8-2011

Combining Procedural and Hand Modeling Techniques for Creating Animated Digital 3D Natural Environments

Hubert Smith
Clemson University, huberts@g.clemson.edu

COMBINING PROCEDURAL AND HAND MODELING TECHNIQUES FOR CREATING ANIMATED DIGITAL 3D NATURAL ENVIRONMENTS

A Thesis
Presented to
The Graduate School of
Clemson University

In Partial Fulfillment
Of the Requirements for the Degree
Master of Fine Arts
Digital Production Arts

---

by
Leopold Smith
August 2011

Accepted by:
Dr. Timothy Davis, Ph.D. Committee Chair
Dr. Donald House, Ph.D.
Tony Penna, M.F.A.

ABSTRACT

This thesis focuses on a systematic solution for rendering 3D photorealistic natural environments using Maya's procedural methods and ZBrush. The methods used in this thesis started with comparing two industry specific procedural applications, Vue and Maya's Paint Effects, to determine which is better suited for applying animated procedural effects with the highest level of fidelity and expandability. Generated objects from Paint Effects contained the highest potential through object attributes, texturing and lighting. To optimize results further, compatibility with sculpting programs such as ZBrush are required to sculpt higher levels of detail. The final combination workflow produces results used in the short film *Fall*. The need for producing these effects is attributed to the growth of the visual effect industry's ability to deliver realistic simulated complexities of nature and as such, the public's insatiable need to see them on screen. Usually, however, the requirements for delivering a photorealistic digital environment fall under tight deadlines due to various phases of the visual effects project being interconnected across multiple production houses, thereby requiring the need for effective methods to deliver a high-end visual presentation. The use of a procedural system, such as an L-system, is often an initial step within a workflow leading toward creating photorealistic vegetation for visual effects environments. Procedure-based systems, such as Maya's Paint Effects, feature robust controls that can generate many natural objects. A balance is thus created between being able to model objects quickly, but with limited detail, and control. Other methods outside this system must be used to achieve higher levels of fidelity through the use of attributes, expressions, lighting and texturing.

Utilizing the procedural engine within Maya's Paint Effects allows the beginning stages of modeling a 3D natural environment. ZBrush's manual system approach can further bring the aesthetics to a much finer degree of fidelity. The benefit in leveraging both types of systems results in photorealistic objects that preserve all of the procedural and dynamic forces specified within the Paint Effects procedural engine.

DEDICATION

This thesis is dedicated to my wife and best friend, Sandra, for her support throughout my program of study. I am forever grateful.

ACKNOWLEDGMENTS

First, I respectfully thank and acknowledge my advisor, Dr. Timothy Davis, for his guidance and support. With your eminent assistance, I was able to conquer my vision for a quality thesis. Thank you for providing me with rich experiences that afforded me the opportunity to strengthen and broaden my skills. Thank you most importantly for imparting your expertise.

I would also like to express my gratitude and sincere appreciation to my committee members for their support and encouragement. Specifically, I want to thank Dr. Donald House for his expertise, feedback, and recommendations. Additionally, I want to thank Mr. Tony Penna for his assistance during my program of study.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

With the high demand of computer-generated imagery (CGI) in film, television, and gaming, producing any given digital effects sequence within the constraints of time and budget, while also delivering a high degree of visual quality, is critical to success. Procedural systems help alleviate some of the overhead by generating a certain amount of modeling. In creating a natural environment scene, automatic generation of flowers, plants, trees and rocks helps the artist save substantial time and establishes a polygonal base on which to build. Unfortunately, the results of generated natural 3D objects are often of rudimentary forms. For example, a single selected tree preset may contain a high branch count with plenty of leaf-textured planes; however, it is likely to be low-polygonal with low-resolution procedural textures. The same example can apply to a plant where each branching element builds to a rudimentary form starting from the stem leading to the branches and leaves. With the addition of low-resolution textures, the quality of such a model cannot hold up in close view of the camera. For high-end results to occur, substantial time and effort are needed to bring the models up to better standards.

The goal is to take advantage of the rapid generation of what procedural systems offer and combine the results with methods designed to enhance the fidelity of all objects produced. Using refining methods on generated base objects will result in noticeably higher quality when compared to non-refining methods. Also, after refinement there is a benefit of sustaining procedural animation enacted through forces. The benefit is further increased through minimizing efforts in rigging and deforming refined models or through the combination of these efforts with forces.

Organic objects are generated to populate a natural environment scene and show how refinement methods can produce a higher quality landscape different from using the procedural process alone. The populated objects are selected from Maya's Paint Effects Visor and painted to custom modeled terrains. The environment setting is a forest plain with painted trees and grass. All elements in the scene are set to react with forces to show natural motion. Subsequent steps involve custom textures and ZBrush hand-sculpting techniques to enhance model fidelity while still sustaining animated effects. Later results will be applied to a short film, *Fall,* using the same processes outlined in research. The need for this approach is the production of animated 3D natural environments where the surrounding scenery serves to aid performance for main characters.

Often visual effects involve creating complex natural environments that set the stage for character performance. Films such as the *Pirates of the Caribbean* series and *Avatar* used procedural systems to generate and design much of the digital environment composited with live actors. These production pipelines used packages like Vue and Maya to minimize much of the overhead in creating an abundant amount of environment geometry. Common within these two applications are processes that quickly predetermine model form and appearance through L-system and fractal functions. Where Vue is defined as a standalone procedural-based application with L-system and fractal function integration, Maya's Paint Effects includes both integration types that benefit from its inherit components: modeling, animation, dynamics and rendering to customize objects from their preset appearance. Unfortunately, from an artist perspective, both L-system and fractal-based computations lack intuitiveness for controlling results, but are efficient for establishing a productive base.

L-systems, or Lindenmayer systems, were created during the 20^{th} century by Hungarian biologist Aristid Lindenmayer to mathematically describe the growth patterns of organisms such as yeast and algae. Early L-systems used simple algorithms to create organism patterns while later algorithms were influential in the development of complex computer graphics and artificial life [KELLER10]. The complexity derived from later algorithms produce digital natural environments containing abundant detail with enriched and seductive characteristics.

Within Vue and Paint Effects, L-systems provide important functions for creating natural digital elements such as trees, plants and flowers. L-systems comprise the form of these elements through mathematical algorithms that recursively run to create branching segments of geometry. The results of repeated application of different organic types can come together to produce a variety of ecosystems.

L-systems are routinely used for leveraging the situation of limited time and manpower for hand modeling organic objects for later use in complex natural scenery. For this reason, production companies organize their workflows to include specialized applications with some level of L-system functionality. Base modeling is an area where L-system processes excel to quickly populate a scene with instances of user-selected vegetation objects. Examples include L-systems inside Vue and Maya's Paint Effects that invoke fast algorithms to build multiple organic presets applied either by global or painted means.

Procedural modeling is directed at representing one to many segmented objects created in a progressive phase. In the case of nature, this would constitute the characteristic of branching. For example, when creating a tree brush stroke in Paint Effects, each tree object references an assigned vector curve that specifies the magnitude of geometric segment growth. Any specified growth parameters are important for determining structural design of multiple parts from main stem to higher-order branches. Modular algorithms determine origin and axes data for producing sequential cylindrical structures along a curve that are separated into nodes of a certain length and size [DEUSSEN04]. The specification of procedural attributes within L-systems and Paint Effects generate graphical entities that are defined and manipulated using function, curve, surface and material editors.

Fractal-based functions are another part of procedural systems that also control certain aspects of generated data. Some procedural systems use fractal functions to define the texture appearance, shape and spatial relations between objects. Terrain generation in Vue is an example of where a fractal can determine the formation of the geometry by mapping the fractal data. A fractal pattern is derived from an initial state, which recursively produces an increasingly more complex growth pattern until a terminating point is reached. All fractal-based functions have a similar basis for beginning construction with an initiator and generator represented in the Koch illustration (Figure 2.1). This example illustrates one type of re-writing system where in this case a constructed broken line is repeatedly replaced with interval copies from the generator that are reduced and displaced to have the same end points as those segment intervals being replaced [PRZEMYSLAW96]. The Koch fractal example is the premise leading to generated elements within digital environments and toward more complex fractal patterns used within Vue to create common elements such as land and vegetation.

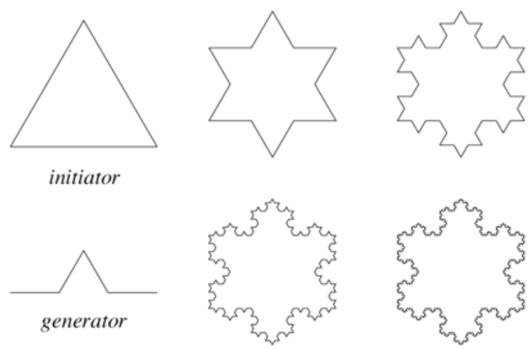

Figure 1.1: Koch recursive snowflake fractal.

An important attribute of fractals is spatial frequency or lacunarity, which can show a variety of fractal scales from input parameters [DEUSSEN04]. Perlin, Voroni, Sine, Linear, Steps, Rectangular are among the most common fractal functions within procedural systems used to develop derived alternatives [DEUSSEN98]. The ability to create and combine fractal functions for defining displacement and formation of ecosystems is another time-efficient method and added layer of complexity for simulating realism. This advantage can lead to the benefits of rapid and engaging prototyping including training simulations, entertainment applications, and movie pre-visualization.

A drawback of procedural systems in creating complex digital natural environments is the amount of polygonal data produced, which can be quite extensive and requires vast amounts of system resources. Unmanaged availability of resources will slow the progress of workspace processors and is increased even more when rendering a sequence of frames. Another is limited artistic control, as procedural systems process unintuitive fractal-based algorithms, and although an expedient approach for populating landscapes, an artist must be free to express his or her creative intent through changing a 3D environment as directed. The artist must have a high level of predictability during this creative phase in order to progress toward an intended look. Without a sense of predictable behavior, experimentation can become exhausting due to the randomized outcomes of multiple productions to arrive at something close to a requirement [SMELIK10]. This drawback can also include limited features to fine-tune procedural objects, especially those representing organic objects, to a greater level-of-detail (LOD) [ONG05].

For some small effects studios, producing high-end digital effects efficiently is imperative to stay competitive among major production houses. As the demand for visual effects grows, so does the importance of being able to deliver quick turnaround. The visual effects industry has evolved to handle the growing complexity of many environment visual effects. Key reasons are advances in technology hardware, better written algorithms and effective workflow strategies for using leading visual effects applications, all of which increase productivity. Even with such advances, deliverables of animated 3D environments with photorealistic quality remain computationally expensive to render.

Using Vue environment modeling package, photorealistic 3D imagery representing different types of ecosystems was rendered for research purposes. Even though Vue has the capability to produce animated scenes with higher levels of fidelity than Paint Effects, it remains inflexible for optimizing with 3rd party sculpting applications. The research in this thesis resulted in Vue best used as a standalone product as explained along with the renderings provided. A more empirical procedural method for developing an animated natural environment was achieved with converted Paint Effects objects that were rendered in Maya's more robust renderer, mental ray. The scene contained three combined sculpted terrains comprised of a foreground, middle, and background with painted vegetation, procedural sky and dynamic forces acting upon all scene objects.

CHAPTER 2

BACKGROUND AND RELATED WORK

Scene development began with Eon's Vue xStream environment modeling application. Vue can create complex digital landscapes resulting in an abundant amount of geometric data. The number of polygons produced, however, often runs into the billions for single frame renderings. Some forethought is therefore required when considering animation output for single system renders.

A couple of common solutions for minimizing render overhead are recommended when developing natural environment scenes in Vue. First, when creating a terrain, the user should employ a standard terrain and avoid using a procedural terrain, when possible. Procedural terrains, especially infinite procedural terrains, have a much higher polygon count. Second, if utilizing the library of base objects within Vue, the user should avoid selecting complex polygonal objects that contain more than 40,000 polygons. Problems can occur using high polygon objects if the objects are to be populated across the landscape. Starting with a base object of less than 20,000 polygons will allow the workspace to run smoother and allow easier modification of geometry for added detail. The consideration of using appropriate parameters and high-resolution texture maps will increase the quality as shown in Figure 2.1, a maple tree model on a hillside, which includes just 10,000 polygons. The surrounding vegetation, however, is comprised of more than 1.5 million polygons.

Figure 2.1: Modeled maple tree and standard grassy terrain environment.

Even with some of the limitations with L-systems, Vue is a useful tool for creating natural environment because it is procedural. The complexities involved in nature require systems employing functions that can adequately build representation. Managing scenes with large amounts of organic data, interacting light and atmosphere are made more efficient overall with concentration applied to detail by the artist. Generated Vue objects are enhanced through a build-up of various stages in terrain sculpting, texturing and atmospherics.

2.1 Vue Sculpting

The approach for modifying terrains within Vue is through a sculpting process available within the Terrain Editor (Figure 2.2). Similar to other digital sculpting applications, a variety of brush tools are presented for altering geometry surfaces. For a broader assortment, the Brush Editor allows for adding and customizing existing brushes. The build-up of brush strokes can add huge amounts of detail and complexity to generated terrains. Sculpting is one of the most important processes in forming the terrain and most commonly used.

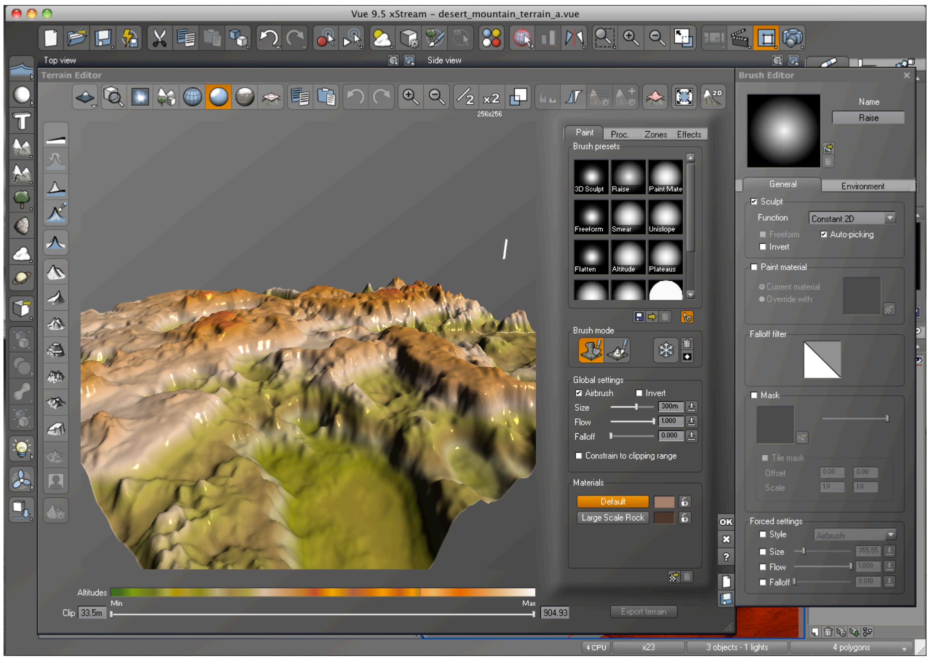

Figure 2.2: Terrain Editor.

2.2 Vue Materials

Applying procedural materials or textures is controlled through the Advanced Material Editor. The number of layers created in this user interface should be limited to as few as possible to reduce calculations. Procedural materials are to be avoided in place of texture maps whenever possible as this will render more quickly. Texture maps containing high quality image detail show a higher level of realism than what is possible with procedural materials.

Figure 2.3 depicts a mountainous desert terrain rendering from Vue showing a before and after process of texture image optimization for a 2k image (2048 x 2048). The image on the lower left has areas of heavy saturation and high contrast. Color correcting and deep shadow areas containing strong blacks were removed in Photoshop. The right image indicates areas where alteration occurred to minimize seams. This workflow between Photoshop and Vue is an iterative process with rendered test results as there are no texture controls in Vue to handle low-level image editing.

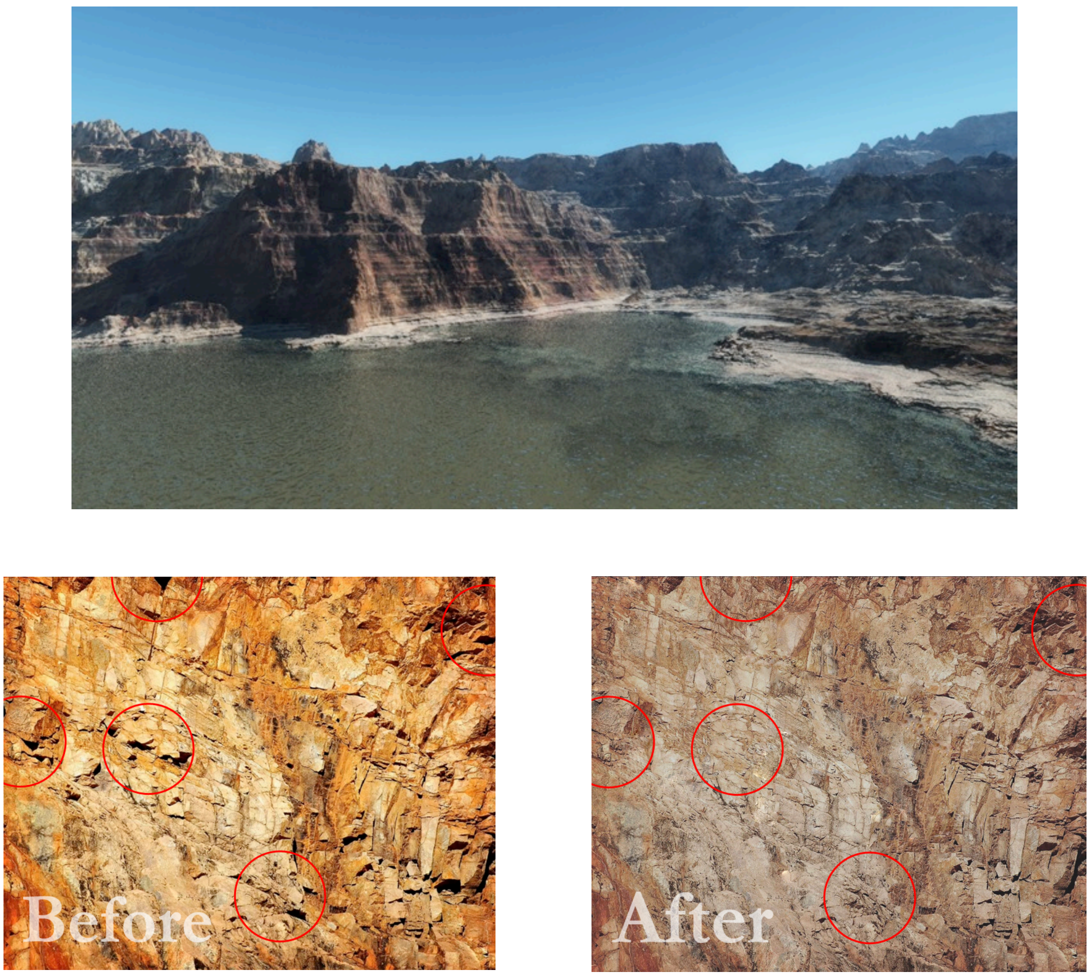

Figure 2.3: Finalize Vue-rendered terrain with optimized textures

The top image (Figure 2.3) resulted from fractal patterns blended in the base color and texture mapping within the Advanced Materials Editor. Mapping was set to Object – Parametric to specify a one-to-one relationship between the texture and the terrain object, meaning, the texture will stretch to cover the entire terrain object. To remove this stretching, Image Scale was then increased to x=2 and y=2 tiling which made seams inconspicuous.

Moreover, seams were not noticeable due to the high-resolution complex rock face. To eliminate obvious seams, Image Scale and Image Offset are the attributes to adjust; however, for more complex situations the image must be re-imported after changes have been made with an image editor. The final adjustment was the addition of a simple material blended with the image texture to fine-tune color contrast and introduce a higher degree of fidelity. The procedural material was adjusted to a neutral dark brown with alpha set at -35% (Figure 2.4).

