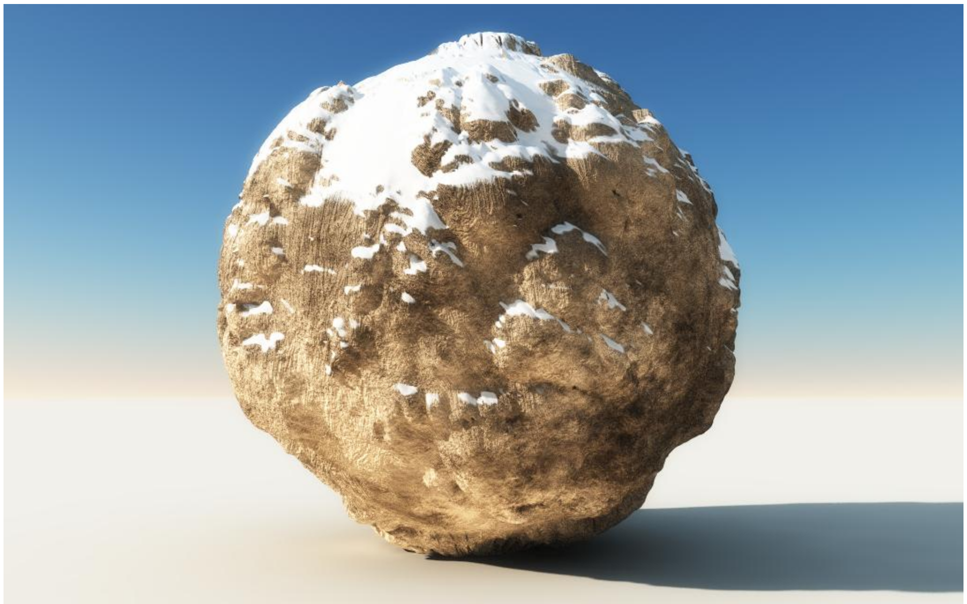
This is an extremely simple distant grasslands material. Its function uses the Color-Brightness Variation color node to use a single color to create varied color patches in the same family. A grainy fractal with a filter to control Brightness and Contrast is used to create the Bump.



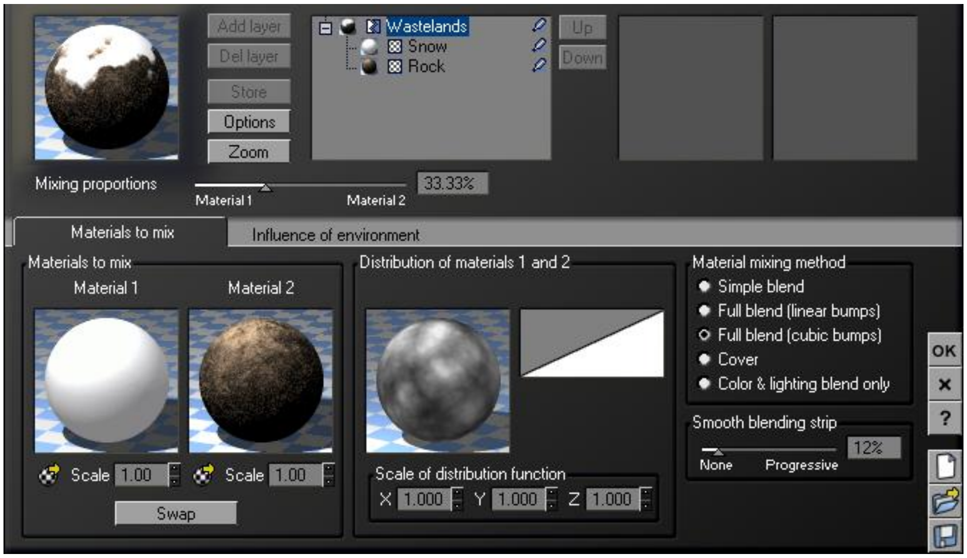
The bump fractal affects the coloration of the color node.

The output given to the Bump channel is also given to the Brightness correction Color node to adjust the output of the Color Map to match the Bump to an extent.

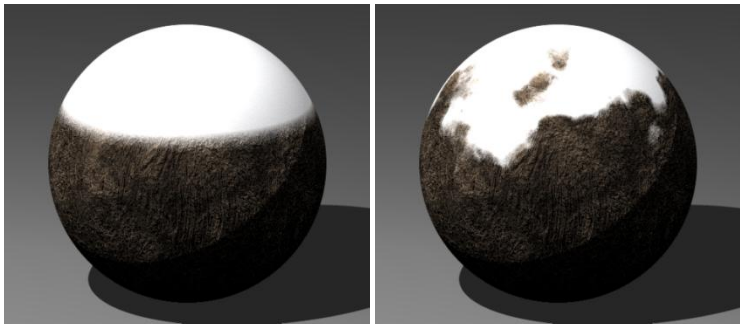
Rocks and Snow



In this snow covered rock material, there are three key things: the distribution, the environmental influence, and the mixing method. The snow is the simple Snow Layer found in the Vue Layers materials, the rock is a QuadSpinner Mineral Infinity material.



The Distribution is based on a fractal. Personally, I always prefer to use a Grainy or Terrain Fractal, but for distant objects a simple fractal would do just as well. In this case, we are using the Grainy Blobs Fractal Preset. The difference of using a fractal versus not using one (which some Vue users often do) can be seen clearly in the images comparing the two methods below.

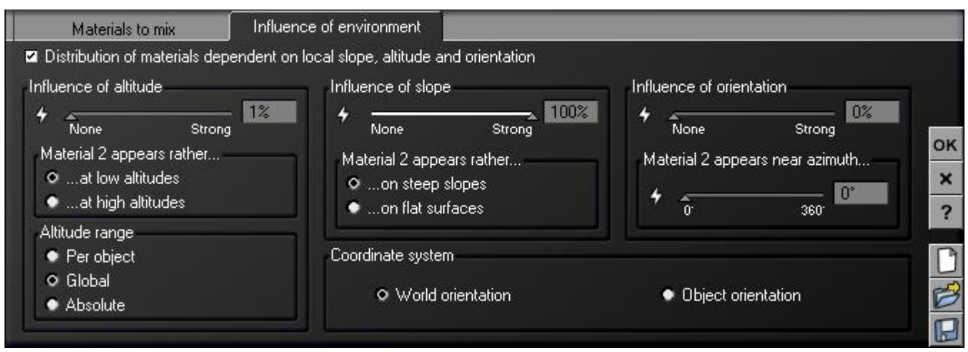


Empty Distribution Function (left) versus a Fractal Distribution Function loaded on a mixed material.

By using a distribution function, you get a little unevenness that looks more natural than a flat line where the distribution parts. This becomes more important when you have a gentle sloping terrain without many crags. Experiment with different fractals (or even Fractal presets) to get the best look for your scene.

The Material Mixing Method of the Mixed Material is set to Full Blend (Cubic Bumps). This instructs Vue to combine the bumps of both materials. In the right image in the comparison above you can see the rock bump edges bleeding into the snow layer.

Finally, below you can see the Influence of Environment settings used to have our snow stay on flat surfaces while steep slopes expose the bare rock.



Displaced Rocks

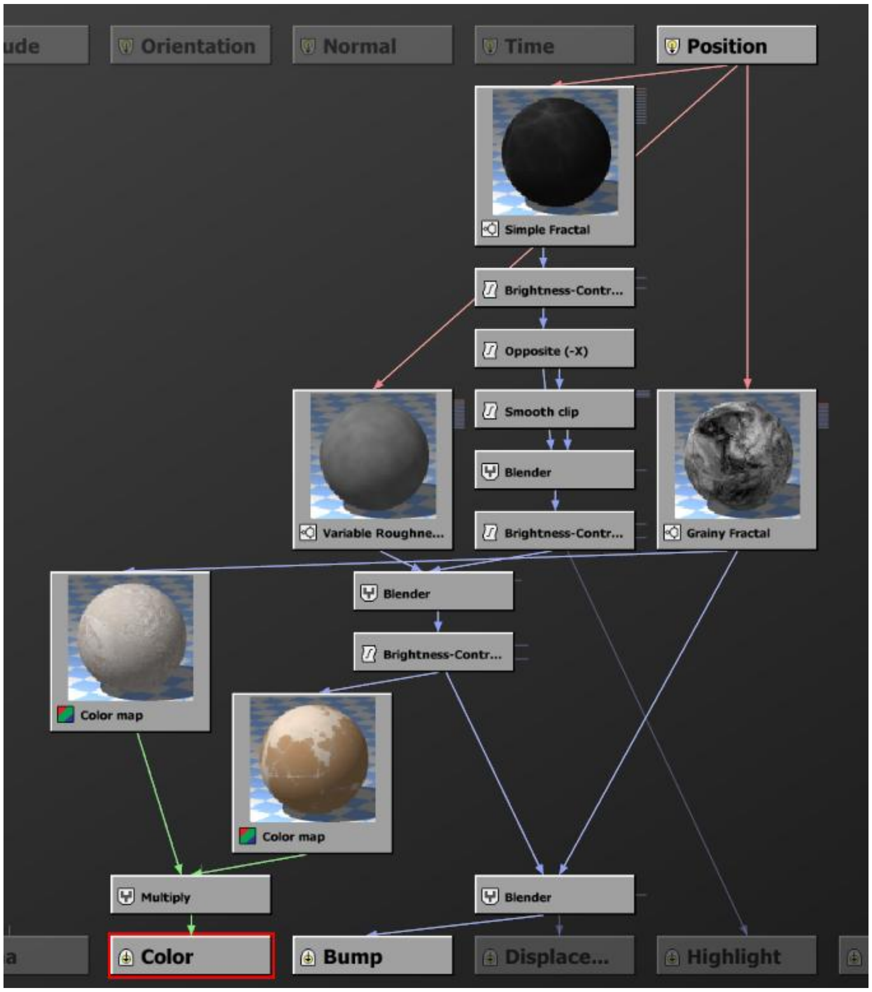
The amount of detail on this material creates an image of an extremely complex function in our minds. But that is far from true. The power of realism is often found in simplicity. The function editor does not really require complex graphs, but rather complex thinking to create good results. And this material is a study in simplicity.

This material is powered by three fractals and two color nodes.

The first fractal is a Simple Fractal using the Voronoi pattern. This creates the “rocks” in our material. The second fractal is a highly distorted Grainy Fractal which creates the “sand” in the material. The third fractal is a Variable Roughness Fractal using the Perlin pattern that creates the flaking/crusting effect in the sand areas giving it a sandstone effect.

The Voronoi fractal is passed through several filters to remove the excess lower areas of the output and creating only the “stone” shapes we need. Mainly, brightness-contrast control is used for this. This fractal is combined with the Perlin fractal using a Blender with the Max blending mode. Another Brightness-Contrast filter stabilizes the output.

This output is passed on to the Color Map that uses a soft cream color and a medium brown that creates the rock-sand colors for the overall material. The output is also passed on to another Blender node which uses the Max blending mode as well. The Blender mixes this output with the Grainy Fractal with a bias towards the Voronoi than the Grainy fractal. The combined output is passed on to both the Bump and Displacement outputs.



A separate color map with soft sandy colors has the Grainy Filter passed as the input. This creates the soft swishes in the sand. The two color maps are mixed using a "Multiply" combiner.

The result is a rocky material with big rock clusters protruding from flat stone and sand.

Sand

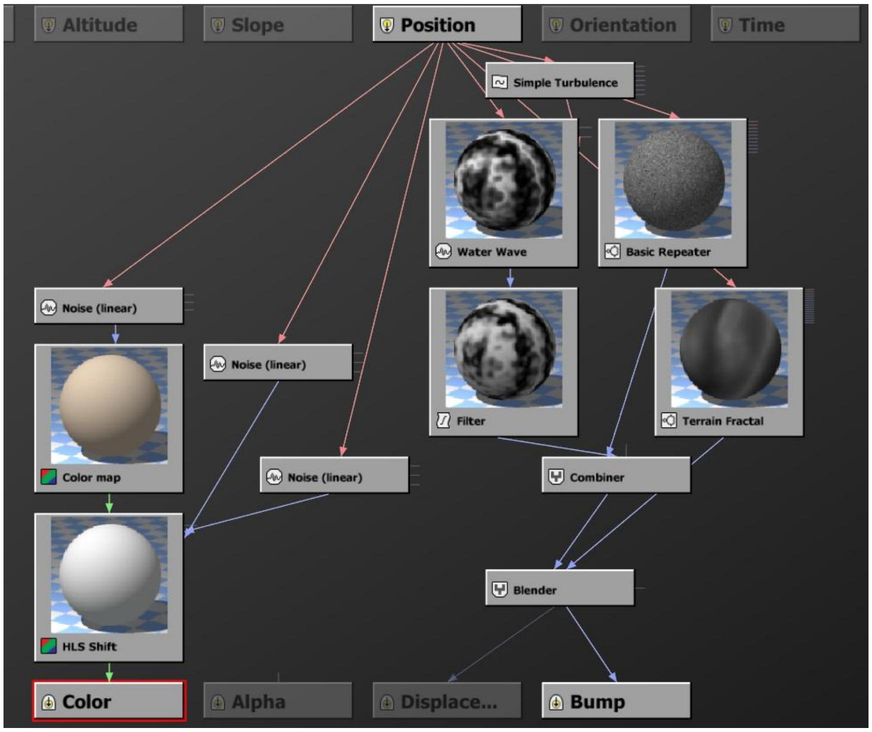
Sometimes the simplest of materials require a complex function to achieve a high level of realism. For this sand material, we are going to use several different types of Noise nodes and two fractals with the goal of creating realistic sand that has cresses and windswept curves which can be used on large scale terrains.

The function for this material starts with a Noise node which uses the Water Wave Math Pattern with a Simple Turbulence node providing the Origin which increases the amplitude and repeating of the noise node. The output is filtered to add more brightness resulting in larger shapes. It is combined with a Basic Repeater that uses very small scale Perlin Noise.

This is the ‘grain’ feature of the sand. The combined output is blended with a larger scale Terrain Fractal with a very high distortion to create the windswept sand features. The bias in the Blender is of 0.89 towards the Terrain Fractal.

The color is created by a very low-contrast Color Map using 3 sand colors with a Linear Noise node as the input.

The color map is then output to a HLS (Hue-Luminosity-Saturation) Shift node with two new Linear Noise nodes providing the Luminosity and Saturation shifts.

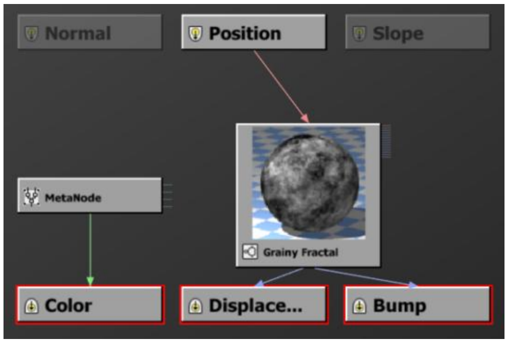


A multi-level bump production with an independent noise-based color production.

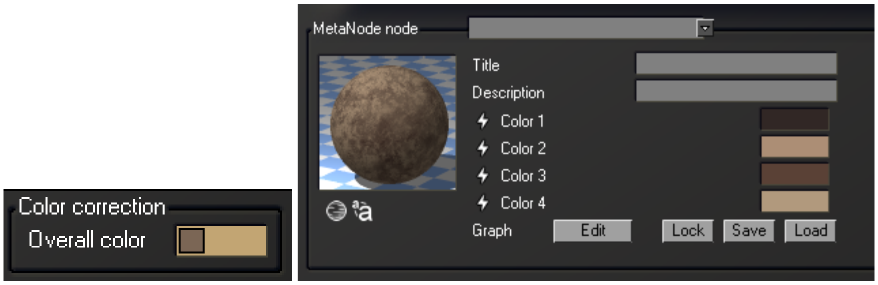
Because of such large scale color and bump features, we have a sand material that is more realistic than a simple patterning material which would look tiled from a distance.

Yellow Minerals

This material is one of the easiest and simplest to create. It uses a single MetaNode – the 4 Color Variations – available in the Color nodes and a single Grainy Fractal with default settings for the bump.



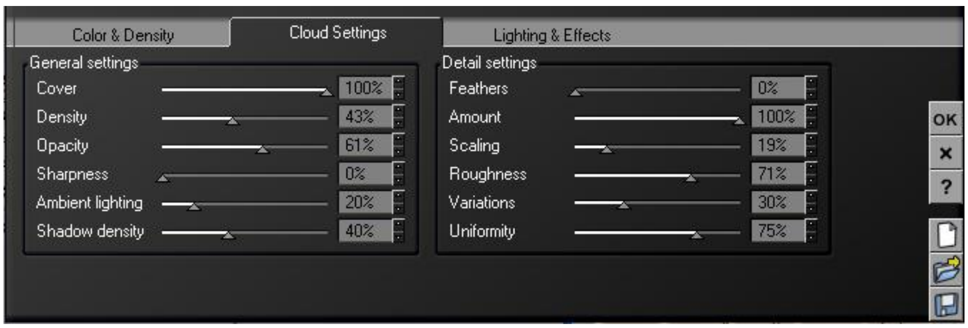
Finally, a simple color correction in the Color Tab shifts the entire brown-stone look to a soft, yellow mineral deposit feeling.



Color production of the mineral material.

Dust

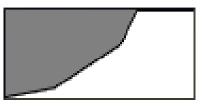
This is the same material we encountered in the MetaClouds section of Chapter 3. By changing the density of this MetaCloud material from 43% as shown in the first image to a very low 9% creates this soft, windblown dust as seen in the second image.



By placing several, small scale simple MetaCloud objects between mountains or even smaller objects like architecture can create a dusty environment. The function of this material is extremely simple: a single Perlin Noise > Gradient noise node with a Scale of 10 and everything else at default values.

Damaged Metal

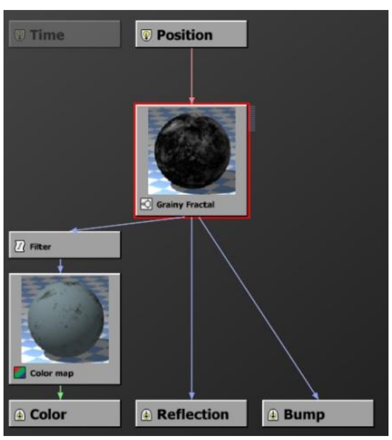
This damaged metal material has a very simple Grainy Fractal passed on the output to the Color Map via a filter.



Filter used to control the non-reflective area's coloration.

