

PROCEDURAL GENERATION AND RENDERING OF TREES AND
LANDSCAPES IN THE STYLE OF EYVIND EARLE

A Thesis

by

LAURA KRISTINE MURPHY

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Chair of Committee, Philip Galanter
Committee Members, Ergun Akleman
John Keyser
Head of Department, Tim McLaughlin

May 2015

Major Subject: Visualization

Copyright 2015 Laura Kristine Murphy

ABSTRACT

In this thesis I develop methods of generating digital 3D landscapes in the style of the artist, Eyvind Earle, who is perhaps most well-known for his art direction and background paintings on *Sleeping Beauty*. I develop a variety of trees and other terrain elements, each tailored to match the graphic shapes and rendered accordingly to match the style of reference artwork. Creation of both terrain and trees can be highly generative in nature – complex in a way that lends to being defined by a logical systematic approach. I provide procedural methods for matching the shapes of the objects, relying on noise, L-systems, and other constraints. In general, the process is divided into base geometry generation and shading details. Shading methods include simple custom shaders and geometry-based stippling and linework. The various systems are implemented in Side Effects Software’s Houdini as its procedural capabilities allows the creation of many scenes with the same tools.

DEDICATION

To my friends and family.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Philip Galanter, for his guidance on this project. I would also like to thank the rest of my committee, Dr. Ergun Akleman and Dr. John Keyser, for their support as well.

The Texas A&M Department of Visualization has been an amazing and inspiring learning facility over my graduate (and undergraduate) career. Thank you to the Visualization faculty, staff, and students for making my time as a student truly wonderful.

I would also like to thank the Texas A&M Association of Former Students, the Department of Visualization, the Sony IPAX program, and the Pixar Aggies for their generosity in providing additional funding for my graduate studies.

Also thank you to the many friends that helped me edit this paper: Cameron Coker, Caleb Kicklighter, Brian Smith, and Daniel Freeman.

NOMENCLATURE

VizLab	The Texas A&M University Visualization Laboratory
Houdini	Side Effects Software's Houdini
VEX	Vector Expression. Houdini's expression language.
SOP	Houdini Surface Operator. A node that operates on geometry.
VOP	Houdini VEX Operator. A node that generates VEX code and operates on each point in the geometry. in parallel.
SHOP	Houdini Shader Operator. A node that provides a shader utility.
VOP SOP	A SOP that contains a VOP network
Scatter	A Houdini operation where points are scattered randomly across a surface or throughout a volume.
Copy	A Houdini operation that copies geometry to a set of points.
Serigraph	A screen printed artwork.

TABLE OF CONTENTS

	Page
ABSTRACT	ii
DEDICATION	iii
ACKNOWLEDGEMENTS	iv
NOMENCLATURE	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	ix
LIST OF TABLES	xiii
1. INTRODUCTION	1
2. BACKGROUND	4
2.1 The Life of Eyvind Earle	4
2.2 Stylistic Journey	5
2.3 Impact on Animation	8
2.3.1 Pocahontas	10
2.3.2 Open Season	10
2.3.3 Frozen	12
2.3.4 Maleficent	14
3. OBSERVATIONS	16
3.1 Painting Technique	16
3.1.1 Painting a Master Copy	19
3.2 Image Survey	21
3.2.1 Lighting, Shadows, and Atmosphere	21
3.2.2 Bushy Trees	24
3.2.3 Leafless Bushy Trees	25
3.2.4 Spruce Trees	26
3.2.5 Angular Trees	27
3.2.6 Forest	28

3.2.7	Cliff	28
3.2.8	Mountains and Hills	29
3.2.9	Ocean and Waves	29
3.2.10	Other Elements	31
4.	LITERATURE REVIEW	33
4.1	Tree Generation	33
4.2	Non-Photorealistic Rendering	35
4.3	Paintings as Reference	36
5.	METHODOLOGY	40
5.1	System Framework	40
5.2	Asset Creation Overview	42
5.2.1	Level-of-Detail	43
5.2.2	Shadows	43
5.2.3	Temporal Coherence	44
5.2.4	Stippling with Point Geometry	45
5.3	Bushy Trees	46
5.3.1	Base Geometry	46
5.3.2	Details and Lighting	48
5.3.3	Shadows	50
5.3.4	Level-of-Detail	51
5.3.5	Interface	52
5.4	Bare Trees	53
5.4.1	Angular Tree	54
5.4.2	Spruce Tree	54
5.4.3	Leafless Bushy Tree	57
5.5	Forest	59
5.6	Cliff	62
5.6.1	Base Geometry	62
5.6.2	Detail Geometry	62
5.7	Receding Mountains	64
5.7.1	Geometry	64
5.7.2	Shading	65
5.7.3	Ocean Rock Outcrops	66
5.8	Ocean	67
5.9	Wave	73
5.9.1	Base Geometry	74
5.9.2	Details and Shading	78
5.10	Additional Elements	83
5.11	Rendering	85

6. CONCLUSIONS	87
6.1 Overview	87
6.2 Still and Animated Work	88
6.3 Problems	92
6.4 Future Work	94
REFERENCES	97
APPENDIX A. L-SYSTEM INFORMATION	102
A.1 L-Systems in Houdini	102
A.2 Rule Sets	105

LIST OF FIGURES

FIGURE	Page
2.1 Pocahontas concept art by Michael Giaimo. Left from [34]. Right from [32]	9
2.2 Pocahontas concept art from [9]	9
2.3 Stills from <i>Pocahontas</i> [10]	9
2.4 Examples from <i>Open Season</i> [2][16]	11
2.5 Top: Costumes from <i>Frozen</i> . Bottom: Costumes from <i>Sleeping Beauty</i> .	12
2.6 Comparison between Eyvind Earle and <i>Frozen</i>	13
2.7 Left: Concept art from <i>Sleeping Beauty</i> . Right: still from <i>Maleficent</i> . [32].	14
2.8 Concept art by Dylan Cole for <i>Maleficent</i> , from [32].	15
3.1 Example process paintings from <i>Sleeping Beauty</i> . Images from [3]. . .	17
3.2 Eyvind Earle’s painting process (1991). Stills from [5].	18
3.3 A master copy I painted of <i>A Touch of Magic</i> (1997).	20
3.4 Eyvind Earle’s use of vibrant colors in his compositions.	22
3.5 Stylized shadows.	23
3.6 Atmospheric perspective and level-of-detail effects.	23
3.7 Bushy Trees.	24
3.8 Leafless Bushy Trees.	25
3.9 Spruce trees and other conifers.	26
3.10 Angular trees are bare leafless trees with straight branches and sharp angles.	27

3.11	Forest.	28
3.12	Cliff.	29
3.13	Mountains and hills.	30
3.14	Water and waves at a variety of distances from the viewer.	30
3.15	A variety of other features that are in landscapes.	31
4.1	Figures from <i>The Algorithmic Beauty of Plants</i> [27].	34
4.2	Landscape showing a variety of levels of details using graftals [24]. . .	37
4.3	Losure’s results: the Maya viewport preview and the corresponding final painterly render [22].	38
4.4	Stills from <i>Open Season</i> [2].	39
5.1	Left: example node network in the SOP context. Right: example node network in the VOP context.	40
5.2	Houdini gnomon and a cube.	41
5.3	Objects casting shadows. Left: a sphere casting a sphere shadow. Center: a teapot casting a teapot shadow. Right: a sphere casting a teapot shadow.	43
5.4	Popping sprites and two solutions.	45
5.5	Stippling technique: scatter and fuse.	46
5.6	Left: trees by Eyvind Earle in <i>Cachuma Ranch</i> (1999). Right: trees generated with my system.	47
5.7	Steps for creating the bushy tree base geometry.	48
5.8	The probability distribution for a direct light, and the corresponding detail points.	49
5.9	Left: Small stretched points. Center: Larger filler points. Right: Final black silhouette.	51
5.10	The tree with special shadow geometry.	51

5.11	Trees at varying distances with different amounts of detail. Top Row: trees from <i>Cachuma Ranch</i> . Bottom Row: trees generated with my system.	52
5.12	The node and interface for the Bushy Tree Digital Asset.	53
5.13	Top: My angular tree results. Bottom: Eyvind Earle’s Trees.	55
5.14	Top: My spruce tree results. Bottom: Eyvind Earle’s trees.	56
5.15	Top: My leafless bushy trees with 20, 40, and 60 generations. Middle: My leafless bushy trees with 12 generations. Bottom: Eyvind Earle’s trees.	58
5.16	Lighting components for the forest trees. Left to right: ambient, rim, direct, final.	60
5.17	A forest at different levels of detail.	61
5.18	Cliff base geometry.	63
5.19	Cliff detail geometry stages.	63
5.20	Top: my receding mountain ranges. Bottom: receding mountain ranges in <i>The Wave</i> (1990)	65
5.21	Mountain Range	66
5.22	The rock outcrop geometry is placed mostly below the water plane so than many outcrops can appear.	67
5.23	Left: my rock outcrops. Right: rock outcrops from <i>Enchanted Coast</i> (1980)	67
5.24	Top: my ocean. Bottom: Eyvind’s ocean in <i>The Wave</i> (1990)	68
5.25	Top: Ocean lines from the main camera. Bottom: Ocean lines from above.	70
5.26	Color ramp used for the ocean lines color palette.	71
5.27	Top: base colored ocean lines. Middle: proximity to coast – sea foam. Bottom: distance from coast – deep water.	72
5.28	Top: Eyvind Earle’s wave in <i>The Wave</i> (1990). Bottom: my wave.	74

5.29	Top: the subdivided plane. Middle: wave crest pattern. Bottom: wave with curl.	75
5.30	Top: major noise pattern. Middle: minor noise pattern. Bottom: major and minor pattern composite.	77
5.31	The wave curl process.	79
5.32	Top: spray points scattered on the wave geometry. Bottom: spray points in their final positions.	81
5.33	Color ramp used for the wave crest shader.	81
5.34	Seafoam Comparison. Top: seafoam in <i>The Wave</i> (1990). Bottom: seafoam render from my system.	82
5.35	Decaying seafoam pattern iteration steps.	84
5.36	Top: ramp for the decaying seafoam color. Bottom: ramp for the ocean water beneath.	84
5.37	Left: my cows. Right: Eyvind Earle's cows.	85
6.1	A cliff surrounded by forest.	89
6.2	A hill with trees.	90
6.3	Coast scene with a wave.	91
6.4	Render artifacts solved by switching from Mantra Micropolygon Render to Mantra Ray Tracing Render.	93
6.5	Examples of similar shapes across different aspects of Eyvind Earle's work.	94
6.6	Three example landscape types in Eyvind Earle's paintings. Left: <i>Purple Mist</i> (1989). Center: <i>Green Forest</i> (1989). Right: <i>A Touch of Magic</i> (1997).	95

LIST OF TABLES

TABLE	Page
5.1 Values used to generate subcanopy geometry at particular distances from camera.	49
5.2 Values used to generate the detail geometry at particular distances from camera.	50
A.1 Houdini L-system rule format terms [23]	102
A.2 Houdini L-system commands [23]	103
A.3 Houdini L-system variables [23]	104
A.4 Bushy tree trunk L-system rules. This rule set also happens to be one of the presets on the L-system SOP.	105
A.5 Angular tree L-system rules	106
A.6 Spruce tree L-system rules	107
A.7 Leafless bushy tree L-system rules	108
A.8 Hill shape L-system rules	109

1. INTRODUCTION

Eyvind Earle was one of the artistic giants of the golden age of animation, perhaps most well known for his work on *Sleeping Beauty* [4]. After many years of odd jobs and freelance artwork, he obtained a job at Disney as an assistant background painter. He was chosen to be the production designer and color stylist for *Sleeping Beauty*. After leaving Disney, he continued to work in entertainment for several more years, and he painted for the rest of his life.

Even though he only worked at Disney for about six and a half years, Eyvind Earle made a lasting impression on the animation community. His artwork and style continues to be referenced and used as inspiration [16][31][37]. Sony Pictures Imageworks used Eyvind’s art as a major stylistic inspiration for their first animated feature, *Open Season* (2006) [16]. Influence can be seen in the shape and branching structure of the trees in Disney’s *Frozen* (2013) [31]. *Maleficent* (2014) presents a new take on the story of sleeping beauty, which of course references Eyvind’s style and his original artwork for Disney’s *Sleeping Beauty* [37]. Further examples and discussion are in section 2.3.

The goal of this thesis is to characterize Eyvind Earle’s artwork into a system. This means to find examples of his work, analyze the style of specific elements, and create a module that will produce a digital 3D version of that element. This will result in digital 3D landscapes featuring a variety trees and landscape features that emulate Eyvind Earle’s 2D graphic style. The technical problems consist of non-photorealistic shading, tree modeling, and creating the scenes as a loose mapping of a 2D source material to a 3D result.

According to [11], “generative art refers to any art practice where the artist uses

a system, such as a set of natural language rules, a computer program, a machine, or other procedural invention, which is set into motion with some degree of autonomy contributing to or resulting in a completed work of art.” This thesis work falls under generative art as I generalize landscape systems with mathematics and randomness. I leverage node-based systems created with the 3D software package, Houdini, to create 3D landscapes that would be far too tedious for an artist to create by hand in a reasonable amount of time. Noise and other designed mathematical functions are allowed to take some control away from the artist.

Basing 3D computer generated images on artwork is a significant topic in computer graphics. The field of non-photorealistic rendering (NPR) covers any topic where life-like fidelity in the rendered image is not the goal. The research covers a variety of different looks, from hand drawn techniques, to graphic illustration, to painterly rendering, and much more. There are two main shading techniques that I employ. Firstly many of the systems have geometry specifically generated to function as a shader that creates stippling points or otherwise descriptive lines. This “detail geometry” typically sits on top of a flatly-colored “base geometry”. Secondly, when a traditional shader does apply, it is typically derived using toon shading concepts where the object color and light color do not necessarily interact in a physically realistic way.

Much work has been done previously on tree modeling, varying from stylized to realistic. L-systems offer a flexible (albeit complex) framework for generating branching structures. Trees with a prominent branching structure, I create with L-systems. Otherwise I define the tree canopies directly, attempting to match the graphic shape of reference artwork.

Geometry creation extends beyond trees to portray the terrain and other landscape features in the correct style. Following an initial study of Eyvind’s paintings

and techniques alongside relevant computer graphics research, I present a visual analysis that characterizes the shapes and textures used in his landscapes. I studied Eyvind's painting process through videos, descriptions, and creating a painting to gain further insight to the 2-dimensional construction of the shapes. This insight manifests itself in the way the shading and geometry creation are intertwined, optimal camera direction, and level-of-detail for some assets.

Combining the above considerations, I derive the most fitting methods of generating and rendering a 3D adaptation of Eyvind Earle's artwork. Ultimately, I present methods for four kinds of trees, a forest, cliffs, mountains, an ocean, waves, and a few general guidelines. Additionally, I present three larger scenes that combine multiple assets into rendered compositions.

2. BACKGROUND

2.1 The Life of Eyvind Earle

Eyvind Earle was born in April 1916 in New York. Although most of his childhood was spent in the Hollywood, California area, he spent three years in Europe in his early teens when his father rashly took him from his mother. Polio from childhood left him with a partially paralyzed face that kept him very shy through the years. He painted constantly through his youth, and even had a few exhibitions.

Eyvind's love of painting consumed him so much that he quit high school just 1.5 months shy of graduation. He got a job at United Artists Studio as an assistant so he could learn from the more experienced artists. Eyvind's pursuit of art lead him many places, including a year in Mexico, a bicycle journey across the United States, and eventually a job at Walt Disney Studios. Ultimately he was able to make a living though his prolific painting career.

In his younger days, Eyvind had joined the Navy where he met Alice Johnson. They married after their commission ended, and their daughter Kristin was born shortly afterwards. Their marriage lasted nearly 25 years until, sadly, Alice passed away from lung cancer. Eyvind met his second wife, Joan through a family friend. Their marriage happily lasted 28 years until Eyvind's death in July 2000. Joan has since passed away as well [18].

Eyvind "was an artist first and foremost" [12]. He worked in many mediums and truly dedicated himself to making great art. He is most widely known for his art direction and background paintings in *Sleeping Beauty*. As a professional artist he created many paintings, serigraphs, and greeting cards that are equally beautiful and influential in the artistic realm.

2.2 Stylistic Journey

Eyvind Earle was an intense worker, a trait that shows through the perseverance as an artist and through the immense detail present in his artwork [12]. At a young age, Eyvind was forced into a rigorous educational schedule where he had to paint every day, along with reading, and scholastic activities. His father instilled in him a drive to work hard and paint constantly [12]. Once free of his father, he continued to paint and train relentlessly with an ardor that persisted throughout his life. He continually worked long hours, driven by his desire to paint, make artwork, and succeed. Eyvind did not have a consistent work schedule, other than keeping himself busy with as much as he could handle. He frequently engaged in ambitious projects that required time much beyond regular hours to bring to completion. He also had many side projects that he would work on nights and weekends after his primary job (like his Christmas card designs and paintings while working at Disney).

Eyvind was exposed to art quite young as he wandered around his father's studio and through the MGM lots where his father worked as an on-set glass matte painter. While in Europe he painted landscapes, cows, barns, etc, as he was surrounded with the French countryside. His father taught him to sketch and to mix paint, but didn't teach much beyond that to allow Eyvind to develop artistically on his own [4].

Through his childhood and young-adulthood, Eyvind had always aimed for realism. A "famous" unnamed artist insulted his work and recommended a more emotional and expressionist approach, referencing Van Gogh, Gauguin, and Cezanne [4]. Eyvind took the advice to heart and "never again copied photographs" [4].

After high school, Eyvind took a job at the United Artist Studio (UAS), where he was a helper and took the opportunity to learn from the artists every moment he could. He was laid off after 10 months from UAS, and he decided to continue

his artistic training in Mexico. He moved to Tasco, Mexico for a year. Everyday he painted and drew for the entire year, just practicing and developing his own personal style. In his biography, Eyvind comments on the lack of form in his paintings at the time, and how throughout his career, he continued to have poor definition of form. In his professional works, however, this lack of form manifests itself as the graphic style for which he is known.

After returning from Mexico back to LA, Eyvind focused on painting watercolors in an attempt to get hired at Walt Disney Studios [4]. After a year of valiant effort painting “the most detailed painstaking watercolors of [his] life” leading up to his interview, the studio declined to offer him a job. Though disheartened, this only motivated him to practice harder and become an even better artist.

Eyvind found himself biking up to 100 miles a day from Hollywood or hitchhiking even farther, all to find a new inspiring landscape to capture with his brushes. He came to the conclusion that he ought to not return home every day or so and continue his journey right across the United States to find more diverse and inspiring scenery. So, he planned a trip to bike across the United States [4]. Starting on October 11, 1937 he rode his bicycle from Los Angeles, California all the way to New York. He biked around 3,000 miles through the American countryside, painting whenever inspiration and mood struck him. During his 45 day journey, he painted 42 watercolors, which he sold in an exhibition upon arriving in New York.

For the next few years in New York, Eyvind designed, printed, and sold screen-printed Christmas cards. He purchased a Chandler Price printing press, and found that the process was “a medium that fit [his] style completely and perfectly” [4]. Over winter 1947, Eyvind’s style took a change to be more like Albrecht Durer – intricate articulation of every detail [4]. In the years following, designing Christmas cards wasn’t always his primary job, but it was a task he kept up for more than 50

years, producing more than 800 cards [8].

Eyvind worked 2.5 years as a draftsman at an architecture studio in Hollywood. Under the guidance of George Vernon Russel, he “learned a hundred wonderful tricks” [4] that became part of his style and technique. Shortly afterwards he took classes at Art Center School in Los Angeles where he focused on improving his weakness in figure drawing.

Eyvind’s style finally solidified while he was working at Disney (around 1951) [4]. He eagerly painted long hours on *Sleeping Beauty* – full work days plus overtime – which gave him ample time to practice and perfect his style. For him, seven years at Disney was “the greatest art school in the whole world” [4]. Eyvind was especially influenced by his friend and peer, Tom Oreb.

After Disney, Eyvind dabbled for several years in commercial animation work, including freelancing and work for John Sutherland productions where he painted animations for industrial films. One film, “Rhapsody of Steel,” was set to music by the Pittsburgh Symphony Orchestra. Along side this, Eyvind became very interested in the fusion of music and art; he spent his free time experimenting with hand painted abstract animation aiming for a visual representation of the music. The trademark sequence for Universal Studios turned out to be the “most important step of [his] entire motion picture career” [4]. After all the work Eyvind put into it, he finally had mastery of the camera for practical compositing effects. He now had the skill to create the twinkling star of Bethlehem effect in his short “The Christmas Story” [4].