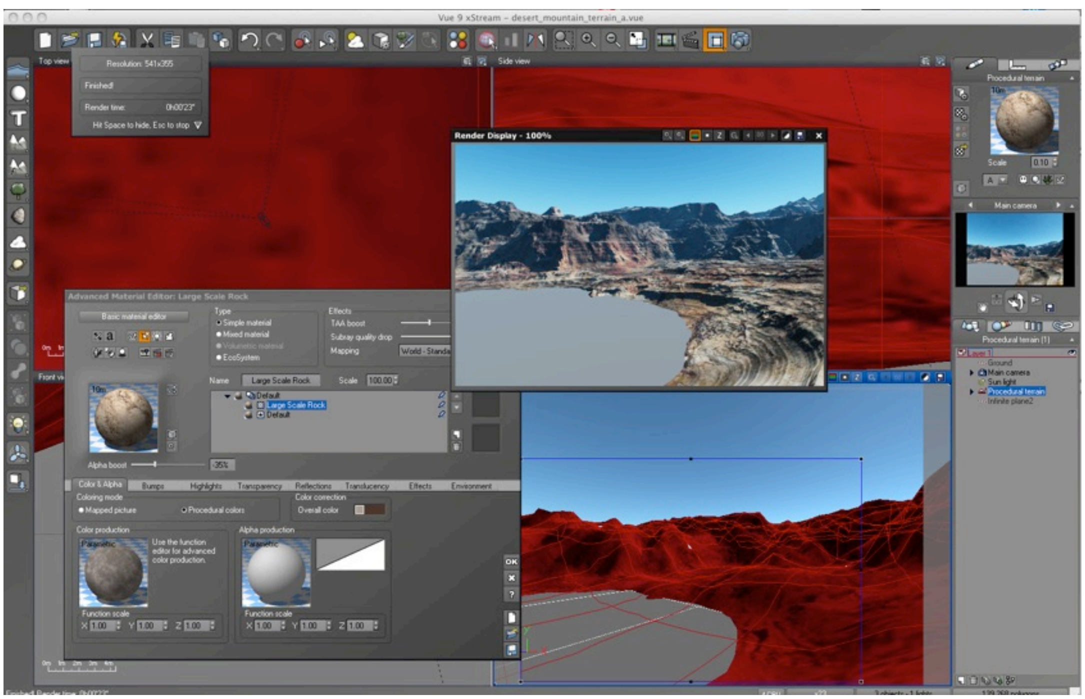

Figure 2.4: Custom material combined with texture and added to procedural terrain.

2.3 Vue Manual and Procedural Object Painter

For controlling placement of procedural objects, the EcoSystem Painter user interface lists objects available for selecting and manual painting within a Vue scene. Matte painters benefit since after locking the camera to a certain point of view, objects can be painted and placed to create the best composition. Unfortunately, if rendering a scene with an animated camera, the scene will show surfaces with unapplied and floating procedural objects. Painting objects in this manner is not effective when creating digital environments because they may be translated to different perspectives; therefore, it is best used for stills or finishing touches where camera motion has been determined. Additionally, some objects from the EcoSystem Painter library do not list total polygon count, leading to an issue of excessive geometry and file size for distant objects. Objects with high resolution should have foreground or middle ground placement as hero objects. Conversely, objects away from the camera should have lower polygon count. Specifying the resolution of procedural plant and rock objects is a limited feature in Vue. If modifying an object's geometry is needed, exporting to another 3D application such as Maya will yield the most customization. Vue has a plugin for Maya that allows a workflow between the two applications; however, Vue scene files take long time to open within Maya. Whereas, a better workflow involves exporting .obj files from Vue to Maya and importing them back into a Vue scene.

2.4. Vue Applied Forces

The addition of subtle movement to procedural leaves, grass and water adds life to 3D environments. Applying forces to procedural objects is dependent on a per object basis. The wind forces on tree leaves, grass and plants requires proper adjustments for perceiving natural motion.

When including atmospherics like haze, fog and clouds, global radisosity tremendously added to rendering time. Using Vue's render passes helped with some of the overhead, but still required a great deal of rendering time for each frame. Vue can produce photorealistic renderings of vast ecosystems shown from a wide-angle perspective viewpoint, such as those in high-altitude fly-overs or distant landscapes (Figure 2.5). Each shot perspective can then be integrated as a straight-to-compositing workflow in the form of a digital matte painting, or imported and projected onto 2D or 3D surfaces before being exported to a compositing application. In each case a feeling of motion is from animating a wide-angle lensed camera across the terrain with the inability to see fewer environment objects reacting to simulated forces.

Figure 2.5: Wide-angle perspective of Vue forest ecosystem rendered with Vue.

When applying forces to vegetation objects close to the camera, Vue consumes an abundant amount of system resources on a single machine. Making entire ecosystems animate naturally takes great effort and should be dedicated to render farms to make work progress easily. Additionally, specific rendering options should be selected to minimize animation flickering is a common problem in Vue. When dedicated to a render farm; however, such flickering is greatly minimized using the optimal settings within both the Render and Animation panels. A solution for minimizing flickering on a single machine is more difficult and is reliant on the subject matter within the composition. If the composition is a dense and complex natural environment with heavy force winds acting upon objects, the resulting render will exhibit noticeable flickering. A common workaround for rendering complex digital environments from a certain vantage point involves using multiple layering of projected mattes on 3D and 2.5D geometry from middle to background regions. This technique is further aided by projecting subject mattes on 2D cards that are rigged to face the camera when the camera moves left or right. Of course, such methods require solving parallax issues, hence the need to include appropriate layering such that the objects appear naturally with camera motion.

When setting up the landscape in Maya, the general method is to choose the level of detail for each object represented in the scene from the perspective of the camera, and place it in the foreground, middle or background. Developing a strategy for breaking down scene elements for placement into complex virtual worlds is important. Emphases on the hero objects and areas of interest, as well as scene organization, are key points for delivering professional results.

All visual effects steps are documented within a given pipeline for use later on the project. A great deal of research and planning is performed in the initial stage to reference real-world physical movement of objects. For dynamic systems, such as wind, video data obtained from field research is often reviewed repeatedly to set the forces correctly under virtual conditions. Terrain shape, vegetation growth, atmospheric conditions and environment lighting are researched and set within Vue's environment engine, which can be adjusted to achieve an inspiring visual effect for a directed shot.

Early testing with Vue delivered results with a high level of detail. Vue is able to produce imagery that appears indistinguishable from photographs. Developing skills for creative terrain sculpting, as well as creatively utilizing materials, lighting and atmospherics components are useful endeavors. A high degree of understanding of these components is necessary for creating scenes that are photorealistic. Vue is widely used for creating environments for games, television and film. Major movie projects that have benefited from Vue are *Pirates of the Caribbean: Dead Man's Chest, Terminator Salvation, The Curious Case of Benjamin Button, 2012,* and *Avatar*. Figure 2.6 contains images from the Cannibal Island scene in *Pirates of the Caribbean* showing a before and after of background removed and replaced with digital environments created in Vue. Vue in this instance was utilized to output the environment in layers with multiple middle and background card projections composited together to give a sense of volume.

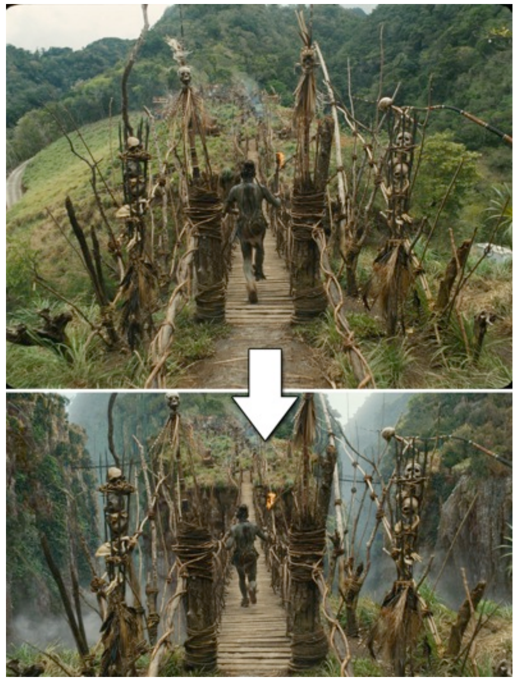

Figure 2.6: *Pirates of Caribbean: Dead Man's Chest* (2006) Walt Disney Pictures. Environment background removed and replaced with Vue renders.

In *Terminator Salvation* Vue was used to create the look and feel of a demolished Los Angles city. The vantage point is from the main character, Marcus Wright, played by Sam Worthington, who is overlooking an apocalyptic landscape. Figure 2.7 shows a rendered digital matte composited over blue screen. In this shot, Vue was able to provide an accurate perception of depth from an integrated procedural atmospheric feature that produces photorealistic stratus clouds, lighting and haze. Another shot from the same scene shows the demolished Hollywood sign that was produced in XSI and imported into Vue to recreate a mountainous terrain around the sign (Figure 2.8). Photographic source images were referenced to judge the mountain shape and its form relation with the sign. Artists used the Material Editor within Vue to create a burnt and destroyed hillside that resulted from the nuclear catastrophe.

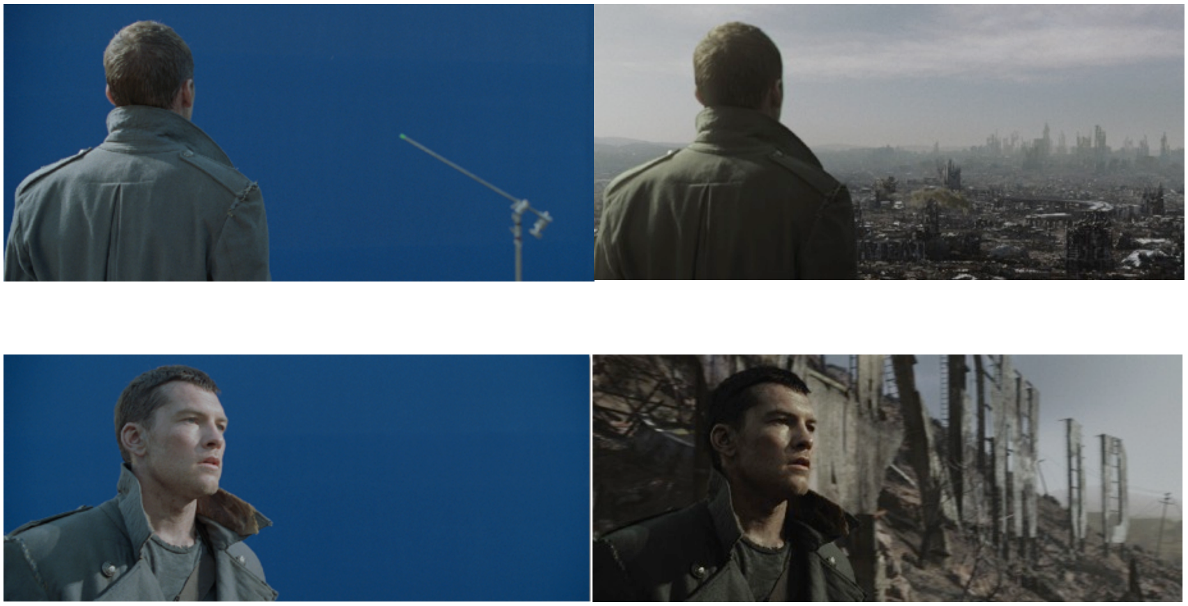

Figure 2.7: *Terminator Salvation* (2009) Warner Bros. Pictures
The Halcyon Company, Wonderland Sound and Vision.

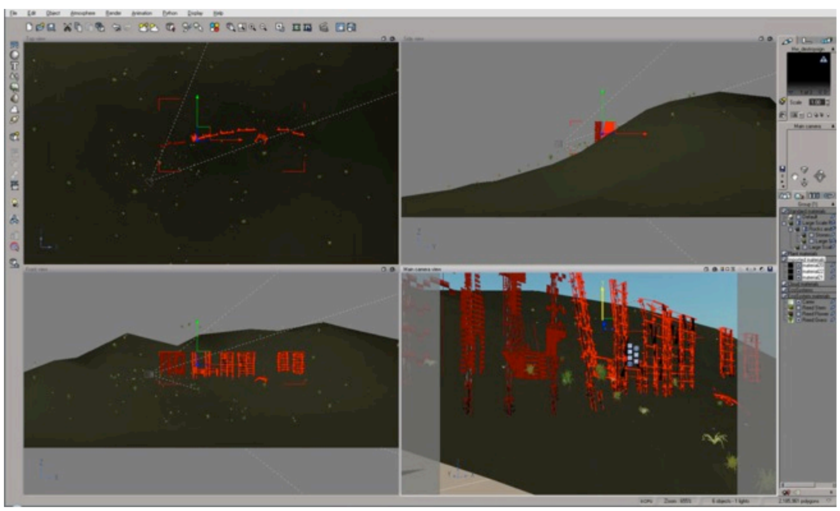

Figure 2.8: Final composite *Terminator Salvation* terrain model with imported Hollywood sign.

In *The Curious Case of Benjamin Button,* artists created shorelines and sky digital matte paintings using Vue. By using this procedural system, renders took on a photorealistic quality that greatly enhanced the shot and provided an authentic environment (Figure 2.9).

Figure 2.9: *Curious Case of Benjamin Button* (2008) Warner Bros. Pictures.
Vue environments.

Artists from Weta Digital created 330 matte shots using the Ecosystems procedural System in Vue. Figure 2.10 shows an example of how Vue's lighting features, which can produce various time of day with sunlight affecting environment elements by scattering light onto other scene objects with correct shadow placement. Bounce light within the clouds illuminates the sky, which serves for adding contrast to the main object, the army aircraft.

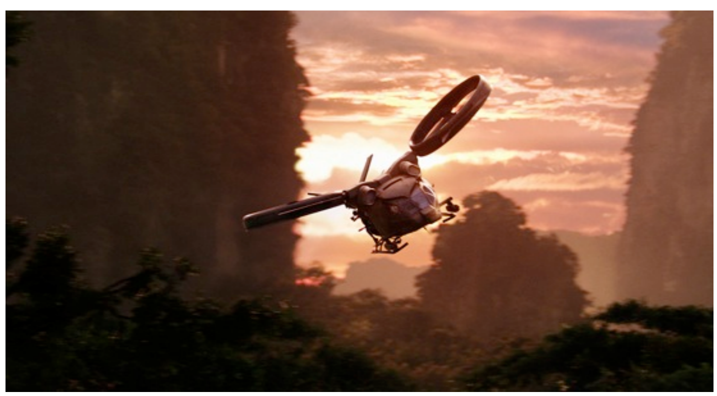

Figure 2.10: *Avatar* (2009) Twentieth Century Fox.
Vue environments

Initial work started with Vue for sculpting the terrain. Two types of procedural terrains that can be generated within Vue are standard heightfield and procedural. Both are generated using fractal algorithms; however, standard heightfield terrains are not as computationally expensive as procedural terrains since they contain significantly less geometry information, and thus less object detail [ONG05]. Heightfield terrains should therefore be placed away from the camera or polygonally subdivided the polygons if any part will be in the foreground. This decision, however, depends on the type of environment needed for each terrain type. For example, since procedural terrains have a higher level of detail, their use may depend on the need to show ground and rock material surfaces. As an added benefit, procedural terrains possess the characteristics of appearing more photorealistic when rendered (Figure 2.11). The mountain was sculpted in the Terrain Editor with rock and snow material later added in the Material Editor. As with imitating anything in nature, photo references allow for adjusting materials to an accurate level of chrominance and luminance. For instance, knowing that packed snow is highly light reflective, increasing the level of luminance will add more light to the material making it much brighter, white and increasing the rock material midtones allows for more accurate and photorealistic materials.

Figure 2.11: Procedural terrain showing abundant details of exposed rock material.

Alternatively, the lower-polygonal standard heightfield works better if the ground cover requires an abundant amount of vegetation (Figure 2.12). Typically, both are used and placed according to the camera field of view.

Figure 2.12: A foreground and background heightfield terrain with little ground exposure.

For render research using Vue, a combination of methods was used to achieve a look inspired from a picture of a landscape taken with a digital camera (Figure 2.13). To model a similar appearance, a standard heightfield terrain was arranged in the viewport according to camera field of view, which had a focal length of 35mm. The next step was sculpting the geometry mesh inside the Terrain Editor user interface using the Paint 2D and 3D brushes together with terrain presets (Figure 2.14). In other cases, an alpha can be imported and used as a brush to sculpt certain areas of a model, which can be useful for sculpting unique land deformations, such impressions in a snow-covered landscape left from a vehicle or person.

Figure 2.13: Reference photo.

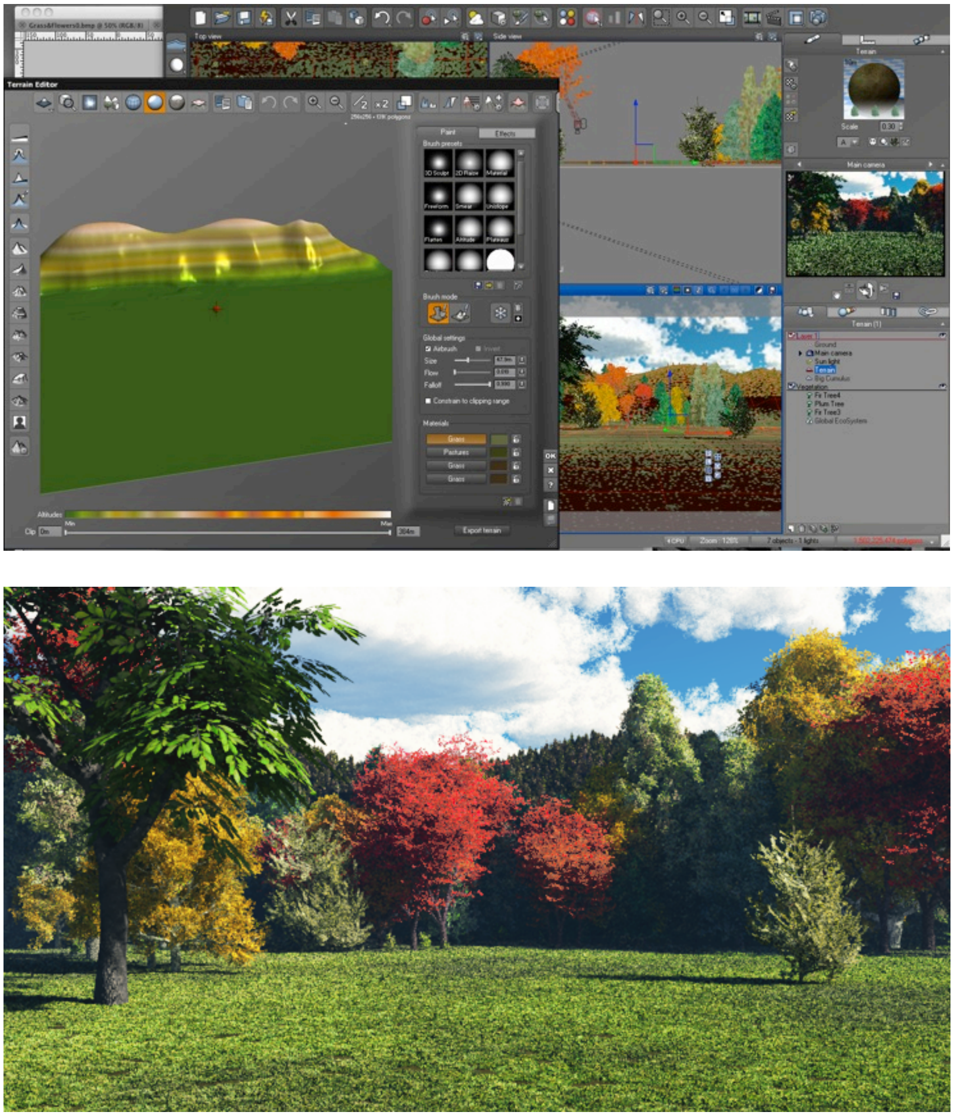

Figure 2.14: Terrain sculpting geometry layout and final Vue render.

A wide variety of terrain types designed within the Terrain Editor. The most commonly used tools during the sculpting process are the 2D and 3D sculpt brushes, which help solidify a mesh design. These tools allow a build-up sculpting method for flatten base mesh or carving away if starting with a mountainous base mesh. In many cases vegetation will cover the landscape; therefore, effect should be focused mostly on creating materials and vegetation. Since populated objects are determined using procedural algorithms each controlled with fractal patterns, relying on the procedural algorithms alone will incur multiple control adjustments and test rendering. For this scene, a Perlin fractal was suited to be the fastest and used for controlling placement of populated objects.