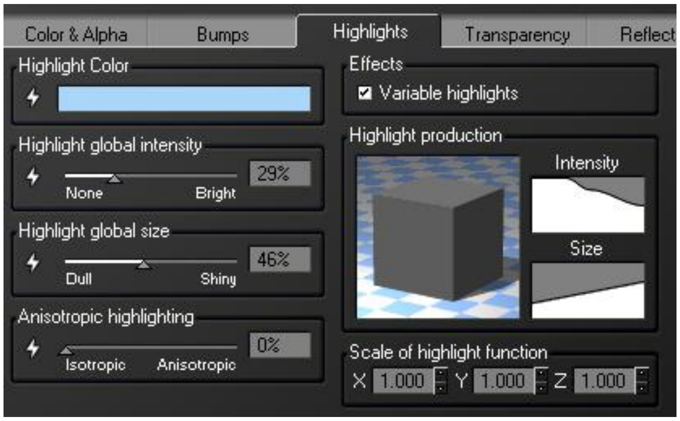
This filter lowers the "damage" area and the orange-brown color in the color map.

The Grainy Fractal is also passed on to the Bump and Reflection nodes, where different settings and multiple filters are applied to fine tune the bump and reflection, as well as a non-Function powered Variable Highlights setting.

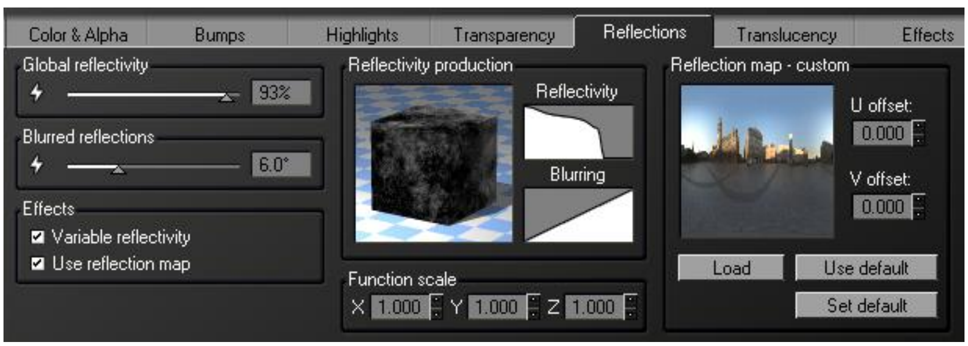


The highlights have a custom intensity filter to create an uncommon type of shine that further enhances the warped nature of the surface of the metal. The Highlight Color uses a light blue color to avoid the overly glossy white highlights. Lastly, the Highlight

Size is filtered to have a high 0.0 value and a low 1.0 value so the highlight is a little stark and contrasted rather than too smooth.

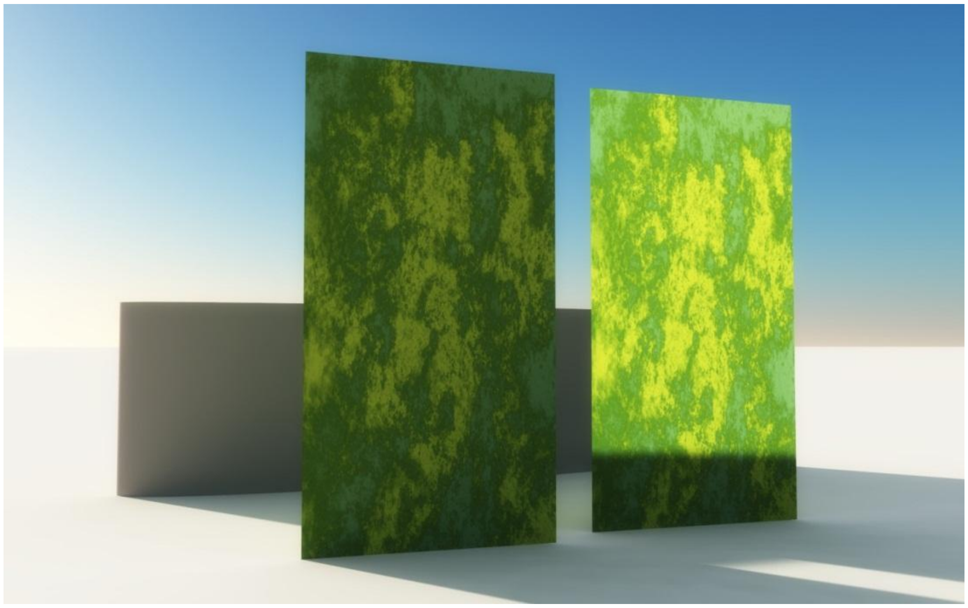


The bump is a very low -0.150 so our surface looks like real metal. The filter applied to the bump staggers the bump and cuts it off at the higher values to keep it from looking too bumpy and blunt.



Finally, the a very high and somewhat blurred Reflections setting, coupled with a HDRI map powered reflection through the Use Reflection Map option create the final touch of realism for our material. The reflections are powered by our Grainy Fractal which also creates the bump however you will notice that we use an extremely inverted filter for Reflectivity. This creates reflections only in the fractally-low but by color the “high” or metallic areas. The damaged looking areas which reside in the high ranges of the filter are cut off completely. This filter has virtually the same effect as using the Invert or Opposite (-X) filter node, however we get greater control over how the reflection is distributed over our material’s surface.

Backlighting



Backlighting is a very special material feature of Vue that often gets overlooked. By using backlighting, you can have light penetrate the object to a degree and simulate translucency.

Let's take a look at the image above to understand a practical example of backlighting. This render has two flat planes in front of a distorted cylinder. Both planes use the same material, except the right plane has Backlighting set to 100%. You can see light penetrate the second plane and even see the shadow thrown by the cylinder.

Unlike Subsurface Scattering, backlighting is much faster but provides a non-sophisticated solution comparatively as it only lets light through the object in a virtual sense. Subsurface Scattering actually calculates complex light bouncing patterns inside the object and takes your specific absorption and refraction index settings into play, while Backlighting only provides simple light penetration.

Backlighting can add realism to leaves, thin clothes, papers, flying light debris, and any thin object that would have light penetrating its surface.

Backlighting for external plants

Sometimes you may wish to use external plants created in other 3D software or custom modeled by yourself. Backlighting is an important aspect of materials in Vue that you have to apply to keep the realism of the plants consistent with Vue's own SolidGrowth plants.

Backlighting, a property available in the Effects tab of the Advanced Material Editor, processes light as it would appear coming through a very thin surface. This is not like

Subsurface Scattering, as it is much simpler method and the resulting effect is not as complex.

Comparison image of a plant without backlighting (left) and with Backlighting (right).

Leaves are thin enough for light to penetrate. So in the leaf material for your plants, you may want to turn up the Backlighting from 50 to 100% depending on your needs, the thickness of the leaf, and lighting conditions. You may also wish to turn on the One-Sided option for the leaf material for faster processing.

Subsurface Scattering

Often overlooked as being too simple or too complex (depending on your point-of-view), the Subsurface Scattering capabilities of Vue can allow you to add a rich layer of detail to your renders.

In a "normal" material, a ray of light bounces off the object and the returning color value of the surface is used to create the render. In a Subsurface Scattering (commonly referred to as SS or SSS) material the ray of light partially bounces off the surface as it normally would, but it also penetrates the surface creating a more complex and physically realistic version of the material.

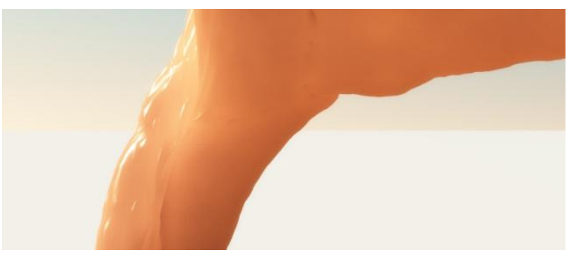
In the real world, a large number of objects have subsurface scattering of light - our skin, wax, the rind of an orange, clothes, marble, Styrofoam, and so on. Almost all organic matter (both flora and fauna) have a good degree of subsurface scattering. Subsurface Scattering works by taking light and partially absorbing it into the object and scattering the rest into the depths of that object.

Absorption-Multiple slider at 0% (full absorption)

Absorption-Multiple slider at 25% (biased towards absorption)

Absorption-Multiple slider at 50% (equal balance)

Absorption-Multiple slider at 75% (biased towards multiple scattering)



Absorption-Multiple slider at 100% (full multiple scattering)

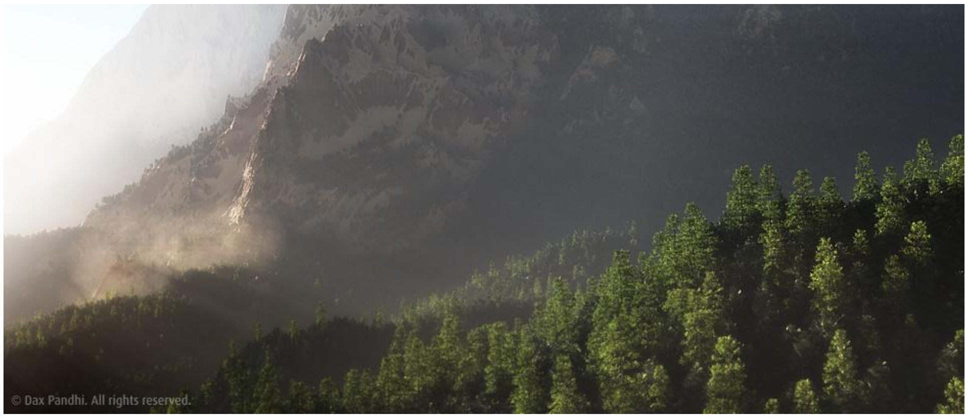
These five materials use Subsurface Scattering at varying degrees.

While still a basic feature in terms of depth, Subsurface Scattering in Vue is still powerful enough to create some fantastic results.

EcoSystems

Chapter 7

Seeding EcoSystems



Coniferous EcoSystem containing 1.3 million tree instances totaling 25 billion polygons

At the core of Vue, lies its best feature – EcoSystems. With a few clicks of the mouse, you can grow an entire tropical jungle, or a coniferous forest, or a pile of rocks, or even a street full of cars!

EcoSystems are one of the things that all Vue users are familiar with; and often because of the ease by which you can create extensive EcoSystems they can be taken for granted.

This technology, when used with a little finesse, can create breathtaking vistas that can rival an actual photograph. But the finesse can often be elusive. There are certain techniques that will help you as you experiment with different EcoSystems.

In this chapter, we will cover techniques that can apply to most types of vegetation and can ease the load of adding millions of objects (and billions of polygons) to your scenes.

Standard vs. Dynamic EcoSystems

Do you use Standard EcoSystems or go with Dynamic EcoSystems? In some Vue circles, the choice becomes an either/or choice, but in reality they go hand in hand. The answer to the question comes by asking another question: what is the purpose of this EcoSystem? The choice should be made on the requirements of the scene and what the hardware is capable of.

	Standard	Dynamic
Placement	Very precise.	Sometimes quirky.
Memory Consumption	Extremely high for large densities; affordable for smaller terrains/objects.	Uniform. Memory consumption is distributed to each piece being rendered, regardless of size.
Rendering Speed	Does not require dynamic population and repopulation during render time resulting in faster renders.	Ramp up time to render very slow for large EcoSystems as instances need to be populated repeatedly.
Instance Capacity	Limited by hardware. RAM/CPU for rendering, GPU for viewport display.	Unlimited. Renders may slow down for extremely large EcoSystems but capacity does not get limited by hardware.
File Size	Enlarges by number of instances populated.	Remains the same regardless of instances. (Except if preview instances are stored)

So again, depending on what your requirements are, you may choose to use either type of EcoSystem population. Personally, if I have enough hardware to handle it, I will choose Standard EcoSystems because of their fast response time when performing lots of quick renders. And sometimes, I use a mix of both population types based on the work at hand. In the following pages we will explore several scenarios and how to use both types of population in them.

Constraining EcoSystems

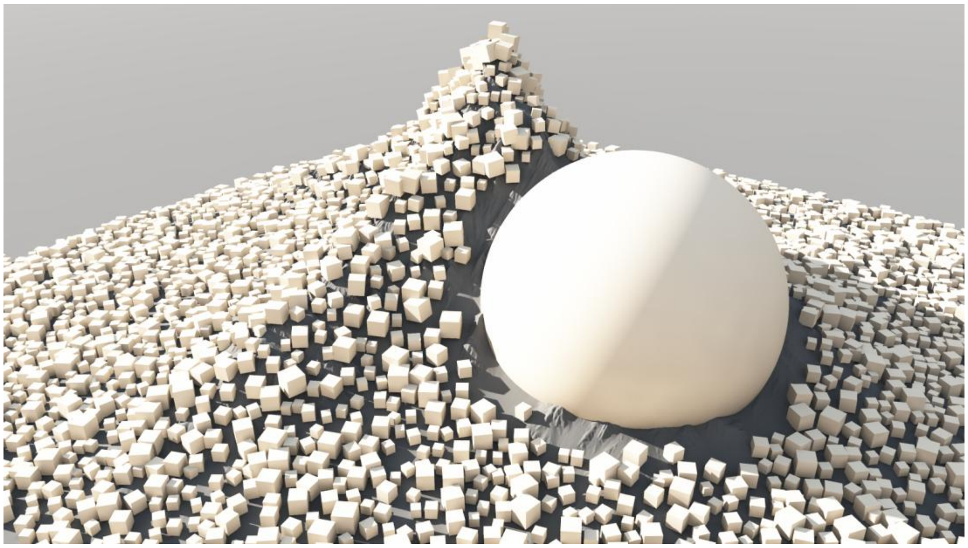
As someone coming from a background in old school 3D (since DOS, I'm sad to say), the ability to grow literally thousands and thousands of objects makes me feel so powerful! I know many people share this feeling with me where you just want to hit 'Populate' and have everything be covered with vegetation.

Unfortunately, that is not how the real world often operates. Even the thick, densely populated tropical rainforests have patterns of growth. If you follow these patterns, your results will be even more realistic.

The main constraint types we will use are: foreign objects, other vegetation, altitude, slope, and direction.

Abundant tropical mountains can still have limited growing areas.

Foreign objects

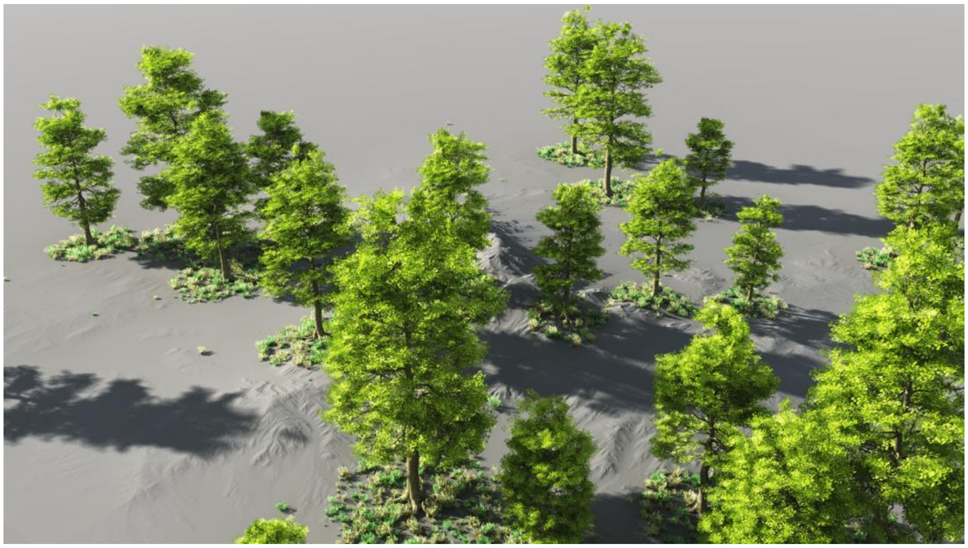


'Decay near Foreign Objects' in action

Rocks, manmade objects, water surfaces, individual tree objects, and the like are all considered foreign objects. Vegetation should not overlap with them. Depending on what they are, the EcoSystem should stop a little far away or just at the edge.

For faster performance, large primitive objects places outside the camera field of view, with Hide from Render checked, can prevent unnecessary population of non-dynamic EcoSystems resulting in less instances in the scene.

Layer Affinity



80% Affinity to Layer applied.

Sometimes you would have more than one layer of EcoSystem. For example, if you have an EcoSystem of large rocks or trees, you may want a second layer of grass to grow closer to these larger objects. The Affinity to Layer setting can help with this.

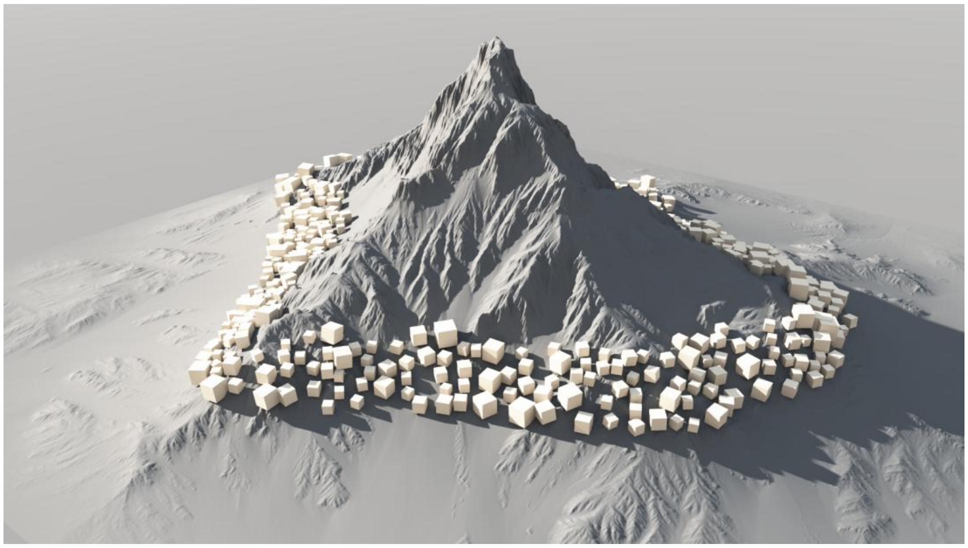
80% Repulsion to Layer applied.

Repulsion from layer can do the opposite and make the instances grow farther away from these objects.

The topmost (or higher) layer should be the one on which affinity or repulsion settings are applied.