In the early 1960s, a friend from Disney Studios convinced Eyvind that he should start exhibiting his paintings again. Eyvind was skeptical, but after an interview on NBC and a successful show in Scottsdale, Arizona, he realized that he could live on painting alone. He didn’t need any other jobs, now he could just paint, all the time.

As his paintings gained popularity, Eyvind’s galleries were demanding paintings

faster than he could create them. In 1973, he decided to start silk screening some original works [4]. He could print many copies of these new works quickly. Again, they sold very well.

Eyvind continued to subtly refine his technique [5]. He didn't paint from life past his early years, only memory and imagination. He often revisited and painted the same mental images over the years. He rarely planned out the specifics of his paintings so that he could discover new technique variations by chance. He went through phases of mediums— casein, watercolor, oils, and often choosing acrylic for its convenience. Eventually he also switched from his previous mode of finishing one painting in a day or two to finishing forty simultaneously over a year [4]. In the video *Painting a Painting*, Eyvind describes his painting process as it stood in 1991, oil on masonite, with an acrylic underpainting [5].

2.3 Impact on Animation

Before Eyvind started his career at Walt Disney Animation, many of the artists were avid followers of his Christmas card designs and regarded his work highly. Eyvind quickly moved up in the company. Walt Disney loved his work [12]. Eyvind's style matched the trend that had been growing among the smaller studios and United Productions of America (UPA), changing from 19th century to 20th century style of animation [13]. As a color stylist (art director) Eyvind took the emerging style and brought it to a new level of quality – first with the short *Toot, Whistle, Plunk, and Boom* (which won an Academy Award), and then with *Sleeping Beauty*. Deeply ingrained in the childhood of the boomer generation [35], Eyvind's styling of *Sleeping Beauty* influenced the artistic style of new generations of artists [36].

Sleeping Beauty changed how the studio worked [36]. Eyvind was given complete control over the art direction of the film, much to the dismay of the animators who



Figure 2.1: Pocahontas concept art by Michael Giaimo. Left from [34]. Right from [32]



Figure 2.2: Pocahontas concept art from [9]



Figure 2.3: Stills from *Pocahontas* [10]

previously had much more control over the color and look of the subjects they were animating [4][36]. Eyvind’s tight control, backed strongly by Walt Disney, brought a completely unified look to the film while bringing the studio into the 20th century animation style. [14]. Tom Oreb designed the characters to match the style of the backgrounds – angular and strong verticals [13]. The total synthesis between background and characters was largely unprecedented [13].

Even though he only worked at Disney for about six and a half years, Eyvind Earle made a lasting impression on the animation community. His artwork and style continues to be referenced and used as inspiration. Below I detail several more recent examples.

2.3.1 *Pocahontas*

For Disney’s 1995 film, *Pocahontas*, art director Michael Giaimo “studied Eyvind Earle’s work to find a graphic language for the backgrounds” [32]. Michael Sporn comments that the forests in *Pocahontas* are taken right from Eyvind’s work, sighting the image in figure 2.1, additionally mentioning Eyvind’s influence on the film several times in his blog posts [36][34][33]. The influence is apparent in other pieces of concept art such as that shown in figure 2.2, as well as in stills from the final film shown in figure 2.3. The tree canopy shape and color palette of figure 2.2 are very reminiscent of Eyvind’s work.

2.3.2 *Open Season*

Sony Pictures Imageworks used Eyvind Earle’s art as a major source of stylistic influence for their first animated feature, *Open Season* (2006) [16]. The designers were drawn to the graphic quality of the tree shapes as well as the graphic light direction and shadows [1]. They attempted to keep the simple graphic quality through all aspects of the film to include the sets, character silhouettes, and effects.



(a) Receding mountains.



(b) Tree shapes and branching structure.



(c) Stylized skunk.

Figure 2.4: Examples from *Open Season* [2][16]



Figure 2.5: Top: Costumes from *Frozen*. Bottom: Costumes from *Sleeping Beauty*.

Figure 2.4a displays two examples of the use of atmospheric perspective with receding mountain ranges. Figure 2.4b has several examples of trees that bare resemblance of shape and branching structure to trees in Eyvind’s works. Figure 2.4c shows two views of a skunk from the movie; its graphic shape is similar to that of the many cows present in Eyvind’s paintings and serigraphs. Its tail, while not similar to the cow exhibits a distinct graphic quality as well.

Some additional discussion on technical aspects of the *Open Season* is in section 4.3.

2.3.3 *Frozen*

Frozen (2013) was also art directed by Michael Giaimo and draws inspiration from Eyvind’s styling of *Sleeping Beauty* [31]. Figure 2.5 shows the influence of the color palette on costume design by employing “analogous hues and subtle temperature shifts balanced with black” [31]. Additionally, the *Frozen* costumes draw from



(a) Tree concept art by Lisa Keene for *Frozen* from [31].



(b) Concept art by Michael Giaimo for *Frozen* from [31].



(c) Eyvind Earle's trees that bare a resemblance to the trees in *Frozen*.

Figure 2.6: Comparison between Eyvind Earle and *Frozen*.

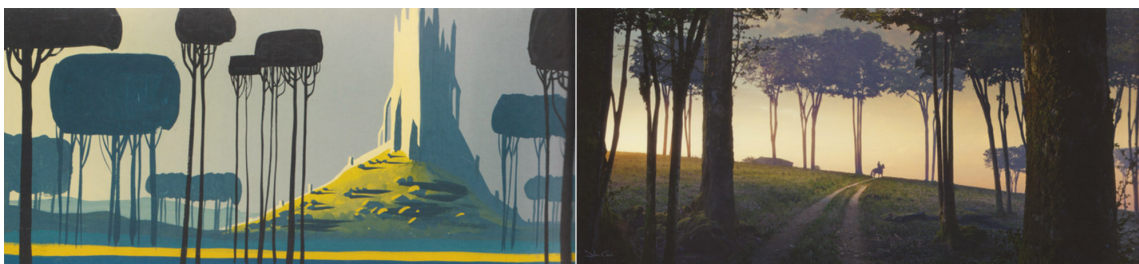


Figure 2.7: Left: Concept art from *Sleeping Beauty*. Right: still from *Maleficent*. [32].

Eyvind’s artwork by consisting of planes of color (especially black) embellished with tiny details.

Eyvind’s influence is also seen in the trees of *Frozen*. Figure 2.6a shows concept art by Lisa Keene that bares a striking similarity to Eyvind’s artwork. The way the snow piles up on the branches and the fractal-like patterns on the edges of the snow is very similar. The shape of each tree is similar as well. Concept art by Michael Giaimo in figure 2.6b again shows Eyvind Earle’s influence on his trees, with the needle-less pines and the rectangular canopies. I collected a few examples from Eyvind Earle’s work in figure 2.6c that bare visual similarity to the trees in figures 2.6a and 2.6b.

2.3.4 *Maleficent*

Maleficent (2014) presents a new take on the story of sleeping beauty, and it couldn’t have been designed without drawing from Eyvind’s style and referencing his original artwork for Disney’s *Sleeping Beauty* [32]. At the 2014 SIGGRAPH talk, presenters from Digital Domain specifically mention Eyvind Earle’s impact and present several side-by-side comparisons. Particularly heavy influence is seen in the trees in the human world (near the castle, and in the forest near the cottage)[37]. Figure 2.7 has one example from [32].

Concept art for the movie reflects this as well. Dylan Cole depicts a more re-



Figure 2.8: Concept art by Dylan Cole for *Maleficent*, from [32].

alistically rendered version of one of Eyvind's eucalyptus groves in his concept art in figure 2.8.1. Figure 2.8.2 displays another work by Dylan Cole that has many angular trees in the forest, again rendered realistically instead of stylized.

3. OBSERVATIONS

3.1 Painting Technique

Like many other painters practicing abstraction, Eyvind Earle had extensive training in representational painting which allowed him to subsequently abandon realism in pursuit of design creativity [4]. He paints from memory and imagination, stating, “Never paint anything until at least a year after you see it” [5]. This lets his intuition guide the painting process [5]. He avoids planning all the details to the end of his paintings as it would stifle his ability to unexpectedly create new effects [5].

His style solidified while working at Disney (around 1951) [4]. Eyvind’s technique of painting a “black shape with multiple tiny geometric patterns of dots” [4] is what earned him the job on *Sleeping Beauty*. The project “fit [him] like a glove” [4], as it allowed him to paint just as much for his personal enrichment as it did for the movie.

His process remained largely the same over the years, with some small variations as he changed materials and learned new techniques. Figure 3.1 shows Eyvind’s process from *Sleeping Beauty* in the mid 1950’s (about age 40). The style and process are very similar to that shown in figure 3.2 which was produced in 1991 (age 75). The mediums and specifics may have changed a bit, but the overall look is highly similar. Each has the patterned intricacy he is known for— planes of color sporting elaborate patterns. The light reveals the details and the shadows are left unarticulated [32][14]. The highly complex and simple graphic shapes fuse into paintings and illustrations that are still considered beautiful, gorgeous, and inspiring today [14][36].

Eyvind Earle goes over his painting process in his video *Painting A Painting* [5]. Figure 3.2 shows some images of him working on a painting of a eucalyptus grove. Eyvind’s painting method consists of these steps:

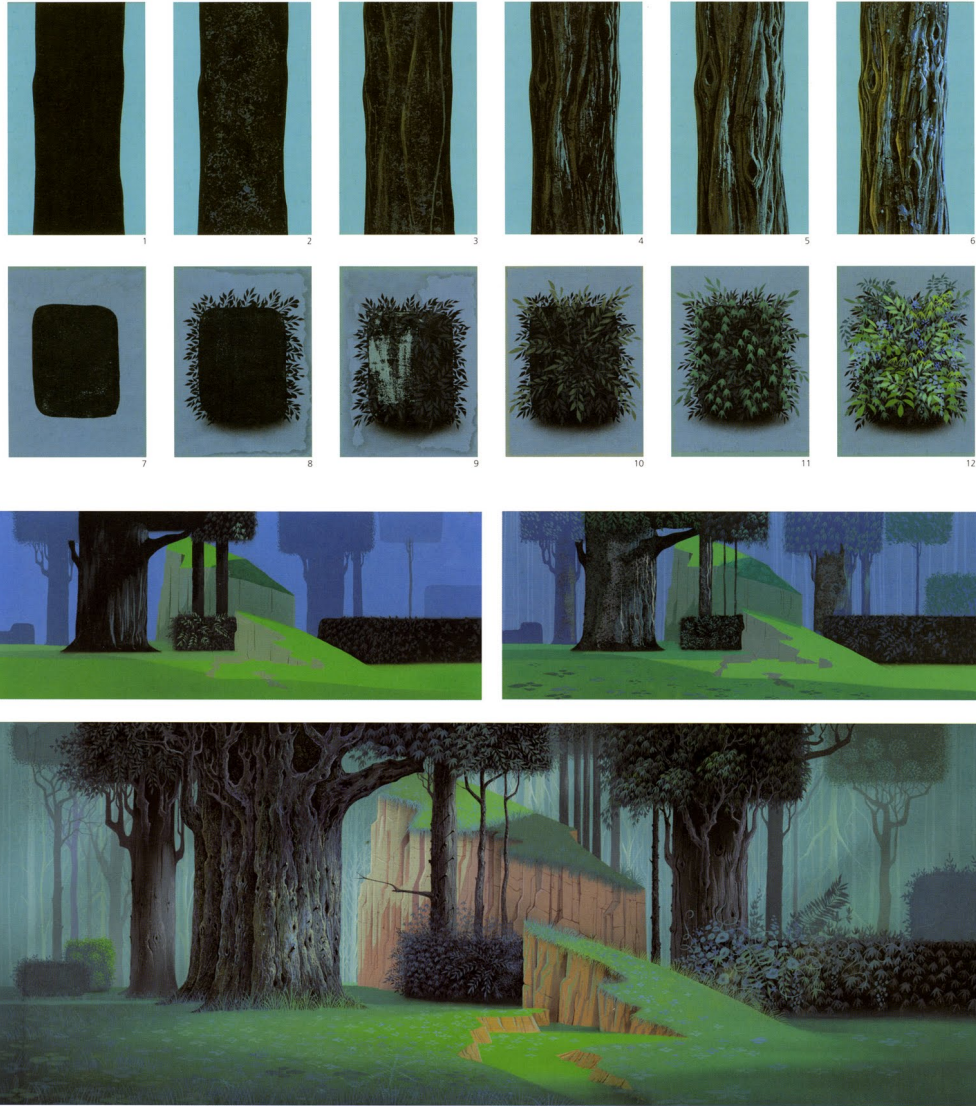


Figure 3.1: Example process paintings from *Sleeping Beauty*. Images from [3].

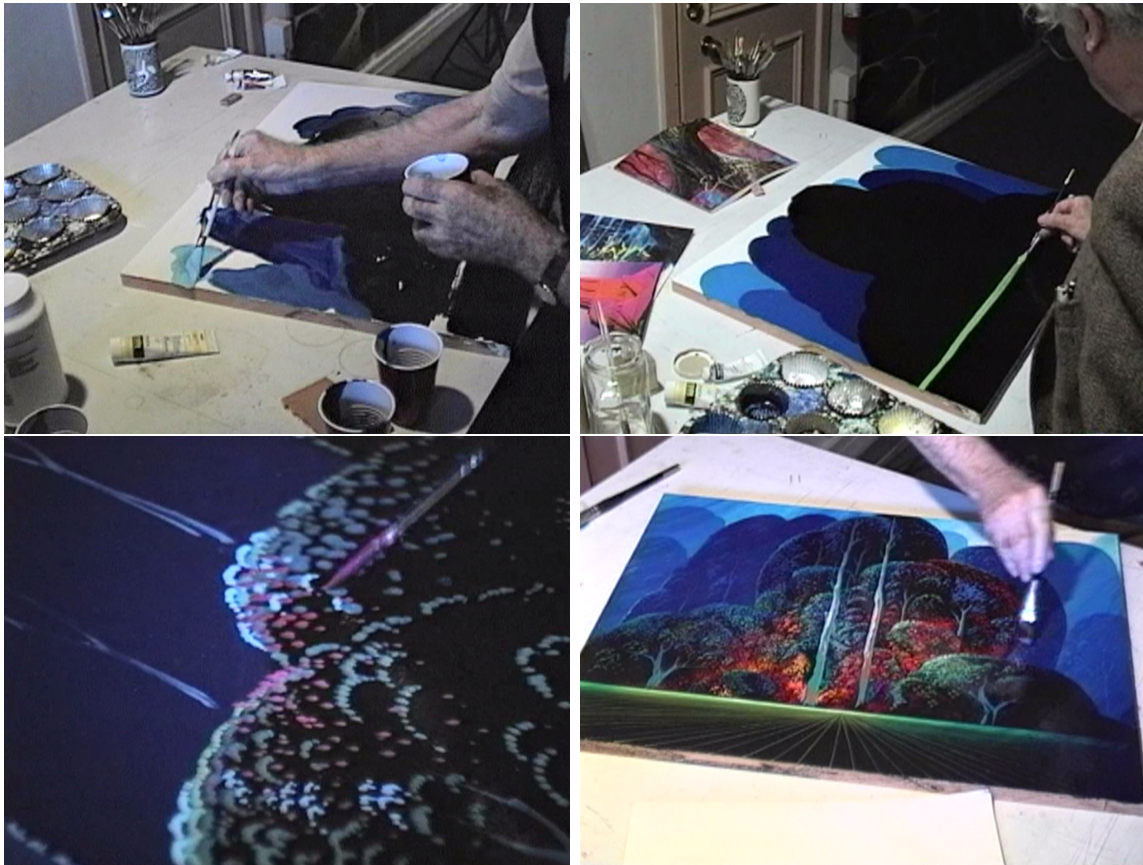


Figure 3.2: Eyvind Earle's painting process (1991). Stills from [5].

1. Rough out the layout.
2. Paint the large flat areas.
3. Add the fine details.
4. Add atmospheric effects and finish.

After initial preparation with gesso, he paints the rough shapes of the landscape with acrylic paint. Black goes first, then if necessary, lighter and lighter colors as the scene elements get farther away. After this rough pass with acrylics is dry, he switches to oil paints and paints over his sketch to refine the graphic outlines of his shapes. He paints the coats smooth and thin to keep the painting as smooth as

possible. Then he switches to a “fine delicate brush” to paint a multitude of tiny dots, ultimately completing the finest details with a pen. Once satisfied, he glazes the painting with a mixture of varnish and oil paints. This process further smooths the painting and grants the opportunity to add atmospheric effects such as light shafts and fog.

Based on my observations, the detail dots are painted with somewhat of an “outside-in” recursive fashion. He starts defining the outlines of the largest shapes within the silhouettes (e.i. separate trees). Once the detail is sufficient, he moves on to smaller subsections and subcanopies. He progressively defines smaller and smaller regions as he stipples the entire silhouette area. Additionally, as he moves from darker to lighter colors, he adds more details in the lit areas. This helps to define the form of the landscape structures. Figure 3.1 also illustrates the detail layering process.

3.1.1 *Painting a Master Copy*

I chose to paint a master copy of *A Touch of Magic* (1997). Figure 3.3 shows my final painting. I attempted to copy Eyvind’s methods as closely as possible, within practical limits. I used Winsor-Newton artist grade paints mixed with Liquin to speed drying times. I painted on a 4’ x 3’ piece of Masonite primed with white satin house paint.

Through producing this master copy, I gained a more thorough understanding of the painting process and technique. It was difficult to keep the painting smooth; my primer coat was slightly textured, and my acrylic underpainting had raised brush strokes resulting from rough paint application. Ideally I should have spent more time sanding, especially when preparing the masonite with primer paint. A few more iterations of painting and sanding would have ensured a smooth starting surface. Further



Figure 3.3: A master copy I painted of *A Touch of Magic* (1997).

difficultly stemmed from achieving clean lines and brush marks. It's important to achieve the right paint consistency along with reloading the brush with paint at the correct frequency. I often ended up with a brush that was too dry and left jagged marks, or with paint that had been thinned too much and dried too transparently. This was especially a problem with the many minute dots.

Besides technique-based problems, much difficultly came from struggling to see the details of the work through available reproductions. Working from a print reduced the color space. Most noticeably, dark colors disappeared and vibrant violets moved towards muted blues. Additionally, the resolution available in reproductions is not sufficient as most of the dots are as small as or smaller than one pixel. Fortu-

nately I was able to see the original in person, and I took a few detail photos which helped decipher some of the ambiguously described areas.

3.2 Image Survey

As part of my research, I visited Gallery 21 in Carmel-by-the-Sea, CA to view original Eyvind Earle paintings and serigraphs. The paintings have so much more detail in person than is visible in reproductions. The intricacy of the work hits you when you can finally see the hundreds of thousands of painted dots clearly and the elaborate layering effects of the paint. The subtlety of the dark colors and vibrancy of the brighter colors seem to elude capture with modern digital technology. The amount of work that Eyvind put into his paintings is almost unbelievable.

Additionally I studied two volumes of works ranging from 1940 to 2000 [6][7]. Digital copies of many of his artworks are also available online at [38] and [8]. The figures in this section are compilations of images retrieved from these stated sources as well as photographs I took of the artwork at Gallery 21.

Eyvind Earle's paintings feature a range of stylized trees and terrain elements. Below are examples of different types of objects that I include in my digital landscapes. I categorized them visually such that things that look similar or are made similarly are grouped together. Keep in mind this is not a survey of every painting, just ones that appeared most typical in style and content, with a slight bias appealing to my aesthetic interests.

3.2.1 *Lighting, Shadows, and Atmosphere*

Eyvind's vibrant use of color and light was one of the things that initially drew my attention to this project. The lighting is fantastical and not realistic. In many cases the lighting is completely unphysical. The colors are often highly saturated and placed in high contrast situations (e.g. bright dots on black). Several examples are



Figure 3.4: Eyvind Earle’s use of vibrant colors in his compositions.

in figure 3.4. Color palette and lighting design is not featured as part of this thesis, but is a necessary step that is considered when designing scenes to be rendered. The color usage certainly adds to the beauty of Eyvind’s work.

Included in the stylized lighting, much of Eyvind Earle’s work features stylized shadows (examples in figure 3.5) that are much simpler than the object casting the shadow. The shadows are often a different shape as well—pointy when the casting geometry is curved or vice versa. Long raking shadows are also quite common and add to the graphic quality of the work.

The introduction of atmospheric perspective and varying levels of detail adds depth to the artwork (examples in figure 3.6). Atmosphere stacks up in the scene making farther objects blend more with the atmosphere color. Level-of-detail adds to the atmosphere effect by making elements that are farther away (and thus occluded by more atmosphere) have less detail and a less complex shape. Each tree or



Figure 3.5: Stylized shadows.



Figure 3.6: Atmospheric perspective and level-of-detail effects.