Metablobs are also useful for constructing unique land formations by combining procedurally generated primitive elements. Rather than using a fractal function to generate landscapes, this procedure creates landmass through math enacted algorithms that intersect, merge, subtract and exclude polygonal objects, much like the functionality in a vector-based application on 2D shapes. A fractal/noise displacement map added to the surface of a metablob object produces finer detail, shown as in Figure 2.15, which was created and rendered in Vue. A node-based function panel shows the combination of two fractal patterns connected to give the resulting image (Figure 2.16).

Figure 2.15: Modeling with metablobs using fractal bump and displacement.

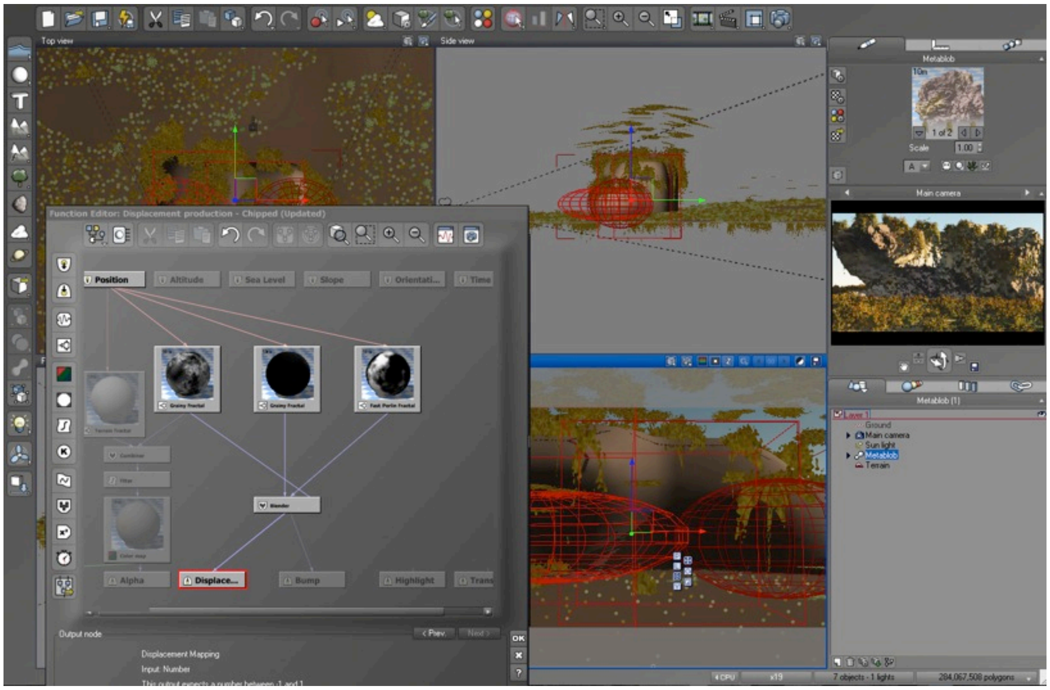

Figure 2.16: Metablob workflow interface using a fractal bump and displacement node network.

2.5 Dedicated Applications

Paint Effects in Maya is both a powerful procedural modeling tool and dynamic particle-based system that allows painting on a 2D canvas for creating images or textures, or paint effects objects in 3D space [KELLER10]. Also related is the capability to create and save custom brushes for adding unique Paint Effects objects. This allows for a combination of procedural modeling efforts, which alleviates much of the effort that would otherwise be used to hand every object within a complex scene manually. Paint Effects contains an array of 2D and 3D brush objects within a library system called the Visor. As brush strokes are applied within the scene, a curve path is created and attaches a new brush stroke, determined by the brush shape [KELLER10], all of which are accessible through the Outliner (Figure 2.17). The brush stroke defines the appearance and behavior of the paint applied along the stroke path (Figure 2.18), and once applied to a scene, its attributes are available within the Attribute Editor. For each stroke applied to the view scene, a new brush node is created in the Outliner, which can be an issue with hundreds of painted strokes. Methods to adjust multiple strokes simultaneously rather than each individual stroke was therefore employed to facilitate modification.

Proper adjustments to these user controls can deliver consistent outcomes unlike many procedural applications that rely upon procedural algorithms that produce somewhat random results. Fundamentally, both Vue and Paint Effects provide strong solutions for producing natural digital environments.

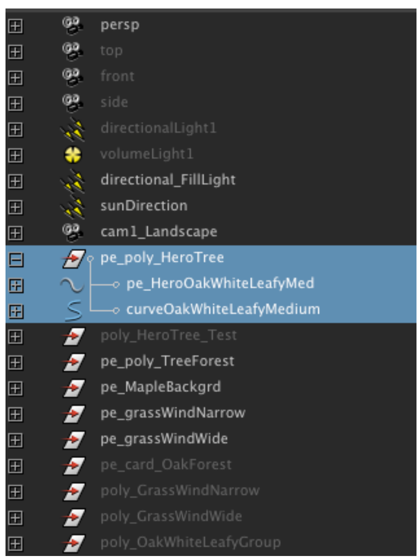

Figure 2.17: Constructed stroke and curve in the Outliner.

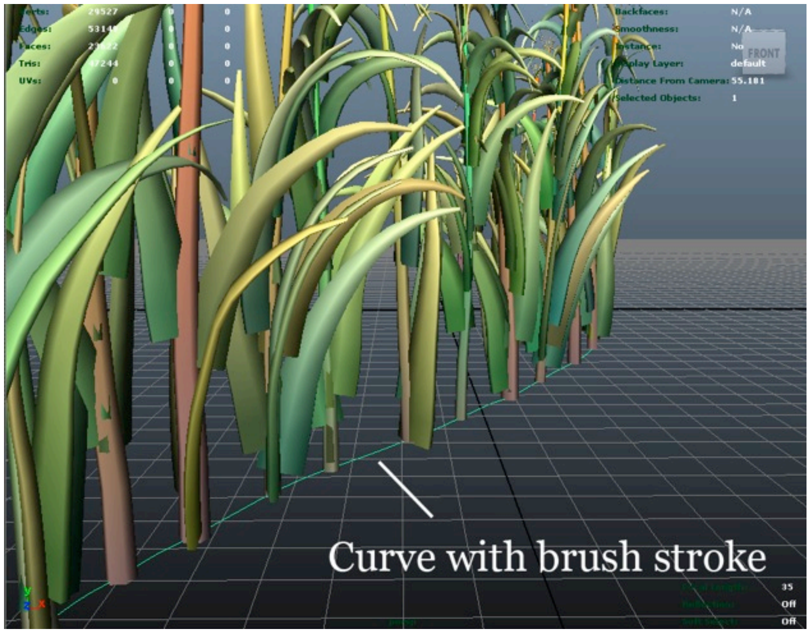

Figure 2.18: Anatomy of a Paint Effects brush.

To help with adding intricate detail to objects, ZBrush was used to bring out organic details on the surface ground and trees. In recent years, ZBrush and similar sculpting applications, such as Autodesk's Mudbox, have become an integral part of many visual artist toolsets. Character modeling from either application can produce hyper-realistic results. Both applications include tools that function similarly to traditional artist sculpting tools for molding digital objects into specific forms. This behavior is preferable for hyper-realistic results since polygonal modeling which often takes more time to complete and lacks the finer details of sculpted models. Standard workflows, however, includes both polygonal and sculpting processes for organic builds. For example, a base mesh object can be completed 70% to 90% using polygonal sculpting before exporting the results to a sculpting program for applying final level of detail. The result can be a higher resolution object or maps for adding detail to a lower resolution object by altering the surface normal or geometry during final render. For this project, ZBrush was used to create fine detail on a base mesh, which was exported as bump, normal and displacement maps.

2.6 Analyzing Reference

The human eye can quickly spot unnatural characteristics within a photorealistic digital environment. For instance, characteristics such as uniformity and organization are rarely found in nature, and when these conditions are introduced in digital natural environments, a feeling of uneasiness can arise, similar to the Uncanny Valley phenomenon [HODGKINSON09]. Most importantly, the viewer must maintain connection with the story, which is hard to achieve if the environment in which the characters act serve as more of a distraction than aid in the performance.

Determining which elements will be foreground, middle and background objects in reference photographs helps with organizing project files and specifying which objects will need the most attention to detail. Objects in the mid to background region will not require as much processing overhead as hero objects in the foreground. Any system can be overtaxed with the burden of processing visual information; therefore this information is often limited to 2D cards or low poly 3D objects with enough texture information to seem convincing.

For this thesis, various scenic images were collected to use as reference information to duplicate the natural world as much as possible. Each presented a distinctive tone due to different types of vegetation, land formations and atmospheric conditions.

CHAPTER 3

ENVIRONMENT DESIGN

3.1 Scene Setup

The initial goal in design was to establish good staging and determine areas of key focus. The retina has limited visual angle, approximated at 2 degrees; therefore, complexity should be concentrated on high-focus areas as the eye moves across a scene [CHALMERS03]. Thus, planning was given to dividing the scene was divided into field of view regions [GLENCROSS06] to identify level of detail for each model. The point of view determines the polygon count for a certain areas, one of which is the hero tree object, placed on the left and the surrounding ground (Figure 3.1). The project settings were set with an aspect ratio of 1.777 at 720p resolution. A wide-angle setting was used on the camera and locked down at 35mm. The project had specified boundaries for the areas of fidelity due to scheduling and available resources. This research will show that using procedural methods of the Paint Effects system delivers a much more efficient means of photorealistic environments.

In the beginning stages of developing photorealistic natural environments, research and planning were performed to build a strategic approach for completing a digital effect. Collected resources, such as source images, provided much needed information to build and arrange objects in the scene with a natural feel.

Figure 3.1: Final landscape rendered scene.

3.2 Scene Design

3.2.1 Landscape Modeling

Modeling began with establishing a base mesh for the ground and hills. The terrain is comprised of three polygonal plane objects arranged to provide foreground mid-ground and background. Each plane was given enough subdivisions to utilize the artisan tool for molding hills and deformations in the geometry. The terrain layout was based on reference images with similar points of view. Since the foreground terrain set the stage for higher resolution objects, the closest viewed ground plane contained more spatial data with smoothing. To increase the smoothness, the artisan brush was selected again to add more detailed impressions. The landscape model was close to completion before it was exported to ZBrush such that the UV layout for texture mapping would be determined. UVs are instrumental in applying 2D textures to objects, as they represent the specifications for coordinates on a surface. Without accurate UV information, visually unsightly artifacts, seams and stretching occur on the object's surface. When using displacement maps, UVs are especially important to allow the geometry to deform properly during rendering.

An export of the landscape object from Maya was created to expand upon later using specific sculpting tools in ZBrush. Once imported to ZBrush, the geometry was divided twice and sculpted. The standard brush was first used to add finer bumps to the landscape. With a smaller brush setting, the positive and negative intensity values of the clay and clay buildup brushes were used to characterize other details of the surface, including finer bumps. Four additional divisions were applied to the geometry with the same brush tools used in combination with a positive intensity value for the noise brush. Sculpting was concentrated at the focus of the camera, while limited surface detail was applied to the back of the plane due to low visibility (Figure 3.2).

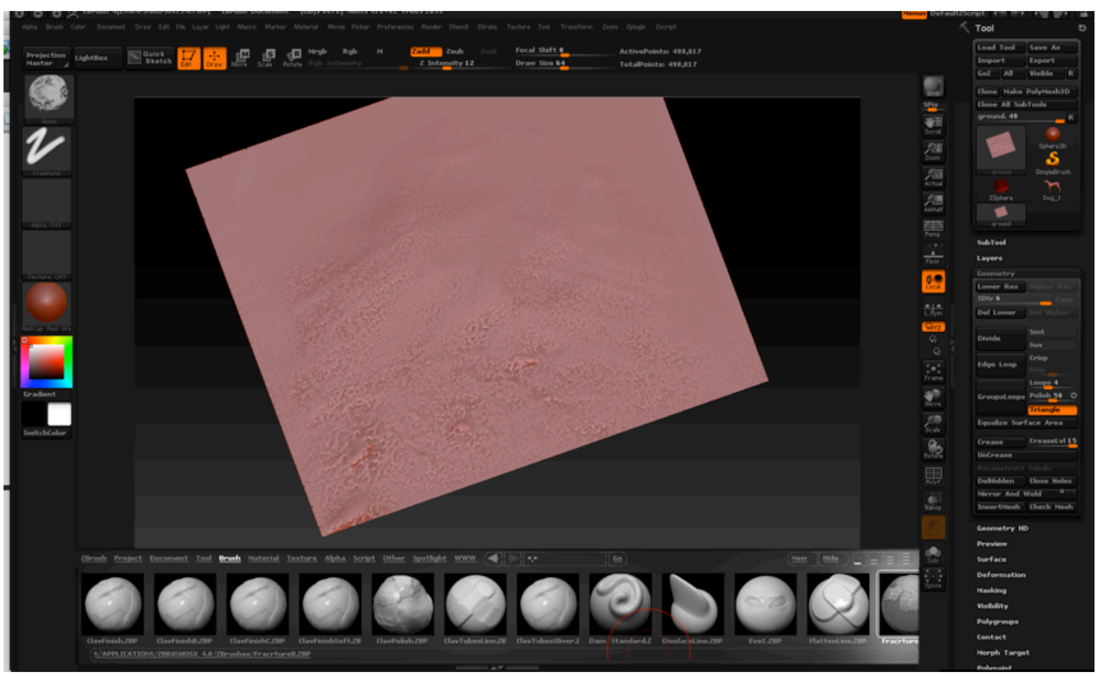

Figure 3.2: ZBrush terrain sculpting.

With the foreground subdivided to level 6, the model contained enough information to be exported either as a normal or displacement map. ZBrush provides three methods for exporting this information, all of which are fairly straightforward. The method selected for exporting used ZBrush's inherited normal and displacement map creation toolset. The other two methods used plugins: GoZ and Multi Map Exporter. All three methods are convenient for exporting maps, with GoZ being the simplest providing a one-click workflow between Maya and ZBrush. Exporting maps is preferable to limit polygon count in the main scene. Unlike displacement maps, normal mapping carries a lower render penalty; however, exporting heavy geometry is far costlier in terms of productivity and render times. Normal mapping was therefore applied to the background terrain, while the foreground terrain utilized a displacement map. The added displacement helped to portray accurate ambient shadows where the surface was altered (Figure 3.3).

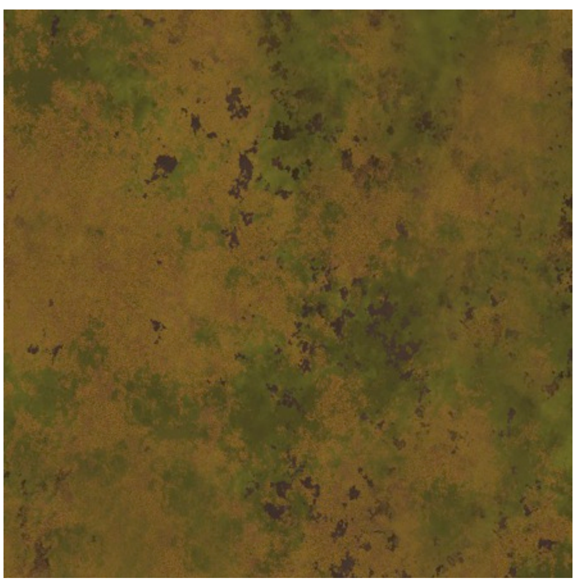

Figure 3.3: Custom created texture applied to terrain.

3.2.2 Applying Paint Effects to the Scene

Before applying paint effects to the landscape surface, the user must specify that the sculpted terrain has paintable surface from the main menu, 'Paint Effects > Make Paintable;' otherwise strokes will appear disconnected from the surface and produce undesirable effects when rendered. Paint Effects can be applied to any type of primitive geometry (polygons, nurbs or subdivision surfaces). The object created for this thesis used polygons.

Paint Effects preset brush objects are stored within a designated library, the Visor, located under the General Editor menu. The Visor displays meshes, fluids, hair, nCloth, nParticles, textures, images and other object types to add to a project scene. For this project, certain plant, tree and grass meshes were selected from the grasses and treesMesh directory. Selected tree objects, oakWhiteMedium.mel, oakWhiteLeafyLight.mel, oakAutumn.mel, maple.mel and redPinesLight.mel were placed around the landscape aesthetically (Figure 3.4). The hero tree, oakWhiteMedium.mel, was placed closest to the camera, while remaining tree objects were distributed within the mid-ground and background regions. The oakWhiteLeafyLight.mel and oakAutumn.mel comprised the mid-ground strokes, while maple.mel and redPinesLight.mel were placed in the background. The land surface was made paintable for adding grassClump.mel, grassWindNarrow.mel and grassWindWide.mel Paint Effects strokes (Figure 3.5).

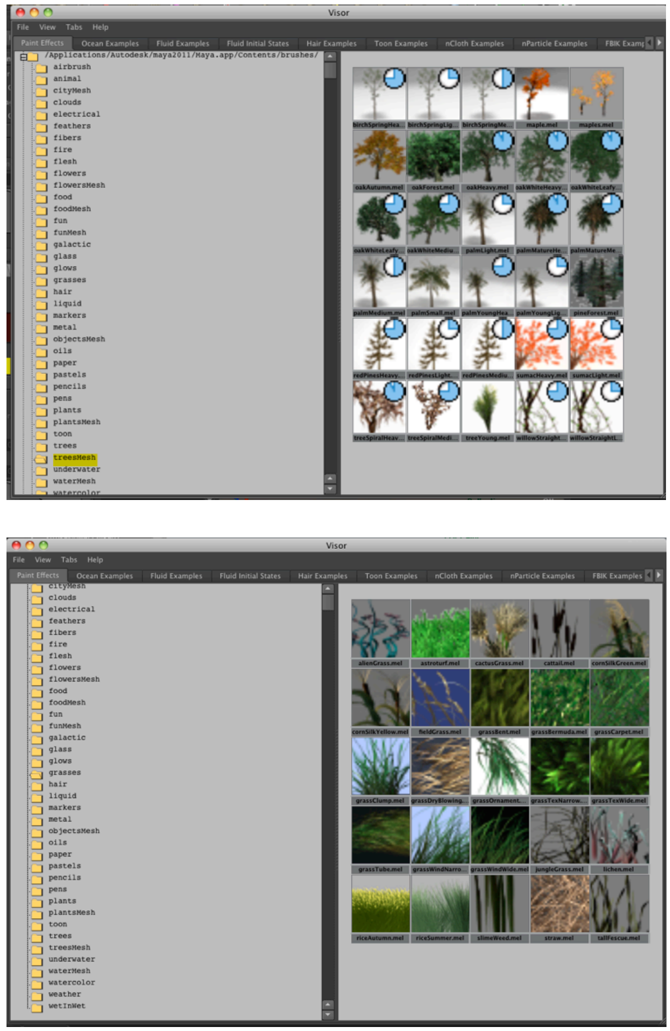

Figure 3.4: Visor displaying selected tree and grass objects.

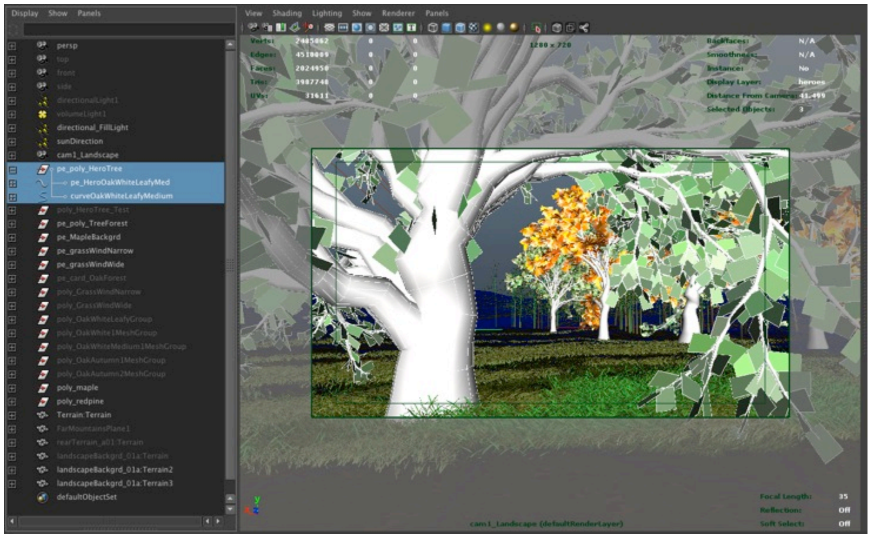

Figure 3.5: Paint Effects scene layout.