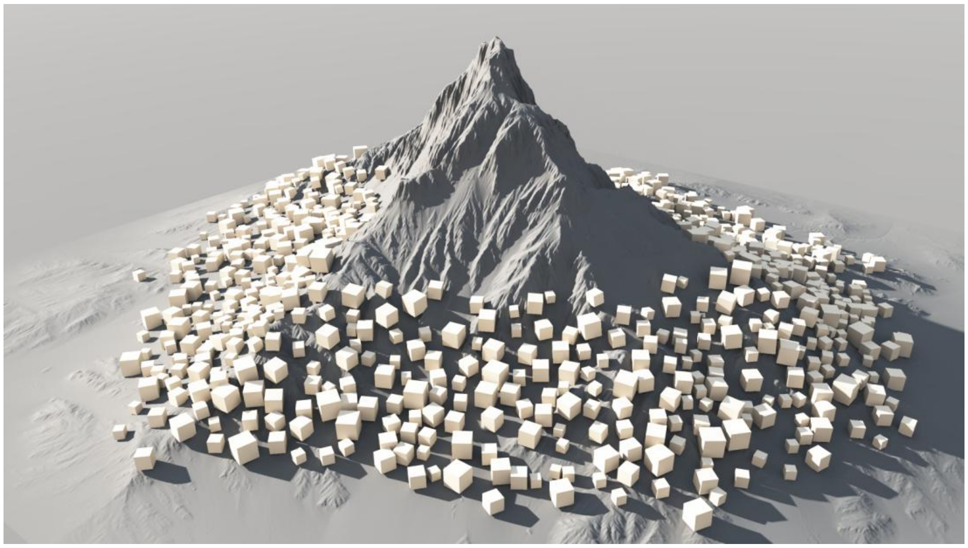
Environment Sensitivity

Altitude and Slope



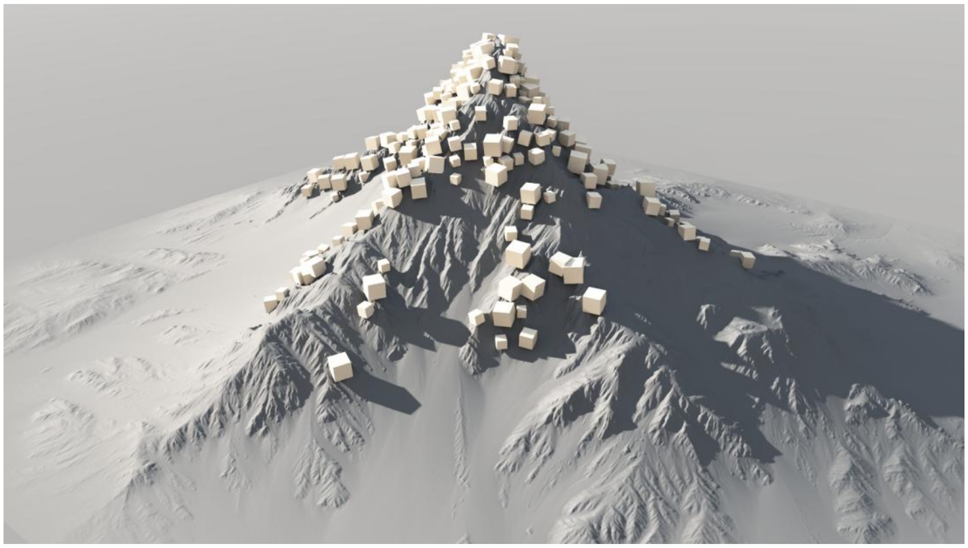
Constrained Altitude levels keep the EcoSystem in the selected zones.

Altitude and slope constraints are more or less self-explanatory. With these options you can choose to control the altitude levels or slope range between which your EcoSystem will be grown. Altitude can be chosen according to the object, the world, the material, or relative to Sea level, while Slope can be chosen according to only the steepness of the object itself, as judged by its current axis.



Constrained altitude with fuzziness (20%) for both Top and Bottom.

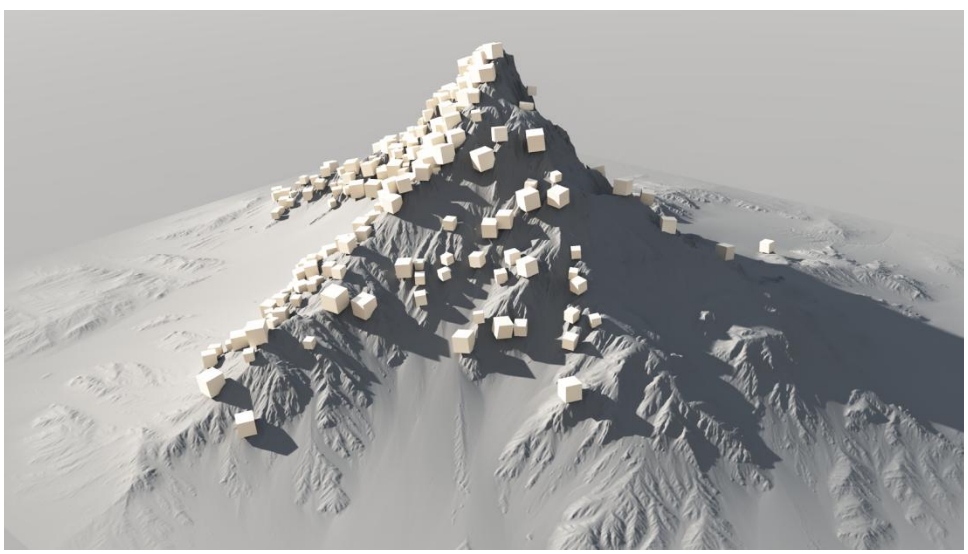
Slope's two entry fields and sliders represent 0 degrees on the left side and 180 degrees on the right side. Essentially, the left side represents how steep a slope can be to grow EcoSystem instances and the right side represents the flatness of the slope. For example, if you would like to have a flat terraced object or terrain, such as a cliff's side, you may want grass to grow only on the flat ledges. To do this, you can set the Slope levels between 160° and 180°.



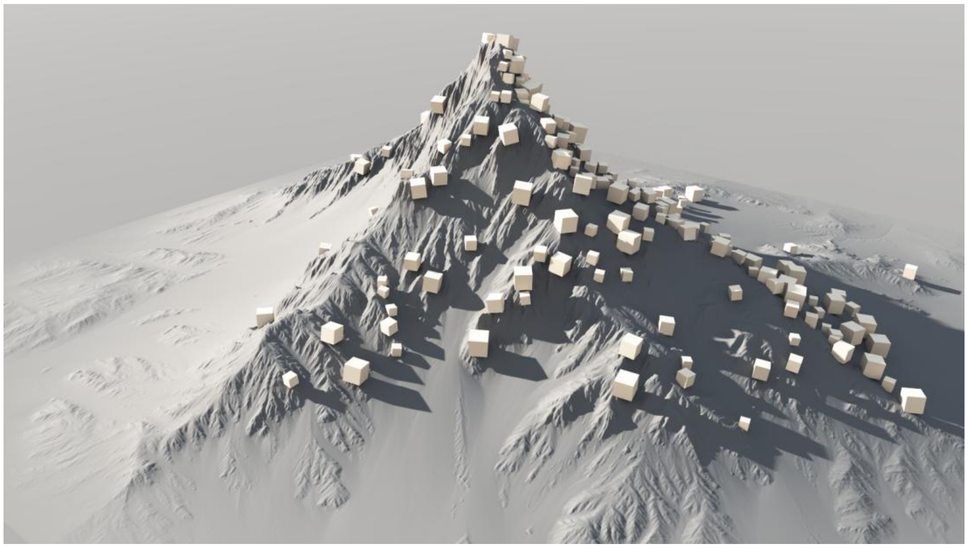
Slope restricted to steeper edges.

Orientation

Orientation allows you to choose the side of the object on which the EcoSystem population will be favored. The intensity of this setting is controlled with the Orientation Tightness, where 0% does not affect the EcoSystem at all and 100% strongly limits the EcoSystem to the side selected by the Orientation angle.

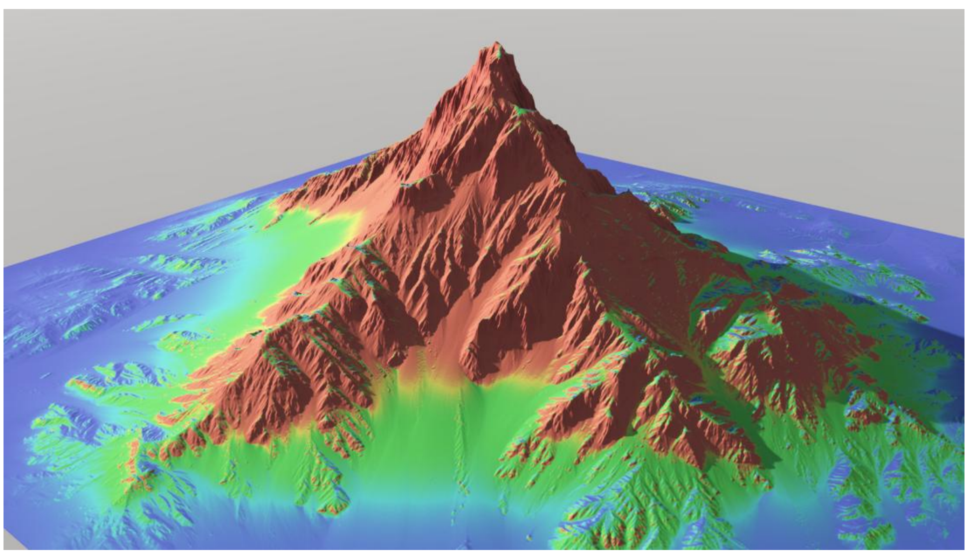


0° Orientation keeps the EcoSystems to the “left” side (if seen from the Top viewport).

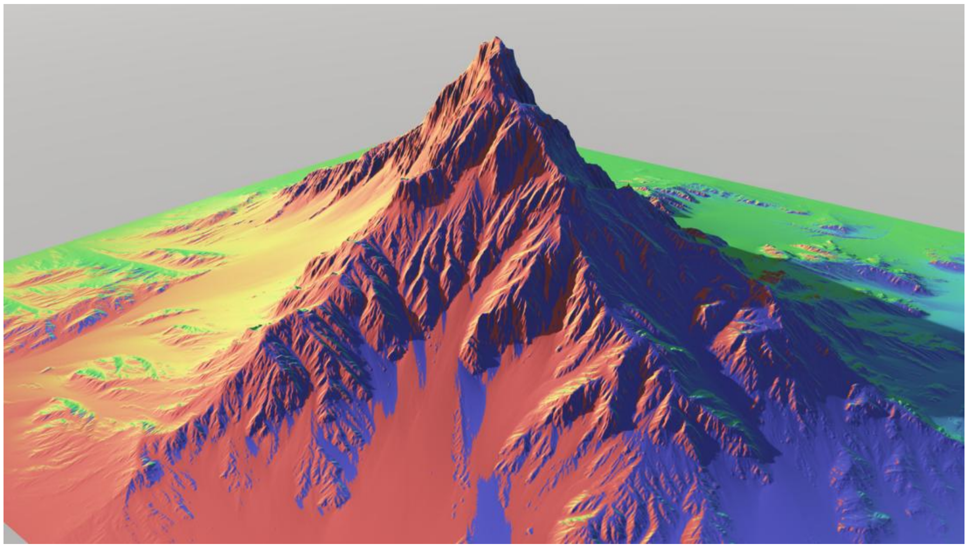


120° Orientation pushes EcoSystems to the "right" side (if seen from the Top viewport).

In nature, certain plants or trees may grow only where the sunlight falls, or the reverse in case of a harsh climate like the desert. Sometimes you may want to have strict control over where EcoSystems are populated or not for artistic reasons. In such situations, Orientation and Orientation Tightness can be used for subtle, yet deep control over the EcoSystem.



Steep slopes are warmer in color, while cooler colors represent flatter areas.



Warm tones start at 0° and lead up to a full 360° circle.

For more advanced uses, you can tie the Variable Density to Slope, Altitude, and/or Orientation nodes and manipulate them by using filters and fractals.

EcoSystem Examples

Rocks

The Rock Ecosystem in the image above uses the standard Vue rocks, but uses a custom material to add realism the default material does not provide.

Instead of creating a few large rocks, a lot of small rocks add a more believable quality to your renders. Thanks to Dynamic Population, you can have as many rock instances as you'd like.

Size

Use a relatively low size. For the scene above, a size of 0.250 was used with a Variable Size of 6.0 which created randomly sized rocks all over the surface.

In addition to the size, in the Density tab of the EcoSystem in the Advanced Material Editor, the Offset from Surface should be set to Proportional to Size of Instance, and the Distance slider below it should be set to anywhere between -35% to -10%. This will cause the rocks to go that much percent, based on their individual size, below the surface. Rocks in real life are often partially embedded into the ground. The default Vue rocks fail to give that appearance.

Distribution

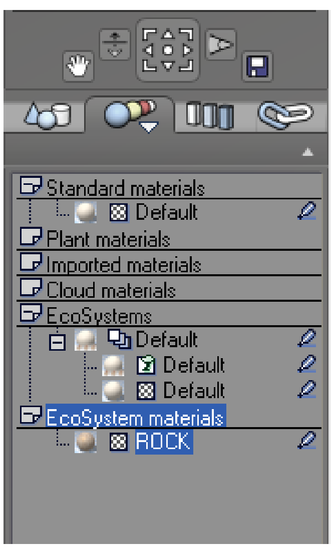
Normal distribution is ok for general situations, but sometimes you may want a little randomness in the rocks. Using any simple fractal from the Fractals function category in the Variable Density can help you achieve that look.

The image above uses the "Stones in Dirt" fractal.

If your terrain has medium or high level steep slopes, then using a Slope Range of 0.8 to 1.0 is recommended as slopes cannot contain rocks. If you must have rocks on the slopes, decrease the Offset from Surface to -35% or higher. You can use multiple layers of EcoSystems to have different rocks on the slope and the rest of the terrain.

Multiple Materials

By using the same method of creating more than one EcoSystem layer (this can be simply done by creating an EcoSystem layer, then going back to the Underlying Material and changing its Material Type to EcoSystem which adds another EcoSystem layer) you can have different types of rocks. One layer can create sparse large rocks with smaller rocks provided by another layer.

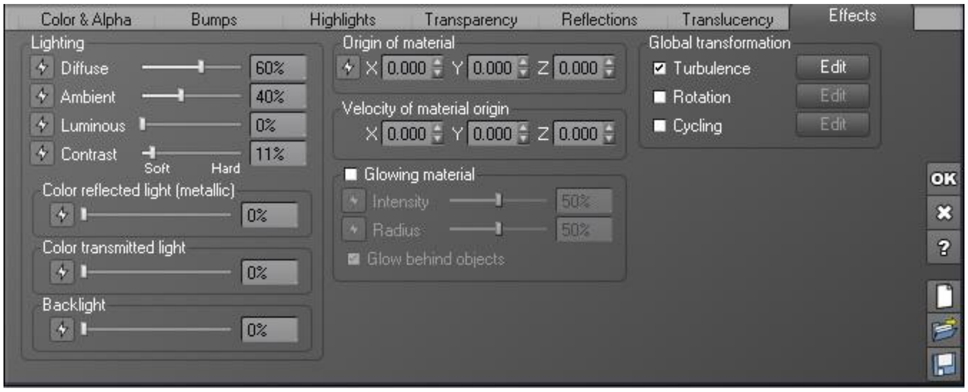


Materials tab of the World Browser allows editing EcoSystem materials

Using the EcoSystem Materials group, you can change the material of the rocks. But if you would like to have multiple rock layers, then it is recommended that you change the material of the first layer before adding the new layer. Otherwise both layers will use the "Chipped Rock" material and changing that in the Materials Tab will change both.

Distinguishing Materials

If you are using the same material for the terrain (or whatever object this EcoSystem is applied to) and the rocks, you may wish to change the Origin of the material. Otherwise you may see the same color variations on the rocks as well as the underlying terrain which may hide some rocks or even make them look artificial, as if created by displacement rather than actual object population.



Origin of Material in the Effects tab of the Advanced Material Editor.

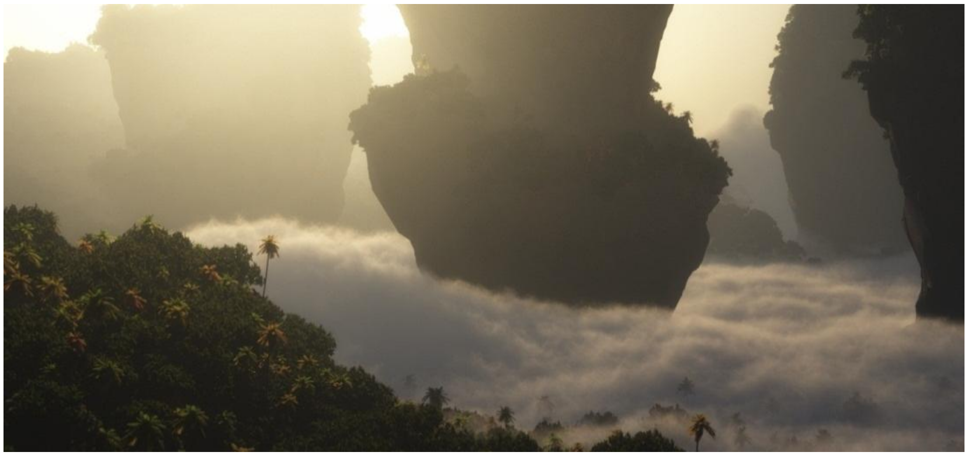
Rocks + Grass

In the same way as with multiple rock layers, creating a mixed EcoSystem of rocks and grass is better when created using layers rather than adding both types of objects in the EcoSystem.

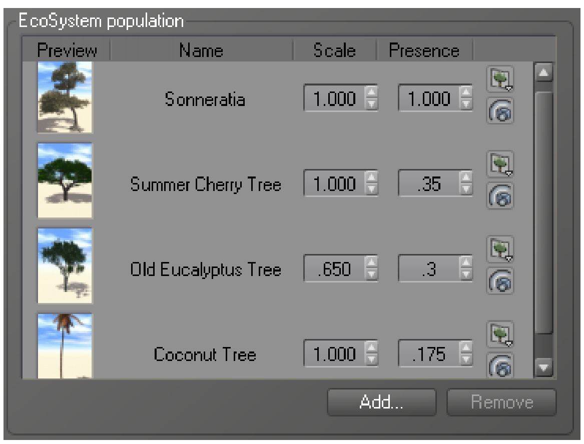
The image above is created of a two layered EcoSystem. The first creates the small rocks over the entire surface. The second uses the “Small Grassfield Plants 1” and “Carex” SolidGrowth plants to create the minute grass, with a 0.005 presence of Long Grass to create the occasional larger stalk of grass.

With Decay near Objects set for the grass, but not the rocks, the grass grows only on the top of the terrain while the rocks are everywhere including inside the water.

Tropical



One of the easiest mistakes and one you will often see, in making a tropical EcoSystem is to use the Coconut Palm only. Such scenes are very rare in real life and occur mostly on very small islands or a section of a beach. Even in such a place, the density of the trees is very low. Due to the nature of palm tree roots, you may find clumps of multiple trees, but a large, dense forest of palm trees is not possible.



For creating a visually rich tropical EcoSystem, try the EcoSystem shown in this image. By using the generic looking trees such as Sonneratia, Summer Cherry Tree, and Old Eucalyptus along with the Coconut Tree gives off the effect of a realistic tropical environment.