Figure 3.7: Bushy Trees.

landscape feature compared to its partner at a greater depth is not just a smaller version of the same thing. It is actually a different form, modeled with less features and a generally smoother silhouette.

3.2.2 *Bushy Trees*

One of the many kinds of trees portrayed in Eyvind's work a tree I refer to as "bushy tree" as it is full and dense with leaves. Examples can be seen in figure 3.7. The bushy trees are generally defined with a flat back silhouette with details given by intricate point patterns and stippling. Generally, the trees are composed of a defined main canopy with a certain number of subcanopies. The canopies show variable amounts of self-similarity, where multiple levels of subcanopies within subcanopies depend on the level-of-detail of the trees and artistic choice for the particular piece of artwork.

For example, the trees on the right side of figure 3.7 are all from the same painting. These trees are also the main reference for the version of the bushy tree model that I created. As the trees recede into the distance they become less detailed. The closest



Figure 3.8: Leafless Bushy Trees.

tree has about 15 subcanopies, many branches, and high-frequency details in the canopy silhouette. Slightly farther trees have 3-5 subcanopies, 2-3 branches, and a much smoother silhouette. Farther trees range from 0-2 subcanopies, to just a rim light, to a gestural streak.

Bushy trees also vary significantly in height, ranging from the shorter half-moons and crescents to the much taller lumpy monoliths. However, generalizing this tool is outside of the scope of this thesis, so I will only be creating a tool for the shorter stand-alone tree version. This tree type also extends to a generalized forest, which I discuss later.

3.2.3 Leafless Bushy Trees

Figure 3.8 shows what appears to be a leafless version of the “bushy trees”, which I refer to as “leafless bushy trees”. These trees have smooth strongly defined silhouettes like the bushy trees. However, they are largely leafless which exposes the interior branches. The branching structure is very regular as the limbs reach out from the trunk towards the silhouette shape boundary. The trees are inherently planar

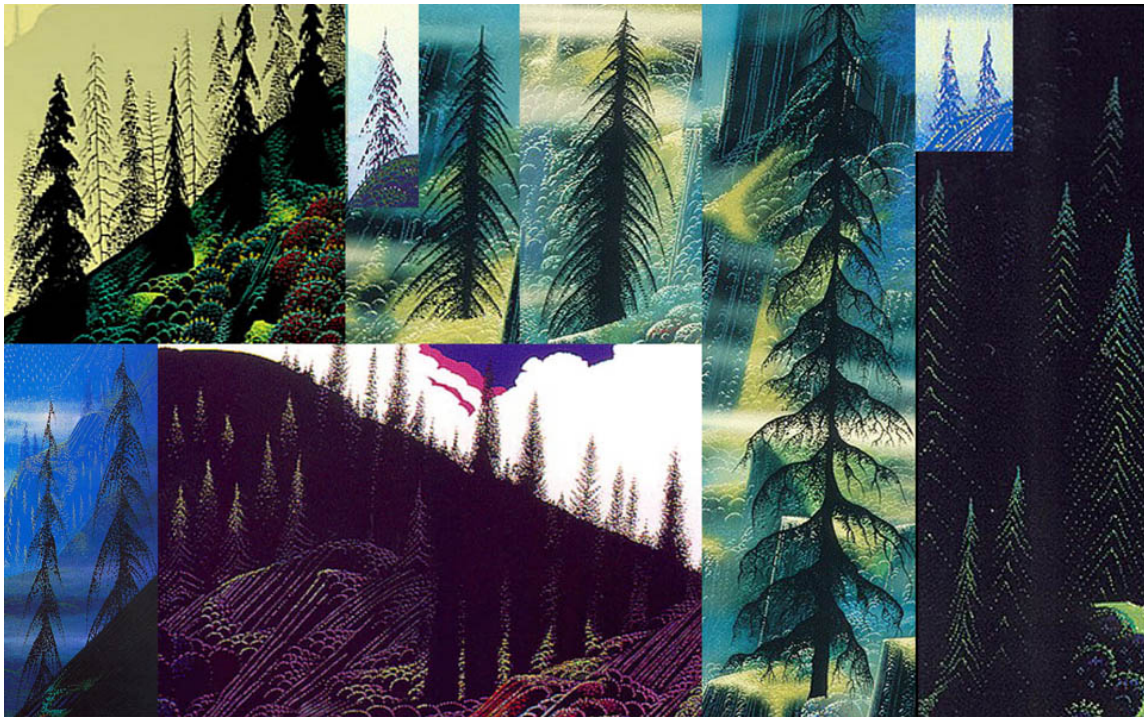


Figure 3.9: Spruce trees and other conifers.

due to graphic quality of their non-overlapping branches, and thus may require some creative license for the conversion to a 3D form.

3.2.4 *Spruce Trees*

Another category of trees are the spruces, firs, and other conifers (shown in Figure 3.9). These trees also have a well-defined visible structure that can be defined with a branching pattern (despite them not technically being bare and leafless). Each is roughly cone shaped, and is composed of lines or dots creating implied lines. Many are purely black silhouettes with no shading or influence from light. Lit spruce trees are typically defined with nested peaks (upside-down “V” shapes) composed of dots that hint at the existence of the many branches on the tree. For simplicity, I do not create a model to blend between lit and silhouetted conifers, just a silhouetted



Figure 3.10: Angular trees are bare leafless trees with straight branches and sharp angles.

version.

3.2.5 Angular Trees

The trees in Figure 3.10 feature long thin branches that sharply change direction at consistent angles. Due to the prominence of the angles in the structure, I refer to these trees as “Angular Trees”. The branches split off at approximately 30° to 45° angles (depending on the specific tree) and tend to reorient back to vertical. This results in mainly vertically oriented branches with short diagonal portions at their bases with an “elbow”. Some are topped with leaf canopies which are similar to the bushy trees explained above. However, in my version, I am only creating the branching structure.



Figure 3.11: Forest.

3.2.6 Forest

Eyvind's forest are defined as a flat areas of color with stippling defining the individual trees (shown in figure 3.11). The base color is black, but pushes towards an atmosphere color as the terrain recedes into the distance. The trees are defined in a similar manner to the bushy trees – many successive subcanopies defined with stippling. Similarly, the forest trees become less detailed with distance. The lighting reveals the trees which in turn describes the shape of the underlying terrain.

Many of the forests feature clearings of grass or mustard plants. The shape of the clearing is always very stylized with alternating curves and points, creating a highly organic graphic shape.

3.2.7 Cliff

The cliffs are composed of steep faces with small foliage-topped plateaus. The rock face is straight and the slope aligns along a main direction for each scene such that all the cliff faces are at approximately the same angle steepness. The cliffs are rendered with a black base color, and defined with colorful lines oriented up the steepest part of the cliff. The detail lines are often a saturated blue or green. Cliff examples are shown in figure 3.12.

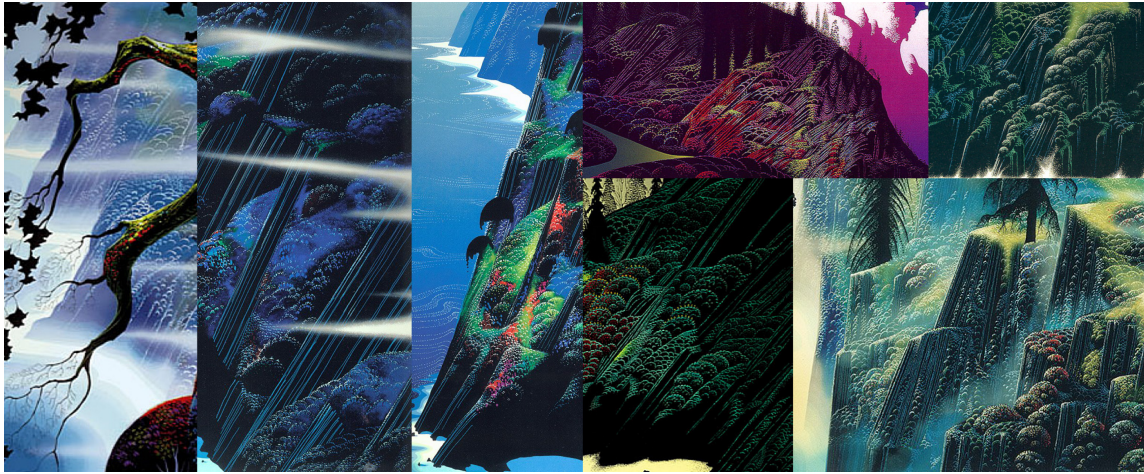


Figure 3.12: Cliff.

3.2.8 Mountains and Hills

Very common in Eyvind's works are receding mountain ranges and successive rolling hills (examples in figure 3.13). These exemplify depth in the scene as each successive piece of terrain changes color with the atmosphere as it retreats into the distance. Often the terrain becomes bluer as it goes farther back in space, but I have also included examples where it becomes yellow or purple.

3.2.9 Ocean and Waves

Figure 3.14 shows several instances of the ocean water and waves. A close view of the shore depicts a breaking wave with many details in its crest, including a fine spray mist splashing off the top created with many points. The decaying sea foam preceding the wave has a clear intricate pattern of its own.

This depiction contrasts with the medium range vantage point where the finer spray and detailed decay pattern are now hidden. Just the overall shape of the ocean spray is visible crashing against the land. The sea foam dissipates farther from shore, and is depicted as a smooth or stepped gradient as it fades to the ocean color.

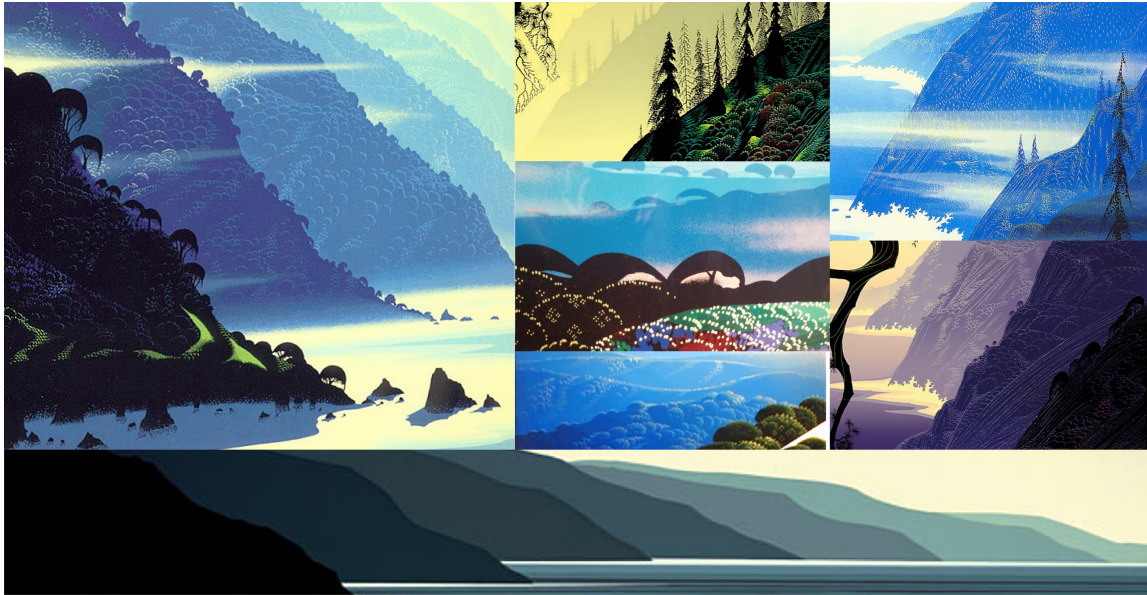


Figure 3.13: Mountains and hills.



Figure 3.14: Water and waves at a variety of distances from the viewer.

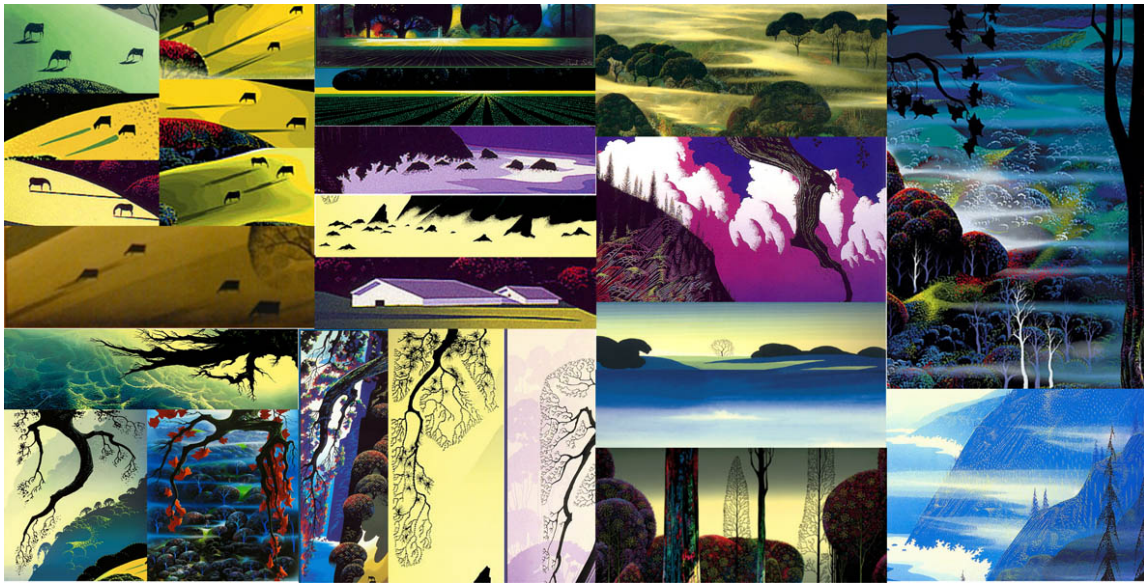


Figure 3.15: A variety of other features that are in landscapes.

From a distance, the water is portrayed as either fine horizontal lines or as a solid color with points defining the lines of wave crests and highlights. The breaking waves along shore are no longer distinguished. However the resulting whitewater is evident as a slight gradient towards white from the ocean color.

3.2.10 Other Elements

Figure 3.15 shows other elements featured in Eyvind Earle's paintings that may be considered "one-off" elements. Many of Eyvind's paintings feature cows or horses. They are painted as black silhouettes, sometimes with a rim light for extra definition. Other land elements are barns or other small buildings. Similarly in the ocean, there are a small amount of jagged rock outcrops protruding from the sea foam.

Many paintings feature flat fields before a grove of trees. This typically presents a dark foreground and that becomes a brightly lit middle ground directly front of the grove. Strong horizontal lines or stippled crop rows define the lit area of the field.

The sky is typically a vertical gradient or simply a solid atmosphere color. Sometimes there are puffy clouds in the sky as well. Many landscapes also feature transparent fog. In the extreme foreground, many paintings depict a detailed branch that shares a similar, but more intricate branching structure than the other trees in the scene.

4. LITERATURE REVIEW

4.1 Tree Generation

L-systems or Lindenmayer systems are a very common way to model branching structures in plants. They rely on iterative character string rewriting to create complex patterns and branching structures with a fractal quality. An L-system starts with an initial rule (axiom) and a series of replacement rules which are applied in succession for a certain number of iterations [27]. The resulting character string is, in turn, interpreted as a sequence of drawing instructions that create the branching structure. The complex interaction between rules makes the resulting structure difficult to predict. Determining rules that will generate a desired structure can be challenging and even counterintuitive.

Stochastic L-systems add a randomness component by allowing rules to occur with certain probabilities. This can make the result seem more natural by disguising the iterative pattern with randomized tweaks. Using a different random seed produces a different yet similar result.

Figure 4.1 shows a visual example of rule replacement, as well as a result derived from a stochastic rule set.

Shek et al. [30] at Walt Disney Animation Studios worked on art-directing procedural trees for Disney's *Tangled* (2013). They point out that L-system grammar is not an intuitive way for artists to create a particular look for a tree. At best, they can hard code rules and give artists sliders to change parameters and random seeds within the rule set.

Shek et al. created a tool to allow artists to first paint curves in the scene for the trunk and major branching structure, then supply a canopy shell used to grow

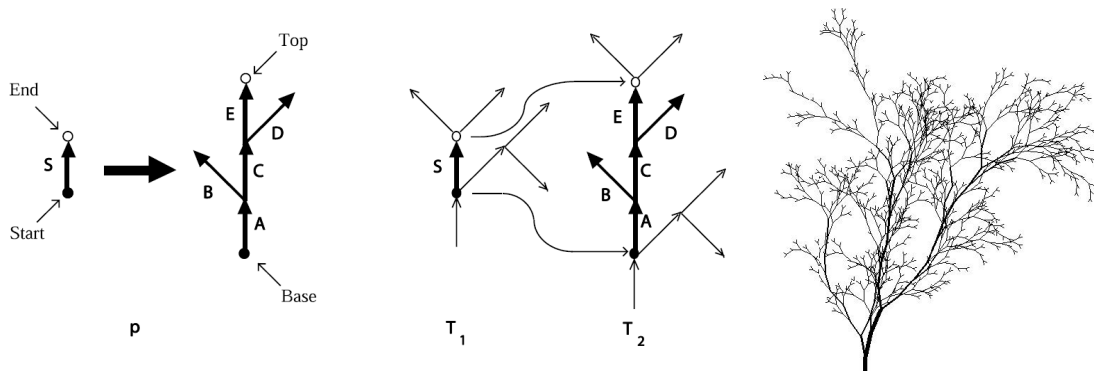


Figure 4.1: Figures from *The Algorithmic Beauty of Plants* [27].

smaller branches to fill the shell. The system, named Dendro, marches particles away from the major branching structure. At each step, the particle can split into two branches, or march out in a direction, filling the shell. The particle data is used to derive branching curves and widths that specify the tree's geometry. The system allows for tropism effects like gravity, and allows the user to specify a random seed to create different yet similar trees. One parameter controls how tightly the branches are clipped to the shell, creating looser or more tightly shaped trees. Dendro also has controls for leaf orientation.

Kuruc et al. [20] at Pixar, used a similar approach to Disney for creating art-directable trees. However, instead clipping branches to create the final stylized shape, they create a mapping that determines the start and end point of each branching route. Particles are marched out from their starting points, and branching occurs until it has reached all the desired end points.

Often canopies are fuller at the top and sparser at the bottom. With Kuruc's system the user can paint a density map on the canopy shells to achieve this effect as well as any other special change in density. The user can also specify leaf orientation using special normals on the canopy shell geometry.

Another method of building trees is derived from reconstructing point cloud data (such as LIDAR or Light RADAR), like as the work by Livny et al [21]. They start with point cloud data from scanned trees and construct a minimum weight spanning tree. The tree graph is then optimized to produce long thin smooth branches. Then the tree geometry is generated along the edges using associated vertex radii. Finally, smaller fine branches and leaves are synthesized using extracted L-system rules.

4.2 Non-Photorealistic Rendering

Using paintings as a reference is fairly common in animation. Much research has been done in the area of non-photorealistic rendering.

Barbara Meier at Walt Disney Feature Animation developed a painterly rendering technique involving particles [25]. Her system generates particles across a surface in 3D space and then renders them in 2D camera space as brush strokes. Particle/brush stroke properties are either defined and stored on the particle or looked up and calculated from a reference image. The image is temporally coherent because the particles (and brush strokes) are stuck to the geometry surfaces. This method avoids other painterly rendering issues such as the shower door problem (where brush strokes or other “painterly” noise is stuck to the canvas while the color moves through it). Each particle has a random seed that so they can have randomly determined property values that are temporally consistent.

Meier’s method presents some good techniques and an excellent basis for temporally coherent painterly rendering techniques. However, I will not have a more in depth discussion on painterly rendering techniques. Although Eyvind used oil paints to create many of his works, his painting methods diminish the character of the medium. He tried to make his paintings as smooth as possible with no brush strokes, and the shapes in his paintings are crisp and graphic and full of colorful stippling,

unlike many other oil paintings. The smooth finish and graphic nature of his works makes illustration techniques much more applicable than painterly rendering.

Gooch et al’s work on technical illustration offers a shading solution that emphasizes shape information [15]. Objects are simply shaded with a warm to cool color gradient instead of light to dark. The low contrast color scheme combined with outlines and clear object boundaries provides a graphic look that clearly communicates the shape.

Isenberg et al. has an overview of many pen-and-ink techniques that create a more graphic illustrative appearance [17]. Among them is Secord’s method of relaxing voronoi centroids to create evenly stippled surfaces [29]. Another is Schlechtweg’s technique which uses individual repelling agents (RenderBots) to maintain a certain distance between stipples [28]. Kim et al. present a newer method for stippling that samples examples drawn by an artist to analyze and synthesize the stippling dots [19], and thus the result is very natural looking.