The application of the grass objects was determined from the camera's point of view. By locking the scene camera's translation and creating a work camera, the user can strategically draw paint effects and avoid over-populating the land surface with grass. More strokes can be avoided using ground displacement maps for distance grass clumps. For efficiency with procedural generation of objects, the fewer instances to apply, the more manageable the scene will be and the shorter the render time.

3.2.3 Optimizing Paint Effects Scenes

The scene presents natural elements, various oak trees, maple trees and field grass. Each instance was adjusted through the level control settings within the Attribute Editor. This process was not intuitive and numerous renderings and control settings adjustments occurred.

Recent releases of Maya allow Paint Effects objects to be rendered using only Maya Software. In some cases, the rendered results may be acceptable; however, employing methods that lead to using the mental ray rendering engine may be advantageous. The mental ray plugin is heavily integrated in Maya providing many contextual tab configurations for optimizing objects. By taking advantage of mental ray, one can simulate predictable and accurate light and shadow by creating combinations of node lighting and shading networks. To render a Paint Effects scene in mental ray, certain steps are followed after the scene has been painted. With mental ray, converting the brush strokes to geometry will allow for better object fidelity than the features within Paint Effects. Maya and ZBrush modeling tools therefore have an advantage in achieving a higher level of detail.

The advantage of using Paint Effects is faster rendering times as compared with either polygons or NURBS surfaces. This point is realized with the placement of the Paint Effects tree stroke placed near the camera, as shown in Figure 3.5. Using the camera's perspective, rendering a digital natural environment with Paint Effects completes fairly quickly (1:04) as shown in Figure 3.6.

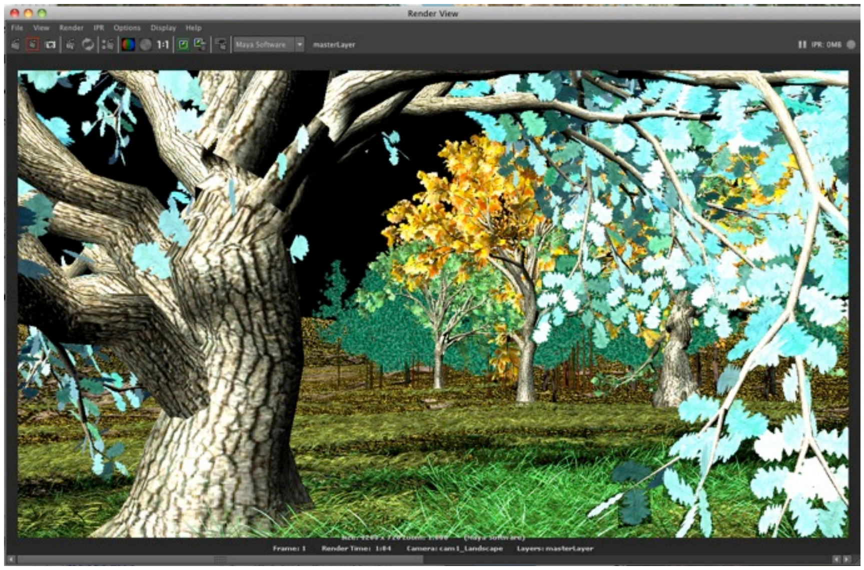

Figure 3.6: Rendered Paint Effects frame completed in 1:04.

Paint Effects offers unique and powerful procedural 2D and 3D generation of various paint strokes and particle effects. Paint Effects template brushes are strokes attached to NURBS curves with attributes for customization. These attributes can be modified to create unique brush strokes for creating highly complex scenery. Further expanding the Visor with personally created brush presets that have saved configurations from the Attribute Editor is possible.

The Maya Software renderer is less robust than mental ray and contains fewer features for improving rendering quality. Many of the options responsible for final image quality are located within the Attribute Editor along with controls to set and share animated procedures for each object instance.

3.3 Animating Scene Objects

A large number of controls for animating strokes in Paint Effects are located within the Attribute Editor. Setting the appropriate controls will simulate natural movement for trees and plants. This capability adds a feeling of life and delivers an impactful mood to a scene. For this scene setup, we set a timeline of 480 frames at 24fps and set Maya preferences to 'play every frame.'

3.3.1 Animating Attributes: OakWhiteLeafyMedium1

OakWhiteLeafyMedium1 was the main tree stroke placed closest to the camera in the landscape. Using this single tree stroke as a starting point for testing values from the Attribute Editor was an iterative process of comparing the latest influences of values entered with previously rendered Fcheck animations. The 'Brush Type' for this object was set appropriately to 'Mesh,' as other types are specifically meant for alternative painting styles not relevant to the scene.

This instance of OakWhiteLeafyMedium1 is a 3D brush type that is composed of tubes constructed with segments. The number of segments a tube comprises determines the length of the tube. While in Paint Effects mode, the individual tubes and stroke elements that comprise the tree shape are not selectable for modification; however, they can be altered through Attribute Editor settings. As the stroke is applied, the Attribute Editor creates the related connection nodes, both a shape and a brush node, each with plenty of configuration options. Within the input connection of the shape node, strokeShapeOakWhiteLeafyMedium1, is the attached brush type, OakWhiteLeafyMedium1. Within this brush node are attributes for animating Paint

Effects shapes: Displacement, Forces, and Turbulence controls. Each panel, starting with Displacement, contained values that contributed to the motion within the landscape (Figure 3.7).

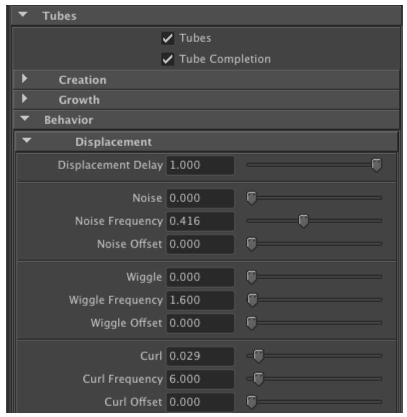

Figure 3.7: Displacement settings for OakWhiteLeafyMedium1.

3.3.2 Tree Displacement Attributes

- **Displacement Delay** = 1.000: This value defined the point along the length of the tube where motion is to occur (Figure 3.8). In this case the point was at the base of each branch.

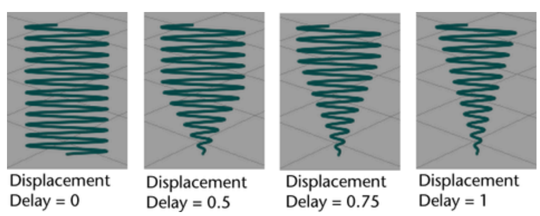

Figure 3.8: Displacement Delay.

- **Noise Frequency** = 0.416: This attribute was set to a low value to define the minimum amount of variance per each tube or branch length (Figure 3.9). This value referenced the noise field as shown in Figure 3.7 as an initial state value that was kept at the default 0.000.

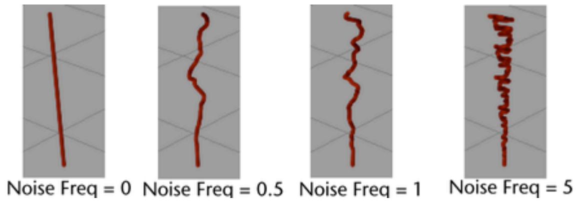

Figure 3.9: Noise Frequency.

- **Wiggle Frequency** = 1.600: This frequency value took into account the data in the Wiggle field, which was set at 0.000. The frequency value of 1.600 was a modest increment from the default 1.000 to introduce more expression in reaction to forces (Figure 3.10).

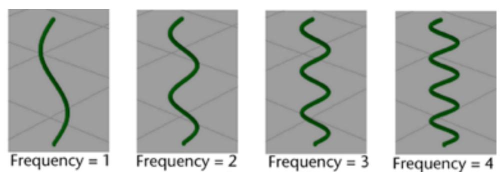

Figure 3.10: Wiggle Frequency.

- **Curl** = 0.029: This setting influenced tube/branch curl width formation (Figure 3.11). A small tubes/branches value minimized this effect.

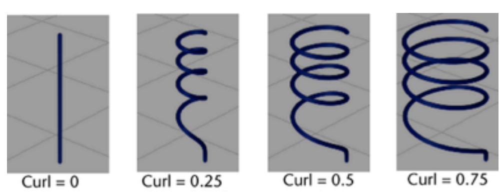

Figure 3.11: Curl displacement.

- **Curl Frequency** = 6.000: From the Curl value referenced in Figure 3.11, each tube/branch maximum length was calculated and applied a frequency within a curl value (Figure 3.12).

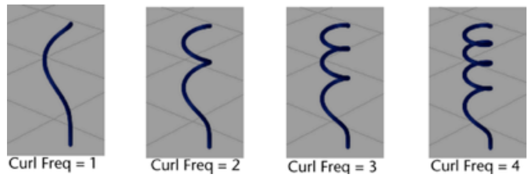

Figure 3.12: Curl Frequency.

3.3.3 Tree Forces Attributes

The attributes within the Forces panel influences applied strokes differently in terms of tube magnitude (Figure 3.13). Where Displacement attributes define and determine tube shape and influences movement, the Forces control panel specifies the tube magnitude of simulated forces enacted upon it. Many of the available levels pertain to the amount of influence along the path; however, the attributes used for the landscape scene defined the magnitude force for tube growth and specified the amount to which forces affect the direction of the tubes.

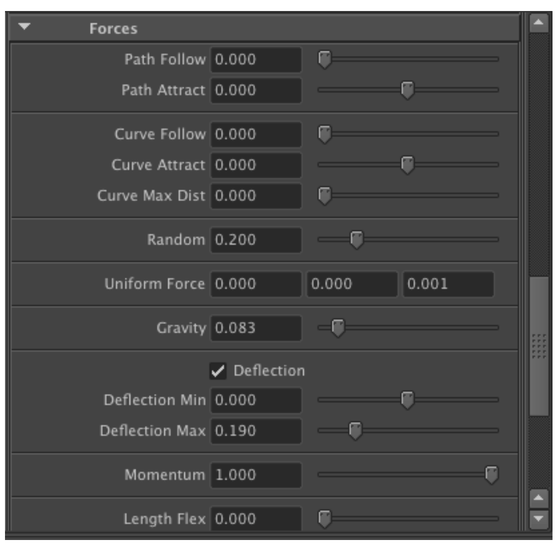

Figure 3.13: Forces settings for OakWhiteLeafyMedium1.

- **Random** = 0.200: This value was set at a moderate influence of force in random direction and intensity, which is applied within the local space of each tube (Figure 3.14).

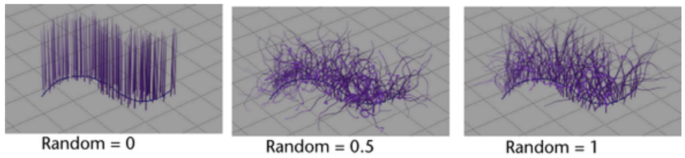

Figure 3.14: Random force.

- **Uniform Force** = 0.0001: This minimal value was set in the z direction field for uniformed forces applied to tubes (Figure 3.15).

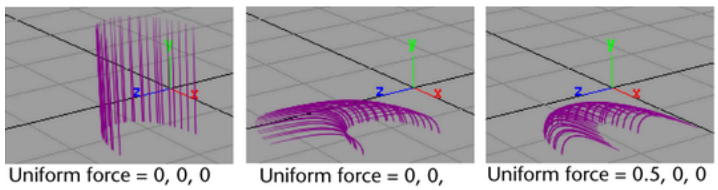

Figure 3.15: Uniform Force.

- **Gravity** = 0.083: This field specified the downward force of magnitude (Figure 3.16). Adjustments to this value were based on a smaller scene scale using centimeters; therefore, this value differs from the real-world value of 9.8.

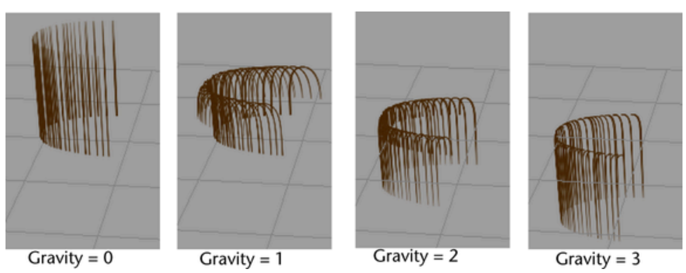

Figure 3.16: Gravity.

- **Deflection =** true: This boolean field was activated for approximation of tube surfaces for deflection (Figure 3.17).

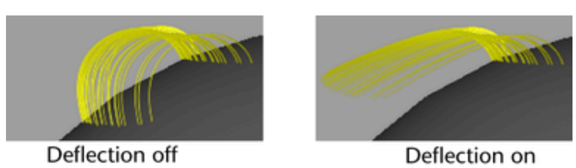

Figure 3.17: Activating Deflection.

- **Deflection Min** = 0.000: This field was the minimum deflection range value that determined the tube distance before contact with the ground.
- **Deflection Max** = 0.198: This range was the maximum range setting for determining the height from the ground before deflection forces.
- **Momentum** = 1.000: No hindrance magnitude forces were applied to the scene.

3.3.4 Tree Turbulence Attributes

The final animation settings were adjusted in the Turbulence section, which were activated by selecting a Turbulence Type. The Turbulence panel lists several attributes that allow configuring turbulent forces and to what extent they affect objects, either in world space or local space (Figure 3.18).

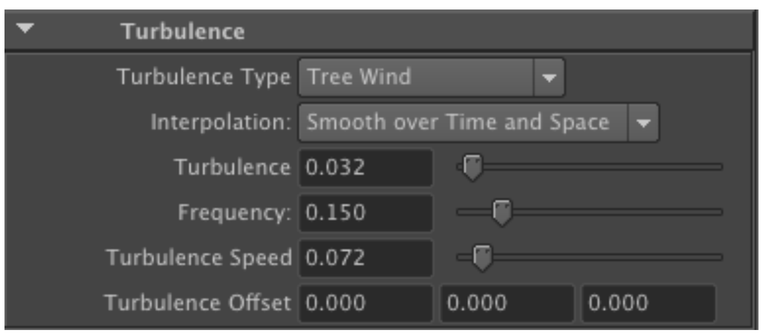

Figure 3.18: Turbulence settings for OakWhiteLeafyMedium1.

- **Turbulence Type** = Tree Wind: This type was chosen because it has a strong influence on tree branch tips, which in turn results in motion similar to wind set in the direction of the tube normal.
- **Interpolation =** Smooth over Time and Space: This setting was selected to smooth the values used to calculate motion to achieve a more realistic movement.
- **Turbulence =** 0.032: This field specified the air speed of turbulence and was set high enough to give the perception of a breezy day.

- **Frequency** = 0.150: This field determined variance in the turbulence and was set rather low to minimize random spatial displacement.
- **Turbulence Speed** = 0.072: This field determined the rate at which turbulence changes over time. This setting required subtle adjustment to allow for smoother tube motion.

3.3.5 Other Utilized Tree Brushes

Other background Paint Effects treeMesh brushes, oakWhiteLeafyMedium.mel, oakWhiteLeafyLight.mel, oakAutumn.mel, and maples.mel, all use similar settings for consistent reaction to world forces (Figure 3.19). Perceived tree mass and branch structure was taken into account when adjusting each of the stroke's attributes. The approach was an iterative process of adjusting parameters until a desired motion was achieved.

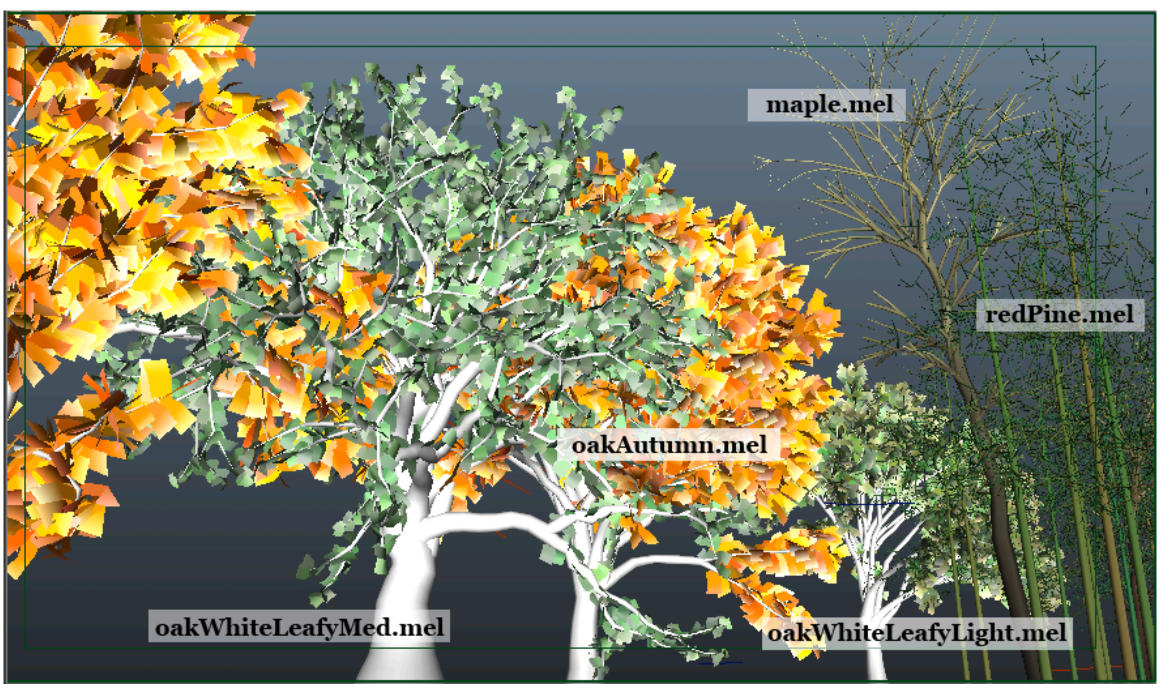

Figure 3.19: Brushes used for middle to foreground areas.

3.3.6 Animating Attributes: grassWindNarrow.mel and grassWindWide.mel

The primary grass brushes selected for painting the terrain were grassWindNarrow.mel and grassWindWide.mel (Figure 3.20). These two brushes cover the majority of the foreground and mid-ground landscape. The background regions included mostly grassWindNarrow.mel with spotted grassClump.mel strokes. Since applying too many strokes to a scene will cause any system to eventually stall, grass was applied to only a small region of foreground for testing. Another iterative process of changing dynamic controls and comparing Fcheck animation renders ensued. Once the final configurations were set, grassWindNarrow.mel covered the majority of the terrain with grassClump.mel painted in strategic areas for added variety. For added complexity, a higher 'Global Scale' value was inserted for selected strokes in the outliner. To deal with the hundreds of grass strokes within the scene, rather than select individual strokes to make changes, a more efficient method was used to group selected strokes under the 'Share one brush' command from the Paint Effects main menu. This command works well for fine-tuning multiple dynamic controls of strokes using the same brush type. In other instances, the Attribute Spreadsheet was desirable for populating property values across all shape, transform and translate stroke instances of the same brush (Figure 3.21).

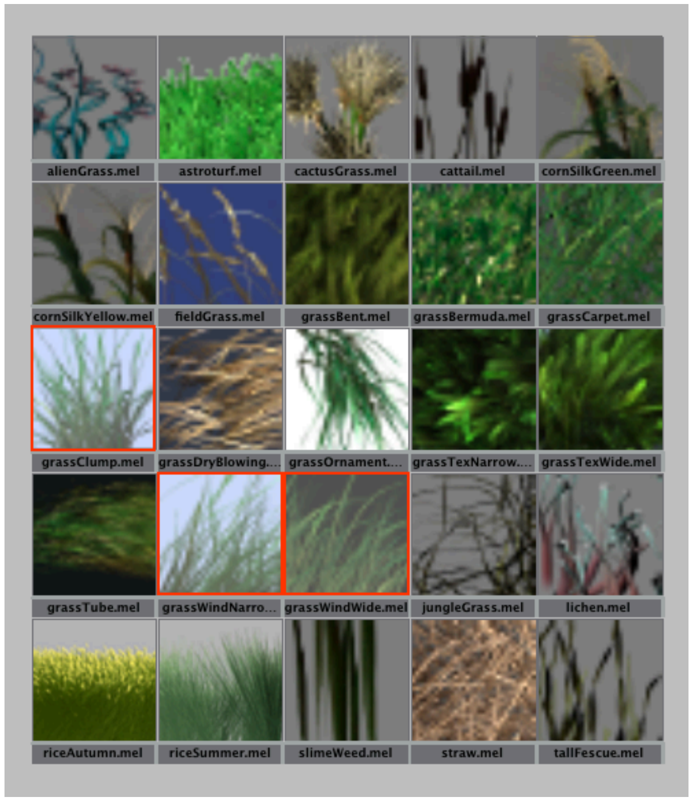

Figure 3.20: Selected grass brushes.