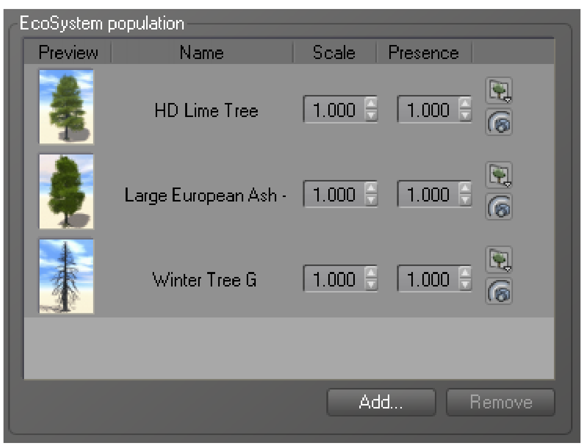
While technically the Eucalyptus may not usually grow in such an area, by lowering the size to almost half, we can make it look a generic gnarled tree. Pay attention to the ratios used for the trees.

The Coconut Tree ratio is particularly very low. The resulting distribution of the trees is quite dramatic looking and visually realistic.

Dead Trees

In the real world, not all trees are green and perfect – in any environment, especially densely packed ones – there are always dead or dying trees. By adding at least a single such tree species in your EcoSystem, you can lift the level of realism in your render.

In this scene above, we have a large northern temperate environment with large broadleaf (Lime and Ash) trees, we have added a 'dead tree' species with only 0.4 Presence. This peppers our scene with a few random dead trees.



This can be done for any type of EcoSystem as Vue has a large dead tree collection.

Try experimenting with different tree types, or if you feel ambitious, try modifying these existing types to be more scraggly and use your own new tree species in EcoSystems.

Aerial View EcoSystems

When your camera is hundreds of meters up in the sky, EcoSystems demand some simplicity for easier rendering. However, simplicity does not have to mean sacrificing quality.

This image shows us a typical single SolidGrowth tree EcoSystem on an island terrain. If you look closely, the EcoSystem is too even. In the real world, such dense growth would be uneven with trees of all shapes and sizes.

If you do not want to add more SolidGrowth plants in the EcoSystem, you can ‘cheat’ a little by simply increasing the Maximum Size Variation in the Scaling & Orientation tab of your EcoSystem to X: 1.0, Y: 1.0, and Z: 12.0 (or in that range between 8.0 to 16.0 based on the tree you are using). This will elongate the trees vertically. While obviously, not something you should do for close-up EcoSystems, it does wonders for an aerial view EcoSystem as follows.

If the size variations seem too random, a fractal can be used in Variable Scaling to create a rhythm for the size changes in the EcoSystem instances.

Multiple EcoSystem Layers

Multiple EcoSystem layers help manage your instances in an easier way while allowing greater creativity. Densities can be changed per layer, and constraints can be altered to suit that specific item's needs.

To create a new EcoSystem, simply switch to the base material layer and click the EcoSystem material type at the top of the Advanced Material Editor. This will create a new EcoSystem layer.

One of the best things allowed by multiple EcoSystems layers is the option to have certain layers to have Dynamic Population while others remain standard and/or manually painted.

A multi-layer EcoSystem in practice: 4 layers separating a complex terrain population.

We will deconstruct the above image to understand the practice of such multilayered EcoSystems.

Grass

With a strong density of 80% and Altitude range ending just near the edge of the water, the Grass layer contributed the largest manually populated EcoSystem. Instead of choosing the Decay near Foreign Objects option, a logical altitude restraint was used to enable faster population. The Grass layer had 3 types of grass SolidGrowth plants.

Flowers

Using the 'Transformed Crystals' fractal, two small SolidGrowth species of flowers were grown sparsely (3% density) throughout the terrain. The fractal was assigned for Variable Density, allowing a non-linear pattern for the flowers to grown on. The different Altitude constraint was used to keep the flowers away from the water. This allows the creation of a clean grass shoreline.

Lavender

The Lavender plant was created in a 3rd party software and wasn't SolidGrowth. It required special rotation and size values than SolidGrowth plants, so it was kept on a separate layer. The Variable Density fractal was a Round Samples fractal with high contrast. This created smaller patches of the plant, allowing them to grow in bunches rather than distributed throughout the terrain.

Rocks

This was the most intense and only Dynamic EcoSystem layer. The rocks were very small, and used a custom material. Manual population was resulting in over 3 million instances. The rocks had a variable size, but no other constraints except for altitude. By unchecking Decay near Foreign Objects, the rocks created a massive ground cover. The goal was to hint that the entire terrain, including the grassy bits as well as the river bed, was covered in small pebbles.

By using an altitude constraint that overlapped slightly with the Grass layer, enough rocks were placed in the foreground grass to hint to more rocks being present everywhere else but the angle and grass being so thick that they weren't visible. This saved the software from rendering several other instances of the rocks had such a restraint not been used.

Waterscapes

Chapter 8

As the life-giving element of our planet, Water becomes an integral part of natural design. With water, the object simulating the shape of the water and the surface material simulating the feel of water becomes more closely interconnected than any other object-material relationship.

In this short chapter, we will explore the main principles behind creating realistic water surfaces in Vue. Whether your aim is to create calm lakefronts, or rolling oceanic depths, a subtle combination of techniques presented here will enable you to get more out of your waterscapes.

In Vue, creating Water Surfaces is the epitome of subtlety working over starkness. Minor adjustments, made with precision, go a long way in creating excellent results. So the best thing you can do to create great water scenes is to pay careful attention to what each setting does, and change them in small increments to experiment with.

This applies especially to the water materials and the foam settings (both in the material and the object itself), because it may seem tricky at first, but you can get fantastic results once you get the hang of it.

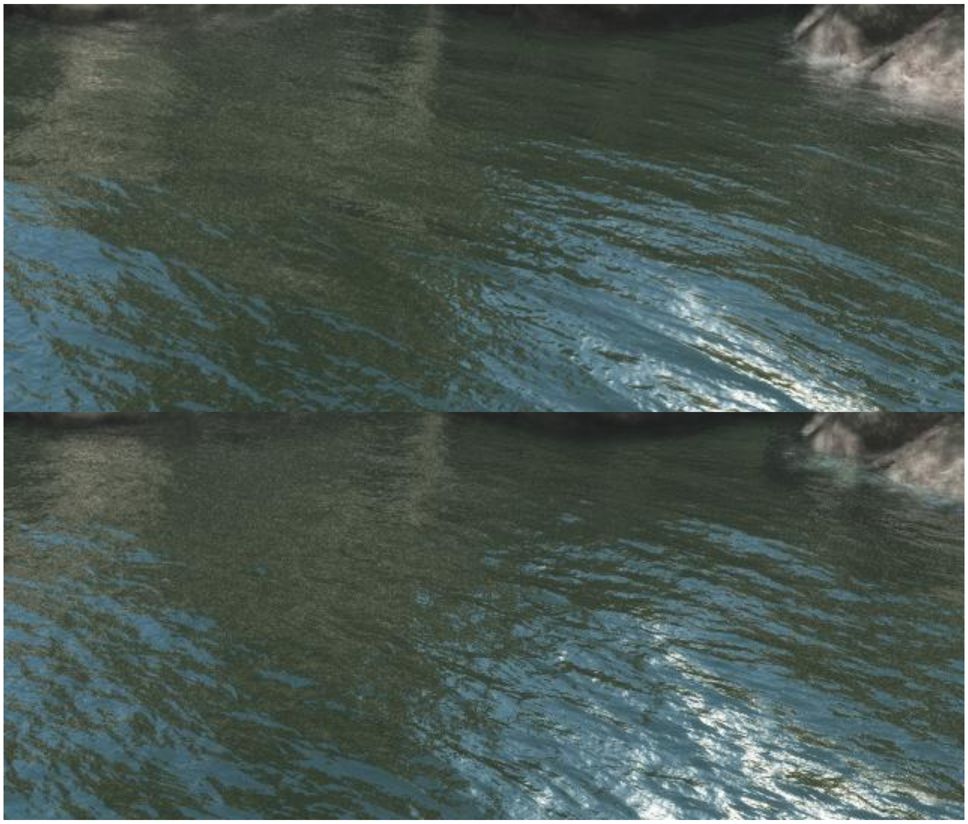
Normal Water vs. Displaced Water

With Vue 7, a new feature was introduced where we could displace water surfaces. Previously, they were just flat infinite planes with bumps simulating waves. The image above shows the traditional water surface.

With displacement the actual surface itself becomes 3D while adding procedural detail that is infinite in nature. This creates a very believable and realistic water surface. The image below is the exact same scene as above, except the Displaced Water Surface option is turned on.

Displaced Water Surfaces work well in both large panoramic images of open oceans and close up scenes of a small bubbling brook or a calm lake.

In fact, when creating a calm waterscape, like a crystal clear lake, it gives you added realism if you create displaced water surface while keeping the Overall Agitation to minimal. Take a look at this example, the first image is a normal, non-displaced water surface while the second image has displacement applied to the water surface.



The tiny details of having physical waves – no matter how small or calm – is better than the painted bump effect of a flat water surface. While not immediately visible to the eye, the added effect enhances your scene's realism factor greatly.

When you create animations, displaced water surfaces become highly important. When the camera moves around the water and the water itself is moving, the difference between a bump-painted water surface and a fully displaced water surface become paramount!

Realistic Water Materials

The material of your water surface is just as important, if not more so, than the shape of the water. To achieve our goal of realistic water, we will create four different water materials based on the Default Water material that comes with Vue. These four materials will be Clear Water, Deep Ocean, Jungle River, and Muddy Water.

Before we create our water materials we will explore four cornerstone conditions to consider when making water materials.

Transparency in Depth

A common mistake seen in Vue renders is the transparency of the water is decreased using the Fading Out controls and pushing the Clarity slider towards Murky. While this control is useful to add murkiness to your water, forcing it to become too dark will make you lose the other helpful properties of water – for example any terrains or objects partially submerged in water will go completely invisible too quickly in the water rather than softly fading out creating the proper illusion of depth.

Reflections

Another important aspect of realistic water is reflected environments. Below the Fading Out controls is a slider that controls reflectivity called Turn Reflective on Angle. This is usually left at around 30%. However, most types of waters have this kind of (30%) reflectivity when seen from higher altitudes close to 75º or higher! For a more realistic "human eye level" reflectivity, you need to push this reflectivity to 65% to 90%.

This may sound like extreme amounts, but take a look at the example images below to see the difference between the default water (left) and the high reflectivity water (right).

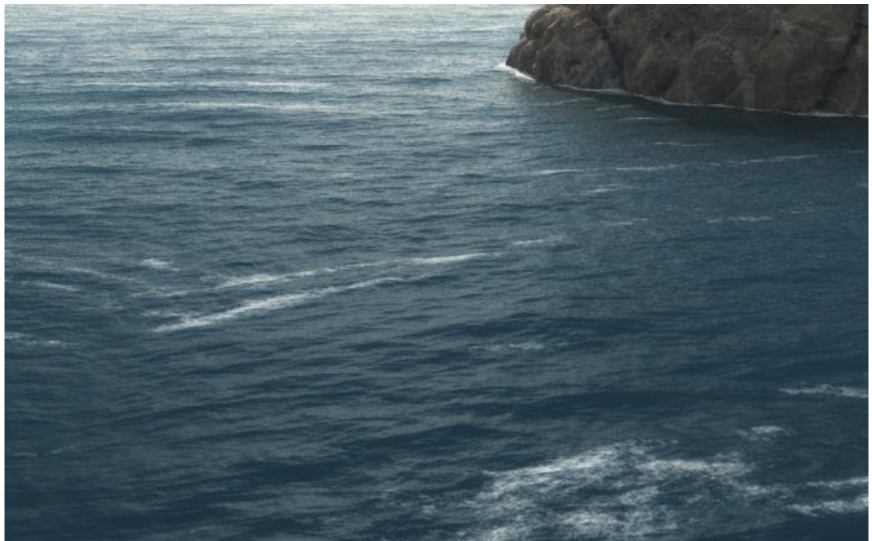
Due the nature of physics, reflectivity also relates to our eye the “liquidity” or viscosity of the water. Therefore, more reflectivity results in our eye believing the water to be fluid and flowing.

Never use Reflectivity itself to increase the reflections on the water as it is indiscriminate and will not take wave angles into consideration. Always use Turn Reflective on Angle.

Specular Highlights

The last aspect of realistic water materials we will look at is Highlight – the tiny specks of light that shine on the water. Just as the reflections contribute to the fluidic look of water surfaces, the highlights contribute to the same factor along with adding the “surface level” illusion.

The surface level illusion is basically the appearance of having the shiny specular dots on “top” along with the reflections and therefore enhancing the effect of depth visible in the other submerged portions where objects are partially visible below the water.

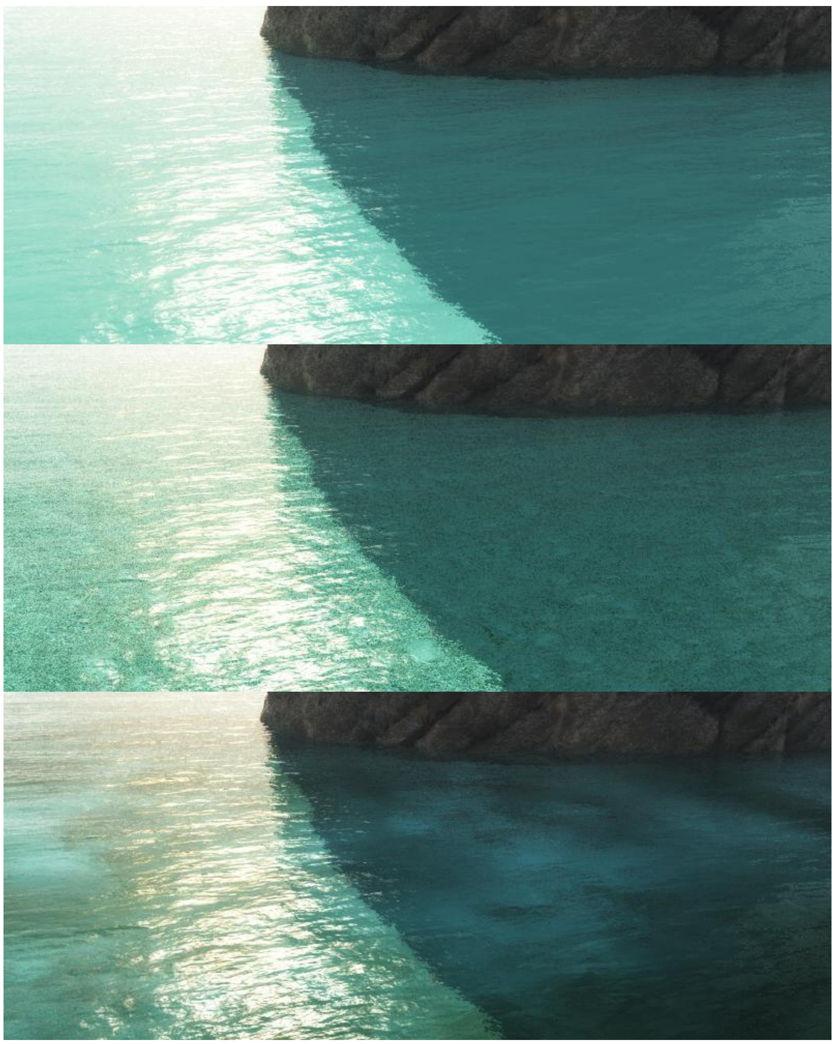


So the “surface level” and the “depth level” are contrasted in the mind of the viewer. Additionally, the shine of the water has to be sharper.

Refracted Depth

It is extremely important to consider what is going to go beneath your water surface. Just like in real life, Vue will refract a distorted image of whatever is below the water onto the upper surface. Even if using a flat surface below the water, make sure you have a proper material applied to that surface.

The differences between a no-material (Default), bad material (Rocks > Sick rock), and proper material (Rocks > Green-Brown Rock, Scale at 5.0) surface resting below the water surface are shown below.



To clarify the difference between a bad material and a good material would be the amount of noise it creates on the water surface. You can tone down the bump value on the underlying material to lower the noise, in most cases.

Clear Water

Like with all the other waters (except Deep Ocean water), I recommend not using the Foam layer unless you absolutely have to. If you do not use it VERY carefully, the Foam layer can make your entire scene look unreal.

So let's start by creating a Water surface and editing the material.

First, we remove the Foam layer by changing the layer Alpha to -100. Then we go to the Highlights tab and change the Highlights by pushing the Highlight Global Intensity and Highlight Global Size to 100% for both. This will make our water shine brightly and make it look sparkly as clear water should.

Next we go to the Transparency tab and change the Turn Reflective with Angle to 78%.

The rest of the settings can remain default. If you want the water to be more clear and tropical, you can change the default Murkiness from 30% to 20%.

This is what our new Clear Water looks like:

Muddy Water

Exactly opposite of our Clear Water, is the Muddy Water.

To achieve a muddy look, we change the same settings as before. Start by removing the Foam layer by setting the Alpha to -100.

This time, the Highlights need not be as high since mud infused water would not have the same sparkle as clear water. By changing the Highlight Global Intensity to 2% and the Highlight Global Size to 65%, we will achieve highlights similar to real muddy water.

In the Transparency tab, we make more changes than before. First we increase the Murkiness from a Clear 30% to a Murky 63%. The Fade out Color becomes a RGB 95, 78, 47 and the Light Color changes to a RGB 210, 215, 195. The Turn Reflective with Angle value is decreased to 40%, as muddy water will not be as reflective. You can turn this down further if your scene requires even more “muddiness”.

Jungle River

A Jungle River water material can be built upon the same principles of the Muddy Water, however we will make slight changes to adapt our water for a jungle environment.

Again, we remove the Foam layer by setting the Alpha to -100.

We change the Highlights Global Intensity and Size to 50% and 80% respectively allowing for the slightly more shiny water found in a jungle river as it contains particular matter just like muddy water, but because it is flowing it will have more crispness.

Next we go to the Transparency tab and change the Turn Reflective on Angle to 32% to compensate for our 'contaminated' feel of the water. We further enhance that by changing the Fade out Color to a RGB 62, 77, 50, and the Light Color to a RGB 166, 186, 117. Both of these colors are very green hued, allowing for the illusion that organic matter such as dead leaves and other such things may be inside the water.

Deep Ocean

Creating an Ocean Waterscape requires a strong material to properly portray the depths of the water.

We will start by using the Default MetaWater, since we will require the Foam layer in this material. All changes mentioned below are to the lower Water layer, and not the Foam layer.

As before, we change the Highlights settings by sliding the Highlight Global Size to 80% to accommodate the sun on the open ocean, and the Highlight Global Size to as well 80% to properly show the shine on the larger waves.