Markosian et al. developed a stylized rendering system that allows dynamic level of detail for a set of defined graphic detail elements placed on geometry [24]. Example implementations include leaves, grass, and fur. The detail elements (“graftals”) are defined as various levels of detail with transitions. They are positioned over simplified geometry, and governed by “tufts” that control the visibility of each individual graftal. Overall, this method provides a temporally coherent way to transition the amount of detail rendered in a scene based on distance to the camera. Figure 4.2 shows a landscape rendered with their system.

4.3 Paintings as Reference

Finally here are two examples of projects that use paintings as reference.

In his master’s thesis, Michael Losure [22] developed a method to generate and



Figure 4.2: Landscape showing a variety of levels of details using graftals [24].

render trees (including atmospheric perspective effects) in the style of the painter Camille Carot. Some of his results are shown in figure 4.3. He made a series of tools for Maya written in MEL, which included parameters for level-of-detail. He used a simple grow and branch scheme: extrude face, potentially branch, and repeat. Leaves are represented and rendered as particles. Leaf particles form implied branches, as they are created along further branching paths, but no new branch geometry is created. Similar particles are also scattered throughout the scene so every object is rendered in the same manner.

The particles are rendered as sprites. The sprite pattern is 2D noise, masked with a softened circle and subsequently passed through a smooth threshold function (smoothstep).

To shade the trees, each particle keeps a lightness value that is used to look up the render color through a color spline. The color spline is determined by the user and functions as the color palette of an object. This also contributes to the painterly rendering style by being able to specify the color of the object as it goes from fully lit to completely in shadow. Lightness is affected by standard illumination, shadow

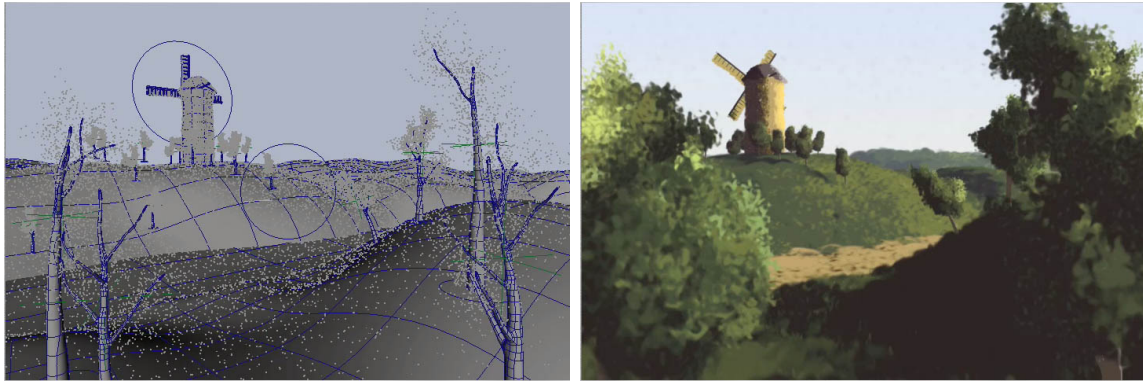


Figure 4.3: Losure’s results: the Maya viewport preview and the corresponding final painterly render [22].

maps, occlusion values, and user bias parameters.

To tackle level-of-detail, particles fade away with distance. By design, the particles that remain have lower noise frequency values. The particles are dynamically adjusted in size and color to account for the distance. While Losure determined a fully functional system for level-of-detail, I will only address level-of-detail for some things with no dynamic level-of-detail. Instead, I decided to focus on creating different kinds of landscapes and landscape elements.

The second example is Sony Pictures Animation’s *Open Season* [2]. Sony wished to try out the simpler graphic backgrounds of the 50’s instead of the lush, highly detailed backgrounds used at the time. Eyvind Earle was a major artistic reference for the creation of the movie [16], and although his backgrounds were highly detailed, they were also highly graphic. Figure 2.4 shows some trees from the movie that implement Eyvind’s style.

Imageworks developed a variety of tools and methods to create “long, raking shadows that focus the eye” [16] that are relatively unrelated to the geometry in the scene. One technique was to use different lights to render the lit elements than the



Figure 4.4: Stills from *Open Season* [2].

shadows. Another method involved rotoscoping and tracking to pin a new shadow to its 3D object. They also created tools to “arbitrarily paint shadows and project them into the scene” [16].

They created “a library of tree types, ages, and levels of detail that would vary based on proximity to camera, proximity to matte paintings, size in frame, director’s compositional notes, etc” [16]. Beyond the trees, the graphic simplicity influenced many character and object designs. They also used atmospheric effects and featured some similar compositional elements. One can also see the similarity in the rendered results to Eyvind’s artwork (see images in the Eyvind Earle Analysis section).

Although Eyvind’s work heavily influenced the style of the movie, it was after all a movie, with a budget, on a limited timetable, and built for entertainment and profit. Consequently, *Open Season* developed its own style for artistic and story purposes that became the ultimate design target, far more than the artwork that served as inspiration.

5. METHODOLOGY

5.1 System Framework

This project was implemented in Side Effects Software’s Houdini. The procedurally-oriented, node-based interface makes it a good candidate for vegetation and geological features where the model can be defined by a system or where a user may need many random variations of an object. Houdini’s widespread usage has the added benefit of existing integration into animation pipelines, like at Texas A&M University’s VizLab. These benefits indicate a satisfactory suite of tools available for 3D asset creation and rendering for this project. An alternative to Houdini was Autodesk Maya, another widely used 3D animation software package. Houdini’s procedural nature and lack of significant drawbacks made it the better choice for this project.

Within Houdini, each asset is created with a node network (see example networks in figure 5.1). Optionally, a node network can be packaged into a Digital Asset residing in an Operator Type Library (OTL). A Digital Asset is a custom-made node that has been saved to disk and can be used in conjunction with Houdini’s other nodes. Nodes, including Digital Assets can have an arbitrary set of control

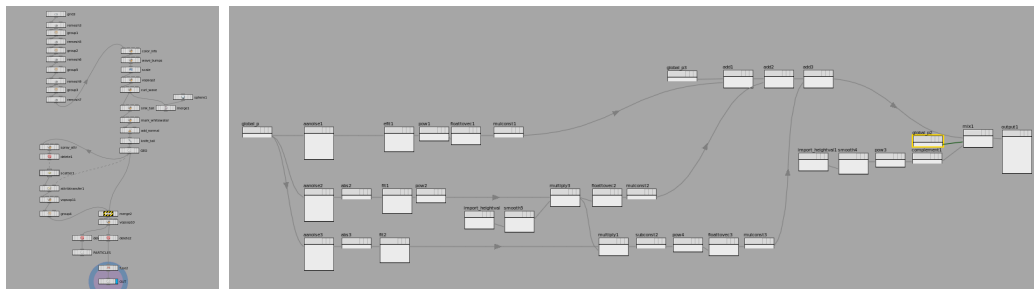


Figure 5.1: Left: example node network in the SOP context. Right: example node network in the VOP context.

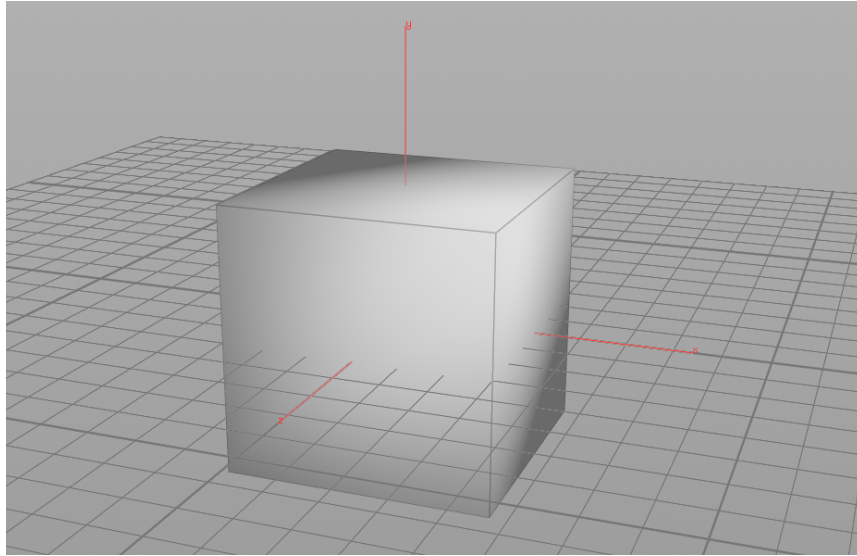


Figure 5.2: Houdini gnomon and a cube.

parameters defined by the creator of the node. It is not always necessary to create a digital asset from a node network since the node-based interface actually makes any given portion of your network highly portable.

The Houdini coordinate space is a right-handed system. The y-axis is up; the z-axis is forward; and the x-axis is right. This is important to note for clarity of my explanations. I may refer to a direction by either its axis name or its direction name. Figure 5.2 shows the Houdini gnomon indicating the x-, y-, and z-axes.

I created a variety of assets that incorporate into a few final scenes for composition and rendering. The assets for each scene are made from a combination of digital assets and node networks with varying levels of specificity to the scene. The networks are composed of SOP (Surface Operator) and VOP (Vex Operator) nodes; see figure 5.1 for example networks of each. Some assets' networks are tweaked per scene; some are uniform across all scenes. Figure 5.5, in the next section, shows a simple example of a node network that is altered and used in several places throughout this project.

The assets and scenes are rendered with Mantra, Houdini’s built-in render engine. For some assets it was necessary to write custom shaders in the SHOP (Shader Operator) context. A SHOP node network builds the VEX (Vector Expression) code for the shader.

5.2 Asset Creation Overview

I have developed systems within Houdini for a variety of trees, several geological features, and a static (not animated) ocean. The result is 3D geometry, including custom shaders or custom geometry with embedded shading information. Each individual system asset is placed manually (or semi-manually) within a scene for rendering. In general, my methods are similar to Eyvind Earle’s in that most of the asset models follow the steps: base silhouette, and then details.

Because the look is so stylized, I chose to create much of the detail with geometry and not within the shaders. Thus, the shading is closely tied to the geometry generation. To achieve stippling and contour lines, point and line geometry (respectively) is generated to create the effect. I will go over the specifics for each asset system individually in further sections of this chapter.

The lighting is closely tied to the geometry generation as well. In many assets, the lighting is indicated only in the details, which means the detail geometry must exhibit the appropriate lighting placement, shape, and color. For example, the details of the trees articulated with stippling are revealed as a lighting effect.

Keep in mind, these are generative systems, leveraging noise and other mathematics to describe the form and details. This way of creation has the advantage of reducing manual labor by allowing the computer to execute many tasks defined by the generative rules, instead of requiring the artist to do everything. Thus this allows for more complex schemes that would be nearly impossible for an artist to complete



Figure 5.3: Objects casting shadows. Left: a sphere casting a sphere shadow. Center: a teapot casting a teapot shadow. Right: a sphere casting a teapot shadow.

without aid due to the volume of data and tasks. Variations of each system's results are driven by randomness, again reducing artist labor to obtain options or variety.

5.2.1 Level-of-Detail

Trees and other landscape elements are generated differently or with different parameters depending on the distance from the object to the image plane in order to address level-of-detail (LOD is the amount of detail depicting an object based on how distance to the camera). The change in LOD matches as closely as possible to the artist reference. Recall figure 3.6 from section 3.2 has a examples of how Eyvind depicts his trees differently depending on their depth in the scene.

This thesis does not include an all encompassing level-of-detail system that considers every case for each landscape item developed, as it is outside the scope of this project. Furthermore, LOD does not change dynamically across the animation. Once an asset instance is created with a particular amount of detail, it remains unchanged.

5.2.2 Shadows

For some assets, it is necessary to create separate shadow geometry; a different object casts a shadow than the object that is visible. As stated in section 3.2 (see

figure 3.5), often the shadows are simpler than the casting object, or a different shape entirely.

In this case, the primary geometry is disabled from casting shadows. The shadow geometry is set to Houdini’s “phantom” setting which means the geometry cannot be seen directly, but it is still available to cast shadows. The lights used in the scene specify which geometry is allowed to cast shadows, and in general needs to be updated appropriately.

Figure 5.3 has a simple example with three shadow casting objects, including one using alternate shadow geometry. See figure 5.10 for an example used for the trees in this project.

5.2.3 Temporal Coherence

Temporal coherence in rendering animation refers to the consistency of the images over time. With many stylized rendering techniques, temporal coherence is a big issue since procedural techniques often make use of noise and randomness. To avoid undesirable sudden changes in the rendered imagery, the creator needs to specify how the pattern should change as the camera or objects move and change.

In my case I only have a moving camera and no changing or deforming objects. The major coherence problem that remains is popping— where one sprite suddenly jumps in front of another as their distance from the camera changes and it is suddenly rendered in front of a sprite it was previously behind. I employ two solutions to combat sprite popping, which are also illustrated in figure 5.4. The first method (enforce maximum color difference) is to make pops imperceptible, rather than removing them. In any pair of points that are the same color (or very close in color), the depth swap will not be noticeable. The second method (enforce minimum distance) is to ensure that all points are farther apart than the point radii; this ensures that no pair

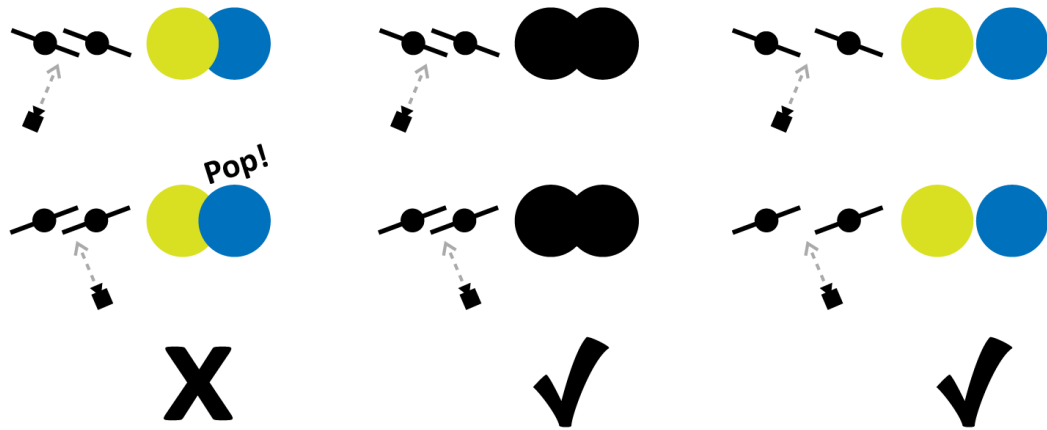


Figure 5.4: Popping sprites and two solutions.

of points will overlap when they change depth order. However this method can only be used when the necessary point density is below a certain level, as it requires a minimum distance between points.

5.2.4 Stippling with Point Geometry

To create stippling, my method hinges on creating relatively uniformly distributed points to act as the stipples across the surface. First generate many more randomly distributed points than are needed. Next, remove points that are too close together. The result is a set of points that are spaced approximately evenly without being in any kind of pattern or grid. In Houdini, the method involves two SOPs: scatter and fuse, for the first and second steps, respectively.

Figure 5.5.1 shows first a 10 by 10 unit square with 5000 points scattered on it. Then 5.5.2 shows the same set of points with a fuse radius of .3 such that no two points are closer than .3 units away from each other. Points that violate the length constraint are deleted. After the fuse node, the square is left with 665 points. Figure 5.5.3 shows the same amount of points (665) scattered across a 10 by 10 unit square.

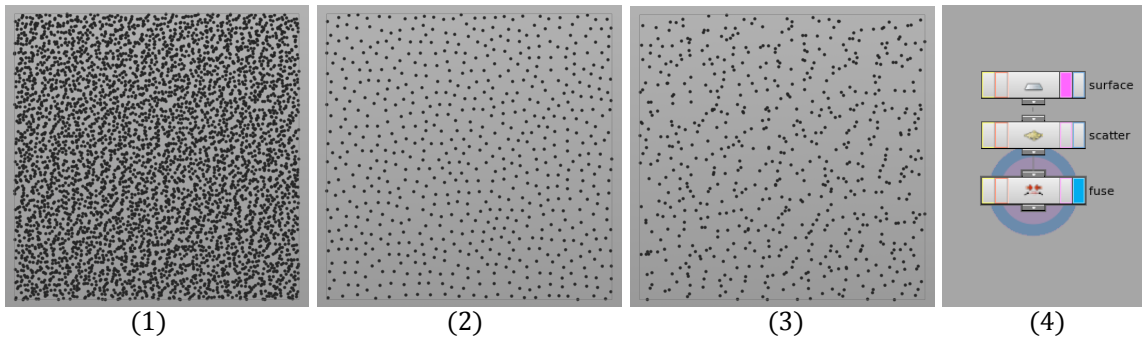


Figure 5.5: Stippling technique: scatter and fuse.

The result of figure 5.5.2 is much more even and thus better suited to simulate the semi-uniformly distributed points of hand drawn stippling than the patchy result of figure 5.5.3. An example node network used in this process is shown in figure 5.5.4. It shows the basic data flow: given the input surface (node labeled “surface”), scatter points with the Scatter SOP, then remove points with the Fuse SOP.

5.3 Bushy Trees

This tree model covers the mid to far-range leafy trees, similar to those in figure 3.7. The close trees and distant ones are excluded from the model. These trees are wider than they are tall, relatively short, and covered with leaves. Figure 5.6 has a side-by-side comparison between the version of this type of tree created with my method versus Eyvind Earle’s version. I presented an earlier version of the bushy tree model ACM SIGGRAPH 2014 Posters [26].

5.3.1 Base Geometry

The tree silhouette is comprised of two parts: the canopy, and the trunk. Both are rendered flat black with a Houdini constant shader. This provides both a background and a base for the detail geometry to build upon. Figure 5.7 shows three stages of the base geometry creation.

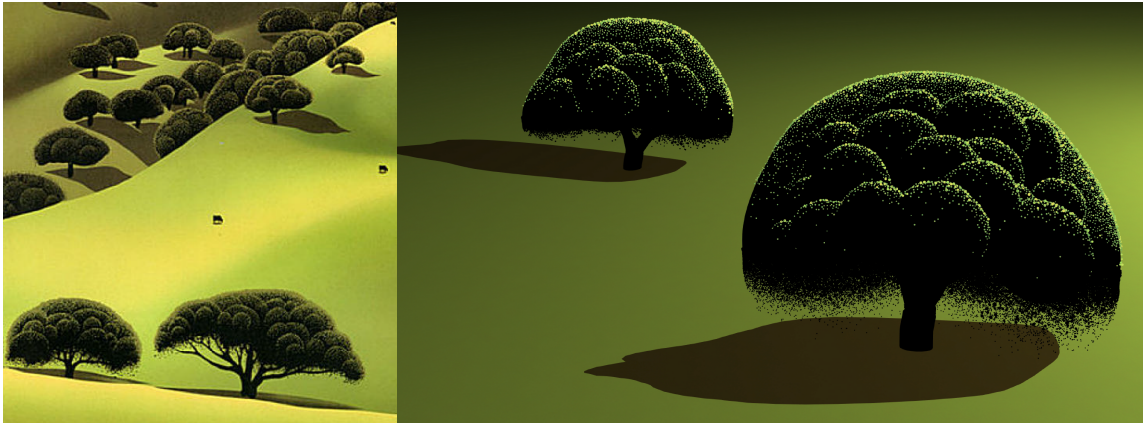


Figure 5.6: Left: trees by Eyvind Earle in *Cachuma Ranch* (1999). Right: trees generated with my system.

The trunk is a simple L-system generated with the rule set shown in table A.4 of Appendix A.2. The rules are fairly simple because the branches are mostly occluded by the canopy. Only a hint of the underlying structure is needed to effectively convey the tree. Tubes are generated along L-system splines to give appropriate width to the trunk and branching structure.

A 3-dimensional metaball is a function of distance in space that can be combined with other metaballs to create a smooth surface. The main canopy is made of three metaballs that are slightly jittered in size and position per tree instance. The specific random values that make each tree unique are based on the position of the tree in 3D space and an optional random seed. Multiple subcanopies are created by scattering spheres across the base canopy geometry and applying a fuse operation to remove points that are too close together (as described in section 5.2.4). The subcanopy spheres are scaled in the direction of the camera such that the spheres become flatter. This prevents undesirable patterns in the detail geometry due to subcanopy intersections.

The number and size of each subcanopy depends of the distance from the camera.