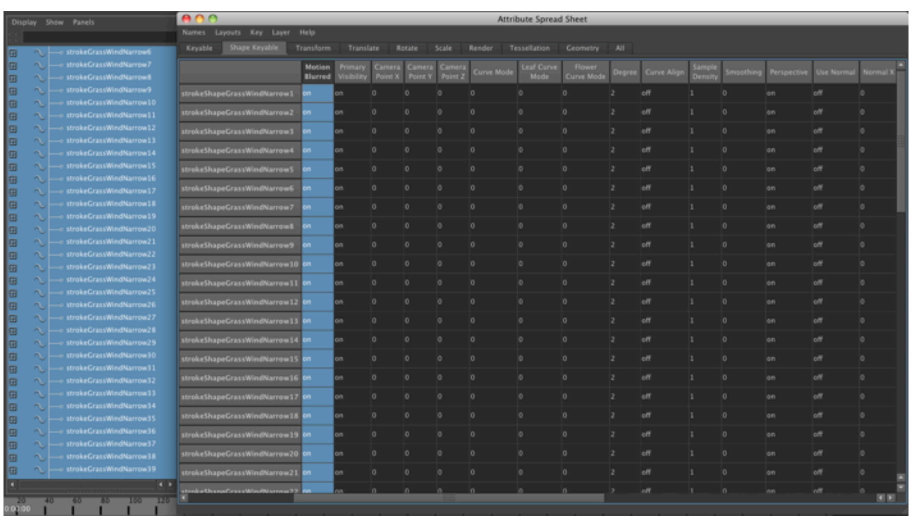

Figure 3.21: Using the Attribute Spread Sheet for stroke management.

3.3.7 Force grass attributes

Below is the outline of control values that were added to the grassWindNarrow.mel brush.

- **Path Follows** = 0.033: This value defined the magnitude of the force that attempts to make tubes conform to their pathes (Figure 3.22). Some regions of tubes/grass strokes contained both positive and negative values to alter grass bend direction and add to the appearance of non-uniform natural characteristics.

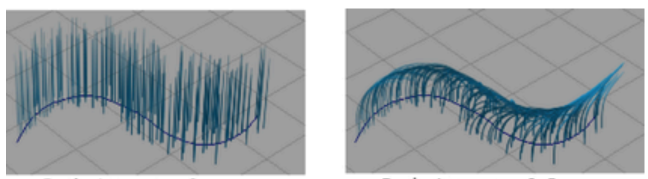

Figure 3.22: Path Follows.

- **Curve Follows** = 0.806: This field controlled the magnitude for each tube's growth along its associated curve (Figure 3.23). For instance, some selected foreground grass curves were set to 0.806 giving some longer blades an arched appearance flowing in the curve bend direction.

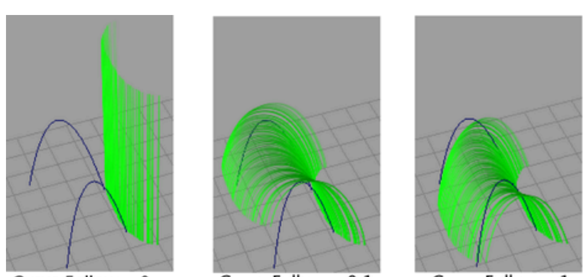

Figure 3.23: Curve Follows.

- **Curve Attract** = 0.262: This field defined the magnitude of force to which tubes are attracted along an associated curve (Figure 3.24). An applied range of values on selected regions introduced another aspect of natural complexity.

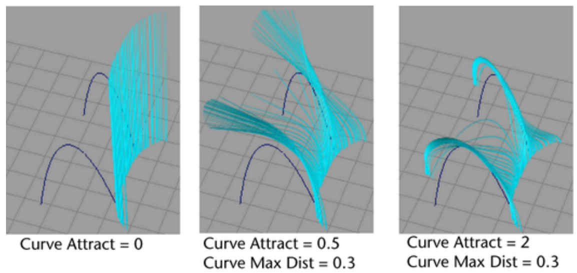

Figure 3.24: Curve Attract.

3.3.8 Turbulence grass attributes

- **Turbulence Type** = Grass Wind**:** This setting was appropriate as the forces involved applied to the grass blade tip and transitioned to the tube base. Much like 'Tree Wind,' this was applied as a world space force that calculated motion based parameters specified within Turbulence control fields.

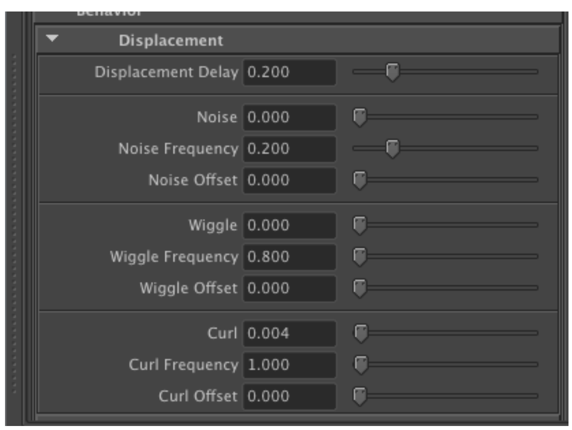

Figure 3.25: Finalized Displacement settings for GrassWindNarrow.mel and GrassWindWide.mel.

The finalized settings for Forces and Turbulence on GrassWindNarrow.mel and GrassWindWide.mel are listed in Figures 3.25 through Figures 3.29.

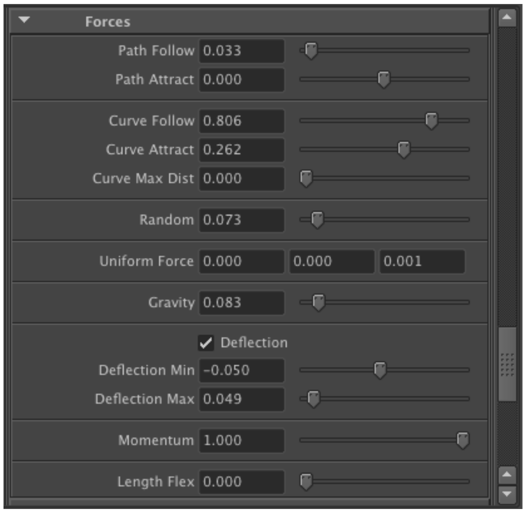

Figure 3.26: Finalized Forces setting for GrassWindNarrow.mel.

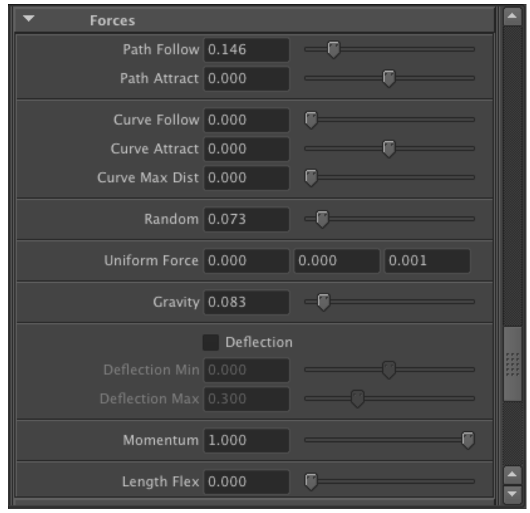

Figure 3.27: Finalized Forces settings for GrassWindWide.mel.

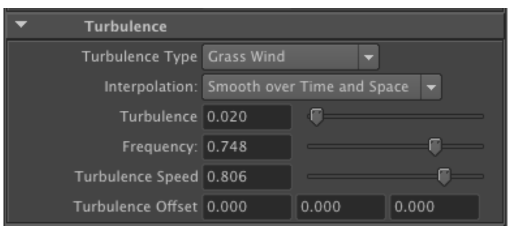

Figure 3.28: Finalized Turbulence setting for GrassWindNarrow.mel brush.

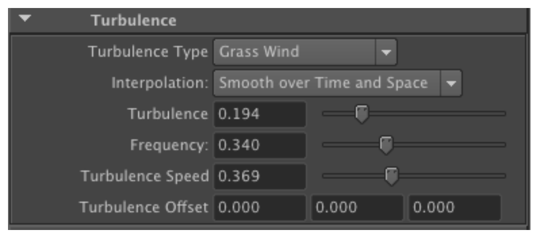

Figure 3.29: Finalized Turbulence setting for GrassWindWide.mel.

3.4 Object Detailing

3.4.1 Paint Effects to Polygons

Taking Paint Effects objects and converting to polygons is a straightforward process using Modify > Convert > Paint Effects to Poly. The more important goal was to develop detailed and refined results once the conversion was complete and above all, preserve connected animated and dynamic processes. After the conversion, asset manageability was necessary for organizing the scene. Three alternative processes were outlined as possible approaches to handle converted geometry:

- increase converted objects to high-resolution geometry

- create and use displacement maps on converted low-resolution geometry
- create and use both bump and normal maps on converted medium-resolution geometry.

Of the three, displacement maps were used for hero objects and normal mapping was applied to selective areas in the mid-ground range.

In certain situations running the Convert Paint Effects to Poly command will need adjustment for objects that will result in a higher polygon count than the default setting of 100,000. This case in occurred for the brush, oakWhiteLeafyMed.mel, to which the Poly limit setting was increased to 1,000,000 with Quad output checked (Figure 3.30). If the poly limit is exceeded after converting strokes, Maya will display warning messages in the output window while rendering. This message is not only an indication of surpassing the poly limit value, but that the total number specified is not high enough to produce details expected in the geometry.

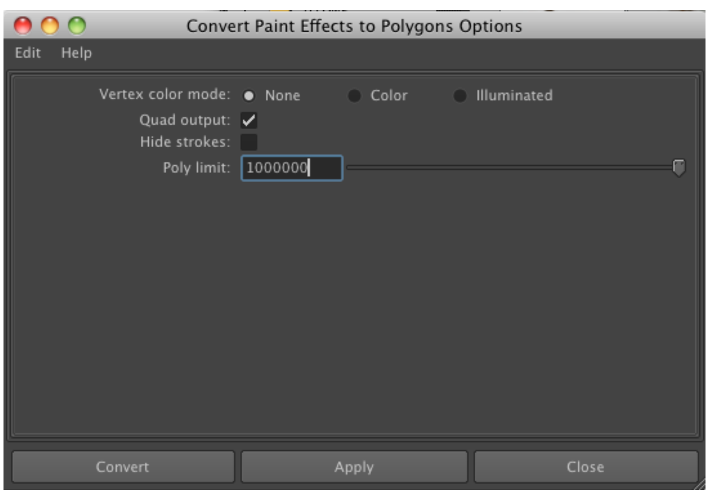

Figure 3.30: Options for converting OakWhiteLeafyMedium1.mel brush.

3.4.2 Managing Paint Effects Polygon Objects

Converting Paint Effects objects to polygons during this translation of objects yielded results that were mostly accurate. Moving from one stroke type to the next, quality can be determined by stroke complexity and conversion settings. Using the main converted tree object, two shape nodes were created in the Outliner: OakWhiteLeafyMain and OakWhiteLeafyLeaf. OakWhiteLeafyMain contained the trunk geometry while OakWhiteLeafyLeaf contained the leaves. To begin the process of refinement, the trunk geometry was separated using Mesh > Separate. This command separated the trunk geometry into numerous geometry segments spanning from the root to the tips. Paint Effects organized the geometry segments from the tree base numbered polysurface1 to the branch tip polysurface2050. With so many geometry segments, some data was eliminated while monitoring a good base from which to build from (Figure 3.31). Determining this threshold resulted in a more efficient workflow with this object due to the high number of geometric segments belonging to the tree tips that could be removed. Figure 3.31 shows an example this process. Even though the tree contains hundreds of geometry objects, selecting multiple polysurfaces and hiding them makes working with such objects easier. Since areas closest to the camera are emphasized, some of the limbs outside the field of view were hidden (Figure 3.32).

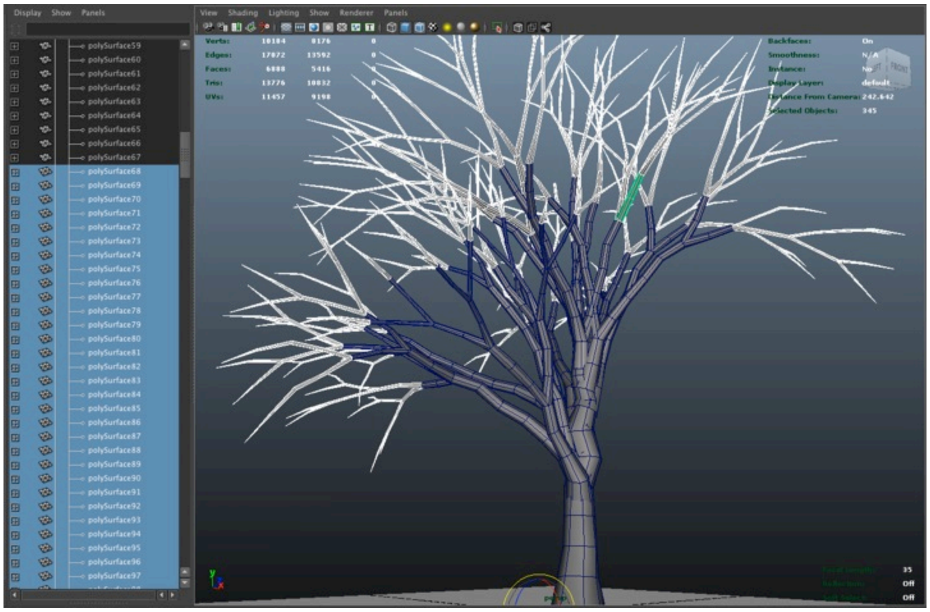

Figure 3.31: Geometry segment selection.

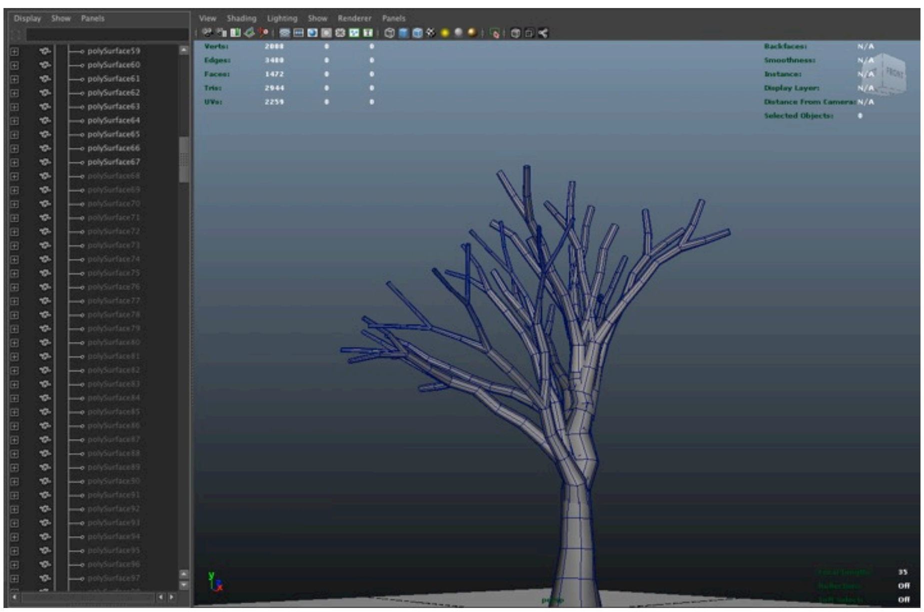

Figure 3.32: Geometry management.

A couple of undesired effects occurred as a result of separating the tree. First, some polygon segments that did not juncture properly along tree tubes/branches (Figure 3.33). Mending the segments was performed in ZBrush using the Standard and Smooth form brushes with different applications of ZAdd and ZSub. Yet another effect was the creation of extraneous surface objects floating inside the geometry protruding outward, or existing around connected segments. In these situations the polygon objects were simply deleted. This remedy lessened the number of polysurfaces, which was beneficial in that it reduced the complexity of the tree model. The second issue involved the management of remaining model segments in need of clean UV mapping, which takes considerable time.

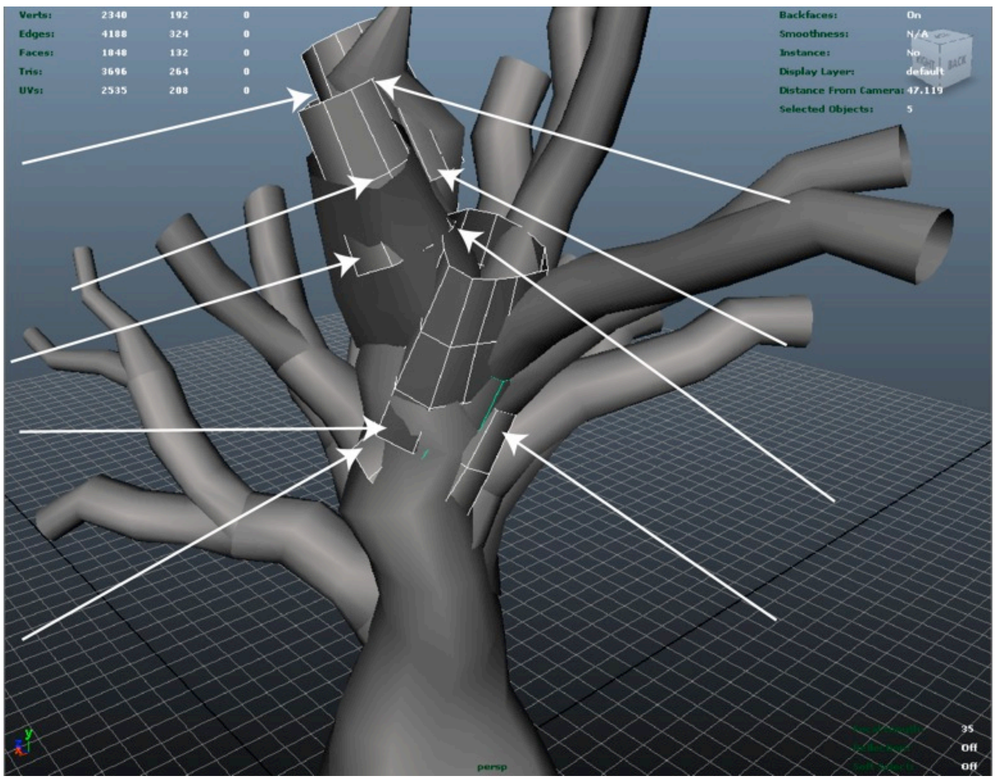

Figure 3.33: Indicated segment issues.

3.4.3 Cleaning Up Object Data

The image in Figure 3.34 shows the oak tree in Outliner after certain segment data have been either adjusted or deleted to form a smoother juncture between neighboring model segments. Within the Outliner are numerous segments comprising the tree components with only a dozen polygon segments highlighted to show how they fit together to form tube/branches. Higher limb objects further down the Outliner list are hidden while working to save system resources.