In the Transparency tab, we change the Clear-Murky setting to 50% to make the water look deep and dark. The Fade out Color becomes a RGB 27, 41, 36, with a Light Color of RGB 157, 216, 252, creating a murky feeling. We move the Turn Reflective with Angle value to an even 80% to reflect either the open blue or gray overcast skies.

Reflecting Illusions

A key element in creating waterscapes is perception. While we will delve deeper into the topic of perception in the final chapter of this book, this specific case of perception is too great to be kept apart from this chapter about creating realistic waterscapes.

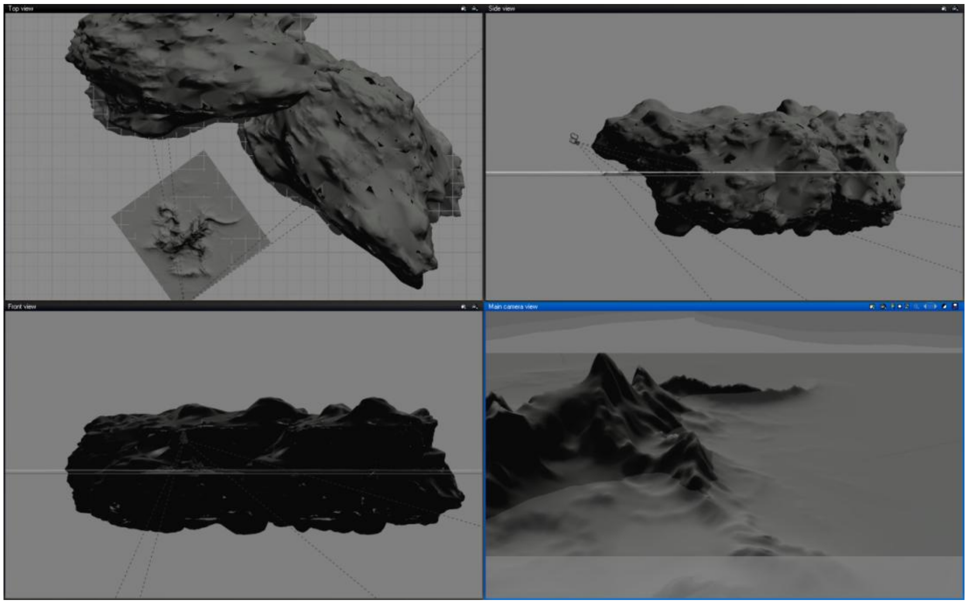
Let's start with a simple question: in 3D, when your camera is above the water not showing the horizon, in fact not showing much of land at all, what makes your water a river or an ocean?

The somewhat mysterious answer is: the parts of the scene not shown. The not so mysterious principle behind it is the reflected world.

When you are looking straight into the water, whether you know it or not (as we discussed briefly in chapter 1) your mind will perceive the reality of the scene by what is reflected in the water. So you must create a "world" around and outside your immediately visible field of view (i.e., off camera) that will be reflected onto your water surface and thereby creating an expanded perception of the world.

Let's take this image below as an example. It is a typical rock outcrop you may find in any water body, whether a river or an ocean. However, this immediately looks like a large water body like an ocean. We need to make it look as if it were part of a river.

We do this by adding a couple of terrains (or HyperTerrains in this case) just outside the field of view of the camera. In the "Main Camera View" you can see the rocks just outside the visible frame. These objects do not require too much resolution, just enough shape that it will create unique reflections. You can use a primitive like a stretched cube, but a terrain or HyperTerrain – even if it is a default Standard Terrain of a 256x256 resolution – will add realism a primitive cannot provide.

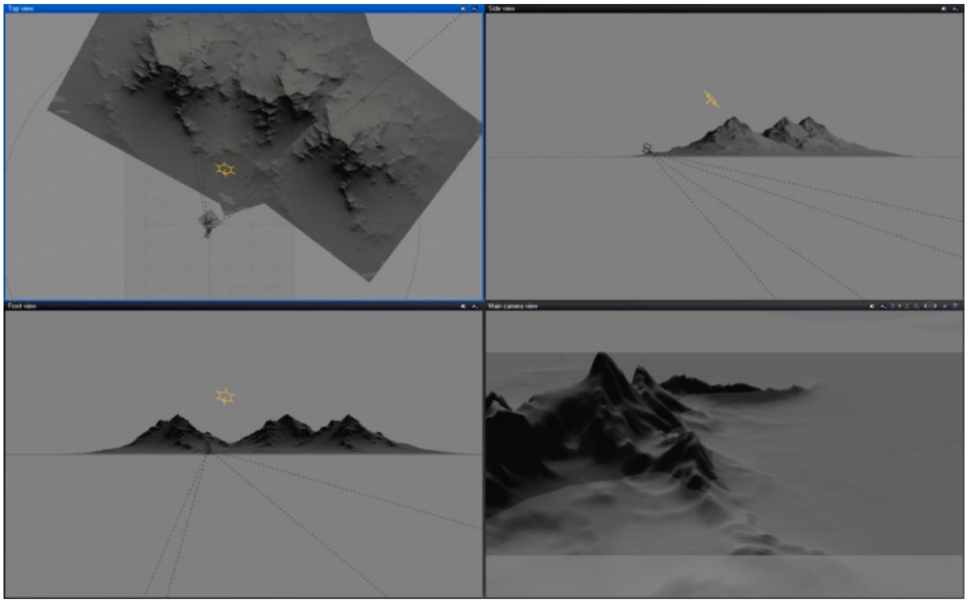


In the case of our scene, a HyperTerrain is used that was decimated from 0.98 million polygons to just a few thousand. It was then duplicated and rotated to create a larger rock field to be reflected, giving a “cordoned off” feel. Giving your reflected world proper materials is also important.

Now, let’s render our image.

The reflected “fence” effect of the rocks makes you believe this scene to exist inside a contained or at least partially confined water body like a river. You can further enhance this effect by bringing some of the rocks (or adding new rocks) closer and into the camera view but only partially, hinting at a world larger than is seen.

Now, if we wanted to turn out scene into a lake, all we need to do is “lessen” the reflections. Instead of our rock formations, let’s use a terrain – the Mt. Cornucopius terrain, for example, from QuadSpinner that is available for free on Cornucopia3D. Of course, any terrain will do if you are not looking for a very specific reflected shape.



With this distant, large terrain looming over our water body, the subtle dark reflections in the top of the frame create the illusion of a lake or a large contained water body.

Reflecting the Sun

If you have a water filled scene with the sun reflecting highlights in the water, they may often appear a little too bright and white. For a little added effect, you may want to increase the Corona of the Sun.

Let's look at an example. The image below shows our scene as it is. The Water surface uses our Clear Water material from the previous section.

You will notice that the highlights are simple white specks. Let's go to the Atmosphere Editor and in the Sun tab, we will push the Corona from 20% to 55%.

In this render, you can see orange glowing edges in the highlights on the water. This is caused by the increased Corona size of the Sun. Be careful, however, since the higher Corona size will make the sun glow brighter and hotter. If the sun is visible in your camera, it will make the sky a lot brighter. Experiment with different values to get the right amount of glowing edges in the water while keeping your sky practical.

Additionally, if you want more – or rather wider – sun highlights in your scene, you can increase the Anisotropy setting in the Highlights tab of the Advanced Material Editor. By default, Anisotropy is set to 0%. Using Anisotropy requires a Displaced Water Surface. Anisotropy scatters the highlights across the surface "horizontally", but requires curvature of some form on the object. A non-displaced Water Surface is a simple plane which does not have any curvature at all, so Anisotropy has to be set to 0% otherwise the highlights will appear incorrectly.

The renders below show Anisotropy values of 20%, 60% and 100%.

Displaced Water Angles

When a Water Surface is displaced, the angle used for the Wind Direction is 45°. If your camera angle is pointing north (‘up’ in the Top View), this angle may not look pleasing.

Experiment with a variety of angles to get the right waves in conjunction with your scene objects.

The images below show different water flows from the Wind Direction.

Perception

Chapter 9

The Heart of Digital Nature

Perception is at the heart of our goal of creating art that mimics nature. In this chapter we will talk technique rather than technicality. We will apply the different aspects of Vue we saw in all the previous chapters in real scene situations.

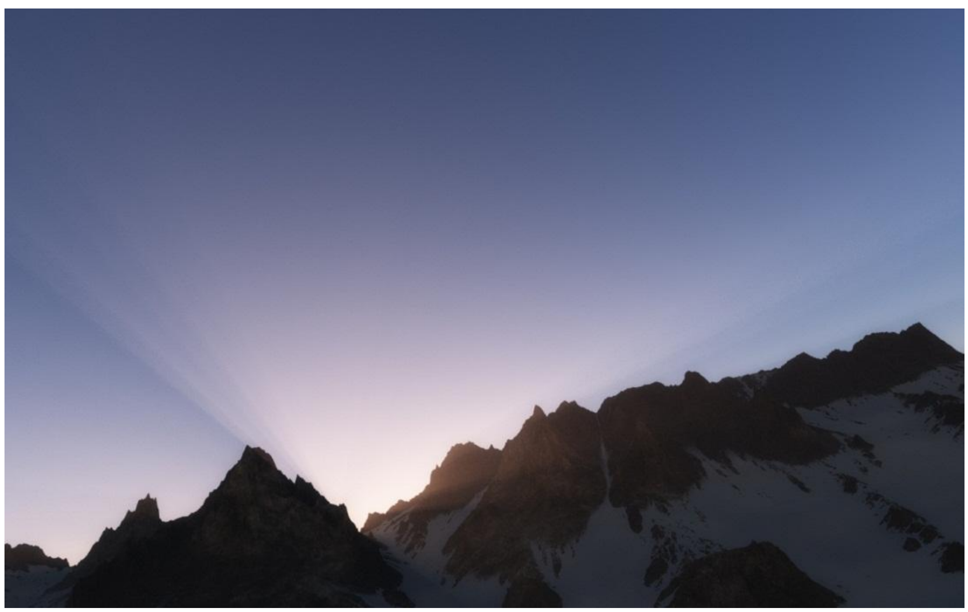
By changing certain minor aspects of your scene, or how a part of your scene is made inside Vue, you can dramatically shift the perception of the viewer and make your scene look so much more than what it is.

Atmospheric Colors

A common phenomenon seen during twilight, in real life, is that the sky color near the horizon and even the clouds take on a color that is different from the sky and the decay – for example, pink or red. Technically, this is still part of the decay of light as it travels through the various particle matter present in the atmosphere. However, duplicating that in Vue is not as simple changing the decay color. Doing so may alter the light color extensively and beyond the level of control.

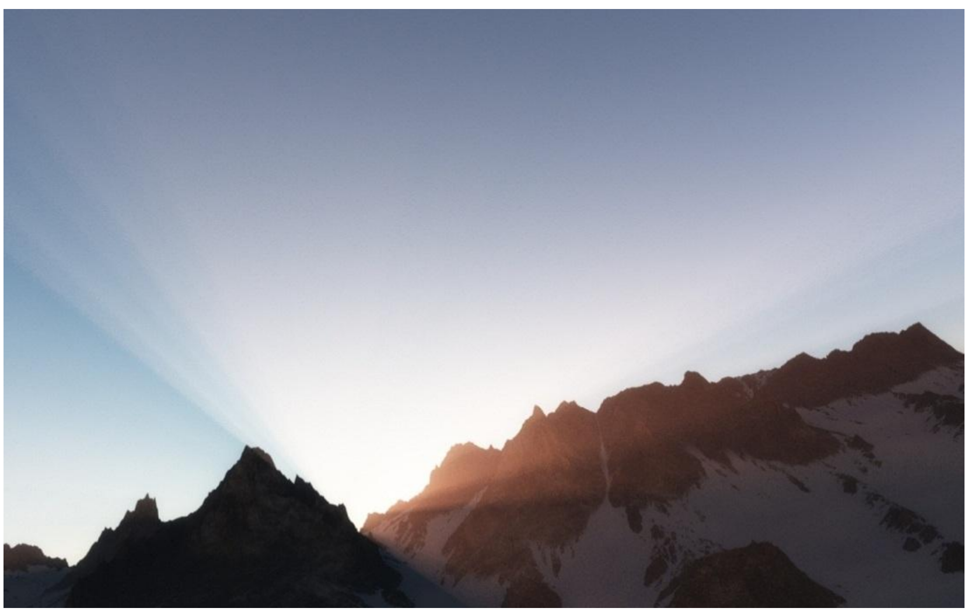
The solution lies in nature itself. This extra light decay or diffraction is caused by an extra or more potent component in the air – moisture, for example. In Vue, moisture in the air is represented by the Fog controls in the Atmosphere Editor and dust or other particles by the Haze controls. Therefore, by changing the color of Haze and Fog, we are able to add such colored sunsets or sunrises in our scenes without affecting the actual sunlight, sky, or decay and retaining greater control of the color shifts.

Let's look at some examples.



Haze of RGB 153, 119, 100 (light purple tone). Default fog (ice blue tone).

Haze of RGB 174, 96, 78 (rich and bright magenta tone). Default fog (ice blue tone).

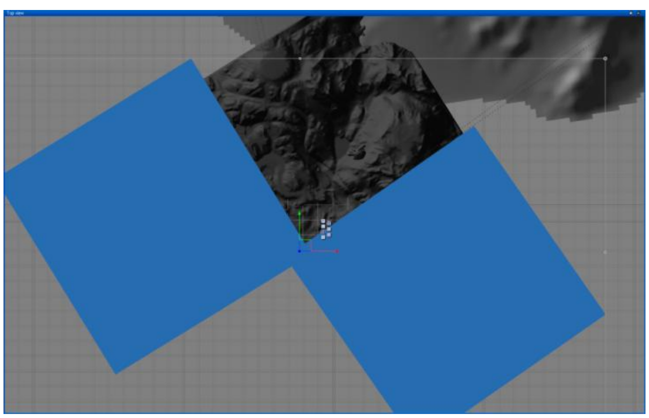


Haze of RGB 109, 137, 142 (dull blue) and fog of RGB 245, 137, 116 (warm tinted).

Small Valley or Deep Abyss

A common scenario found in 3D scenes would be of a canyon terrain, with the camera stationed inside the canyon looking outside towards a larger landscape, or perhaps another section of the canyon. The question is how to make the canyon seem deep rather than a small gully. By changing the perception only slightly – and using only shadows of offscreen objects – we can achieve this effect.

This image above shows our render as is. The image below shows the viewport where we have now added a simple pair of cubes. These primitives are not visible in the camera frame but cast shadows on the walls of the canyon that *are* visible in the frame.



Large light blocking cubes placed outside of the camera's field of view.

As you can clearly see in the image above, this shadow technique shifts the perception of the viewer into thinking this is a deep canyon.

To take this a step further, we add a flattened cube on top of the area where the camera is. This blocks skylight and extends the depth of the canyon, making it seem more like a cave.

Extreme Decay

We talked about Light Decay in Chapter 2, and why it happens. It is common to find an overuse of decay among newer Vue users. The effect looks good often, and is rightly warranted in some situations, but generally we do not see much Decay except for times of precipitation near dawn or dusk. High decay renders also convey a negative or oppressive feeling.

Below we can see a comparison between a high decay and a normal decay sunset setting.

Avoiding high decay lighting can help boost the realism of your scene.

Decay can be a subjective thing – both for artistic and realistic purposes – because the light decay and its form vary in different parts of the world. The best way is to find references for the type of environment (or the part of the world where that environment might belong) and use a decay suitable for that place rather than what you see in your own backyard.

Offscreen Tree Shadows

A render is never limited to just the objects in the camera frame. As we saw in the previous sections, objects that are out of the frame can still affect the scene. One of the best scenarios to use this technique for is to throw shadows of trees on to the edge objects inside the frame to portray a larger grove of trees beyond.

In this image below, a simple resized cube (in the shape of a flat platform) has an EcoSystem of a few HD Ash Trees that throw their shadows on the architecture.

Counting Lines and Layers

When we look at a natural landscape in real life, our depth perception is able to figure out different layers. However, in a render you have to be very careful that your layers are prominently displayed, otherwise the viewer may not be able to see everything in the image properly.

We already saw how the Golden Ratio and the Fibonacci Spiral can help us create better compositions. Similarly, by practicing an economy of 'lines' or overlapping areas can help you compose your images in a more efficient and viewer-friendly manner.

To understand this concept, let's look at a bad image. The render below, first of fails in composition because there is no proper point of focus as our concentration is divided in more than one areas which seem somewhat prominent but not exactly. Secondly, this render has too many unclear areas of overlap. In this light, some of those layers become too difficult to see.

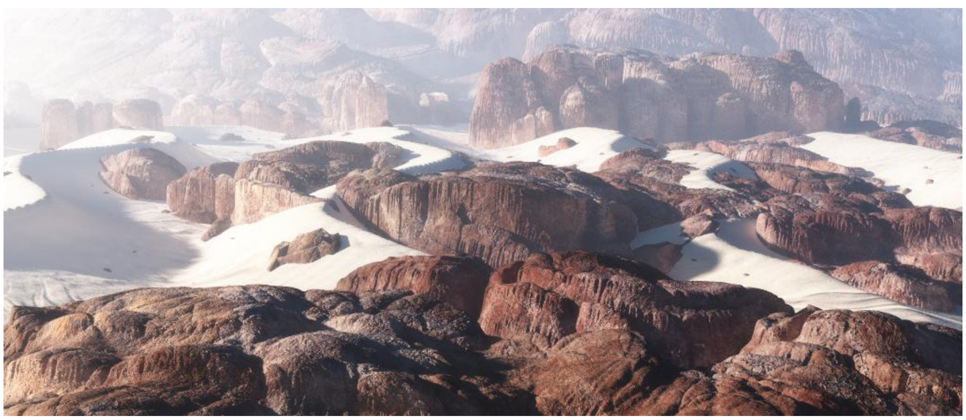
Overtly ambiguous overlapping objects, uneven lighting, and no clear central focus area.

Let's analyze this image by drawing actual lines on the major overlap areas. The red lines show where a large curve or depth occurs in the image creating a visual overlap zone.

On the far right, you can see that some of the elements are difficult to see because of the lighting and the ambiguous layering of objects. The central area is relatively cleaner, but the clarity is lost because of bad composition. The worst part of the image is the far left where we have several HyperTerrains overlapping to create natural bridges and large rock cliffs. While the objects themselves may be proper in form, their position and lighting is extremely bad and the viewer is unable to clearly define 'what is where'. This creates a somewhat disorienting effect which will instantly lose the viewer's attention.

As with most of the other techniques and concepts presented in this chapter, this is not a set rule. This is just common logic. By practicing this technique and counting these ‘overlapping lines’ in mind while making your scene, you can have a better final image.

Even if you have too many lines, making sure they do not overlap excessively you can be assured a good render as illustrated in this desert scene below.