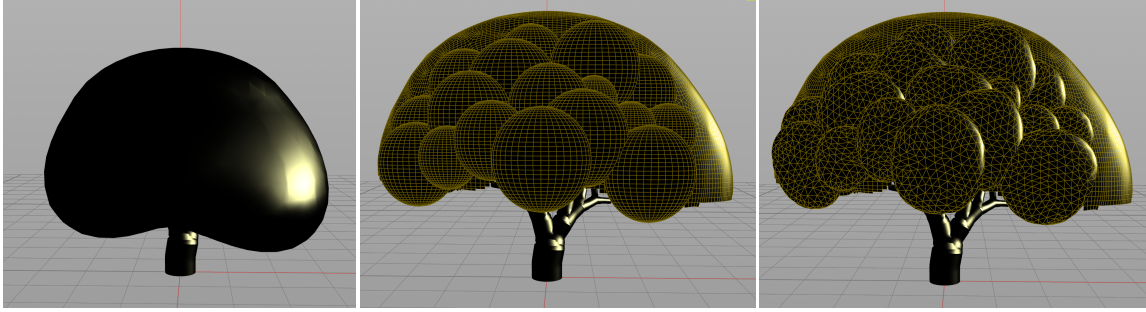


Figure 5.7: Steps for creating the bushy tree base geometry.

The more distant the tree is in the scene, there are less subcanopies, but they are bigger to account for perspective. Table 5.1 shows the different values used for different distances. Distance from camera is measured from the specified camera location to the origin of the tree.

The initial three-metaball base tree canopy is scaled slightly larger in the camera image so it overtakes the silhouettes of all the subcanopies. It is also flattened in the direction of the camera so it does not obscure or intersect the subcanopies. The bottom is trimmed such that any face with a downward component is removed to allow for detail geometry hanging off the bottom of the tree.

Finally noise is applied to add variation to the many spherical surfaces present in the tree. It is important that the subcanopies not be spherical since Eyvind did not paint them exactly round, but as imperfect spherical shapes. The displacement is applied in the direction of the surface normal, and the noise dictates the amount (executed via the Mountain SOP). I do not update the normals after displacement so that the original smoother normals can be used in the detail geometry calculations.

5.3.2 Details and Lighting

The base geometry including the subcanopies functions as a scaffold for the detail geometry. The points act as the leafy green colored and lit part of the trees. The

Camera Distance	Fuse	Scale
9.3	0.40	0.99
21	0.60	1.5

Table 5.1: Values used to generate subcanopy geometry at particular distances from camera.

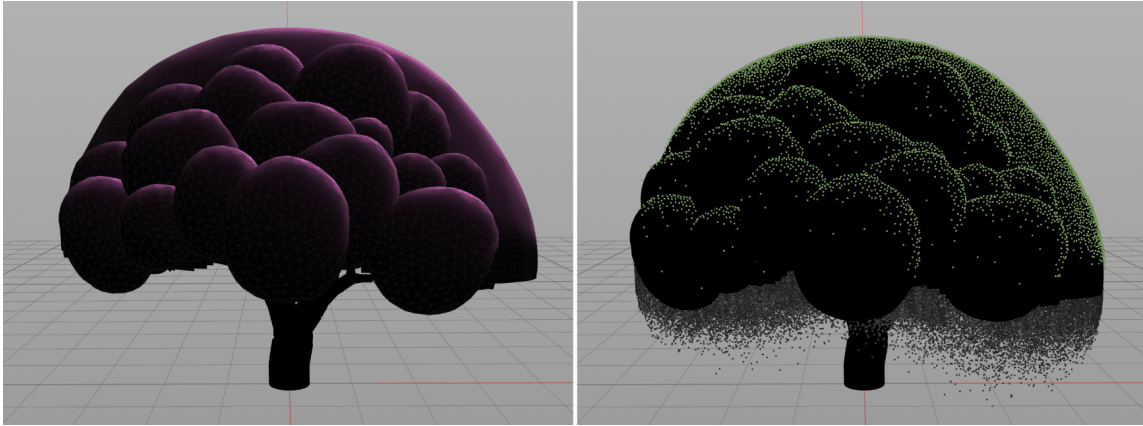


Figure 5.8: The probability distribution for a direct light, and the corresponding detail points.

lighting for this type of tree is split into two kinds: direct light, and rim light. Each is described visually with point geometry stippled across the surface.

A probability distribution is created for the detail points of each light scenario to be used as input for the scatter SOP. The points need to be biased towards the edges (rim light) and in the direction of incoming light (direct illumination). Because the scatter and fuse method requires many more points for the most dense regions, the probability distribution needs to be much more extreme than it would need to be otherwise. Additionally, the distribution is biased away from the bottom of the canopy to avoid a harsh line where the canopy has been cut off. See figure 5.8 for a comparison of the probability distribution for direct illumination and the corresponding detail geometry.

Camera Distance	Direct Light		Rim Light	
	Fuse	Scale	Fuse	Scale
9.3	0.024	0.017	0.029	0.029
21	0.044	0.032	0.054	0.054

Table 5.2: Values used to generate the detail geometry at particular distances from camera.

The color of the points is described with a base color, and variation amount applied to the hue. The color for the rim lit points is a light lime green (0.768,0.881,0.366) with 4.1% hue variation, slightly different than the direct illumination points' medium green (0.504,0.688,0.344) with 3.6% hue variation. In different lighting scenarios, the base color and variation amount can be easily changed to accommodate different color schemes.

The second kind of detail geometry are the points suspended from the bottom of the canopy. Points are scattered across the bottom part of the main canopy that was culled to make room for these details. About 80% of these points are stretched downward so they appear more like dangling branches. The rest remain and are given a larger radius to ensure there are no large gaps in the points near the bottom edge of the main canopy. Figure 5.9 shows the two kinds of points individually, and together in the final silhouette.

5.3.3 Shadows

This tree model features separate shadow geometry. Each tree instance has a distorted, irregular sphere for casting its shadow. First, the sphere is placed centered on the tree origin, roughly the diameter and height of the tree. Then the surface is displaced with perlin noise using a mountain SOP. Simple, but effective! As long as the light angle doesn't become close to parallel with the ground, this ground-

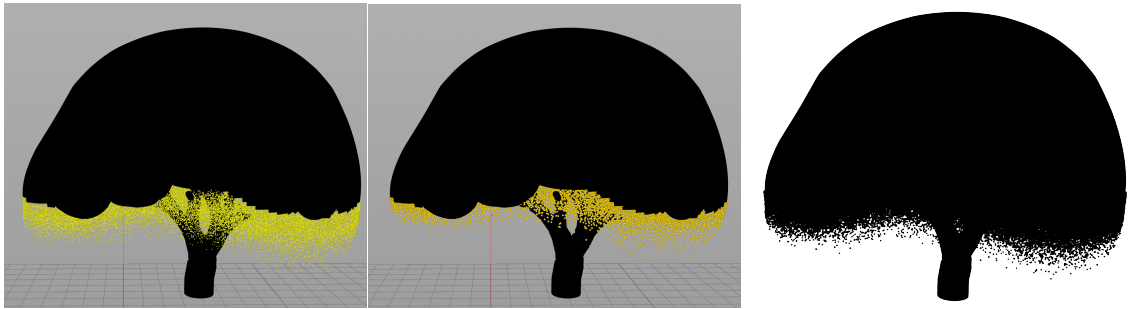


Figure 5.9: Left: Small stretched points. Center: Larger filler points. Right: Final black silhouette.

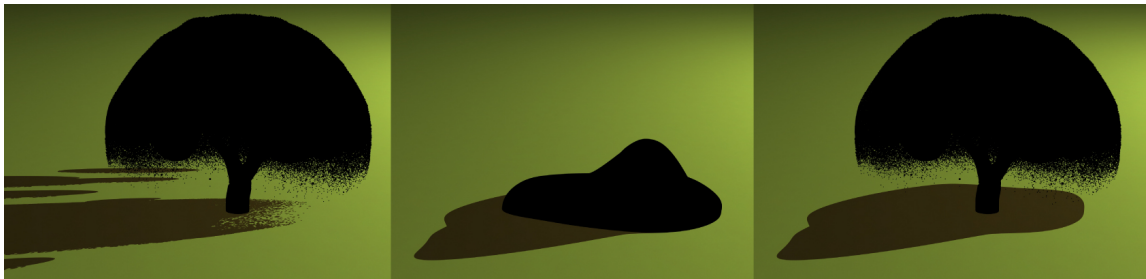


Figure 5.10: The tree with special shadow geometry.

intersecting shadow approximation works well.

5.3.4 *Level-of-Detail*

The closer a tree is, the more sub-canopies it has, resulting in greater detail. Also I attempt to maintain a fairly constant size of points and distance between points in the rendered image plane, regardless of depth in the scene. This means that more distant detail geometry is larger.

Figure 5.11 shows examples of varying depth and level-of-detail from my system compared to examples from Eyvind Earle. See tables 5.1 and 5.2 to review the key properties and values that change the level-of-detail present in the trees.

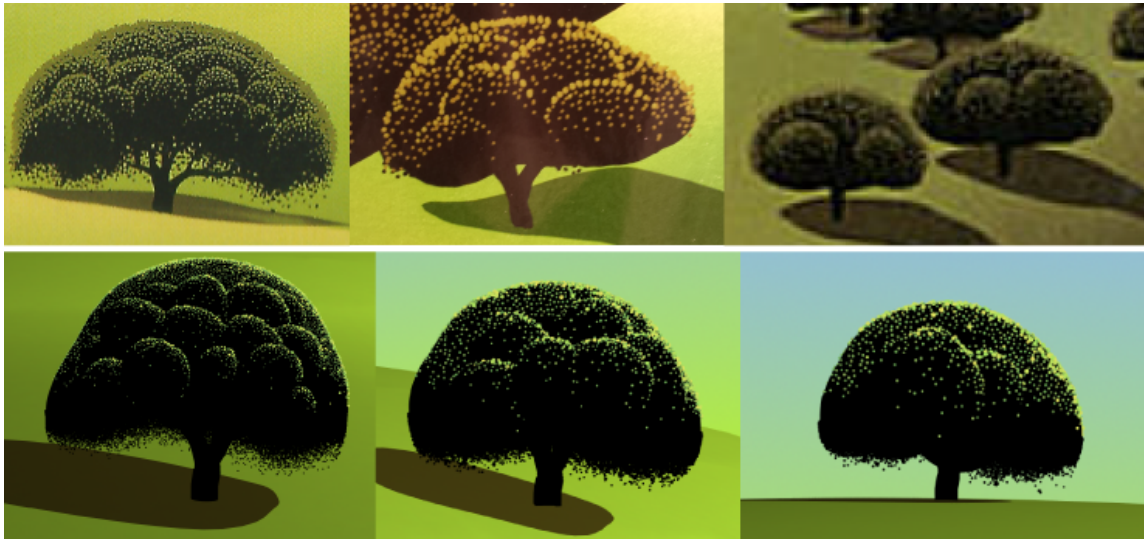


Figure 5.11: Trees at varying distances with different amounts of detail. Top Row: trees from *Cachuma Ranch*. Bottom Row: trees generated with my system.

5.3.5 Interface

This asset is packaged into a Digital Asset. Figure 5.12 shows the Bushy Tree node, as well as its custom parameter tab. There are four parameters for the Bushy Tree: *Position*, *CameraPosition*, *RandomSeed*, and *UseWindAnimation*.

The *Position* parameter is a standard parameter, so it is featured on the Transform tab. *CameraPosition* is used in conjunction with the tree position to calculate a distance value that is fed into many properties within the node network to specify the level-of-detail. *CameraPosition* should be a static value that is near the actual camera position. This prevents tree geometry from re-evaluating at every frame with a moving camera, while allowing the LOD to be approximately correct. The *RandomSeed* parameter is fed as a seed to any random function within the node network. It is not strictly necessary to change the random seed because varying distance values will inherently cause trees to evaluate differently. Finally,

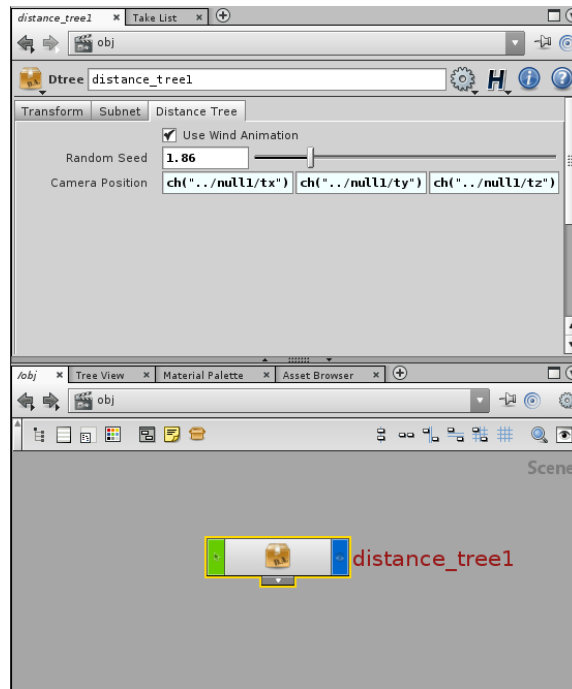


Figure 5.12: The node and interface for the Bushy Tree Digital Asset.

the *UseWindAnimation* parameter is a toggle that can turn the wind animation calculations on or off.

The whole node interface is, of course, much larger than these four parameters. A user can open the digital asset and change anything in the sub-networks to accommodate specific needs.

5.4 Bare Trees

The angular trees, spruce trees, and leafless bushy trees in this section exhibit a highly prominent branching structure, and are defined with stochastic L-system rules. The rule sets are crafted specifically to convey each of these trees. For simplicity, I chose to only create these trees to visually function from a distance, rather than from a closer vantage point. Specifics on L-systems in Houdini can be found in Appendix A.1.

Because the branches are thin and I am only modeling distant trees, there is effectively no distinction between the base geometry and detail geometry. Similarly, the lighting and shading is fairly simple as well – each a solid color appropriate to the specific tree.

5.4.1 *Angular Tree*

Example results of my angular tree L-system are in figure 5.13 (look back at figure 3.10 for more reference artwork). The L-system rule set for angular trees can be found in Appendix A.2 in table A.5.

The tree consists of the main trunk with branches splitting off. Each branch initially departs at a 36.8 degree angle, grows some amount, and then reverts upright. The branch is then eligible for further growth. It grows upright, and then ends and spawns zero, one, or two branches that continue the same process. Some random angle variations are applied throughout the process, along with the probabilistic branching and growth. Gravitational tropism combats the angle randomness and biases the branches towards vertical.

5.4.2 *Spruce Tree*

Example results of my spruce tree L-system can be seen in figure 5.14 (more example references are in figure 3.9). The L-system rule set for spruce trees can be found in Appendix A.2 in table A.6.

While there are some instances of lighting details in the spruce trees, I chose to only model them as dark silhouettes. Effectively the tree is composed of branches spiraling up and around a central trunk. Each branch has sub-branches that grow on both sides of the branch. Since the spruce trees are spiral in shape, they tend to look good from any angle.

Randomness is imbued into each aspect so that the trunk, each branch, and each



Figure 5.13: Top: My angular tree results. Bottom: Eyvind Earle's Trees.



Figure 5.14: Top: My spruce tree results. Bottom: Eyvind Earle's trees.

sub-branch grow probabilistically. The probabilities of each rule are tuned such that the trees achieve a spade-like profile. Tropism controls how much the branches droop towards the ground, so its effect contributes to this shape as well. The branches advance around the trunk at random angles (up to a maximum specified angle) in order to form the spiral.

The spruce tree has several tunable parameters that work within the above system process. The number of generations controls the height of the tree. An angle parameter controls how tightly the branches spiral around the tree. Angle values from 18 to 60 degrees work well. Figure 5.14 shows angle values of 60, 22, and 30 degrees (from left to right). The tropism value due to gravity also has a range of options from 1 to 10 (though a value of 2 is what I used).

5.4.3 *Leafless Bushy Tree*

The leafless bushy tree type (as seen in figure 3.7) is inherently 2D, distinguished by its distinct silhouette and non-overlapping branches in Eyvind Earle's artwork. A direct conversion to 3D is planar; however, after trying this method, I decided it was not sufficient in a 3D environment. After some experimentation I decided on a version with relatively planar (with small angle variations) branches spawning off at any angle around the trunk. This turned out better than a third method I developed with nonplanar branches.

The second method (a nonplanar tree with planar branches), was the most successful of the three. Figure 5.15 shows results of my leafless bushy trees model (at various heights) along side reference art. The L-system rule set for leafless bushy trees can be found in Appendix A.2 in table A.7.

The trunk of the tree follows a cycle of grow, make a branch, and repeat until it reaches the desired height specified by the number of generations. The top of the

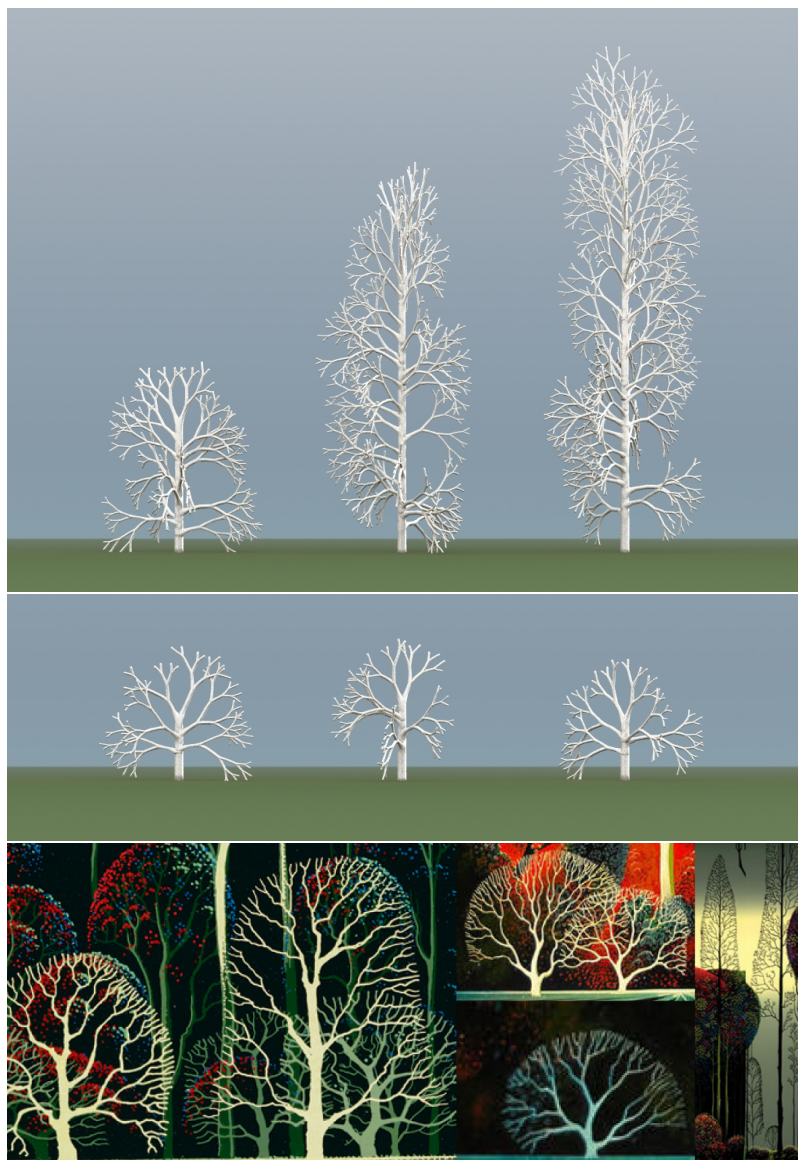


Figure 5.15: Top: My leafless bushy trees with 20, 40, and 60 generations. Middle: My leafless bushy trees with 12 generations. Bottom: Eyvind Earle's trees.

trunk is handled by creating two more branches at the end point. The branches follow the pattern of grow, make two branches, and repeat until the rule has been executed five times. Randomness in the amount of growth and in the angles of branches about the trunk ensure variation between trees. Also, a small random angle adds increased diversity. Tropism has a greater effect on older generations than newer ones; the branches at the bottom fan out towards the ground; the branches at the top reach unhindered towards the sky.

Despite this method being the most successful method I attempted, it still leaves much to be desired. Instead of defining the tree with an L-system, it would be better defined by other methods such as Shek's or Kuruc's particle marching approach that take a goal shell into consideration [30][20]. As this tree type is a small part of this thesis, it is deemed good enough; a better system would be much more sophisticated and is outside the scope of the project.

5.5 Forest

The forest system is similar to the bushy tree system (section 5.3), but simultaneously simpler and more complex. The rules are simpler in my implementation; there is only one level of canopy and not much variation in shape. The complexity comes from the sheer number of trees in a forest. Level-of-detail is not just a choice, but a necessity in a large forest scene. The memory footprint would be too large to handle if the geometry didn't simplify with distance.