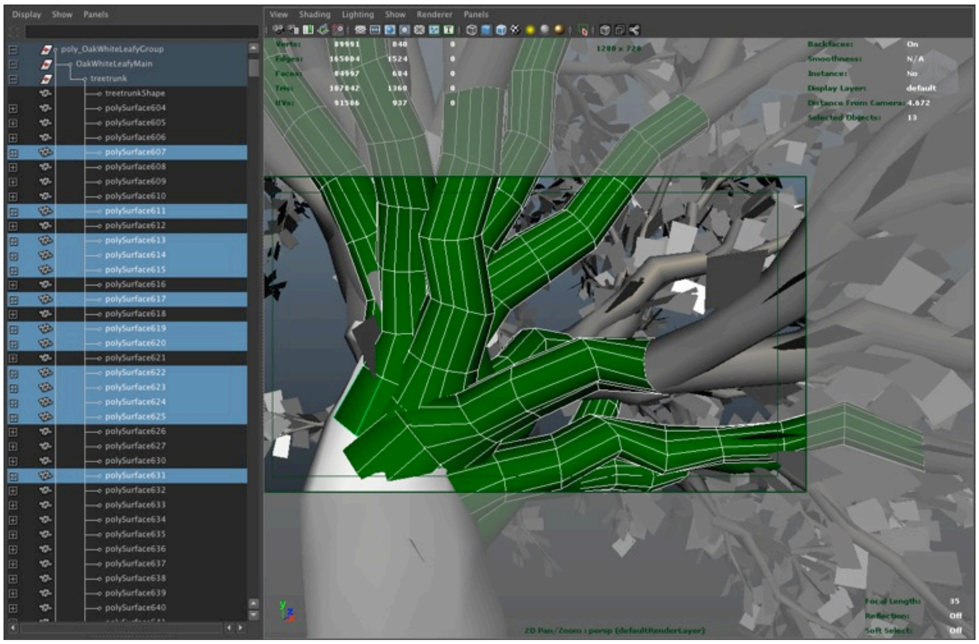

Figure 3.34: Certain geometry segments selected for refinement.

Once the geometry data has been defined in the Outliner, areas of the model were combined into tree sections. Working with the entire model in its separated form will not allow proper texturing due to the generation of unreadable UVs (Figure 3.35); therefore, multiple items were selected for grouping based on geometry proximity. This action resulted in separated sections of the tree, which allowed for efficient UV layout and object groups that were later imported into ZBrush.

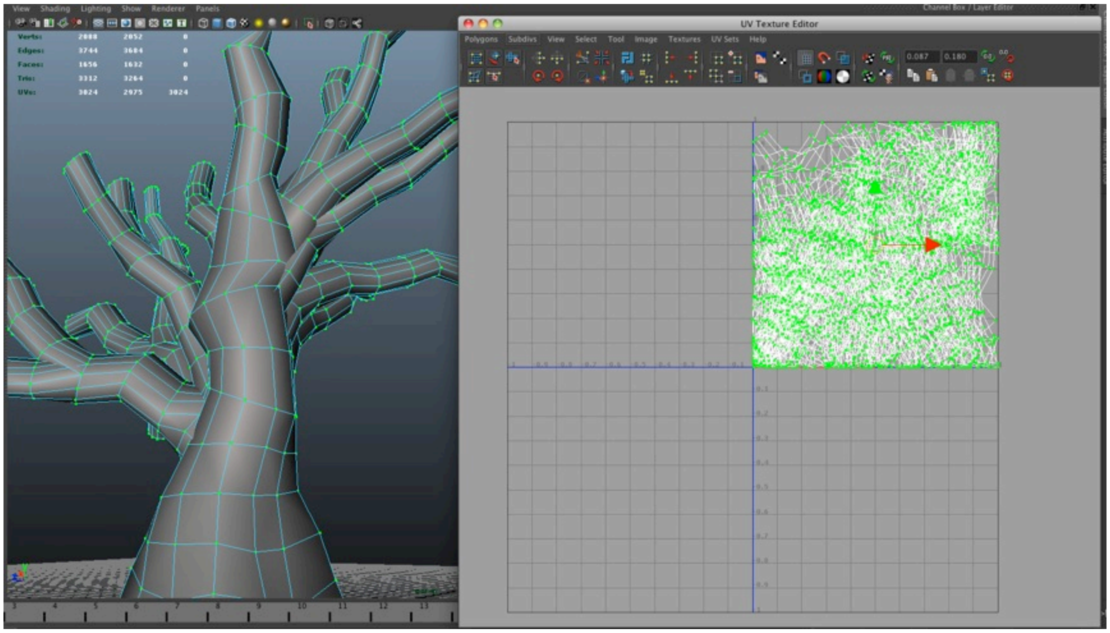

Figure 3.35: Unreadable UV Shell.

Using the tools in the UV Texture Editor, a readable UV layout could be generated. This process consumed a considerable amount of time but was necessary for predictable results. As groups of objects were created, an Automatic Mapping command was applied to the group from the main menu Create UVs > Automatic Mapping. The result was an inefficient placement of UV shells within the [0..1] UV space that was later sewn, scaled and tightly arranged within the texture coordinates (Figure 3.36). This process started from the bottom polygon group representing the tree base and progressed outward along branch groups (Figure 3.37).

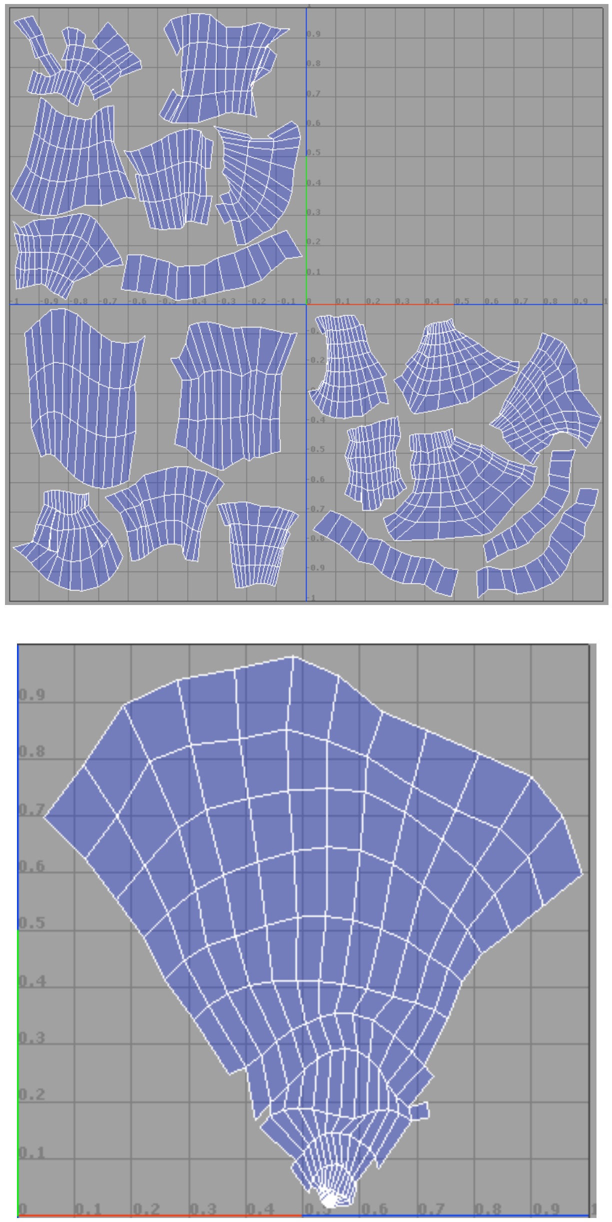

Figure 3.36: Layout of trunk and branch group UV shells.

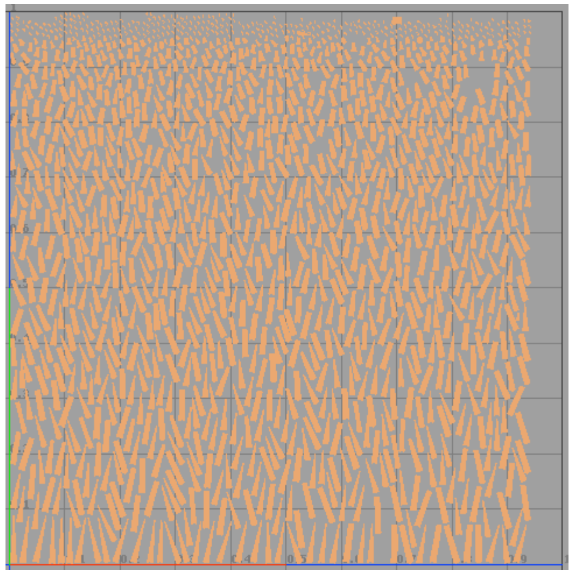

Figure 3.37: Tree branch tips UV layout.

3.5 Lighting

Lighting is a complex process, but leads to creative and impactful imagery. Lighting a modeled outdoor scene with simulated sunlight is possible using the Physical Sun and Sky network within mental ray (Figure 3.38). This network provides built-in functionality that can be configured for a wide range of outdoor lighting moods that emulate the time of day. The Physical Sun and Sky network consists of special interconnected nodes each with its own set of parameters accessible through the Attribute Editor. The Physical Sun and Sky network for this scene was created using parameters found under the Indirect Lighting tab of the Environment section. Once created, mental ray generates three shader nodes: mia_physicalsun, mia_physicalsky, and mia_exposure_simple (Figure 3.39). From this process, sunDirection, a directional light, controlled the sun position when adjusted in the Transform Attributes. The transformed value of -100 units in the X coordinate adjusted the source of emulated sunlight to shine from the left and indicated a time of around noon. Values entered into the Y and Z coordinate fields produce different time of day effects. As a standalone lighting solution, the Physical Sun and Sky network produced results that worked well for lighting the entire scene. The indirect lighting interacted nicely with diffuse surfaces and featured soft shadows across the landscape and models. An additional directional light aimed at foreground objects showed more contrast within texture color. A fill light with a higher intensity value of 1.9 provided deeper ambient shadows and lessened the coverage of ground shading from the main and surrounding trees. Initially, spotlights were used in place of the directional lights, but resulted in unnatural lighting.

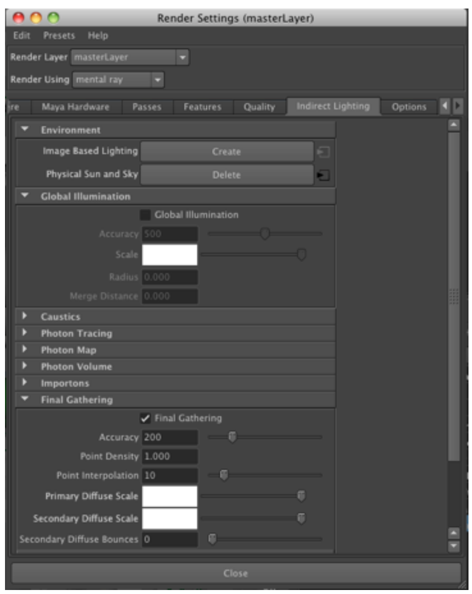

Figure 3.38: Render settings for the Physical Sun and Sky network.

Important nodes such as mia_exposure_simple and mia_physicalsun were left with default values, as the results produced the intended effect. The mia_exposure_simple shader affects the overall white balance of the scene and produces a smaller range of values that are clamped resulting in reduced scene quality. The mia_physicalsun shader provides adjustments to the sun environmental effects that help produce accurate daylight rendering. The core node, mia_physicalsky, affects the sky quality with attributes to control color and introduces atmospheric conditions and sun visual representation.

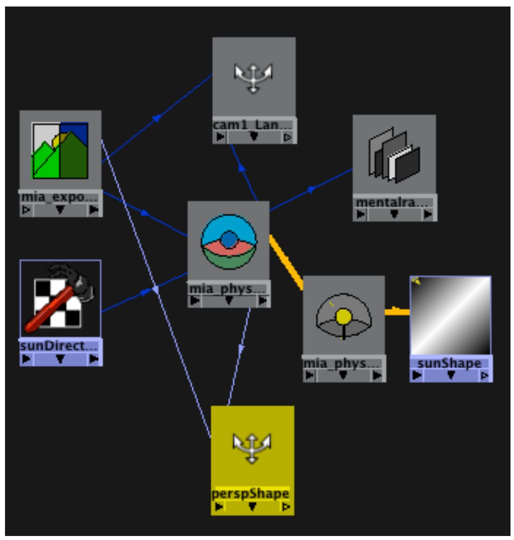

Figure 3.39: Physical Sun and Sky connection nodes.

Once the lighting had been established, a quick render was produced to determine the general quality of the lighting. The limb objects were un-hidden and rendered playblasts were produced to check for proper limb movement. The rendering also indicated areas where direct sunlight highlighted the surface and where soft shadows were formed (Figure 3.40). More geometry cleanup could have been performed; however, issues of protruding branches were minimal and remedied using ZBrush and surface refinement. Lighting in the scene was noted for determining the quality and resolution of textures.


Figure 3.40: Preliminary test renders with lighting network.

3.6 Color, Bump and Displacement Texturing

Paint Effects models use low-resolution textures that are evident upon close scrutiny; therefore, high-resolution textures were created from high-quality images. While online references sometimes work, they are often forced to fit a particular circumstance. Photographs produced specifically for a model work best due to the deliberate intentions of the photographer to capture the source and angle of an object rather than forcing an arbitrary source image to fit a particular model circumstance. For this work, various source photos were shot and altered before being applied as textures to scene models. The reference oak tree model utilized two high-resolution image bark photos, which were taken through a series of processes to arrive at a seamless pattern of 2048 x 2048 resolution (Figure 3.41). As with an earlier example of texturing in Vue, problematic pixel elements were removed from localized soft light with shadow areas and deep blacks for a consistent level of tone.

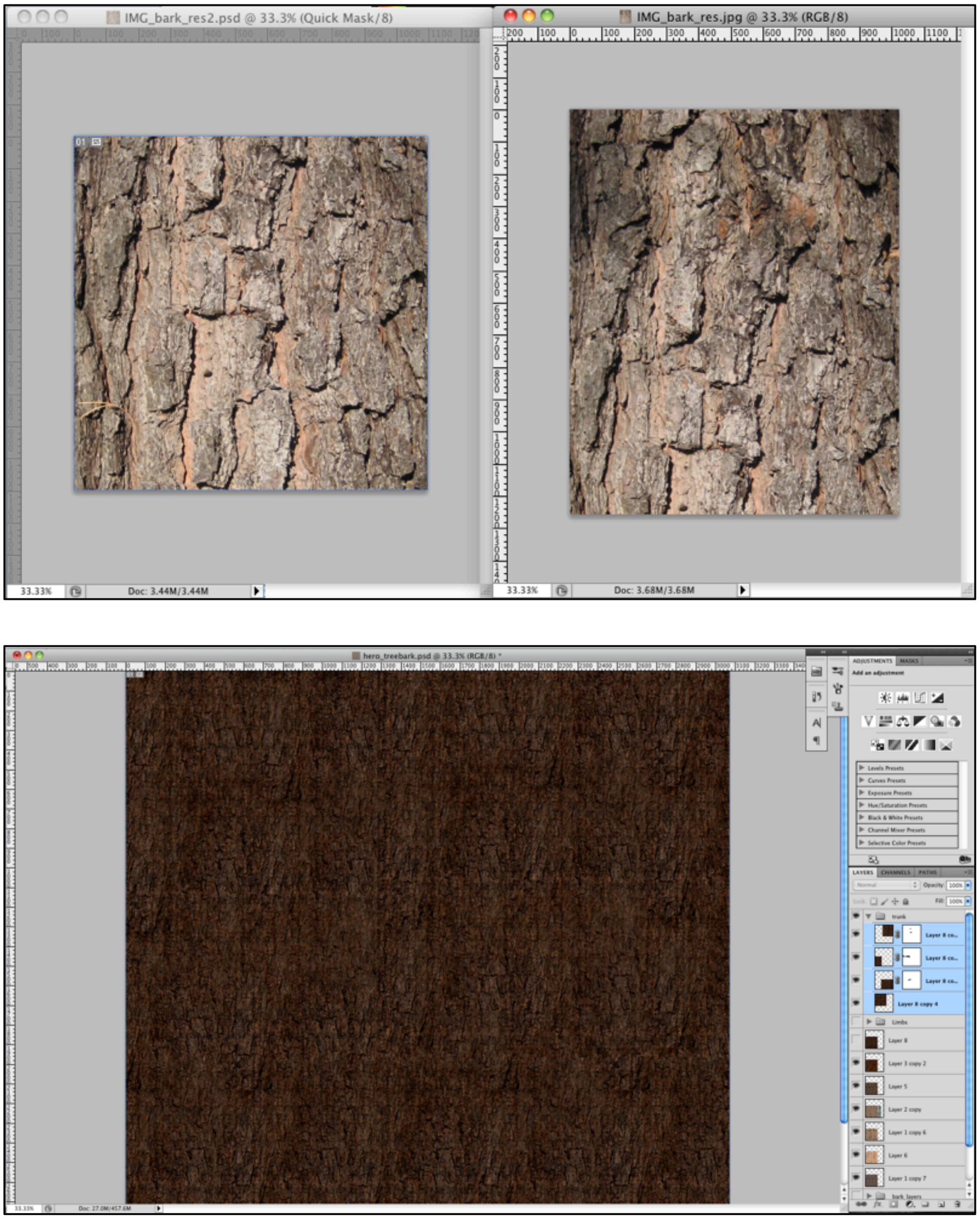

Figure 3.41: Photo references and Photoshop workflow texture development.

3.6.1 ZBrush Bump and Displacement Workflow Methods

The models created in Maya were exported to ZBrush through .obj files (Figure 3.42). The models themselves had a low number of polygons, which were subdivided and sculpted until a high detail was realized. The featured trunk model was divided several times for a polygon count of 639,744. This value is not a substantially high polygon count by ZBrush standards; however, the point was to sculpt the model surface with as much detail as necessary to export both displacement and normal maps. Exporting these types of maps are preferable to exporting ZBrush high-resolution geometry objects, which can impose a heavy burden when imported into Maya scenes. Highly sculpted ZBrush models can contain millions of polygons, and if the same high-resolution objects are duplicated within a scene, Maya's workspace may begin to respond slowly and render time may increase.

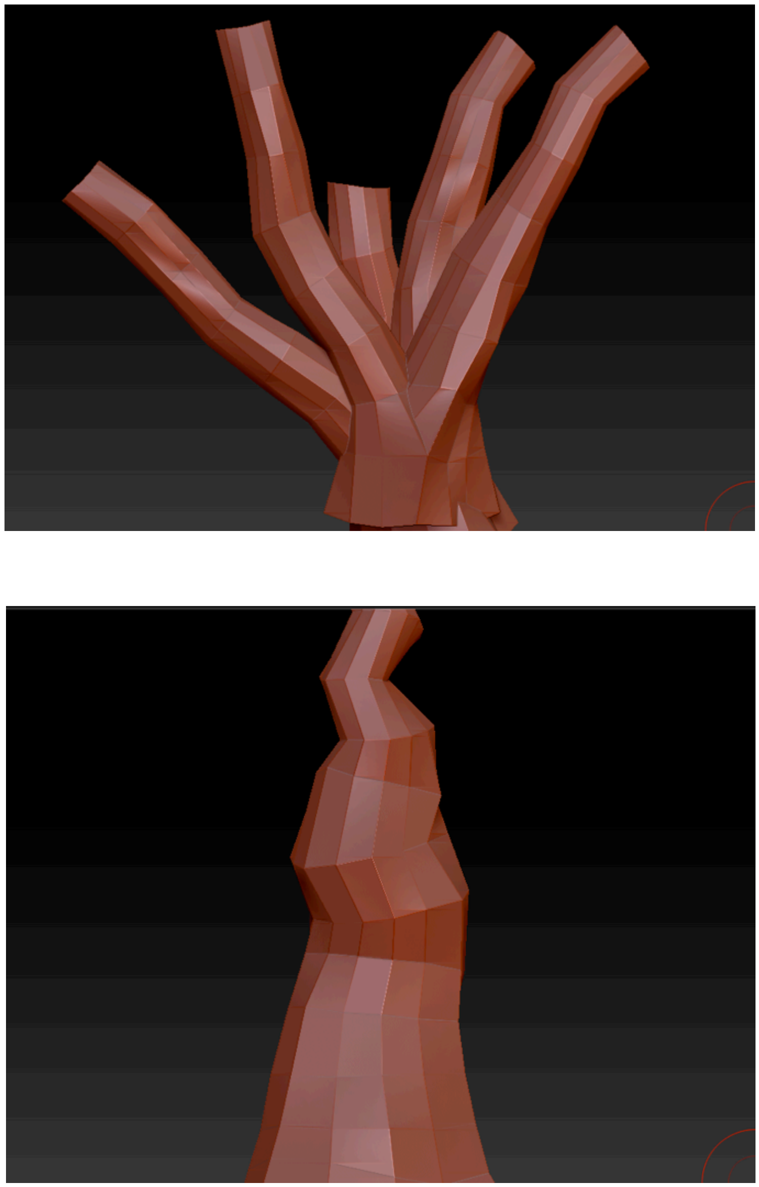

Figure 3.42: Low-resolution branch and trunk import.