Softer Shadows for Twilight

Sunrise and sunset, unless it is an extremely clear atmosphere like that found after rainfall, the sunlight is softer than at daytime. As a result, shadows are soft at twilight.

For a normal daytime scene soft shadows of 2.0 degrees may be enough, but for twilight renders settings of 3.0 or higher may be required.

A sunset in Nevada, USA depicting soft shadows during dusk.

Volcano Arenal, Costa Rica showing soft shadows as the sun heads towards the horizon.

Rocks, Rocks, and More Rocks

The real world is full of rocks, stones, and pebbles. By adding a Rock EcoSystem as we saw in Chapter 8, your terrains will come to life.

Adding rocks does not have to be large boulders. In fact, the Vue rocks are best suited for small rocks and pebbles (HyperTerrains being the apt choice for creating boulders and larger rock formations). Relying on Dynamic Population, you can add millions of small stones and pebbles to enhance the beauty and realism of your renders.

By using Variable Density fractals, you can make them look even better with non-linear population. The best fractals for this in the Vue library are the Alien Rock, Asteroid, Fast Rock, and Lava Flow. Always remember to rescale the Function according to the size of your terrain otherwise you may have too small a fractal shape and the populated rocks may not look good.

Bad Lens Flares

Nothing screams CGI as a lens flare. Lens flare is an optical defect (yes, it is not an intentional effect for the most part!) created by multiple glass optics in the lens of a camera. The only lens flare the human eye sees is the distorted glow rays which are created like Godrays – by some object, often our own eyelashes, obstructing the sunlight. The secondary rings and multiple circles are camera effects.

As a general rule of thumb, lens flare effects should be avoided completely when you are aiming for realism in a 3D scene. If you must have a lens flare, it should be subtle and not intense or glaring.

As mentioned before, a lens flare makes the image clearly identifiable as something computer generated. Our goal is realism – to make the viewer think of the render as a photograph rather than a CG image.

Look at the above image without (top) and with (bottom) lens flare. The bottom image automatically looks 'fake' because of the sharpness and intensity of the lens flare.

Bad Depth-of-Field

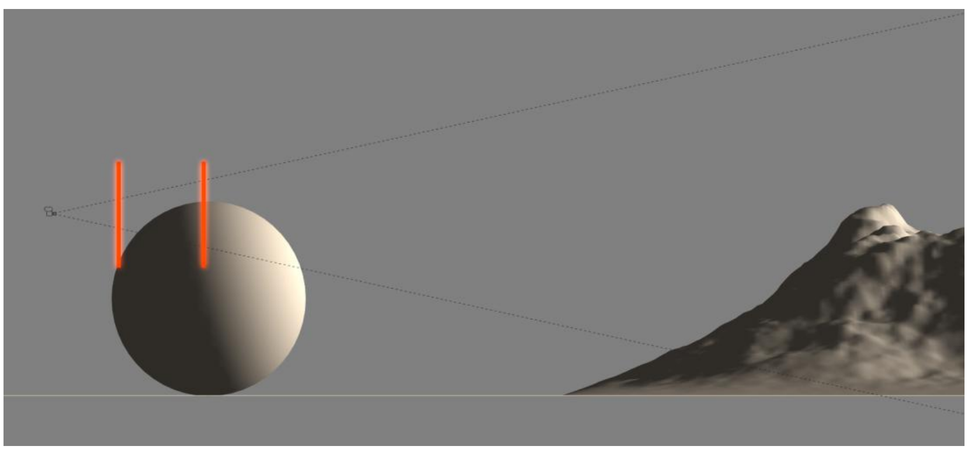
A common "mis-effect" often found in novice renders is bad depth-of-field. This rarely happens when you use the in-built Depth-of-Field effect in Vue, but rather when a Z-Depth map based post-process effect is applied in a paint program.

The source of the problem is not enough understanding how depth-of-field or depth perception works in a camera. Let's explore three common scenarios where depth-of-field is prominently visible. If you apply the depth-of-field of one scenario to another, it will alter the perception of your image. For example, if you use a Macro type depth-of-field to a wide angle image, it will make your vista look miniature instead of adding a feeling of depth.

Macro

In a macro (extreme close-up) situation, a very short and tightly focused focus range is set. Often parts of the subject itself beyond the tight focus range are blurred as well.

Everything other than the subject itself is mostly blurred as well. Because of the high amount of blurring, the distant objects are extremely blurred.



Example of the range of focus in a macro render.

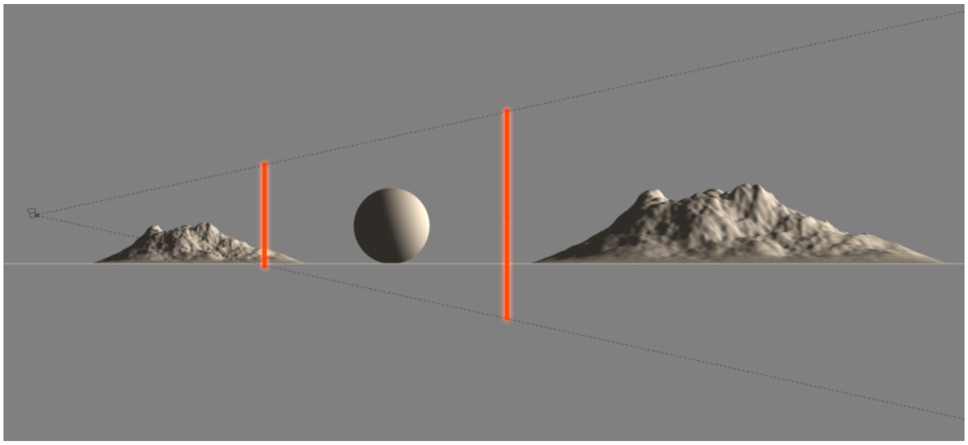
Macro depth-of-field is best used for character close-ups or showing small, detailed objects. A bad depth-of-field in this range (for non-macro renders) can often make your environments look excessively smaller than they should be.

A macro shot of needles of a plant. Notice how anything beyond the immediate tight focus range is blurred.

Telephoto

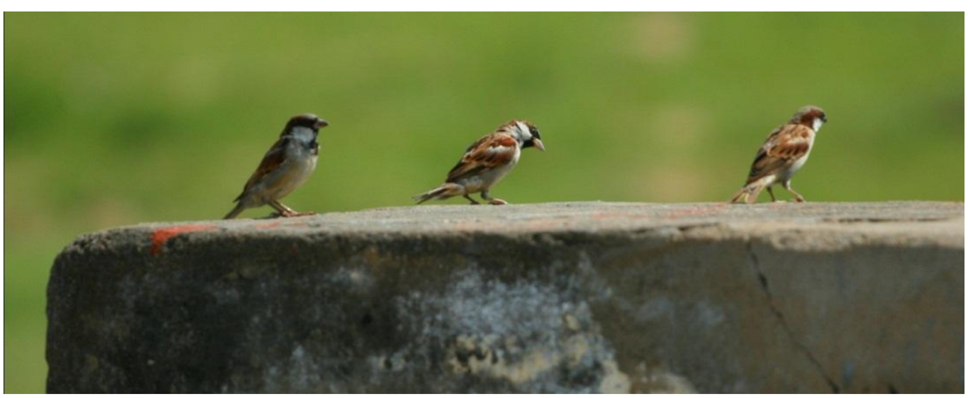
With a telephoto shot (70mm or more), the area between the camera and the subject, as well as the areas beyond the subject towards the horizon (infinity) are blurred too.

The focus range can be flexible and not necessarily as tight as the Macro scenario. However, remember that the greater the distance or Focal Range of the camera lens, the higher the blurring or intensity of the depth-of-field.



Example of the range of focus in a telephoto render.

Focus on the center hill while the foreground grass and the distant hills are blurred out.

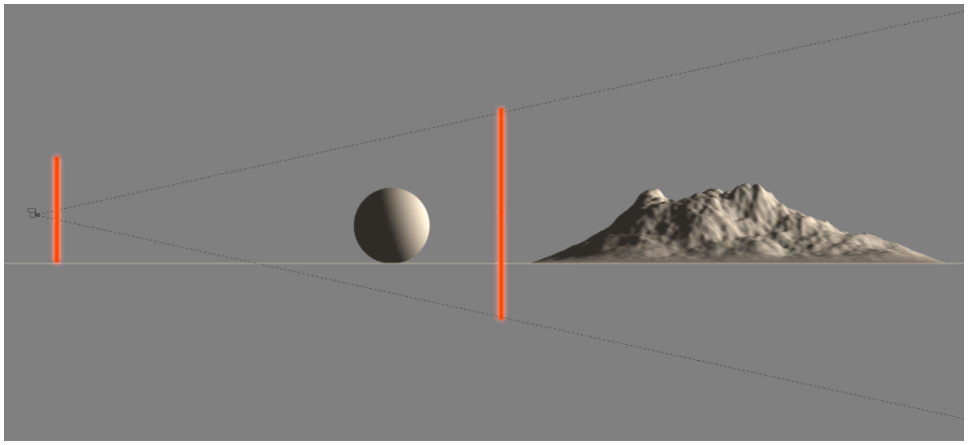


Telephoto depth-of-field is best used for showcasing a distant object, like a character or tree on a hill, a ship on the open ocean, etc.

Normal Distance

Unlike the Telephoto and Macro scenarios, Normal Distance scenes have very little, if any, depth-of-field.

Because of the nature of a wide lens, the camera does not distort the focus field and only the extremely distant objects are blurred.



Example of the range of focus in a wide render.

This wide shot has clear focus on the rock and only a slight blurring of the distant objects.

In most cases, using no depth-of-field blurring or very little blurring is recommended.

Avoid Scale-less Scenes

A ‘scale-less’ scene would be defined as a scene where the viewer is unable to figure out the size of the terrains, objects, etc. in the scene. In 3D renders, the size is of objects is relative unless they are compared to something of known size. Then the entire scene is set into perspective for the viewer. For example, if you have a terrain-only scene like the one shown below, you know that it is a large terrain but the exact size is not apparent.

We can easily add a human or a vehicle so as to show the scale of the scene. That would be a problem in situations where you want a natural environment only. By adding a few cleverly placed natural elements you can effortlessly show the scale. For this scene, we will add a rock in the foreground and some dead grass.

A foreground element in this scene adds a sense of depth and scale.

Black-and-White Renders

Creating black and white renders can be quite creative! However, the main concern when creating monochromatic images is to create realistic color intensities in the grayscale range.

A common practice is to excessively saturate, brighten, or change the color of the materials in a way that they come out more starkly as required in the final black and white image. The problem with this approach is that you are limited by your own perception of the world as black-and-white. So if you cannot match the colors exactly, a more seasoned artist, or someone with a better eye for monochromatic images may be able to see the image as being unrealistic.

The obvious solution is to design the scene as if it were supposed to be rendered in color. With this method, everything will be created to absolute realism and when the image is finally converted to black and white, it will seem that much more realistic.

You can decrease the Saturation option in the Post Effects dialog to have black and white previews of your render.

Realism through Simplicity

There is a devious aspect of Vue that can ruin your scene. Vue is fun! Like a sketch artist over-sketching, it is easy to add too much to a scene. Too many trees, too many elements (We already talked about this in ‘Counting Lines’), or even wanting unnecessary complexity where none is required.

Usually the best scenes you will see in Vue are very simple in nature. My personal favorite (and possibly every Vue users’ favorite) is the example scene *Cerro Verde* by Eran Dinur that you can find on the Vue disc. You will notice that the scene uses very standard terrains. The placement of the terrains, the materials and EcoSystem, and the lighting is what makes that deceptively simple scene look fantastic.

In the opposite effect, sometimes being able to have the ability to create an extremely complex object or terrain in Vue can make us forget that so much complexity may not be required. For example, if you have a distant standard terrain, then a 1024x1024 resolution may be enough. You need not go to 4096 x 4096. This will save render time and file size. Similarly, just because a HyperTerrain can be made up to 4 million polygons in density does not mean every HyperTerrain should be that complex. (On a side note, I always save my MetaBlobs before baking – you can easily go back and bake a denser model if needed) Most of the HyperTerrains you have seen in this book are around 1 million polygons.

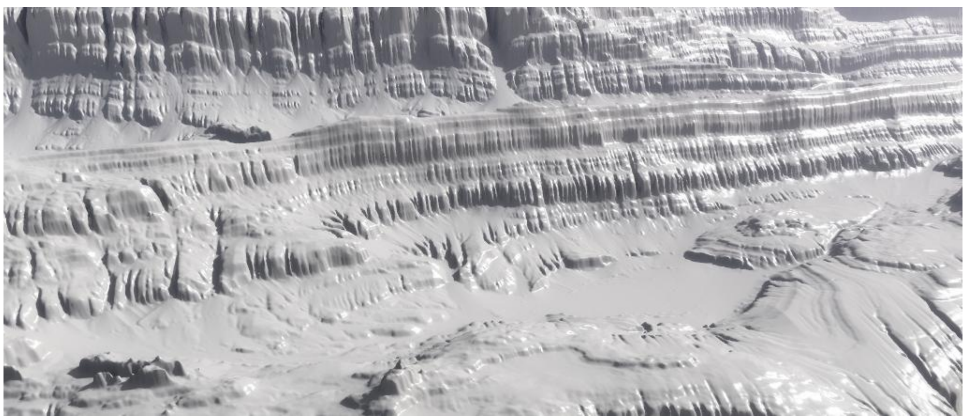
All the resolution, polygons, and render time we save in these medium polycount models can be put to better use – in materials! The complexity should be in the material, not necessarily the model. Use more layers, more fractals, more details in your material. This will paint over the seemingly low-polygon models and create striking renders!

Let’s take a look at some examples. These not-so-high polygon density models use materials to their advantage. Remember what we discussed in Chapter 4: Materials add extra shape perception to all objects, changing how something appears completely.

This HyperTerrain may not look like much when it has a flat material on it, but with a complex material like QuadSpinner's Mineral Infinity, it takes on a whole new look.

Instead of a computer devastating EcoSystem of billions of grass instances, this terrain uses a high bump made of small scale noise to create the look of distant grass fields.

A procedural terrain was used to make this scene. However, since it was to be covered in a dense tropical rainforest EcoSystem, it was converted to a 512x512 Standard Terrain saving rendering time without loss of quality.



This extremely low resolution 1024 x 1024 Standard Terrain looks like a plastic model, but when we add a complex eight layer material on it, it takes on a different look.

The Subtle Touch of Plants

Sometimes adding just a hint of plants can change the entire mood of the scene. Clearly, adding plants suddenly to a scene may not be practical as it may affect the objects in the scene. This is even more troublesome if the scene has been completed.

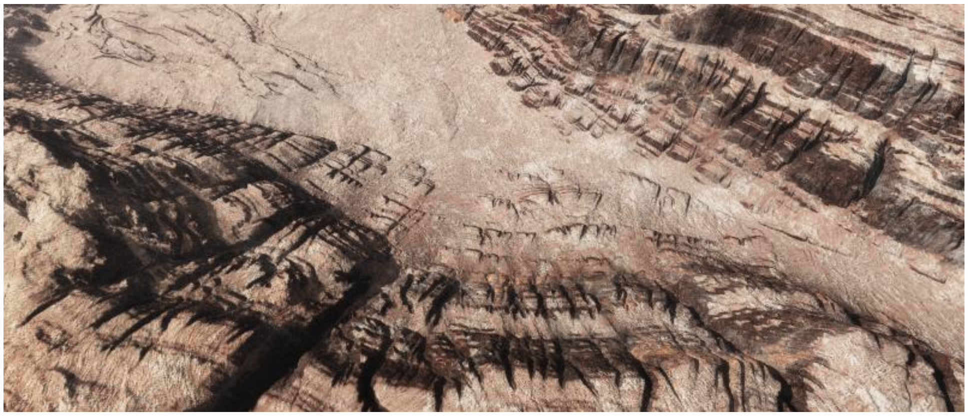
In such an event, you can add a very short (vertically squashed) terrain and place it just outside of the camera frame at the bottom. An EcoSystem populated on this terrain would be visible in the frame and create a sense of lushness in the scene. The image below depicts such a situation.

Additionally, you can place an individual tree or shrub in a corner or elsewhere in the scene to add the sense of vegetation existing all over the scene without actually having to add an EcoSystem to the main scene elements.

Unleveled Cameras

By default, Vue cameras have 'Always Keep Level' turned on. You can see this option in the Object Properties panel as shown here.

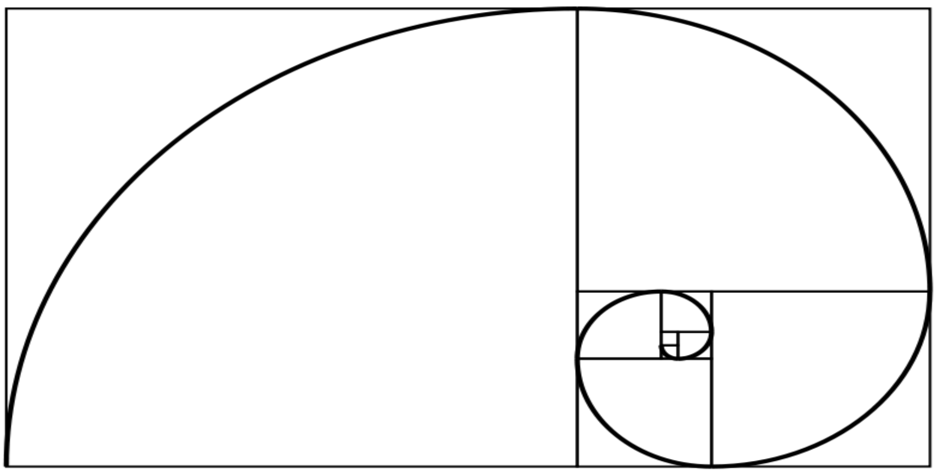
When you uncheck this, you can rotate – or more specifically: roll – the camera. This can help you create a diagonal horizon. Free rotating cameras can help you find creative compositions. It can also help you overcome terrain 'walls' where you may want to show objects behind the 'wall' in question (as seen in the second image below) without sacrificing the view of the 'wall' itself.



One of the biggest helpful features of having a slightly tilted camera is that it creates a sense of “eye sight”. When we tilt our heads (which is a *lot* of the time!) our own eyesight is not exactly leveled. Our brain processes the tilting of objects and compensates for it. When you have a slightly (less than 15 degrees) tilted camera view in your render, the brain will process the image similarly, which hints to our subconscious that this might be a real view. And of course, it sets it apart from the typical perfectly leveled renders out there.

Golden Section

One of the most famous and useful concepts for artists is the Golden Section or Golden Ratio. In mathematics and the arts, two quantities are in the golden ratio if the ratio of the sum of the quantities to the larger quantity is equal to (=) the ratio of the larger quantity to the smaller one. That might sound a little technical, so let's just take a look at the Golden Section image below. In rough estimated terms, the image is divided into 1/3 and 2/3 portions, with the larger (2/3) portion divided again into 1/3 and 2/3, etc.