The forest pattern is initially specified with a field of point locations and corresponding sizes for individual trees. For the closest distance of the forest (seen in figure 5.17.1) the base geometry is composed of a multitude of ellipsoids, each copied to a tree location and scaled appropriately. The trees are a bit smaller towards the edges of the forest.

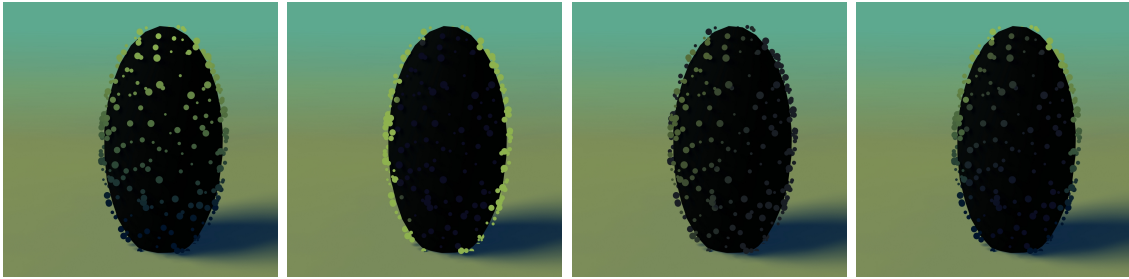


Figure 5.16: Lighting components for the forest trees. Left to right: ambient, rim, direct, final.

Other distinct types of trees can also be included at the closest range, where there is enough detail for them to be distinguished from the mass of trees. In figure 5.17.1 there are leafless bushy trees scattered through the region.

The detail geometry consists of points scattered across the surface of the many ellipsoids. The lighting scenario here consists of a strong rim light, a dim directional light, and ambient occlusion, each shown in figure 5.16 along with the final look. Geometry normals are transferred from the base geometry to the detail points for shading purposes. Each factors into an illumination value for the detail points. A custom shader written at the SHOP level receives illumination from any light sources. Then it identifies whether the point is on the rim of the ellipsoid by taking the dot product of the point normal with the camera. Finally, using a gradient from the bottom to the top of the tree, the illumination is dimmed to simulate ambient occlusion of the forest. The illumination value determines which color value the point should have in the spectrum of illuminated yellow-green to shaded dark blue. This matches the deep blue shadow color used in the figure 5.17 example.

Farther back in the scene, the colors blend towards light blue as the atmosphere color is taken into account. This holds true for the base geometry points, as well as the detail points. The colors shift in steps as a stylistic choice mimicking the

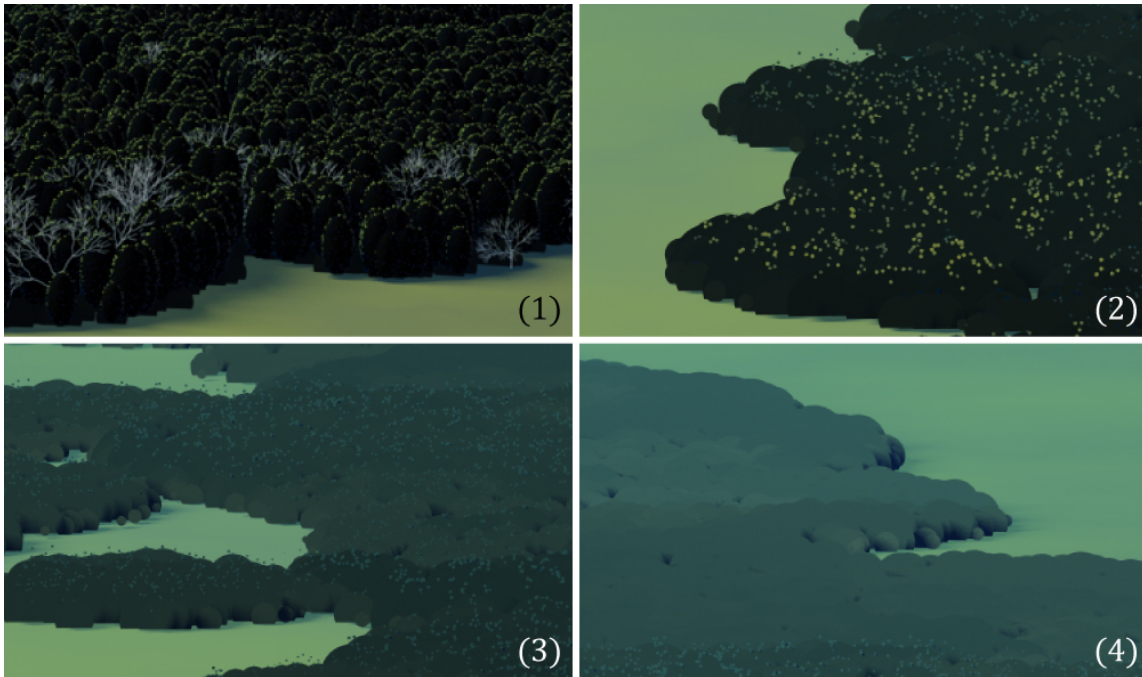


Figure 5.17: A forest at different levels of detail.

atmospheric perspective across rolling hills found in Eyvind Earle's work.

Past a certain depth in the forest scene, trees are no longer represented with the ellipsoid tree base geometry. Each tree is rendered only as a base point sprite with detail geometry points scattered above the base point to continue a hint of the lighting information. This is the scenario in figure 5.17.2.

Farther back in the forest, the detail points transition from yellow-green illumination towards the forest atmosphere color (seen in figure 5.17.3). Finally the detail geometry disappears entirely, and only the base geometry points remain, as seen in figure 5.17.4. As previously stated, the geometry at this point is also closely approaching the atmosphere color.

5.6 Cliff

The cliff procedure can be applied to any piece of landscape geometry, modeled or otherwise generated. Cliffs are formed on steep hills. The cliff procedure consists of altering the input base geometry to create rocky jagged ridges and generating the line and point detail geometry to better describe the form within the silhouette.

5.6.1 Base Geometry

The base geometry of the cliff is an alteration of a given input geometry that is part of a larger terrain. Figure 5.18 shows an example final base geometry. The steep areas of the input geometry are marked as part of the cliff, and are altered as part of the cliff procedure. Any faces on the geometry that are passed a 10° angle with vertical are marked as cliff.

The system allows for a “shelf” normal to indicate the orientation of the small plateaus that form on the steep face of the cliff. Shelf ridges are formed by first displacing geometry vertices downward according to a noise function, and then pushing them inward to amplify the width of the ridges. At this step the *ridges* attribute is created for use with the detail geometry creation.

5.6.2 Detail Geometry

The detail geometry for cliffs is a combination of two components: vertical lines and horizontally aligned strings of points. The vertical lines show the sheer steepness of the cliff. The horizontal points define edges of landings on the face of the cliff.

Creation of the vertical definition lines requires the use of the original base geometry marked as cliff. Points are scattered across this simpler surface. A plane is copied to each point such that one edge of the plane is aligned with the cliff normal, and the other edge is aligned in the direction vector of the steep cliff slope. Figure

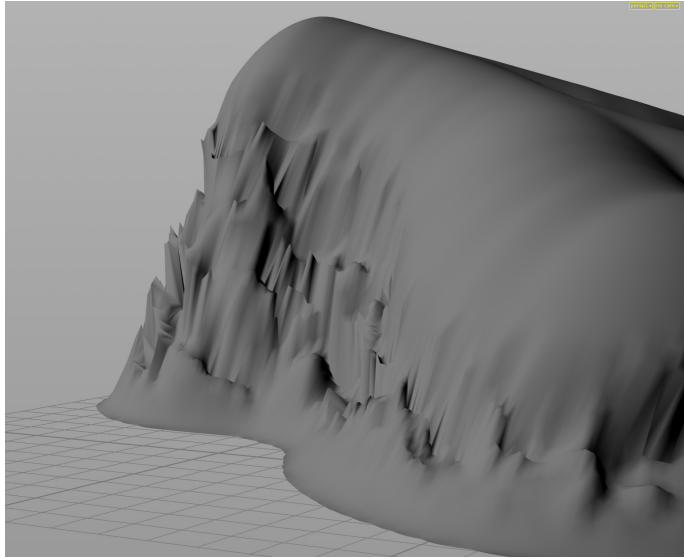


Figure 5.18: Cliff base geometry.

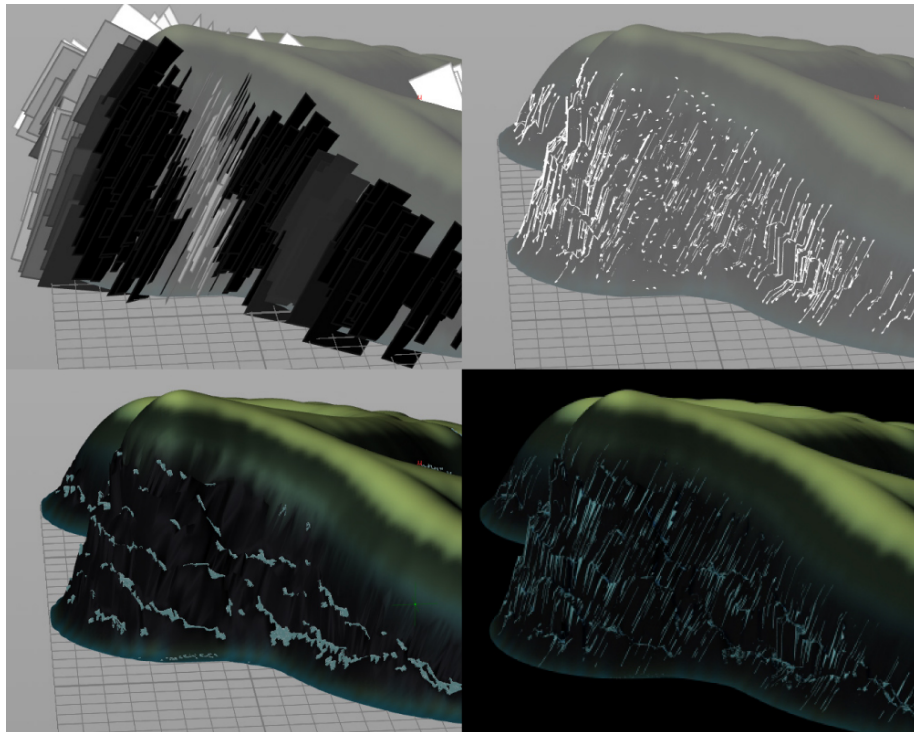


Figure 5.19: Cliff detail geometry stages.

5.19.1 shows an example of how these planes are placed and oriented. Intersecting the planes with the final base cliff geometry produces a line for each plane hugging the surface of the cliff (see figure 5.19.2). The lines expose the form of the cliff as jagged vertically oriented cross-contour lines. The lines are given a color and varied line width.

The horizontal point strings are placed along the top edge of shelf ridges, denoted by the *ridges* attribute. The points are colored to match the lines, and each point gets a size to add variation to the stippling effect. Figure 5.19.3 shows the points scattered along the shelf ridges.

Figure 5.19.4 shows the line and point detail geometry rendered atop the cliff base geometry.

5.7 Receding Mountains

5.7.1 Geometry

I made some receding mountains for use with an ocean. Each mountain starts as a polygon sphere and is processed through a series of noise and geometry filters to arrive at the final graphic look. Figure 5.20 shows a side-by-side comparison of my mountain ranges with an example from Eyvind Earle's work.

The initial sphere for each mountain range is elongated and remeshed to obtain large polygons. Remeshing (in this case using the Houdini Remesh node) reinterprets a polygon mesh surface to obtain a more evenly distributed and sized polygons. A Mountain SOP displaces the surface along the normals with perlin noise. The scale of the noise is tuned manually for the size of the mountain range. The polygons that compose the mountain range geometry are large enough such that this process results in a clear jagged silhouette.

The final step is to create flat landings in the mountains, implemented using a

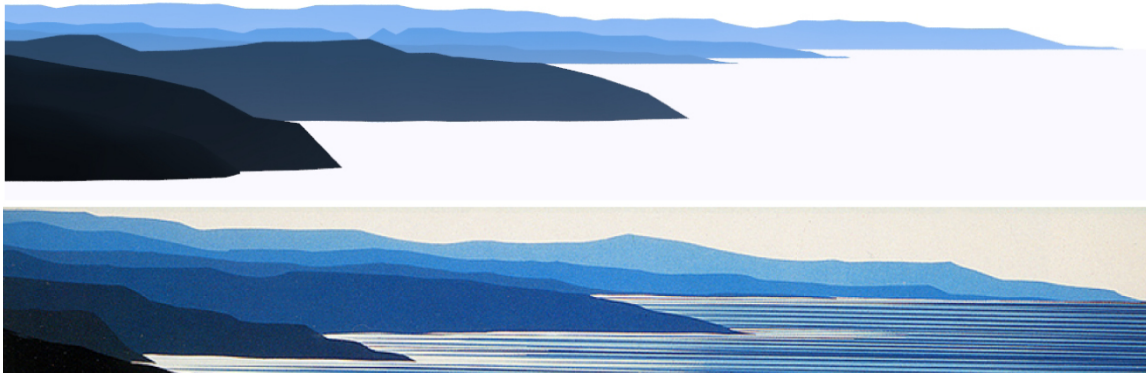


Figure 5.20: Top: my receding mountain ranges. Bottom: receding mountain ranges in *The Wave* (1990)

custom VOP SOP to straighten out some of the jagged points. Each point's height is compared to all of its neighbor's and if the height difference is within the specified range, then the point's y-value is updated to match. No change occurs if the difference is too small, or too big. The difference range allows for subtle curves while capping the maximum change to prevent overly drastic slopes. The example in figure 5.20 uses a value range of .1 to .3 units.

5.7.2 Shading

In order to match the style in Eyvind's artwork, the mountain ranges are shaded as silhouetted shapes that ignore lighting information. The closest rocks and mountains are black. As the depth increases, the mountains change color to reflect the increased amount of atmosphere between the surface and the observer. Ultimately the color depends on the atmosphere color and the overall color palette of the scene. As atmosphere increases, ground fog does as well, which makes the bottoms of the mountains lighter than the tops.

I created a custom shader that computes the color of each mountain based on the stylized effects of atmospheric perspective and ground fog. The shader has a ramp

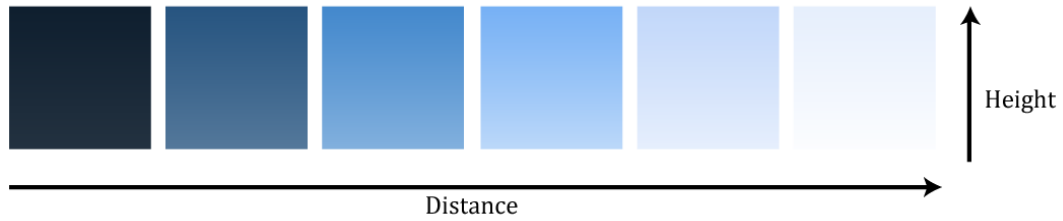


Figure 5.21: Mountain Range

parameter that specifies the color progression from near geometry to far geometry. By default, any geometry with a depth closer than 20 units is clipped to the near color (black), and any geometry farther than 100 units is clipped to the far color (white). To account for ground fog, the shader blends the calculated distance color with white based on height from the ground plane. Ground fog is also attenuated with proximity to the camera. Figure 5.21 shows color swatches for the mountain ranges using an example color ramp.

5.7.3 *Ocean Rock Outcrops*

The ocean rock outcrops are an extension of the receding mountain ranges. They are modeled nearly identically, but with different values (mainly a smaller scale, and larger noise displacement compared to the scale). However, the key difference is the height at which they are placed. The geometry is placed lower in the scene so that most of the mesh is below the water plane. The displacement noise allows portions of the mesh can protrude up through the water to create several small outcrops clustered together. See figure 5.22 for a diagram on the placement of the outcrop geometry. See figure 5.23 for a side-by-side comparison between my system and Eyvind Earle’s artwork.

The same shader applied to the mountain range is applied to the rock outcrops

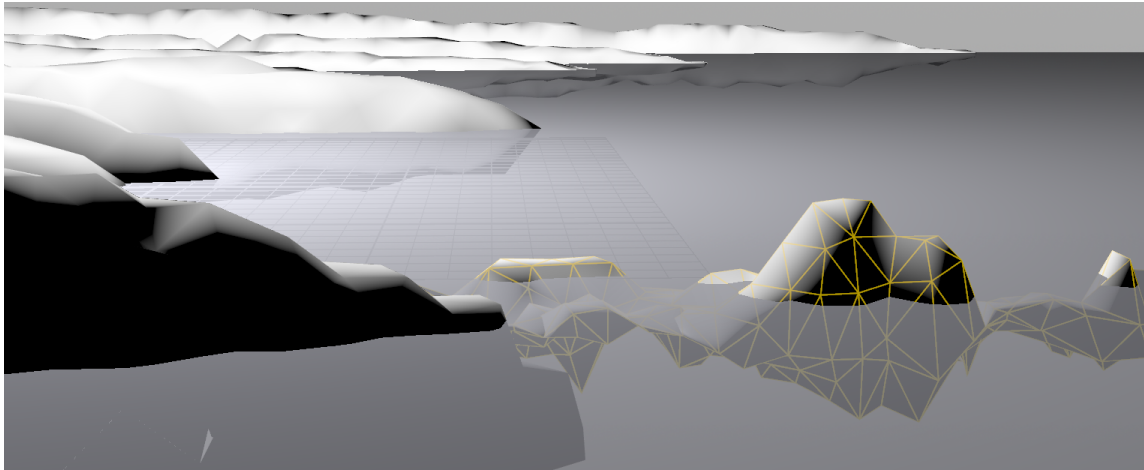


Figure 5.22: The rock outcrop geometry is placed mostly below the water plane so that many outcrops can appear.

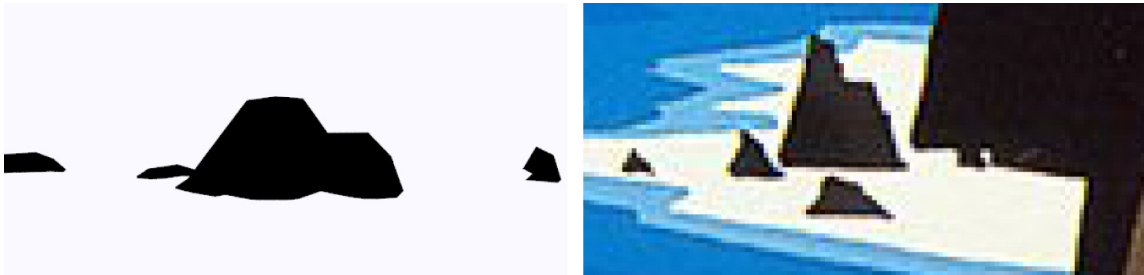


Figure 5.23: Left: my rock outcrops. Right: rock outcrops from *Enchanted Coast (1980)*

as well. The outcrops are extensions of the mountains and need to agree in color to appear integrated in the final image. The gradient on any outcrops that are farther back in the scene will not be noticeable since they are so small compared to the mountains.

5.8 Ocean

Eyvind Earle drew several different styles of ocean, however this system describes just one— a look that consists of many horizontal lines. The lines are colorful shades



Figure 5.24: Top: my ocean. Bottom: Eyvind’s ocean in *The Wave* (1990)

of blue, spaced evenly on the surface of the image. There is also an indication of seafoam along the coasts where terrain protrudes from the water. Figure 5.24 shows an example by Eyvind Earle alongside results from my system.

The ocean is modeled as a flat plane. All the details are described in the shader, which uses the following procedure: 1. distort coordinates to account for perspective, 2. add noise to the coordinates, 3. assign the coordinate to be a light or dark line, 4. assign the line a color, 5. alter the color with distance to shore.

To end up with evenly spaced lines when rendered, the texture for the ocean plane must be distorted to account for perspective. For simplicity I assume the camera is

looking approximately down the z-axis, and I write my equations and descriptions as such. Ultimately, this means that z-values closer to the camera are mapped closer together, while z-values farther from the camera are mapped farther apart. Each integer z-value corresponds to one band of color in the final image. Figure 5.25 shows the lines rendered from the scene camera as well as from above.

It is not necessary for the inverse perspective transformation to be exactly correct. It just needs to look close enough. In fact, I decided to hint at perspective in the ocean by allowing the lines at the back of the scene to be very slightly smaller when rendered. The equation below is the transformation I formulated.

$$z_{distort} = 700 \cdot (z - 20)^{-.5} + noise$$

In the equation, “-20” translates the divide-by-zero discontinuity behind the camera such that its undesirable effects are hidden. The “700” is a multiplier to change the spacing of the integers (and thus the line width of the bands); a higher number would result in smaller bands, while a lower number would result in larger bands. The noise term adds small variation to the lines— plus or minus a maximum value of .25. A line is at most 150% standard width, at minimum 50% standard width, and a maximum 25% offset outside its integer bounds.