A more appropriate way of placing high-resolution models within a Maya scene involves using a method of instancing whereby one object file is referenced and duplicated as many times as needed without penalty to system resources. Since the landscape scene utilizes models derived from Paint Effects strokes, the most important goal was to maintain each object's animation and dynamic settings, which were unobtainable using ZBrush in the previous fashion. Exported normal maps from ZBrush to Maya thus detailed the surface normals on mid to background objects while exported normal and displacement maps delivered a much finer level of detail on the geometry closest to the camera. Displacement calculations did exact a cost during rendering, but not as much as rendering objects that contained millions of polygons. More importantly, using mapping methods sustained connections between Maya polygonal objects and Paint Effects strokes; therefore, working with high-resolution ZBrush models in Maya was not only resource intensive, but also required deformers or rigging to animate. Additionally, specific ZBrush sculpting techniques must be taken into consideration and utilized when creating models intended for rigging, which often involves exporting lower resolution objects along with bump maps.

The three common methods for exporting maps out of ZBrush are all efficient, depending on the project. For an effortless back and forth pipeline between Maya and ZBrush, a downloadable plugin from Pixologic, GoZ, allows importing and exporting maps between both applications seamlessly with a click of a button. The preferred method used for exporting, however, were the integrated tools within ZBrush, both the Displacment and Normal tool panels, which provide flexibility in both viewport visual representation and file management.

ZBrush contains an abundant number of brushes for specific uses. This project used several brushes consistently for refining detail and two that handled form (Figure 3.43). If the purpose of using ZBrush is to export maps, especially displacements, the imported model should be at least 70% complete. Since displacements alter the geometry at render time, too much disparity between the low-resolution form and the created displacement will result in undesirable surface effects from misplaced geometry that drastically alters the model's form. The amount of displacement required is reliant on the level of subdivisions produced, then lowered at time of export. Each of the objects imported had up to 7 subdivision levels and were lowered to 1 to obtain as much detail as created.

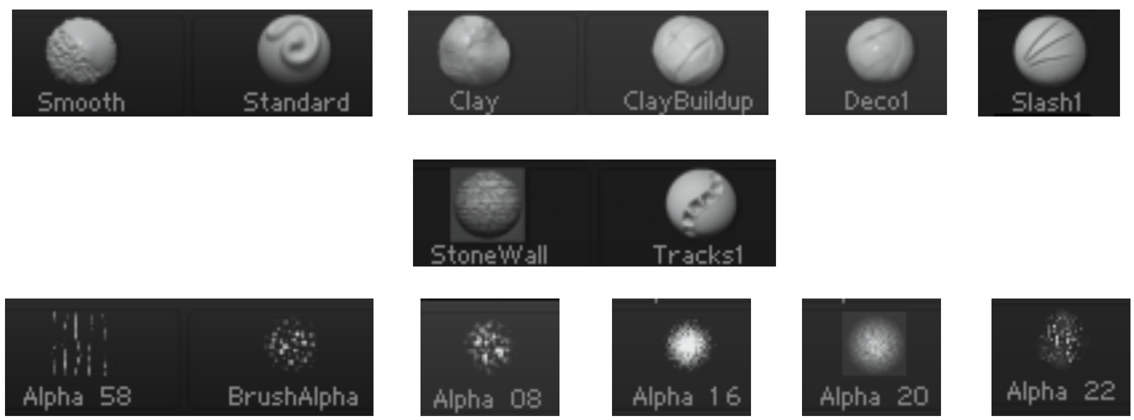

Figure 3.43: ZBrush form, sculpt and alpha brushes.

Alpha maps were useful with individual brushes and strokes to apply variety in detail by changing the brush shape for custom strokes and stencils to add aesthetic features. Similarly, added detail using the Projection Master provided an interface for painting finer unique alpha patterns that were later transformed into deformations on the model's surface.

Before detailing, the base mesh was subdivided to apply general detail. As more polygons were needed, the model was subdivided another level to allow finer detail.

Each stroke was applied using either the Freehand or Dots tool, allowing a wide variety of details to be made in combination with brush and alpha types. All models were processed iteratively between these tools until a determined fidelity was achieved (Figure 3.44).

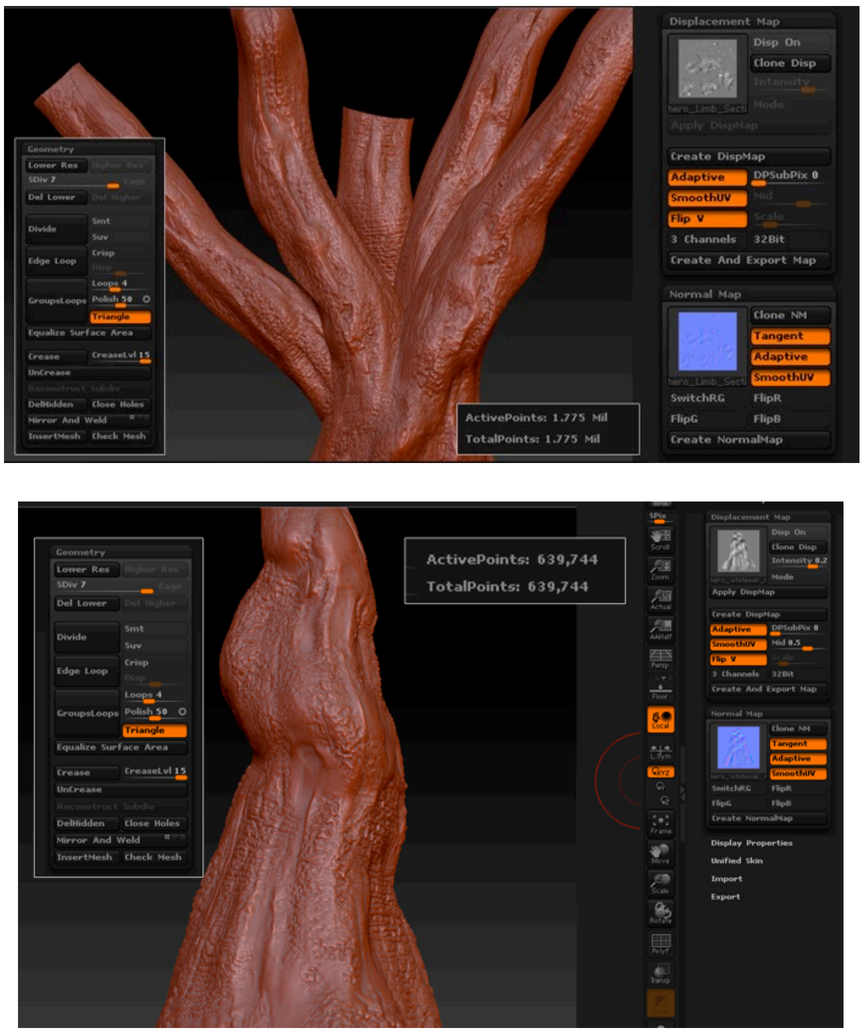

Figure 3.44: Branch and tree trunk refinement.

When the model’s normal map was ready for export, the Geometry panel’s Subdivision option was reset to level 1 and settings in the Normal Map panel were set to

Tangent, Adaptive and SmoothUV (Figure 3.45) to ensure the map would be generated correctly with the best settings. Clicking Create NormalMap generates a map of the object's normal space and direction information. This information was used in conjunction with the object's low-resolution surface to deform the surface based on the blue and purple regions of the normal map. The targeted surface's XYZ vector was mapped to the RGB color of the sourced normal map to determine the direction of the normal faces.

The selections in the Normal Map panel were chosen to generate the best normal map possible. Figure 3.45 shows the normal map settings at time of export. The Tangent option was selected as the most common and versatile method for deforming meshes [KELLER10]. Using Adaptive concentrates on subdividing only areas of high detail during map generation and processes more quickly when activated [SPENCER11]. SmoothUV smooths the UV coordinates when generating the mesh. Before applying the normal map to the target image in Maya, the image was flipped vertically in Photoshop; otherwise, the normal information will incorrectly match what was sculpted in ZBrush. Alternative export methods, such as the Multi Map Exporter menu allow for flipping the normal map inside ZBrush.

Figure 3.45: Normal map settings.

The process for creating the scene's displacement maps was similar to the normal map process. Displacements perform differently when applied to models in that they actually modify geometry within the polygon object. The deformation is based on the light and dark values of a grayscale texture that determine the height and depression of the model surface. As with the Normal Map panel, the same options were activated, which were the Adaptive and SmoothUV buttons that provided the same purpose as before. The effects of the displacement textures were increased using the Intensity level raised from 0.1 to 0.2. The Displacement Map panel has a Flip V option, which was activated before exporting the displacement texture. As in normal mapping, the SubDivision level was set to 1 to achieve the full range of sculpted detail. The final resolution for both map types was 2K (2048 x 2048) with the displacement maps containing 16-bit information (Figure 3.46).

Figure 3.46: Displacement map settings.

Figure 3.47 shows the texture network for the trunk and branch model. A displacement map and a normal map were connected to the custom bark texture. Maya's displacement settings included an Alpha Gain value of 0.403, which increased the contrast within the map according to the amount of deformation. A ramp and fractal pattern was also connected to the texture to break up the bark pattern. The render process included the Approximation Editor to appropriately set subdivisions on polygon surfaces using mental ray (Figure 3.48).

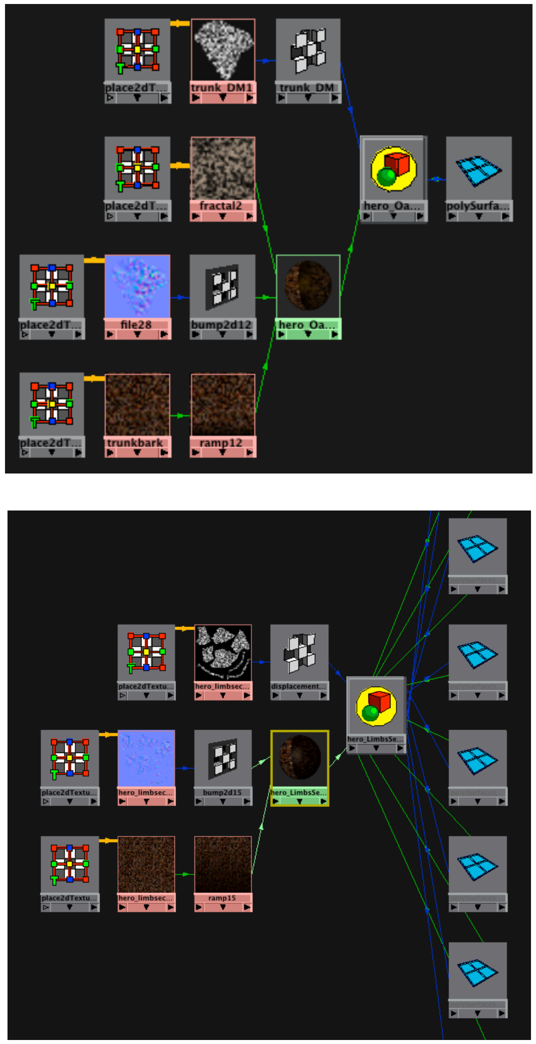

Figure 3.47: Tree trunk and branch section.

Figure 3.48: Approximation Editor.

The tree branch tips had a more simplified network comprising fractal bumps connected to a noise pattern and ramp, with both using the same bark images, all connected to a Phong shader (Figure 3.49). Considering many of the branch tips were obscured by foliage or too small to show surface texture, the use of displacement and normal maps were avoided. There were benefits for adding this simple network to the tree model. By calculating neither displacements nor normal maps, render time was conserved. Also, it is easier to see updates from the Hypershade through changing texture and color for multiple objects simultaneously.

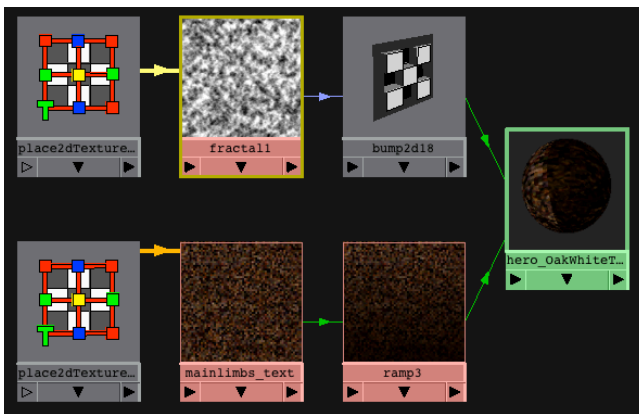

Figure 3.49: Tree branch tips.

Textures were adjusted and reapplied until the desired look was achieved. As changes were made, multiple angles were viewed to observe placement and to note any artifacts on the model (Figure 3.50). Rendering alternative views allows for checking, the interaction between the model and the surrounding environment, including shadows and ground deformation and how well the model blends into the environment. Certain surface areas on each object were selected and tweaked until finalized to add to the rest of the model (Figure 3.51). Remaining branches were unhidden in Outliner and the overall model optimized for best light and shadow interaction through displacement adjustments (Figure 3.52).

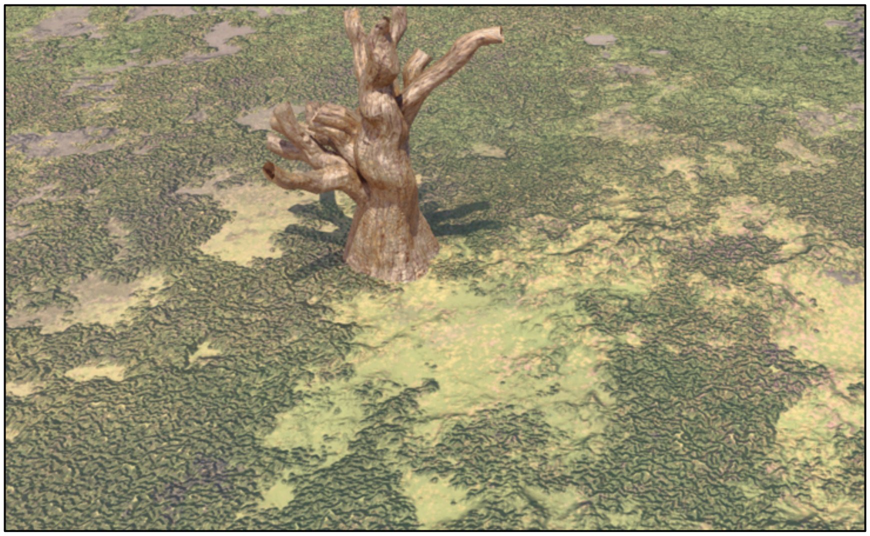

Figure 3.50 Checking environment characteristics.

Figure 3.51: Color, normal and displacement applied to trunk and main limb base.

Figure 3.52: Final mappings applied and adjusted.

Custom leaf textures were created using photographic oak leaf images and optimized in Photoshop. Procedural noise was introduced during the material node building process to add a little more variation (Figure 3.53). Graphic curve panels that determine the silhouette shape of each leaf were adjusted as well.

Figure 3.53: Created leaf textures.

3.7 Changes to Animation Connections

Slight adjustments were applied to the Turbulence control settings allowing for natural movement of the grass and tree limbs. Leaf cards on the tree were activated to face the renderable camera for optimal viewing (Figure 3.54). Animation adjustments to middle ground tree objects were based on similar values entered for the foreground objects. Higher values for smaller background trees and plants were entered for heavier influence.

Figure 3.54: Tree render results.

3.8 Cloud Particle Simulations

The added effects from particle systems are instrumental in providing realism to a scene. The use of particles systems is widely used in situations such as simulating clouds, explosions, fire, rain, and dust disturbed by a moving vehicle. Recent advances in software have allowed for easier implementation by helping the artist establish a base effect and creatively implement calculations with provided settings for achieving a desired effect. Methods for directing stochastic processes, for any given visual effect, however, remain a difficult challenge.

To show how particles behave throughout their lifecycle, a particle lifecycle was developed by [REEVES83]. The steps of this process are as follows: (1) new particles are generated in the system; (2) each new particle is assigned its individual attributes; (3) any particles that have existed within the system past their prescribed lifetime are extinguished; (4) the remaining particles are moved and transformed according to their dynamic attributes, and finally; (5) an image of the living particles is rendered in a frame buffer. Each of these stages describes the particle itself aside from the particle system and is needed to clarify differentiated attributes. For this work, the particle system was developed using the steps listed above and through classifying the particle system using the diagram proposed by [CHEN99] (Figure 3.55).

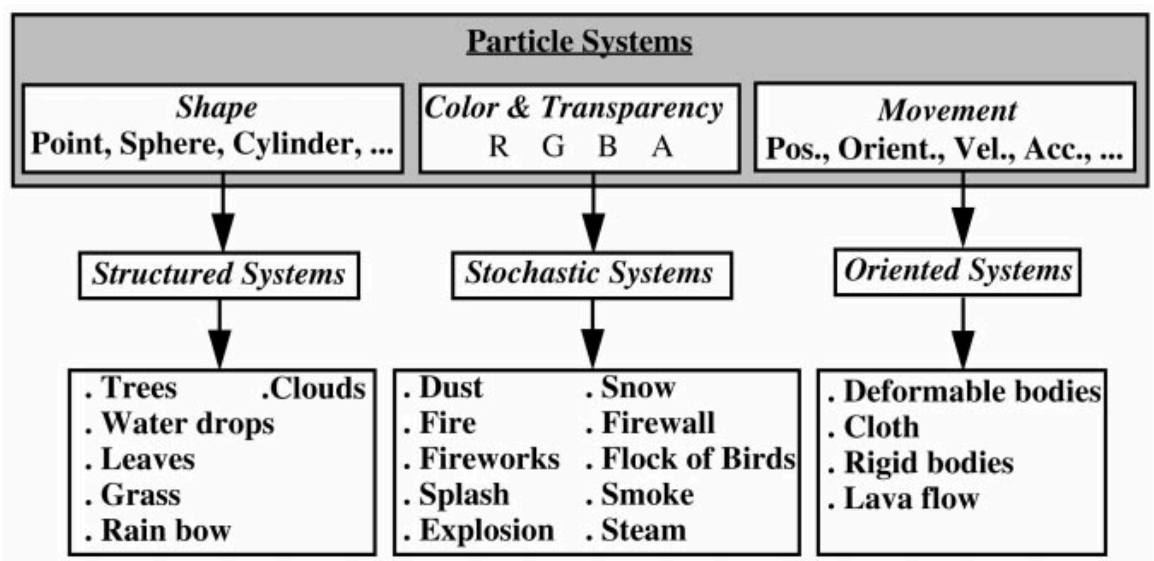

Figure 3.55: The applications of particle systems as illustrated by Chen.

Maya's traditional Dynamics system has been used for creating many types of visual effects; however, Maya 2009 introduced nDynamics that added an enhanced layer of features for particle systems. Under nDynamics is the nParticles system that has the same functionality as Particles, plus added features to make it more powerful [Keller10]. The ability of nParticles to interact with each other from multiple emitters by such means as event detecting, colliding and influencing is a major and powerful feature. nParticles are driven by a Nucleus solver with settings that control how simulated forces interact with emitted particles. Adjustments to gravity, air density and wind are at the core of the Nucleus solver and have profound effects on particle behavior. Both nParticle and traditional particle systems are extensive topics that cover a broad field of particle usage, including fluids, gases, soft/ridgid body dynamics and the destructive forces that create debris, fractures, cracks and shatter.

Both systems can also be used to create the natural phenomena of clouds. The nParticles system employs a more direct approach using emitters with different sets of attributes; however, the number of emitted particles needed for producing the effect of moving thick clouds is quite large. The use of nParticles for finer types of wispy or cirrus clouds is more appropriate and less processor intensive. A procedural method of volumetric fluids was therefore considered for modeling higher density natural phenomenon. Volumetric procedural models use a 3D volume density function that defines the density of a continuous 3D space [DEUSSEN04]. Ultimately, the Dynamic systems approach using fluid dynamics produced impressive and favorable results.

3.8.1 Adding Cloud Simulation

A 3D volumetric container with fluid was created in the Viewport and scaled to units large enough to surround the landscape by setting the grid size to (325, 150, 325) in the Container Properties fields. The 3D volume container shape was renamed "landscapeHaze" and was used to add atmospheric haze to the background. The 3D volume container was duplicated and translated above the landscape for use as clouds. The duplicated container was renamed "clouds1" and scaled and rotated to a desired perspective (Figure 3.56).