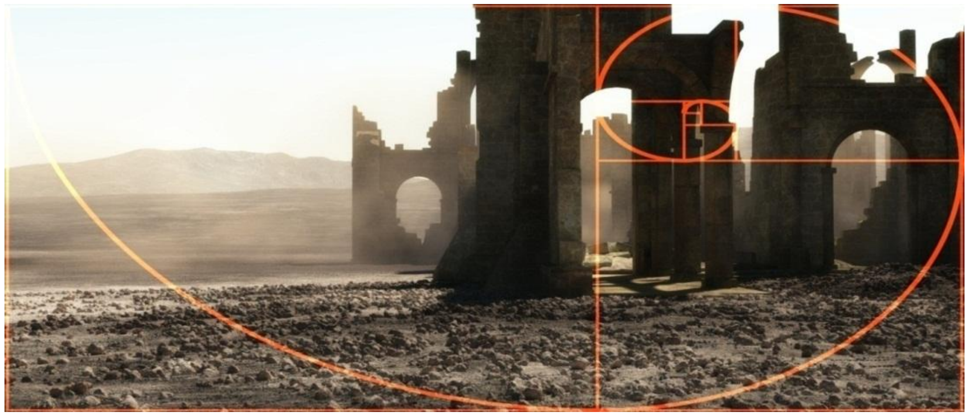


Golden Ratio with a spiral following divisions of the image is known as a Fibonacci Spiral.

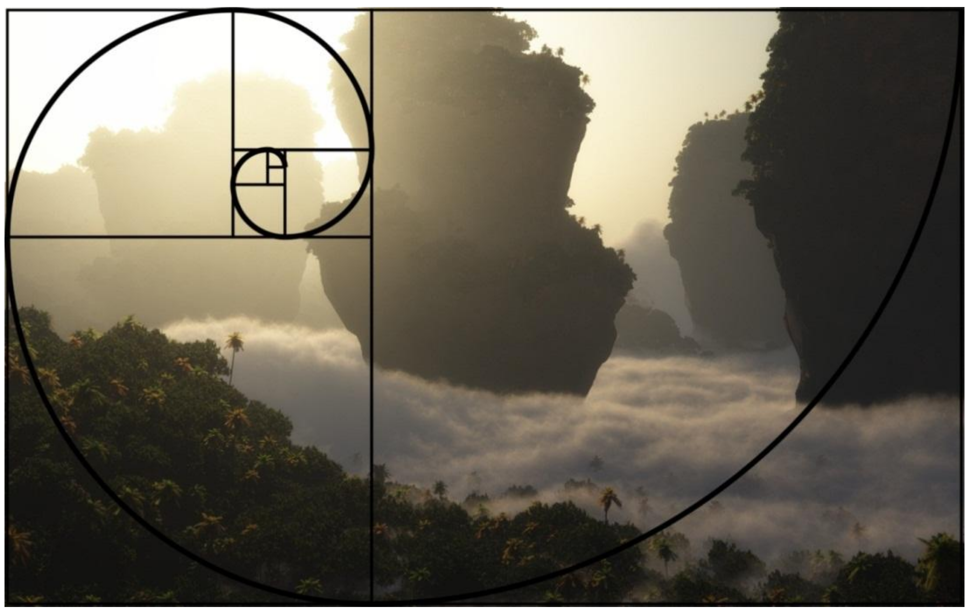
We will be using the Fibonacci Spiral to understand how to better compose images. When you use this method to compose your images, the render automatically becomes pleasing to the eye. Never underestimate the power of composition. A good image of quality can look bad if it is not composed properly, and a low quality image can still look good if it has a proper composition.

Artists through the ages have used the Golden Section to create masterpieces. Leonardo Da Vinci made it very famous through his 'Divine Proportion' image.

If you apply the same principles from the start when create a new scene in Vue, you can be assured to have a properly composed image. This can be done in multiple ways: density, light/color, position of the main subject, or combinations thereof. Let's see a few examples to better understand this.

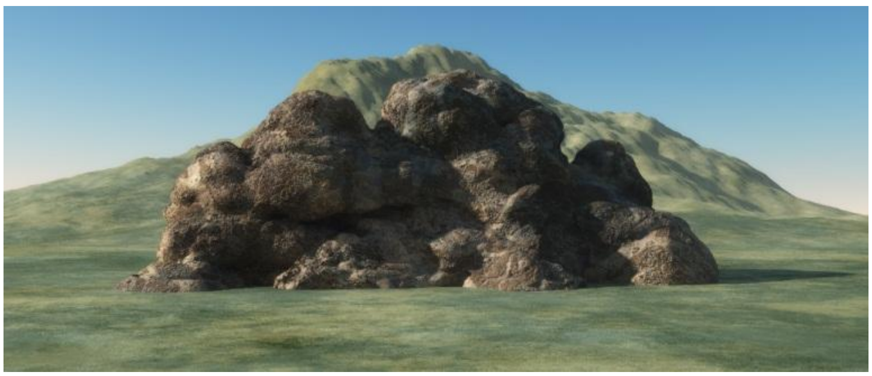


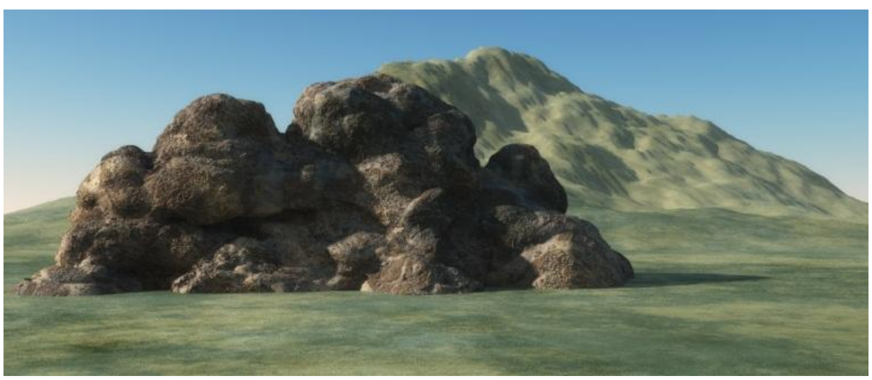
This image features a barren landscape which is sparse when you look at the number of elements. However, at the spiral point of the Golden Ratio, the density of the number of elements becomes very strong. This automatically draws the eye to that specific point. The main subject (the ruins) is also placed at the spiral core.



Conversely, in this image the main subject is in the center, which is generally not advisable under the Golden Ratio rule, however we still the Golden Ratio effect controlled by Light rather than the subject, which draws the attention to the main subject without betraying the aesthetics of the composition. Sometimes the lighting can be changed for better composition if the subject cannot be moved.

In this image, the position of the main subject – a character – is placed at the spiral core of the Golden Ratio. This leaves room for creative and non-Golden-Ratio composition of the background, and still keeps the overall composition completely intact and attractive.





Finally, a centered image (top) compared to a Golden Ratio composition.

When dealing with multiple objects, the scene should be composed with care so that they do not come across in a linear line. They should be juxtaposed like the rock formation and the mountain in the image above. This can go a long way to achieving a truly beautiful render.

Image Flipping

Often, after a render has been completed, you can try one very small but effective method is to flip the rendered image horizontally. It may not always work, but sometimes you may find a flipped version to be more aesthetically pleasing.

A case in point: the top image below was the actual render. It was flipped and published as seen in the second image. The flipped image (bottom) resulted in better composition.

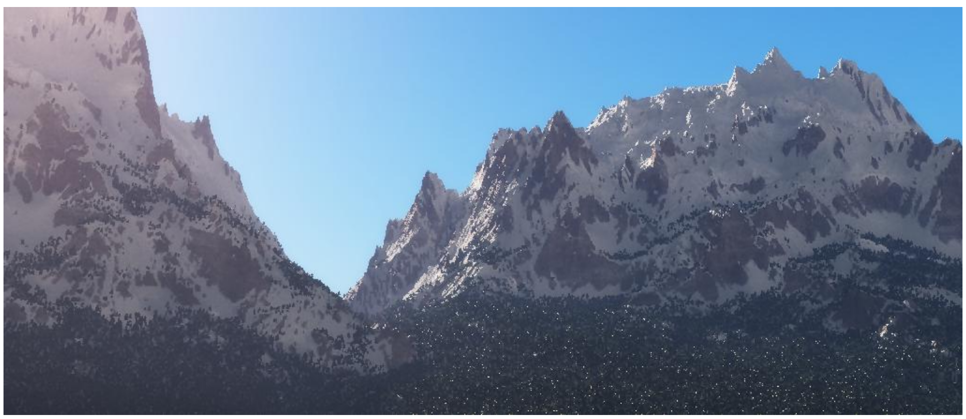
High Quality with Previews

Think of this as endurance training one goes through in a physical fitness regime. The simple concept is: if you can get your render to look good at Preview Quality, it will always look good at Final or higher quality. If we look at the physical training metaphor, if you lift 40 kilos every day, picking up a box of 20 kilos looks very easy.

Of course, for details such as textures, or sampling the antialiasing of items in the scene, a final quality render may be required. However keeping yourself restricted to a Preview quality for renders will help you become stronger in both your Vue skills and general artistic skills as well. If needed, try area renders of small portions to see how things appear, but not for the entire scene unless absolutely necessary.

One of the biggest skills you will acquire (or enhance if you already possess it) is approximation. You will be able to tell how something will look like in a final render by just looking at a preview render. When you are short on time for urgent projects, it will help you immensely. In general work, you will be able to cut down on your render time by not having to do high quality renders.

The image below was rendered at Preview quality.



Massive Terrains

Sometimes Vue users are afraid to make a seriously massive terrain, instead depending on Aerial Perspective, vertical stretching, or some other aspect of Vue as a 'crutch' to make the image look larger than it is.

However, if you truly want grandness to show in your image, make your terrain extremely large – especially the far away terrains. Do not be afraid to move out of the grid and circle "world" shown in the viewports.

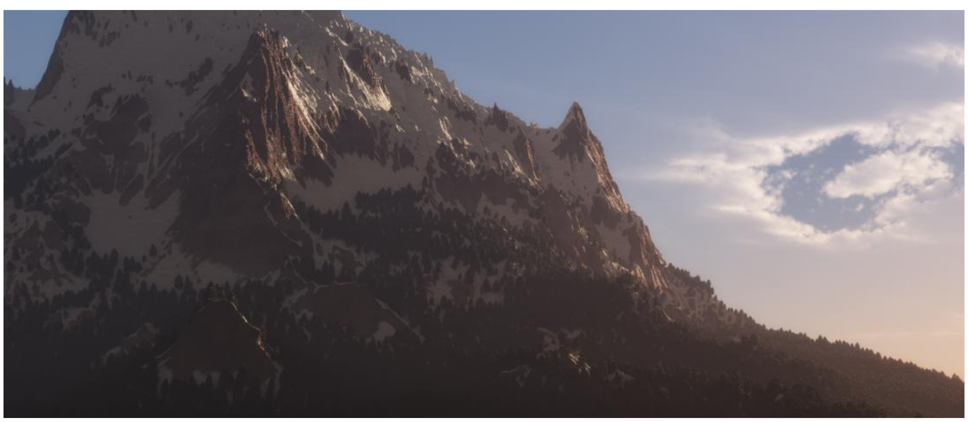
For example, the image below has a 4 x 4 kilometer terrain in the foreground. The background terrain is also 4 x 4 kilometers in the first image, but it does not look grand enough. We resize it 8 x 8 kilometers and it achieves the sense of grandness that was missing from the first image.

In this image, a single terrain is sized to about 12km which would be a realistic size for such a large landscape.

The same terrain in the foreground which is crisp and clear can also be seen curving out into the background several kilometers away, accumulating haze and depth of perception as it goes farther away from the camera.

There is simply no alternative to display grandness without making the terrain actually grand itself.

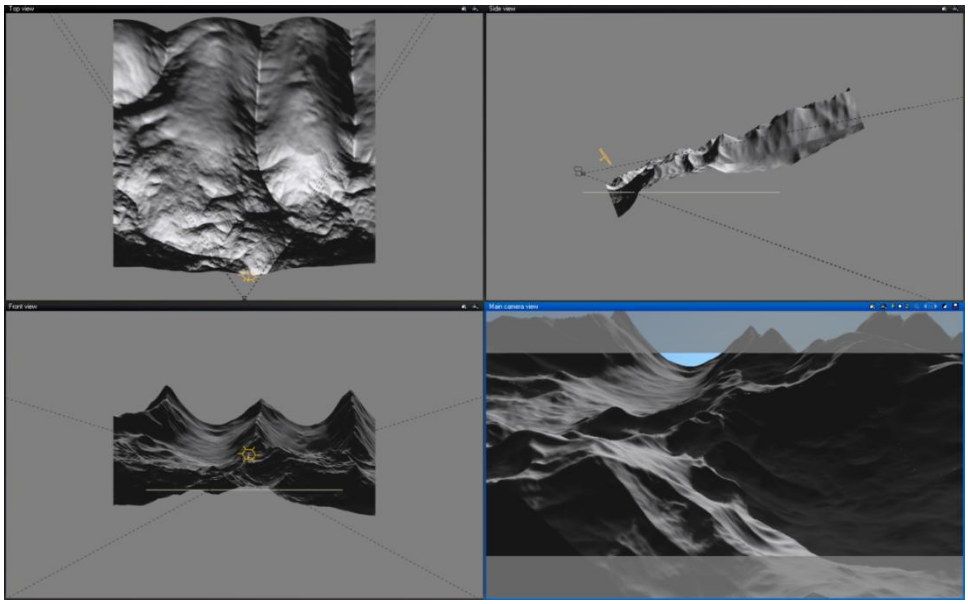
If even after making your terrain several kilometers large, you still feel you are missing depth then the Haze, Aerial Perspective, and Light Balance should be consulted and tweaked as we discussed in Chapter 2.



Massive Tilted Terrains

A mountain in Vue can be literally as large as you want it to be. With Procedural Terrains you can have infinite amounts of detail no matter what size the terrain is. But sometimes you may wish to show your terrains to be not simply large, but massive!

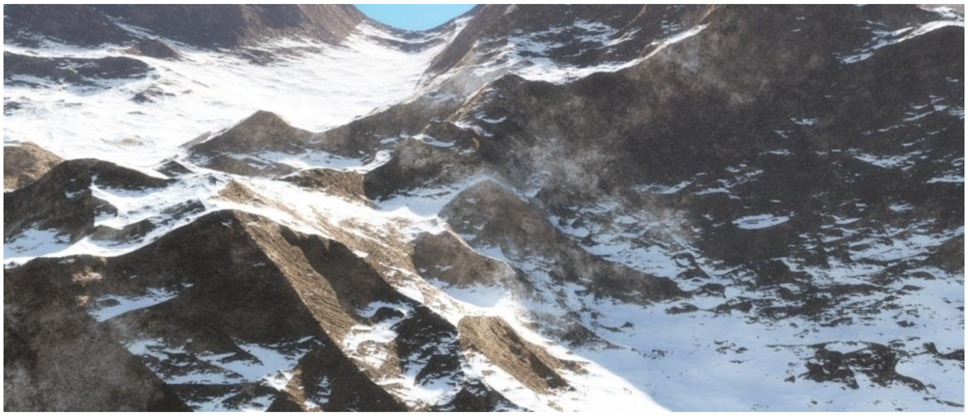
Creating a looming mountain is not too difficult. Let's make one:



The screenshot above shows a standard 'Peak' preset style Procedural Terrain that has been titled 20 degrees on the X axis (Pitch). It has been scaled and placed so that the side edges are outside of the render frame. The top edge cuts off abruptly but since the shape of the terrain has sharp ridges, it looks like part of the design.

If you are using EcoSystems, remember to populate it *after* you tilt the terrain otherwise the environment factors may be slightly different and the population may not reflect what you had intended.

This technique works well if there is another foreground layer, like a HyperTerrain or some other model that is the focus of the scene and is closer to the camera. Any minor imperfections of our titled terrain can be safely ignored as the viewer will be too focused on the foreground. Some minor depth-of-field, depending on the closer object's distance to the camera, can help overcome the imperfections of the background terrain even more.



In this render you can see the effect of our titled terrain. While not a 'physically accurate' terrain as the farther side is up in the air, but still a good backdrop!

Eye Level

If you'd like the viewer to feel a stronger connection to the image they are looking at, you might want to put your camera at Eye Level. This is usually somewhere between 1.5 to 3.0 meters from the ground.

This eye level placement helps create a sense of grandness for the scene. In real life, we look at a tall building or a large mountain from the ground by looking up and that is programmed into our brain as a sense of grandness. The same building or mountain would not look as grand if you saw it from the air in an aircraft.

Staring at the Sun

One common trend you will see, especially in novice Vue renders, is to have a Sunset or Sunrise position for the Sun and have it pointing straight at the camera. It is a common misconception for novices to think having the Sun in the shot so starkly can make the image look dramatic.

In reality, while this concept may work in certain scenarios, having the sun appear so prominently in your images may make your gallery or portfolio appear too predictable and mediocre.

Try experimenting with new Sunlight angles where the Sun isn't visible in the frame. This will help you grow your lighting skills and make your images more interesting.

Antialiasing in External Software

One way to quickly render images with low anti-aliasing and still retain good quality is to resample the render to a lower size in a photo manipulation or paint program.

For example, if the desired final size is 2000 pixels wide, and the desired antialiasing is of 'Broadcast' quality then you would render using 'Final' quality render settings at 2500 pixels wide. The output should be downsampled to 2000 pixels. This will provide a good enough quality of texture and object antialiasing.

This technique is useful for creating fast mockups for a project. The advantage comes from the Final render quality which is faster to render than Broadcast since there is no texture antialiasing taking place.

Below is a magnified quality comparison of a render created with the settings mentioned above.

'Extra' resolution Final render (top left) magnified to 500% (top right), compared with the downsampled version (bottom left) and magnified 500% (bottom right).