Using the reference art in figure 5.24, I observed that there is a high amount of contrast between adjacent bands of color in the ocean. Despite the reference ocean bands not strictly alternating between a light and dark color, I decided this was an appropriate way to approximate the effect of the pattern and maintain contrast. Thus, $z_{distorted}$ modulus 2 determines whether each integer value should be assigned a dark (remainder 0) or light (remainder 1) color value.

A dark to light color palette is chosen via a color ramp parameter for the shader.

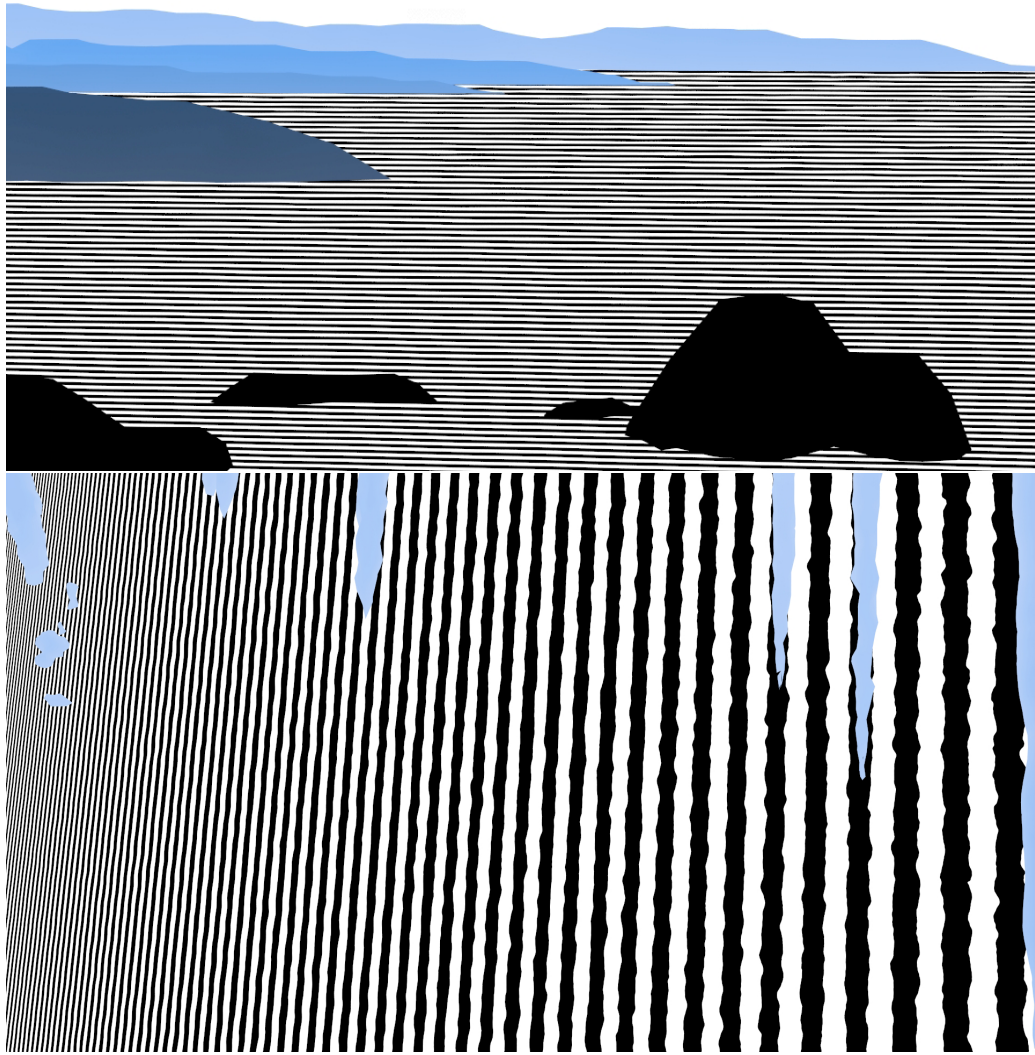


Figure 5.25: Top: Ocean lines from the main camera. Bottom: Ocean lines from above.



Figure 5.26: Color ramp used for the ocean lines color palette.

The ramp (shown in figure 5.26) starts from black as the darkest color, features several blues, and ultimately ends on white. The lower half of the ramp is considered to be the dark palette, and the upper half the ramp is considered to be the light palette. The ramp shown is used in all the ocean examples in this paper. However, it is one many possible color palettes that could be used for the ocean.

Each dark and light line picks a random color from the corresponding half of the color palette. The equation (shown below) starts with 0 for dark or 1 for white, adds a random value between 0 and 1 and multiplies by .5 to obtain the base color value for *ramp_key* between 0 and 1. The *ramp_key* value is passed to the ramp and results in the base color lines which can be seen in figure 5.27.1.

$$ramp_key = .5 \cdot (z_{distort} \bmod 2 + \text{random}(\lfloor z_{distort} \rfloor))$$

Next, I consider proximity to the coast. A point cloud indicates the coastline location by specifying many points along the intersection of the terrain geometry with the ocean plane. The shader uses the point cloud to determine an influence value based on how close the shaded point is to the coastline. The value is 1 at the coast and falls off as is it gets farther away (shown in figure 5.27.2). This value is added into the ramp key and pushes the resulting colors towards the lighter values and white. The final ramp key equation is shown below.

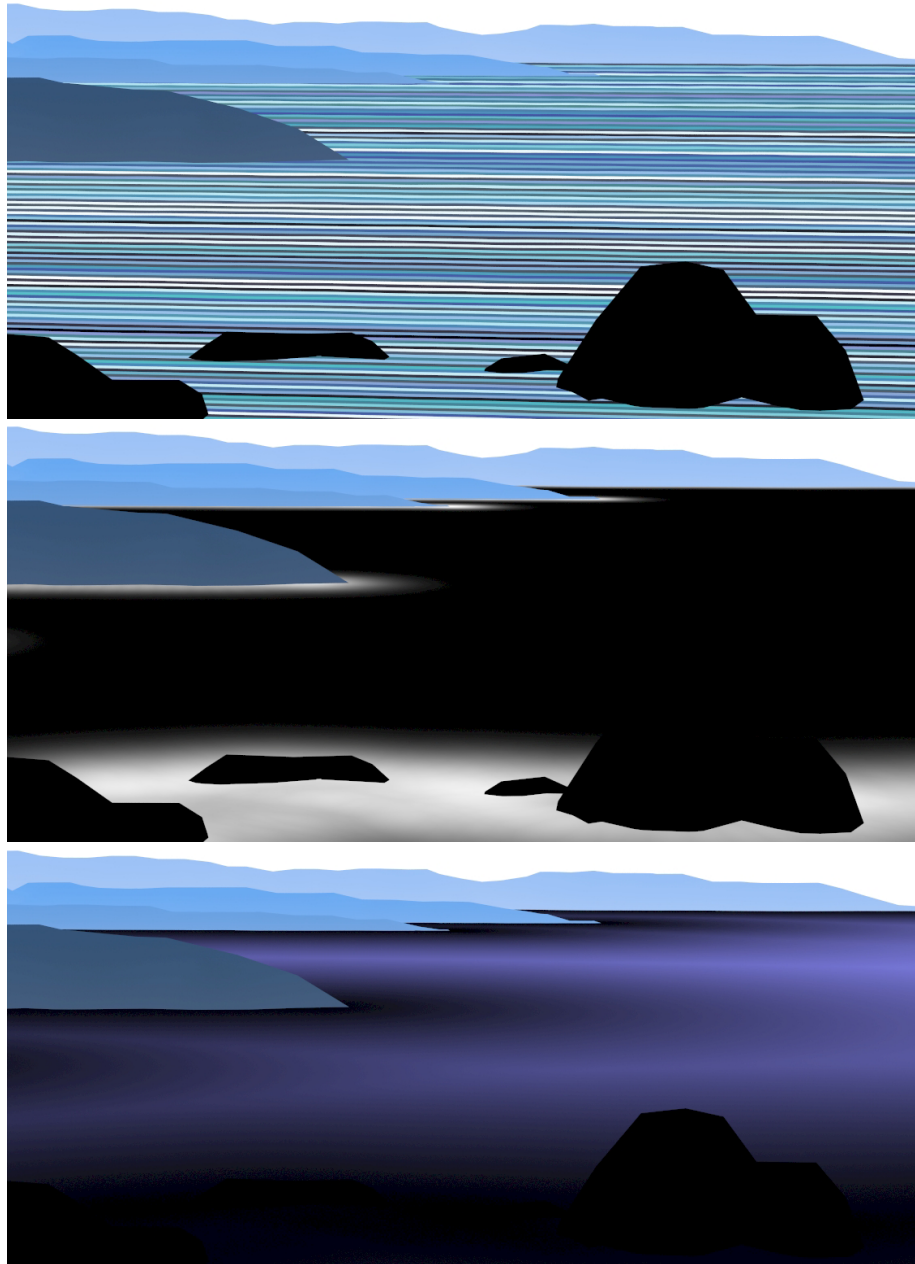


Figure 5.27: Top: base colored ocean lines. Middle: proximity to coast – sea foam. Bottom: distance from coast – deep water.

$$ramp_key = .5 \cdot (z_{distort} \bmod 2 + \text{random}(\lfloor z_{distort} \rfloor)) + .6 \cdot \text{proximity_value}$$

The coastline proximity value shown in figure 5.27.2 is also used to blend the ocean color with white to secure solid white along the coast. Similarly, figure 5.27.3 shows the deep water value indicating the distance from the coastline. This value is used to apply a deep blue tint to the ocean color far from shore.

$$color_{temporary} = \text{mix}(\text{ramp}(ramp_key), \text{white}, \text{proximity_value})$$

$$color_{final} = color_{temporary} * \text{mix}(\text{white}, \text{deep_blue}, \text{deep_water_value})$$

The result of these ocean steps can be seen in the final image shown in figure 5.24.1 at the beginning of this section.

5.9 Wave

For simplicity, the wave is modeled separately from the ocean. This system is intended for a wave that is close to the camera. The interface between the wave geometry and ocean geometry is occluded by the wave itself, so integrating the two is not a problem. This model is not used for waves at a distance. Instead, the wave sea foam is incorporated into the ocean shader. Ideally, the wave would be integrated into the ocean system such that waves are generated arbitrarily when LOD calls for it. However, such a thorough treatment of level-of-detail on this complex system is outside of the scope of this project.

Figure 5.28 shows the result of my wave modeling system compared to an example from Eyvind Earle's work. Note the key differences, particularly in the the underside

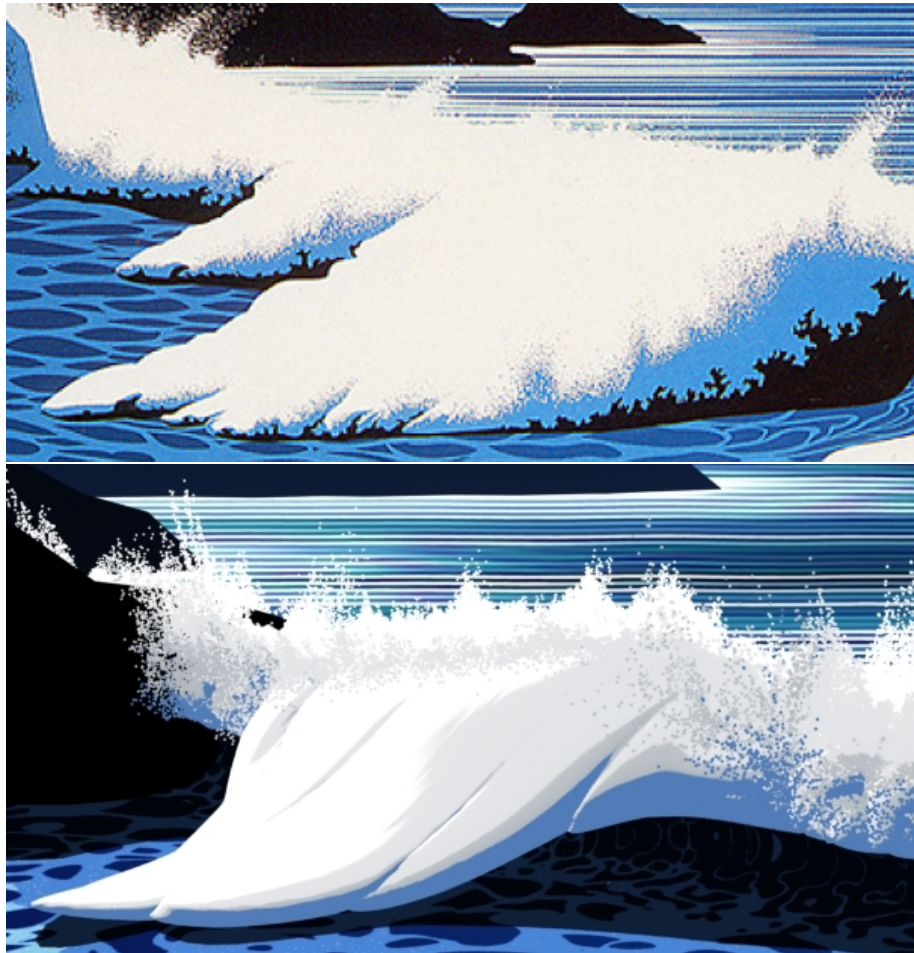


Figure 5.28: Top: Eyvind Earle’s wave in *The Wave* (1990). Bottom: my wave.

of the breaking wave. There are minor differences in every system, but they are deemed close enough, especially since the images I generate are intended to be stand-alone—inspired by Eyvind Earle, but not a direct 3D conversion.

5.9.1 Base Geometry

The wave geometry is modeled starting with a plane. The process can be partitioned into three steps: 1. subdivide to create appropriate resolution in the mesh, 2. protrude upwards to make the wave crest pattern, and 3. curl to make the collapsing wave shape. Figure 5.29 shows each step.

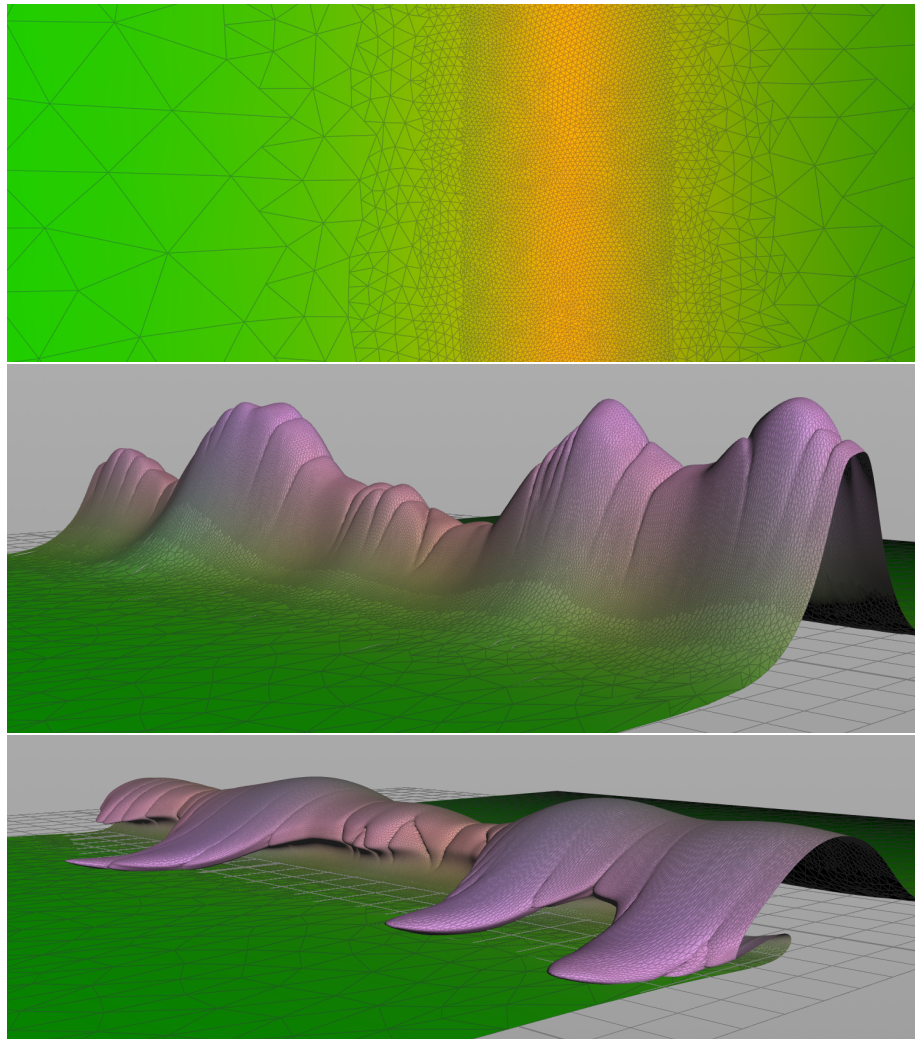


Figure 5.29: Top: the subdivided plane. Middle: wave crest pattern. Bottom: wave with curl.

5.9.1.1 Add Geometry Resolution

There needs to be enough resolution in the geometry to support the shape transformations. The plane is remeshed to varying densities, with the densest portion along the line where the wave is going to be.

Additionally at this stage, two extra properties are added to the geometry points for use in later steps. The *localz_value* (visualized in the green channel) specifies the how far in front or behind a point is to the wave centerline. The value *wave_amt* (visualized in the red channel) is a mask specifying how much of the plane will protrude to make the wave.

5.9.1.2 Wave Crest Pattern

The wave crest pattern is composed of two simplex noise functions masked by *wave_amt*. The first noise (pictured in figure 5.30.1) has a frequency of .55 units; it is translated upwards and processed so the major shape of the wave crest is a smooth oscillating curve. The second noise (pictured in figure 5.30.2) has a frequency of 2.25 units; I take the absolute value so it develops sharp downward points between the variably sized noise bumps. The two noise patterns together, masked with *wave_amt* creates the final wave crest pattern (pictured in figure 5.30.3) before the wave curl is added. Different offsets for the noise will seed a different pattern and generate a new wave shape. Additionally, the attribute *height* (visualized in the blue channel) is added to the geometry points which will be used later; it simply stores the y-value of the geometry points for this state of the geometry.

The equations below detail the mathematics that create the *major_pattern* (large noise), *minor_pattern* (small noise), and final *wave_crest_pattern*. See figure 5.29.2 for the wave crest pattern applied to the geometry. The equations are equivalent to the code executed within a VOP SOP that I created to form the wave

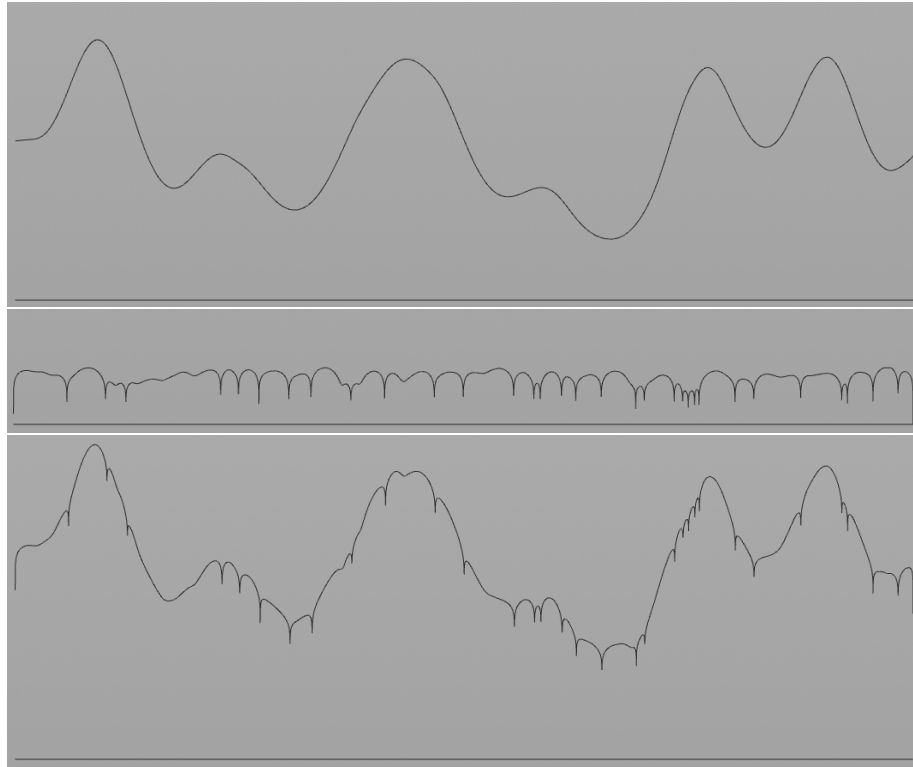


Figure 5.30: Top: major noise pattern. Middle: minor noise pattern. Bottom: major and minor pattern composite.

crest pattern.

$$major_pattern = 3 * (\text{simplex}(.55, p) + 1)^3$$

$$minor_pattern = (3 * |\text{simplex}(2.25, p)|)^3 * wave_amt$$

$$wave_crest_pattern = .5 * (major_pattern + minor_pattern) * wave_amt$$

5.9.1.3 Wave Curl

The final step of the wave base geometry is to curl the geometry in the shape of collapsing wave. The idea is to take the geometry that is protruded upwards and

rotate it about an imaginary centerline. To get a transition from the ocean ground plane to the curling wave shape, the greater *height* is, the more rotation is applied. As the curling geometry approaches the ground plane, the rotation tapers off by applying a reverse rotation after the initial rotation. Additionally, the wave tendrils are pushed forward (towards shore) according to the *height* attribute. At this stage greater *height* values will be closer to the ground plane. This simulates the wave crashing down on the surface of the water and flowing forwards. The result of this process can be seen in figure 5.29.3.