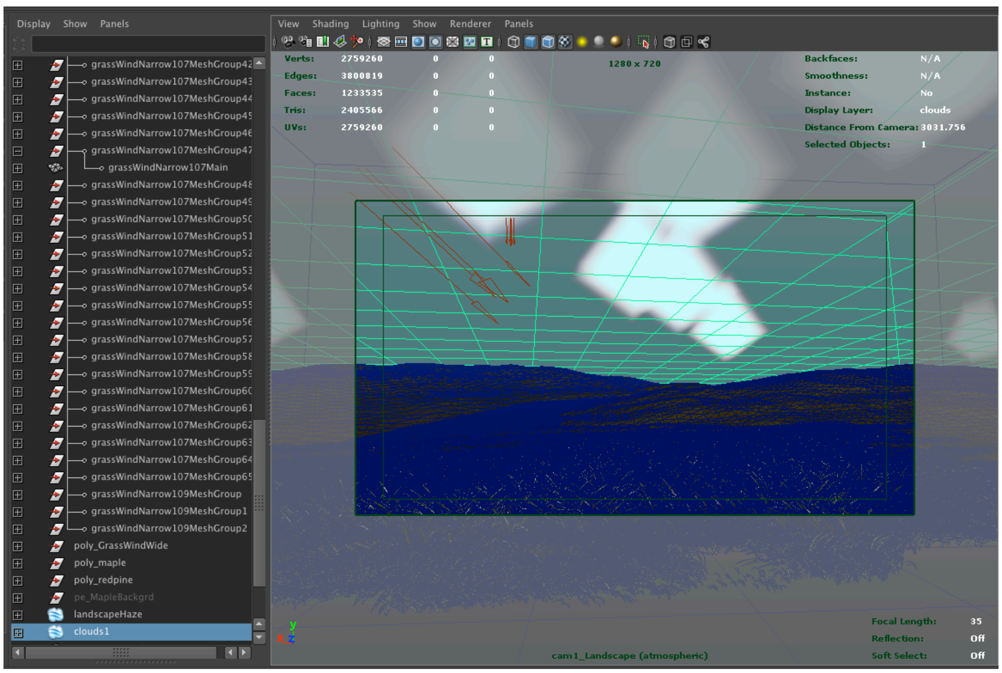

Figure 3.56: 3D grid container for cumulus cloud effects.

Creating realistic animated cumulus clouds is dependent on sensitive parameters to adjust a volume texture determined by a fractal pattern. The texture was created as a light gray to dark gray gradient ramp set in Y by the Incandescence panel, while its shape was graphed using the Opacity graph control. Three points, starting with the first input set at 0, allowed the texture to be transparent while the second and third set the threshold and position at which the texture formed. This graph is sensitive to adjustment; therefore, a wide range of undesirable results was expected across numerous test renderings. The

Textures panel includes most of the parameters needed to simulate the core of the clouds. The texture fractal type was appropriately changed from the default Perlin setting to Billow along with activating the booleans, Texture Incandescence and Texture Opacity, for extended control options. The following parameters within the Textures panel were most influential for controlling the shape of the fractal texture, most notably frequency and scale characteristics:

- Opacity Tex Gain .818
- Amplitude .728
- Ratio .486
- Frequency Ratio 3.081
- Depth Max 8
- Billow Density 1.5

Slight adjustments to these parameters produced drastically different results.

A constant value was replaced by a MEL expression for animating the cloud volume that was specified within the Texture Origin field along the Z axis. Texture Origin is the position of the texture within the container Z = -(time/20). This expression controls the Z position according to time, but changes slowly by dividing by a large constant. Figure: 3.57 shows the same expression for animating the cloud pattern while movement in Z was specified within the Texture Time attribute: clouds1Shape.textureTime = time/3. Documenting values throughout the test rendering was necessary for achieving final cloud form. An image of the final render is shown in Figure 3.58.

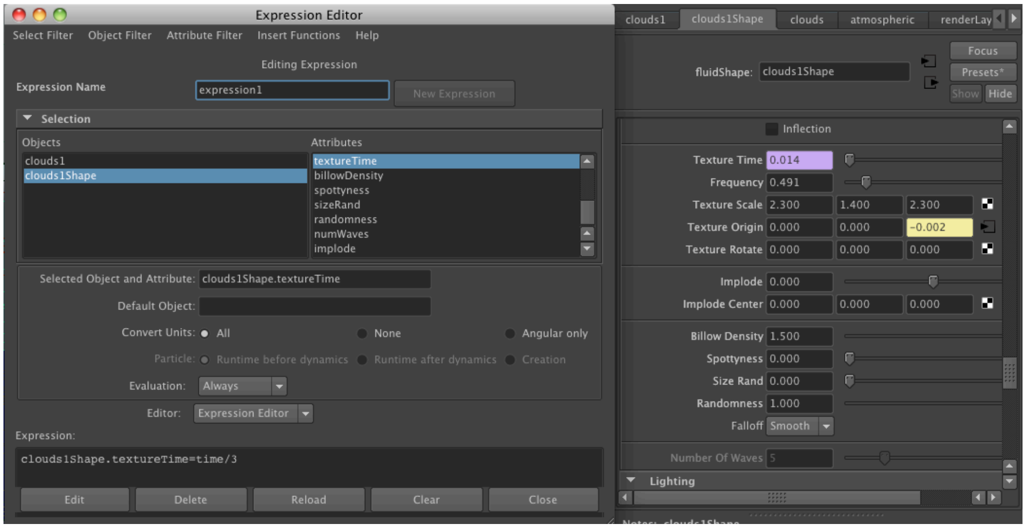

Figure 3.57: Expression settings for animating Texture Origin and Texture Time.


Figure 3.58: Rendered cloud results.

3.9 Rendering Workflow

This particular project utilized mental ray with three render layers, each containing shadow and diffuse passes. Each render layer contained contribution maps sharing the sunlight node for Physical Sun and Sky network. This sharing allowed for consistent lighting as frames were rendered.

3.9.1 Compositing

To avoid excessively long render times, some of the objects were rendered separately and composited later in Premiere. The top layer, consisting of the oak tree and grass, was composited over the mid-ground distant trees and hills, which was layered above the distant mountains and sky. In particular, the distant mountains had textures with baked illumination and Paint Effects Fog to simulate haze and enhance distance.

CHAPTER 4

CASE STUDY: FALL

The film *Fall* had a message of temptation relating to the third chapter in Genesis. The allegory within *Fall* called for creating an environment with a "Garden of Eden" look and feel referenced from photos of scenic desert views. Source images for the Eden look contained lush environments that were visually appealing. Procedural methods were used to develop much of the environment; however, most of the ecosystem felt barren with limited color. Pinpointing the significant transition to act II was difficult since both before and after environments were similar. Adding more Vue objects at the expense of slower system response time, and the inability to accurately place vegetation in high visual areas of the composition that would look aesthetically natural were issues with certain shots. The application of lush vegetation with added forces gave the landscape more life and helped to place the story. Using Paint Effects to procedural functionality, coupled with traditional modeling methods, provided a higher level of image fidelity. Also, these features in revised shots showed overall consistent lighting and cast shadows, which increased scene quality (Figure 4.1).

Figure 4.1: Procedural system usage in *Fall*.

As in the previous process, Paint Effects brushes were selected and organized for painting in certain regions that correspond to the perspective layout in the original opening shot (Figure 4.2). Modifying each species shader and applying forces to each object were later performed after converting the strokes to polygons. Multiple playblast animations were created with each force adjustment until the desired amount of simulated natural motion was achieved.

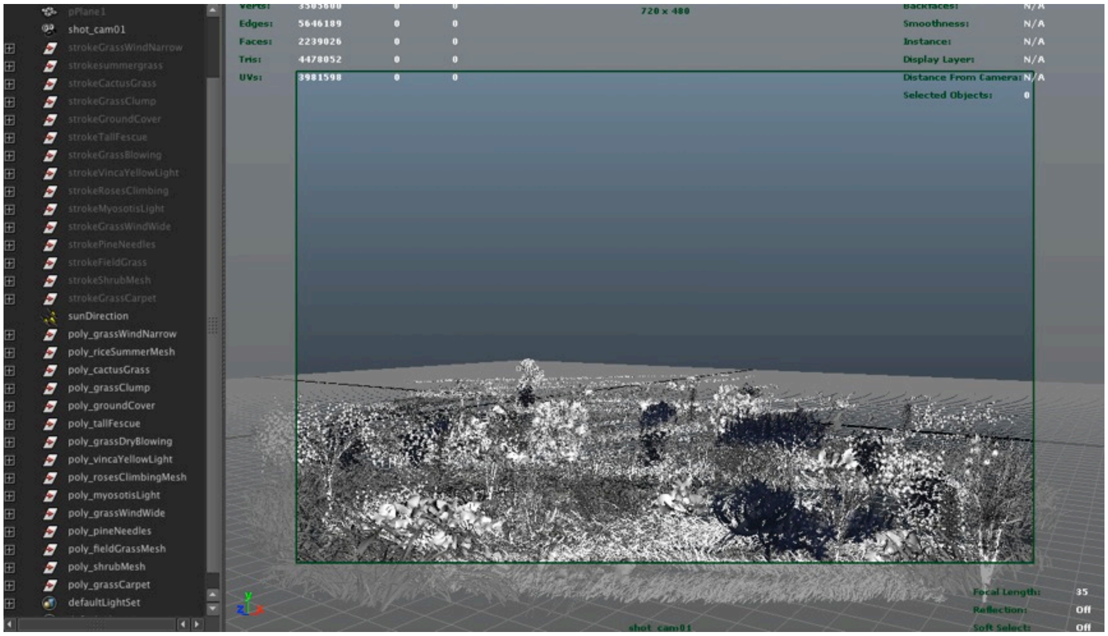

Figure 4.2: Perspective angle of opening *Fall* scene.

Initial geometry development consisted of first modeling the mountains, rock formations and cacti, then painting various Paint Effects strokes on specific render layers for organized output. Since Paint Effects can be rendered more quickly, creating dedicated render layers that contained brush strokes data instead of polygons objects, which would create a much higher mesh, allowed processing of each layer with override settings. The low-quality Paint Effects strokes were distant entities on the landscape; however, even at a distance, the disparity across strokes caused problems with lighting and color grade consistency. The use of simulated atmospherics, such as fog, to minimize contrast may have helped, but would not be appropriate for this scene. This issue arises in the rendering of Paint Effects in Maya Software since indirect lighting and raytraced shadows cannot be calculated. Without such qualities, the renderer produces an image similar to a vector illustration, with highly saturated color and hard-edged black shadows.

Using polygons for all objects and applying forces to each proved to be more efficient and allowed for more control overall within the composition (Figure 4.3).

The original shots in *Fall* did not include a Physical Sun and Sky lighting, but did include a similar type of directional lighting along with some strategically placed spotlights around the main character. Using directional lighting simulates the same lighting and shadowing conditions as the original scene. Inputting the same time of day as *Fall* produced interactive lighting results among animated objects (Figure 4.4).

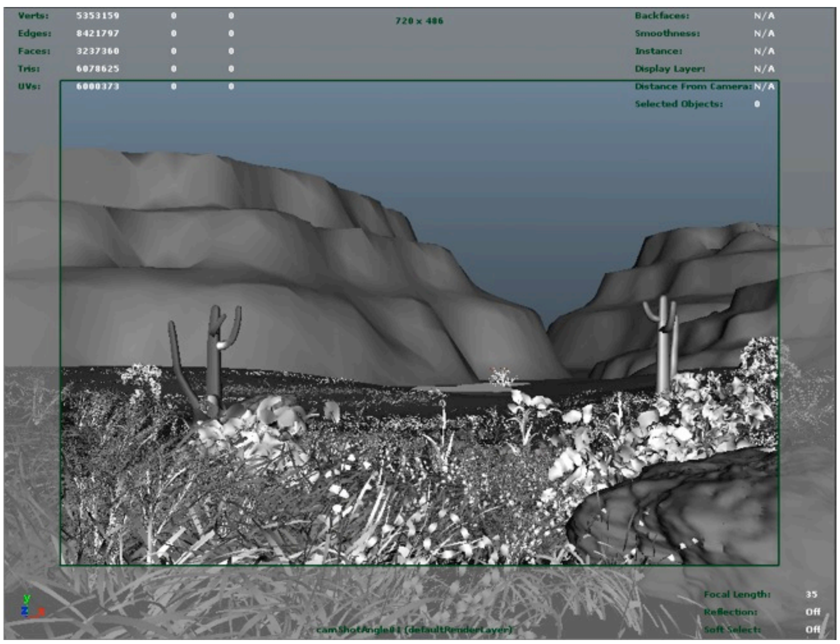

Figure 4.3: Scene with Paint Effects geometry with combined traditional modeling methods and forces applied.

Figure 4.4: Rendered frames composited into original *Fall* shot.

The creation of the main cactus tree was based on the tree brush stroke, treeSpiralMedium, and modified to an interesting degree of intertwining by adjusting parameters for Mesh, Twist and Tube panel using growth, segment control and displacement attributes (Figure 4.5). The idea was to give the cactus more character and relate it to the biblical reference by introducing a snake form that could naturally bring about a feeling of inquisitiveness from the main character. After separating and combining nodes of the cactus, the UVs were aligned for a cleaner layout. The geometry was then exported to ZBrush for sculpting finer detail. Deliverable sources from ZBrush included a painted texture map and displacement that were added to a custom shader network and rendered in mental ray. The color palette used a base neutral green that varied from slightly saturated olive tones to deep dark browns. The same lighting network and parameters were used to maintain consistency in lighting and color (Figure 4.6).

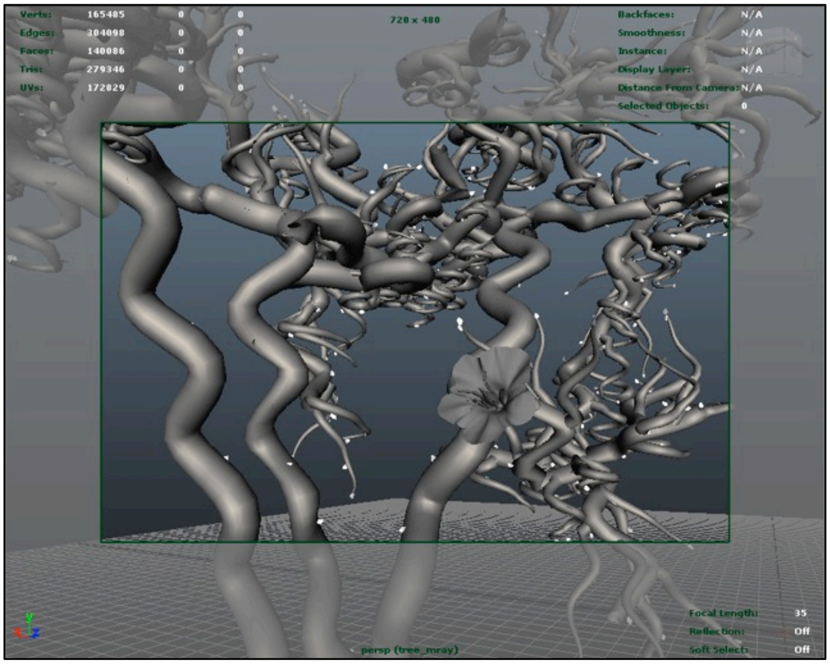

Figure 4.5: Paint Effects modifications to hero cactus tree.


Figure 4.6: Cactus tree with color and displacement map rendered in mental ray.

CONCLUSION

Examples shown from current films indicate how important procedural systems are in efficiently setting up natural environment scenes and the level of support needed to optimize them with live actors or CG characters. This thesis presented an approach for enhancing 3D natural environments starting with procedural objects that can be extended within Maya and combined with features in ZBrush. By leveraging the advantages of generated procedural objects, it is possible to establish a solid base from which to model vegetation. Also, establishing this base alleviated extensive modeling efforts and allowed focusing on refining other objects with higher visibility.

This thesis outlined opportunities for applying creativity to procedural objects for control within a natural 3D environment scene. Results from procedural application led to situations where optimizing textures, geometry, and lighting were needed to bring the scene to a higher level of detail. Later additions of procedural clouds within a 3D volume container helped to create a sky. Also, the preservation of forces on converted scene objects showed natural motion and served as an enhancement to scene fidelity. Altering the texture and geometry from applied Paint Effects brush objects showed creative control available for modifying model appearance. Procedurally generated textures were replaced with customized high-resolution photographic textures. Low-resolution models were exported from Maya and imported to ZBrush for added sculpted detail. The final models resulted in higher levels of fidelity from the application of high-resolution normal and displacement maps. Environment lighting was controlled inside the Physical Sun and Sky network with a strategically placed directional light for emphasis on foreground objects to cast deeper shadows. The combination of textures and lighting within the scene were rendered to photorealistic quality made possible through the extended capabilities of mental ray.

Fall benefited from the extended capabilities of Paint Effects by using the outlined methods from the previous research model. These methods resulted in the addition of elements that enhanced scenery and aided story context. Vegetation was aesthetically modified through custom shaders and textures maps. Forces were activated and set to add motion to the scene. The lighting was controlled through the Physical Sun and Sky network with settings to match the time of day in the original scene. Rendering in mental ray contributed to producing a higher quality composition that was later composited with the original footage.

REFERENCES

[CHEN99] Chen, Jim (1999). "Real-Time Simulation of Dust Behavior Generated by a Fast Traveling Vehicle." *Computer Graphics*. ACM TOMACS '99. April 1999. pp. 81.

[CHOPINE09] Chopine, Ami (2009). Vue 7: From the Ground Up: The Official Guide. Burlington, MA: Focal Press.

[CHALMERS03] Chalmers, Alan (2003). "Visual Attention Models for Producing High Fidelity Graphics Efficiently." *Computer Graphics*. ACM SCCG '03. April 2003. pp. 39-45.

[DEUSSEN98] Deussen, Oliver (1998). "Interactive Visualization of Complex Plant Ecosystems." *Computer Graphics*. ACM SIGGRAPH '98. July 1998. pp. 219.

[DEUSSEN98] Deussen, Oliver (1998). "Realistic Modeling and Rendering of Plant Ecosystems." *Computer Graphics*. ACM SIGGRAPH '98. July 1998. pp. 275.

[DEUSSEN04] Deussen, Oliver (2004). "The Elements of Nature: Interactive and Realistic Techniques." *Computer Graphics*. ACM SIGGRAPH '04. August 2004. pp. 7-15.

[GLENCROSS06] Glencross, Mashhuda (2006). "Exploiting Perception in High-fidelity Virtual Environments." *Computer Graphics*. ACM SIGGRAPH '06. July 2006. pp. 1-168

[HODGKINSON09] Hodgkinson, Gray (2009). "The Seduction of Realism." *Computer Graphics*. ACM SIGGRAPH ASIA '09. December 2009. pp. 1-4.

[KELLER10] Keller, Eric (2010). Mastering Autodesk Maya 2011. Hoboken, NJ: Sybex.

[KISACIKOGLU06] Kisacikoglu, Gokhan (2006). " Directing the Plant Interactions for 'Over The Hedge'." *Computer Graphics*. ACM SIGGRAPH '06 Papers. July 2006. pp. 49.

[ONG05] Ong J. Teong. (2005). "Terrain Generation Using Genetic Algorithms." *Computer Graphics*. GECCO '05. June 2005. pp. 1463.

[PRZEMYSLAW96] Przemyslaw, Prusinkiewicz (1996). The Algorithmic Beauty of Plants. New York, NY: Springer-Verlag. http://algorithmicbotany.org/papers/#abop (2004).

[REEVES83] Reeves, William T. (1983). "Particle Systems – A Technique for Modeling a Class of Fuzzy Objects." *Computer Graphics*. ACM SIGGRAPH '83. April 1983. 17:3. pp. 359-376.

[SCHRAND09] Schrand, Richard (2009). Vue 7: Beyond the Basics. Florence, KY: Course Technology PTR.

[SPENCER11] Spencer, Scott (2011). ZBrush Character Creation: Advanced Digital Sculpting. Hoboken, NJ: Sybex.

[SMELIK10] Smelik, Ruben (2010). "Integrating Procedural Generation and Manual Editing of Virtual Worlds." *Computer Graphics*. ACM PCGames '10. June 2010. pp. 1-8.