Avoid Overkill

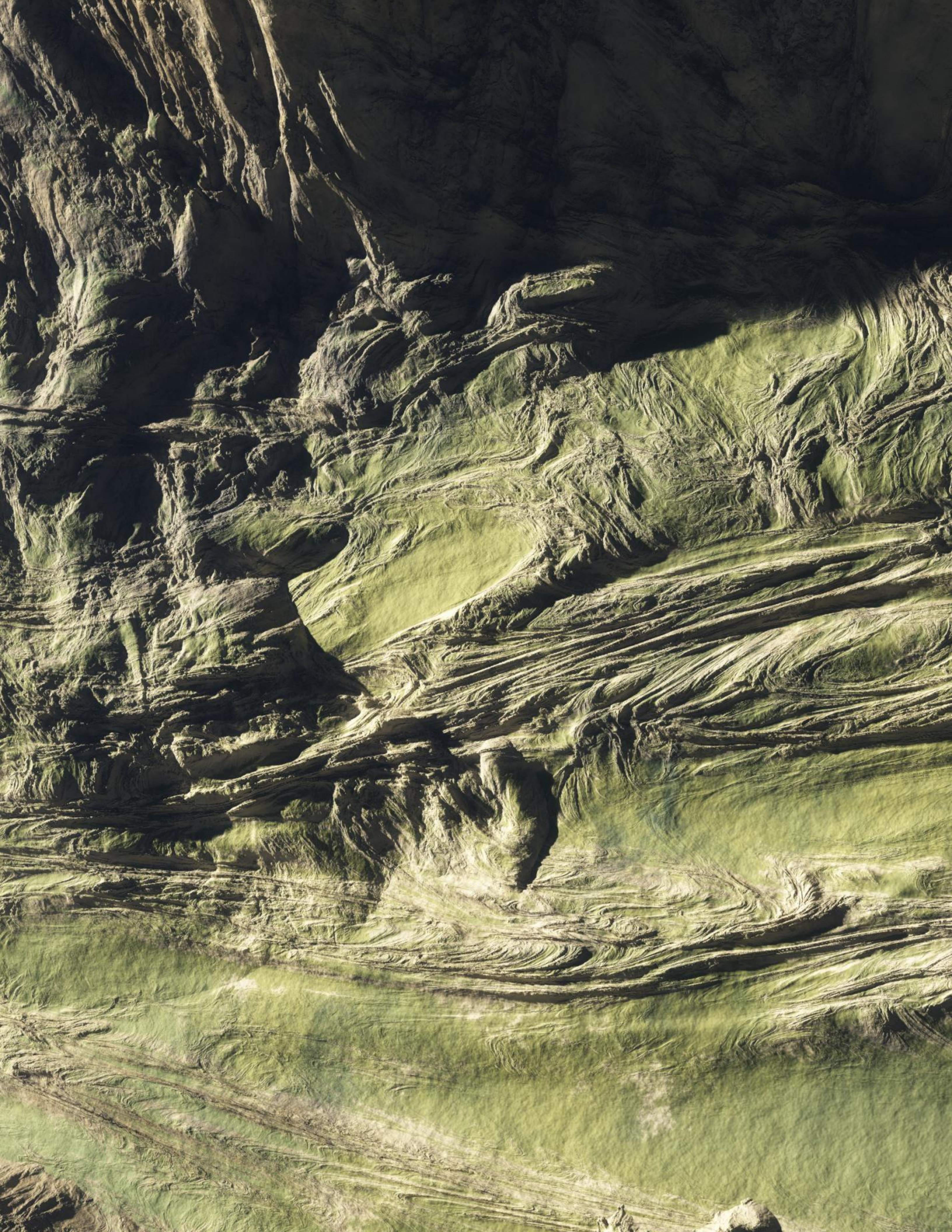
When training or helping Vue users, a common complaint I've heard is "My render didn't turn out good, even though I rendered it at 4000 pixels and Ultra!"

While this is not a blanket rule that may apply to all scenes, it is a common enough scenario in my experience: the problem here is overkill. A render will not look good just because the render settings were pushed to the maximum and the resolution was enlarged. It may look cleaner and sharper, but it will not make it more realistic.

The time and effort put into rendering at such extreme settings are possibly better spent in creating richer materials, stronger atmospheres, and lots of testing of the light. By reallocating energy into these tasks, the render will automatically become better – no matter what render settings or resolution you use.

Also, keep in mind that the larger the output resolution the more mistakes or "CGI" will be apparent in your render. If your resolution is large, then your fractals – both for shaping terrains or HyperTerrains and for creating the materials – will need to have a massive amount of complexity in them. Remember: the larger the output resolution, the higher the required complexity.

Going to Ultra render settings is also not required all the time. In most cases, Broadcast can suffice if you work hard on the actual content and quality of the scene elements. Refer to the Appendix of this book for tips on how to better customize Render Settings to save time while still achieving quality results.

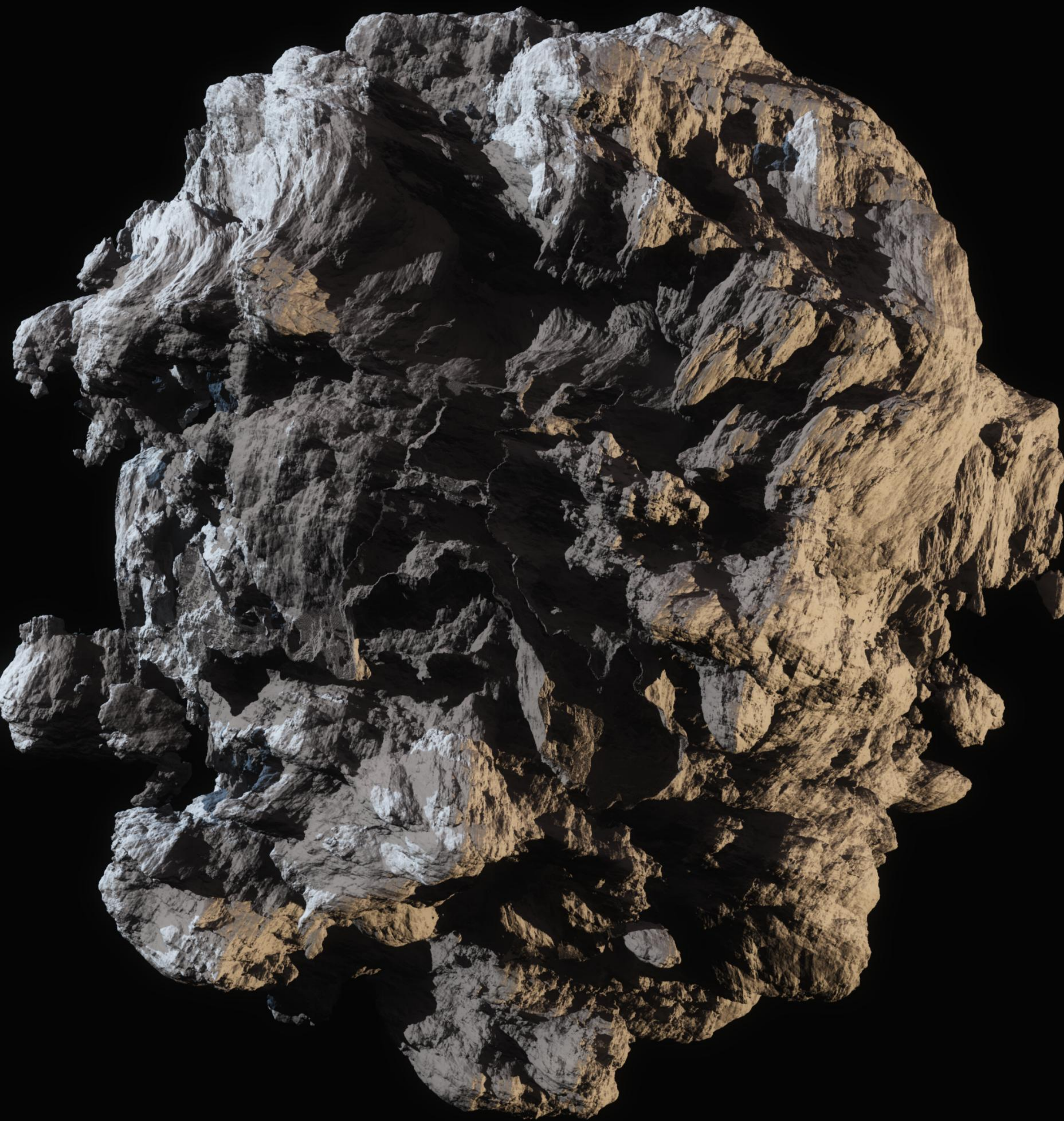


Let the renders begin...

Here, we end our discussion and hit the render button.

You have just explored some of the most pivotal discoveries gained through several thousand hours of experimentation in Vue and study in the field. These core concepts can now be expanded upon by your own trials and research – whether through books, photography, or nature herself.

Thank you for reading this book, and may your renders always be short! :)



Precision Antialiasing

Appendix A

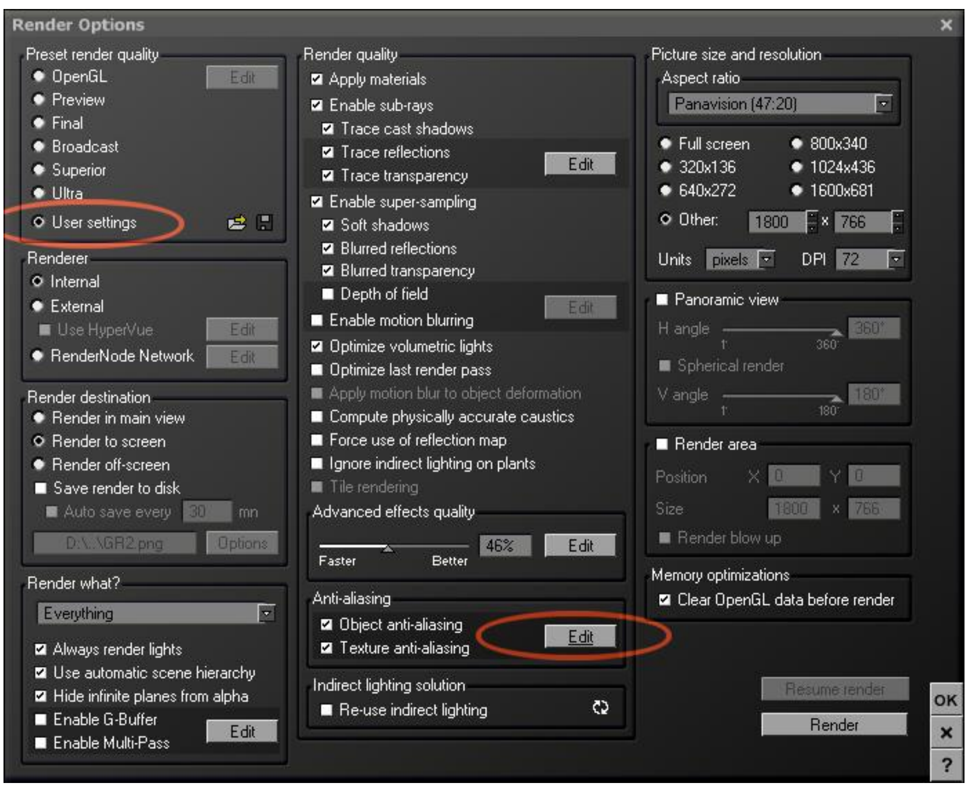
Customizing Render Settings

To achieve better results from Vue, sometimes it is required to go beyond the preset render settings (Final, Broadcast, etc.) and try customizing Vue's render settings. This is often overlooked by many Vue users, resulting in subpar quality in their renders. Sometimes the user will needlessly go to settings like Superior and Ultra, which may not really be required for the scene as such results can be easily achieved by some minor customization of the defaults.

Antialiasing is one basic area that needs customization for most scenarios. This is where almost all detail calculations come from. In Vue the antialiasing is divided into two categories: object antialiasing and texture antialiasing.

Object Antialiasing is for smoothing the edges of objects and the boundaries of shadows. The higher the antialiasing the more soft and 'integrated' the edges appear, and the shadows appear less grainy.

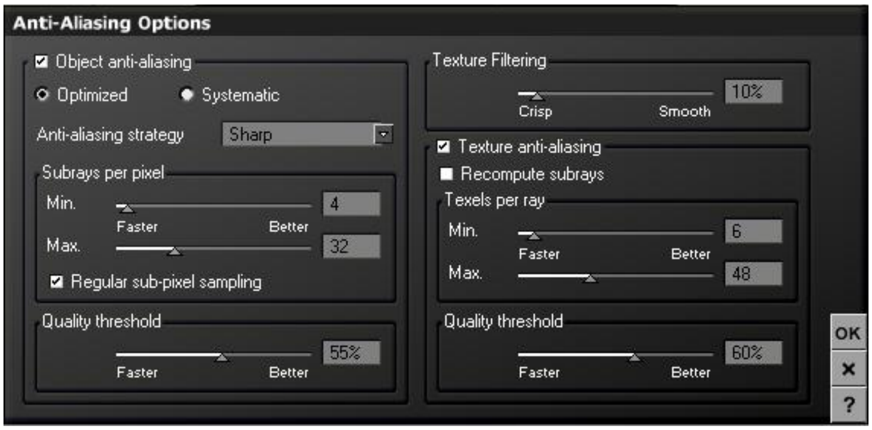
Texture Antialiasing goes beyond Object Antialiasing by super-sampling (simply put, cleaning up and smoothing) both bitmaps and procedural textures. This is most effective for cleaning up noise in textures. In Chapter 4, Materials and EcoSystems we saw details on intentionally creating noise as the real world has a lot of noise and moiré. However, to also keep the noise from looking jagged or pixelated, a good Texture Antialiasing needs to be used.



To enable custom settings, select ‘User Settings’ instead of one of the preset modes in the Render Settings dialog. Use the Load User Settings button to load up the base settings of any presets so you are not required to manually set everything.

Once you have loaded your base settings, you can click the ‘Edit’ button in Antialiasing.

Object Antialiasing



Supersampling

You can choose between Optimized and Systematic antialiasing strategies.

Optimized

This supersampling method works faster and is recommended for most situations. It will intelligently decide which areas of your objects need antialiasing and to what degree.

Systematic

This supersampling method does not use selective antialiasing, but instead will supersample every single pixel in the render. Systematic antialiasing may provide smoother results, but it will occupy more memory and take longer to render.

Antialiasing Strategy

There are multiple anti-aliasing strategies you can call upon depending on your scene.

Automatic

Depending on whether you are rendering a still shot or an animation, this will automatically select Sharp or Soft respectively.

Crisp

Crisp is the recommended and most accurate antialiasing strategy. It requires a lot of antialiasing to remove the noise produced by this strategy; however the results are most impressive.

Sharp

Sharp is best for producing stark details. It works like Crisp but produces less noise.

Soft

Soft is used for animations but can also prove useful for still renders. This is useful when creating cloud renders where you may have too much noise which can be eliminated by this antialiasing strategy.

Blurred

Blurred is not widely used, but can be useful for certain render scenarios where you do not require detail – like a distant background matte painting or skybox, for example.

Subrays

Put simply, Subrays are the rays of light fired by the render through the light source (or sources) – as they hit (or bounce around) objects, they calculate the color (and subsequently, texture) of the object. Just like the resolution of an image, the higher the number of rays, the better the quality of your render.

However, unlike an image's resolution, Subrays are not a fixed number unless you choose to manually force them.

Min and Max

These two sliders represent the minimum (or initial) number of Subrays and the maximum number of Subrays. Initially, Vue will fire the "Minimum" amount of Subrays. If the object's antialiasing is not good enough with those initial rays, Vue will keep firing Subrays until the quality improves, or it hits the Maximum number of Subrays allowed.

Do note that just because the Maximum is set to 1000, it does not mean it will send 1000 Subrays. If 4 Subrays suffice, Vue will not fire more than that.

Quality Threshold

While Vue's renderer is able to decide intelligently whether more Subrays are needed or not, the bias for that calculation is controlled by the Quality Threshold. The higher the Quality Threshold, the more Vue is compelled to send higher numbers of Subrays.

Remember, if you are increasing the Maximum Subrays, then it is better to keep the Quality Threshold conservative (60% or less) as just the increase in the Maximum will automatically cause more Subrays to be used.

Regular Sub-Pixel Sampling

When this option is selected (which is the normal, recommended action) Vue will fire subsequent Subrays at the same location where the initial Subrays were sent. This works well in most situations. However, if you have a scene where a normal pattern is being stretched a lot or even to infinity, and therefore creating distortions, unchecking this may cause the render to improve. When unchecked, Subrays are fired at random.

Texture Filtering

Texture Filtering allows you to ensure whether you want your material textures to appear crisp or smooth, and to what degree.

Texture Antialiasing

Texture Antialiasing works just like Object Antialiasing, and the Texels per Ray work similar to Subrays. The main new feature here is Recompute Subrays. When this is turned on, Texture Antialiasing is applied to reflective and refractive objects. However, if your scene does not have any major reflective or refractive objects or materials, it is best to keep this option turned off. The Quality Threshold also works exactly as with Subrays, and the same caveat for using conservative values with a larger Maximum number applies here as well.

Per-Object Texture Antialiasing

You can use the TAA Boost for a specific Material in your scene if you do not wish to apply Texture Antialiasing to everything in your scene. However, you will still need to turn on Texture Antialiasing (not available in Broadcast or below) for it to take effect.

Antialiasing Tactics

Let's discuss three essential Antialiasing tactics developed from over 2000 render experiments. Each has its own advantage and disadvantage, and of course, some tradeoffs. So use them as you see fit, and always remember to experiment on an area render (keeping the overall render size to the final resolution) to make sure the tactic fits your scene's needs.

Orthodox

Subrays set to Minimum 4 and Maximum 16. Texels set to Minimum 4 and Maximum 16. Quality Threshold between 45% and 60%.

This results in the traditional style of Antialiasing and may prove useful in most situations, especially with renders below the size of 1200 pixels.

Rogue

Subrays set to Minimum 4 and Maximum 256. Texels set to Minimum 4 and Maximum 256. Quality Threshold set to 45%.

This tactic allows for greater quality but may cause slower renders compared to the Orthodox method. This can work well (and fast!) for both small and large renders.

Blunt Force

Subrays set to Minimum 36 and Maximum 36. Texels set to Minimum 36 and Maximum 36. Quality Threshold set to 80%.

This extremely slow rendering tactic promises impressive quality as antialiasing is forced to use the maximum number of Subrays and Texels. To experiment, you can use any number in the Min and Max fields, as long as they are the same. Blunt Force tactics are good for small renders, or if you have massive rendering capacity – like a large render farm – otherwise your renders may literally take days to complete.

Large Scale or Cloud Renders

Often with renders above 1600 pixels, or when rendering just clouds (regardless of resolution), a "Soft" anti-aliasing strategy can result in greater image quality.

Index

Product Index

The following QuadSpinner products were used in the making of this book.

- Mineral Infinity
- Master Trinity (Rock X, Rock Y, Rock Z)
- Natural Monuments HyperTerrains
- Supernoi HyperTerrains
- QuadSpinner Material Development Kit (MDK)
- Arenite Warriors
- The Dark Knights
- The Defenders
- Distant Frontier
- River Rock and Sand
- Mt. Cornucopious
- El Gordo
- Sunbeams
- Procedural Terrains Pack
- Tropical Vale
- Sculpted Clouds
- Dark & Stormy
- Liquid Skies
- Summer Dreaming
- High Altitude Clouds
- Sunset Magic
- Volcan Arsolis
- Grasslands
- Toffee Delight
- Various terrain products
- Various individual scene products

All QuadSpinner products can be purchased from **www.QuadSpinner.com**