The equations below detail the mathematics executed in a VOP SOP for the wave curling process described in the previous paragraph. Figure 5.31 shows each of the steps described above: initial, curl, reverse curl, push forward.

$$curled = (\text{rotationmatrix}(1.05 * height, axis) * p)$$

$$reverse_curl_matrix = \text{rotationmatrix}(\exp(-curled.y) * height * -.125, axis)$$

$$push_forward = \exp(-curled.y) * height * .253 * wave_forward_dir$$

$$tapered = reverse_curl_matrix * curled + push_forward$$

5.9.2 Details and Shading

The detail geometry of the wave system consists of three parts: detail spray geometry (points), the wave crest shader, and the decaying seafoam shader. The shaders are implemented as a CVEX node network in the SHOP context using the Material Shader Builder (an empty material node).

There are two more attributes added for use in creating details. Attribute *spray* on the base geometry denotes where spray particles for crashing whitewater should

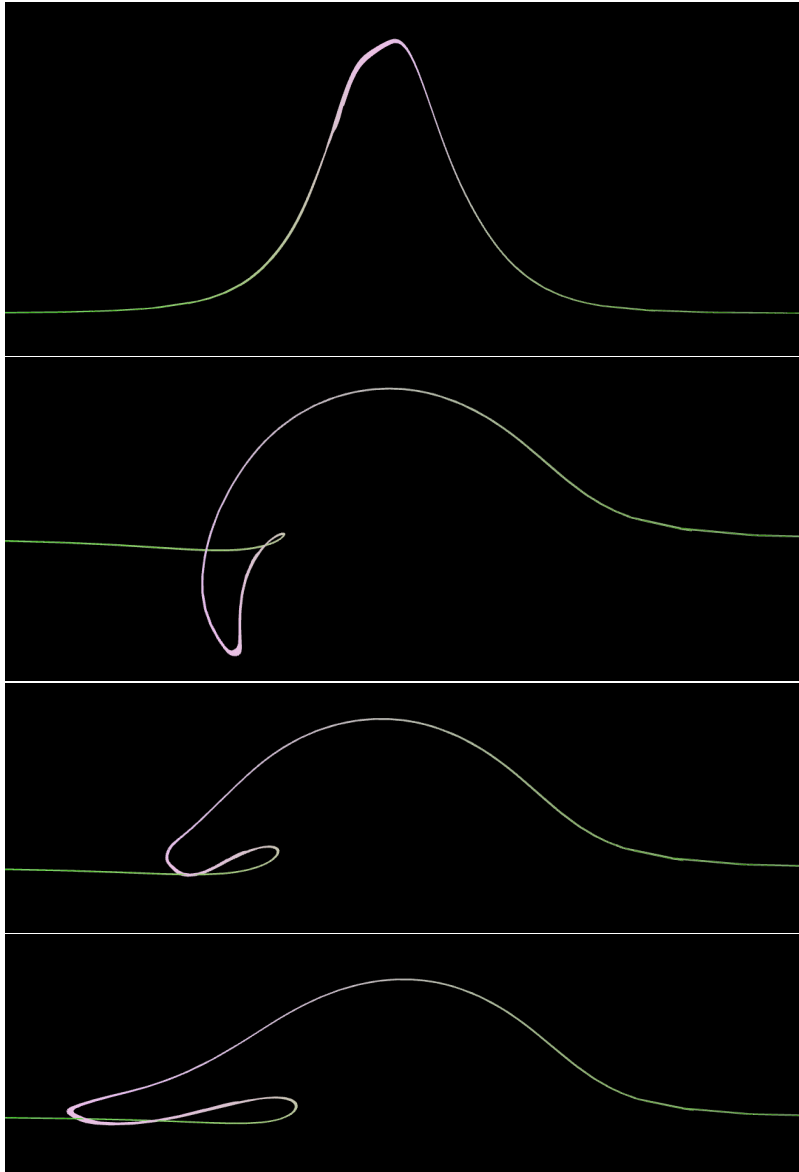


Figure 5.31: The wave curl process.

be emitted (the highest section along the crest of the wave). Attribute *whitewater* indicates where the geometry should be shaded with the wave crest shader vs the decaying seafoam shader (anything where *height* > .75 on the front or *height* > .4 on the back).

I scatter particles over the geometry denoted as a whitewater area, using *spray* as a probability distribution. The normals and the *spray* attribute are transferred to the points.

For each point, I generate a maximum value to create a spikey pattern for the spray (see *max_spike* equation below). Then I randomly move the point a maximum of *max_spike* along the normal, with the values biased towards the base of the spikes, not the points (see *spray_point* equation below). Additionally, I jitter the normals so there is a bit of variation in the shading. These operations are executed within a VOP SOP.

$$max_spike = 30 * (noise(2.55, p) + 1)^7 * normal.y * rand(pt_num)$$

$$spray_point = max_spike * (.5 * noise(11.5, p) + .5)^5$$

Figure 5.32.1 shows the points after they have been scattered on the wave geometry. Figure 5.32.2 shows the points after they have been subsequently displaced to their final positions.

The wave crest shader is applied to the spray point geometry as well as the portions of the base geometry denoted as a white water area. The shader is implemented as a CVEX node network in the SHOP context. Look back at figure 5.28.2 for a rendered example.

Illumination is calculated similarly to standard diffuse illumination. The illumi-

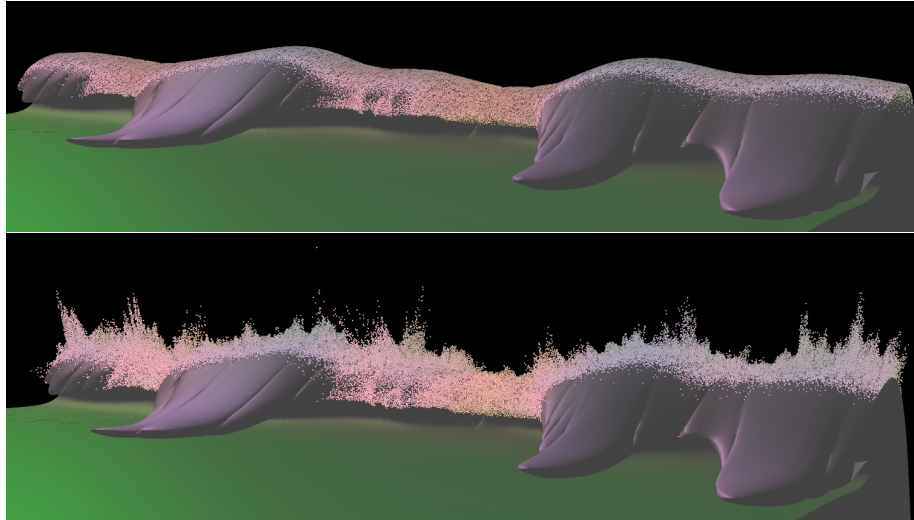


Figure 5.32: Top: spray points scattered on the wave geometry. Bottom: spray points in their final positions.



Figure 5.33: Color ramp used for the wave crest shader.

nation is the sum of contributions from all the lights, where the contribution based on the dot product (or how much the surface is facing the light direction) is remapped from $[-1,1]$ to $[0,1]$. The illumination value is then used as the ramp key to determine the final color for the wave crest seafoam. The equation for illumination is shown below. The color ramp is shown in figure 5.33.

$$I = \sum_{lights} (N \cdot L + 1)/2$$

The decaying seafoam shader is shown in figure 5.34 along side reference drawn from Eyvind Earle.

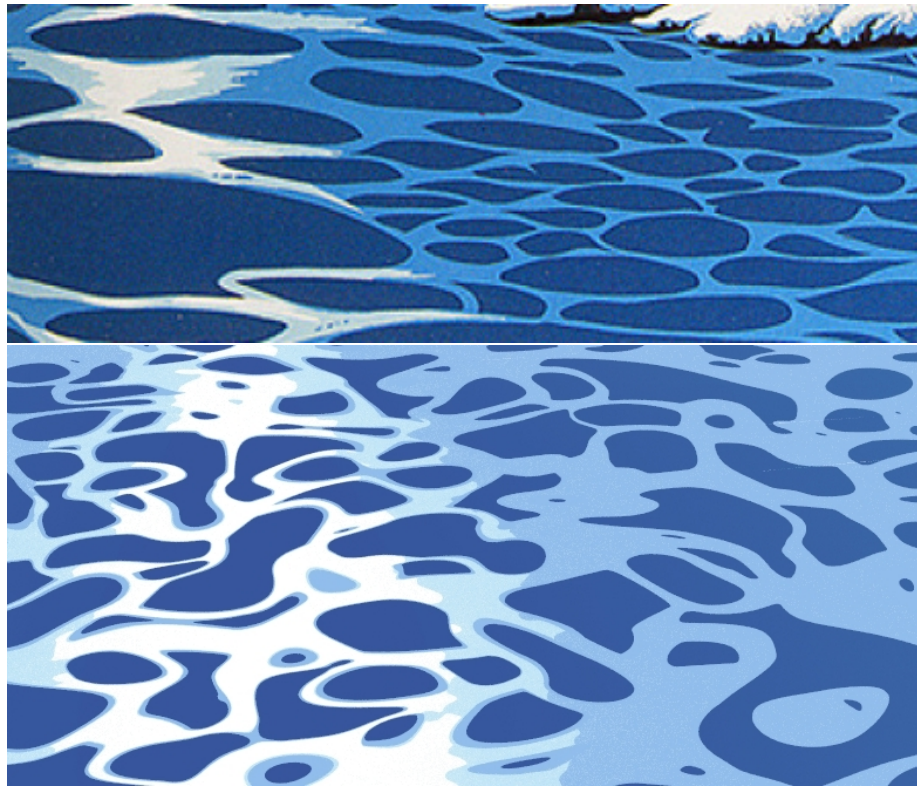


Figure 5.34: Seafoam Comparison. Top: seafoam in *The Wave* (1990). Bottom: seafoam render from my system.

The main component of the pattern is a perlin noise (frequency 2.5 units) with a smooth threshold (low .03 to high .13) that makes distinct blobs and dots. One of these perlin noise dot patterns is not sufficient to fill the surface with the desired pattern. To fill the texture space with the desired pattern, I use multiple iterations of compositing the noise. Each iteration uses the previous iteration as a mask; the result is that there are no overlapping blobs. I used 6 iterations to get a pattern that seems well covered (shown in figure 5.35).

More concisely, the decaying seafoam pattern is composed iteratively with noise interacting with the previous iteration. First starting with zero, generate noise, make adjustments to the noise, subtract the previous iteration's result, then add to the result. This is repeated six times, and then finally a threshold is taken to create the final pattern.

At the base of the wave, a similar smaller pattern is used. A similar masking method is used to blend between the large and small pattern.

The decaying seafoam pattern above differentiates the near-shore ocean water from the white foam sitting on top of it. Based on the example artwork in figure 5.34, I model the foam with a highly anisotropic specular component. A ramp controls the color of the foam based on the specular intensity. The ocean beneath the foam becomes a deeper blue around the specular highlight of the foam. The ramps for each can be seen in figure 5.36.

5.10 Additional Elements

Ultimately everything I have created is meant to be used in a completed scene that looks like one of Eyvind Earle's artworks. Special one time use features must also be created in addition to each of the carefully crafted systems to bring a scene together. These "one-off" items fill in the gaps that aren't generalizable within the

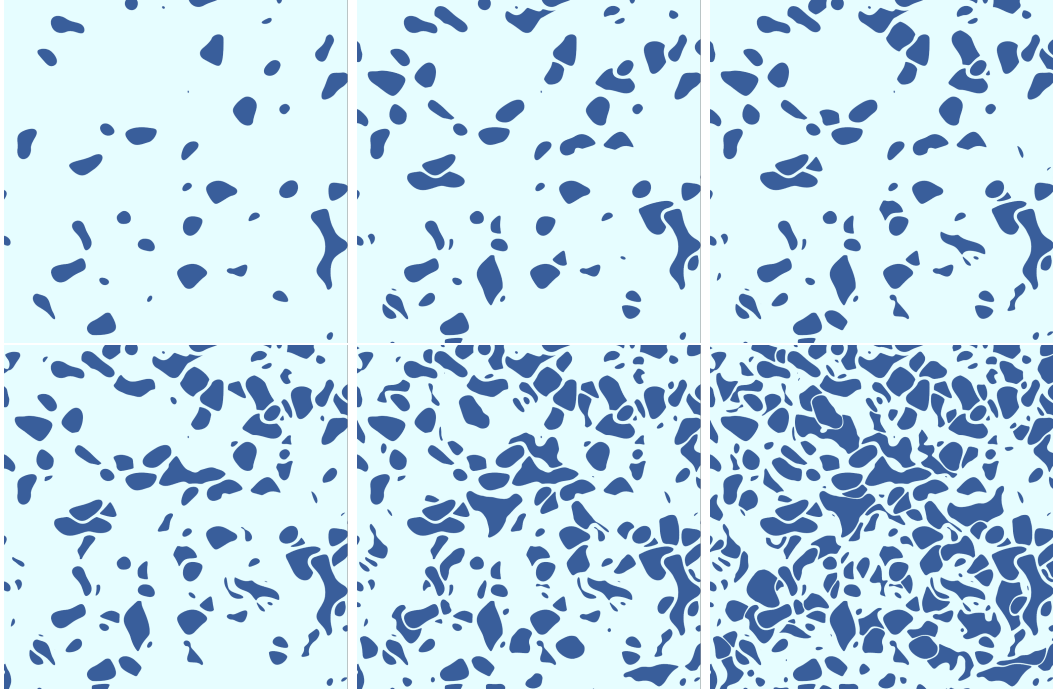


Figure 5.35: Decaying seafoam pattern iteration steps.

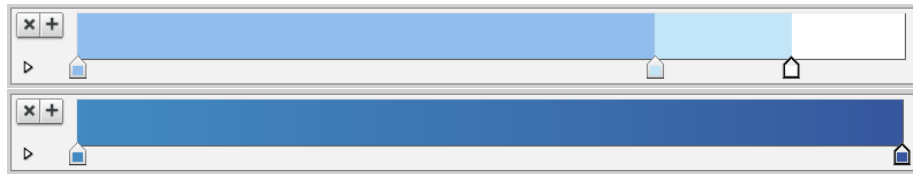


Figure 5.36: Top: ramp for the decaying seafoam color. Bottom: ramp for the ocean water beneath.

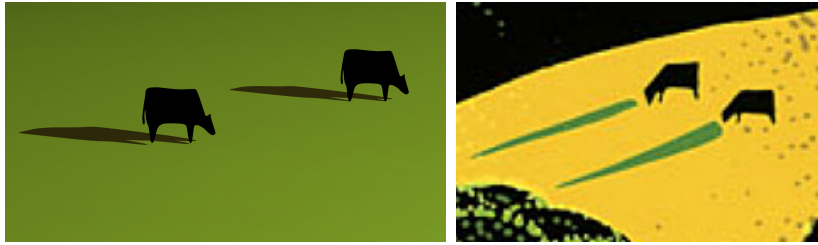


Figure 5.37: Left: my cows. Right: Eyvind Earle’s cows.

scope of this thesis.

Basic terrain such as a ground plane, or a smooth hill that fills the frame are examples. They are modeled, textured, and lit specifically for the scene. Another example is a sky. Typically, the sky is a solid color, or a gradient. A simple sphere or dome with the texture suffices.

Cows are featured in much of Eyvind Earle’s artwork. I modeled a graphic 2-legged cow that can be placed in scenes where necessary. The cow should be rendered flat black. My cows can be seen in figure 5.37 along side an example of Eyvind’s cows. More cow examples from Eyvind’s artwork can be seen in figure 3.15.

Potentially additional kinds of items maybe be needed as well. Any custom-modeled geometry should have a flat color or smooth gradient derived from diffuse shading and ambient shadow color.

5.11 Rendering

As previously stated, I rendered everything with Houdini’s built-in render engine, Mantra. Specifically, the Mantra Micropolygon Renderer or the Mantra Ray Tracer were used depending on the needs of the scene. Images of the forest and the wave needed to use the Mantra Ray Tracer to mitigate render artifacts (more details on the artifacts are in section 6.3).

Scene animations are rendered at 1920 by 1080 pixels (Full HD resolution) to

accommodate today's typical screens. Printed stills are rendered an order of magnitude larger at 25600 by 14400 pixels. They are printed at 720 dpi on the Epson Stylus Pro 7900 on a 24 inch roll of matte art paper, which results in prints sized at 35.6 inches by 20 inches.

To ensure the still frames of the animations remained crisp, I did not use motion blur. Developing a motion blur technique that works with the rendering style is outside the scope of this project.

The final render scene animations are meant to be landscape vignettes; the camera moves are slow much like a documentary. The slowness also helps prevent judder, since no motion blur is used. Motion judder occurs when a viewers brain has a hard time interpolating the motion of an object.

6. CONCLUSIONS

6.1 Overview

For this thesis, I present an analysis of aspects of Eyvind Earle's work and methods of creating corresponding assets, including a variety of trees and landscape features. I developed procedural methods for matching the shapes of the objects, relying on noise, L-systems, and other constraints. Additionally, I combined these assets into digital 3D landscape compositions that emulate Eyvind Earle's 2D graphic style.

I studied Eyvind Earle's life and methods to gain insight into the construction of his images. I gathered many examples of Eyvind's work, categorized the assets by visual similarity, and analyzed how they were generally composed.

Eyvind notably uses graphic base shapes, overlaid with intricate details, typically in a complex graphic pattern, that reveal the form and lighting of the object. He uses solids or simple gradients to color individual sections; the complexity arises from the patterns that are used. The colors Eyvind chooses are also vibrant and highly saturated which, when also combined with heavy use of black, creates a high contrast composition.

Atmospheric effects in Eyvind's work include atmospheric perspective and level of detail changes. Objects tend to push towards an atmosphere color as they recede into the background. The way objects are portrayed also changes with distance. As depth increases, their shapes simplify and the number of details is reduced.

Asset implementation was done in Side Effects Software's Houdini. After a careful analysis of each asset's shape, details, and coloring, I assembled a node network that procedurally defines a representation in 3D. The node networks are parameterized to produce various versions of the same asset with minimal effort, or they can be

adapted to create different similar assets.

Despite the extensive use of noise and point sprites, I maintain temporal coherence within the animations. Computations based on distance to the camera, are locked to a single representative value so they don't re-evaluate at every frame. The methods I employ for stippling (minimum distance and maximum color difference) keep sprite popping to a minimum.

The geometry is rendered with constant colors or simple toon-based shading techniques. Any intricate details are portrayed with point or line geometry instead of a traditional shader to mimic fine illustrative brushwork. Carefully chosen color palettes, separate shadow geometry, and level-of-detail rules also add to the likeness of Earle's style.

For example, the stippling in the bushy trees achieves a similar effect to that found in the tree canopies of Eyvind's work – defining the shape and lighting quality of the trees. The bushy tree system also takes into consideration the amount of detail appropriate for their distance from the camera. In the cliff scene, the forest changes in amount of detail and in color as it recedes into the distance. The shader of the decaying ocean seafoam in the coast scene matches the corresponding look in Eyvind's work by matching the way the shapes pack together and the transition between colors.

6.2 Still and Animated Work

Resulting from the analysis and methods presented in this thesis are a series of rendered stills and fly-throughs. Figures 6.1, 6.2, and 6.3 show rendered images of the final scenes. These stills as well as video files of the scenes are included as a digital download with the thesis.

The scenes exhibit a variety of compositions and content with a high degree of

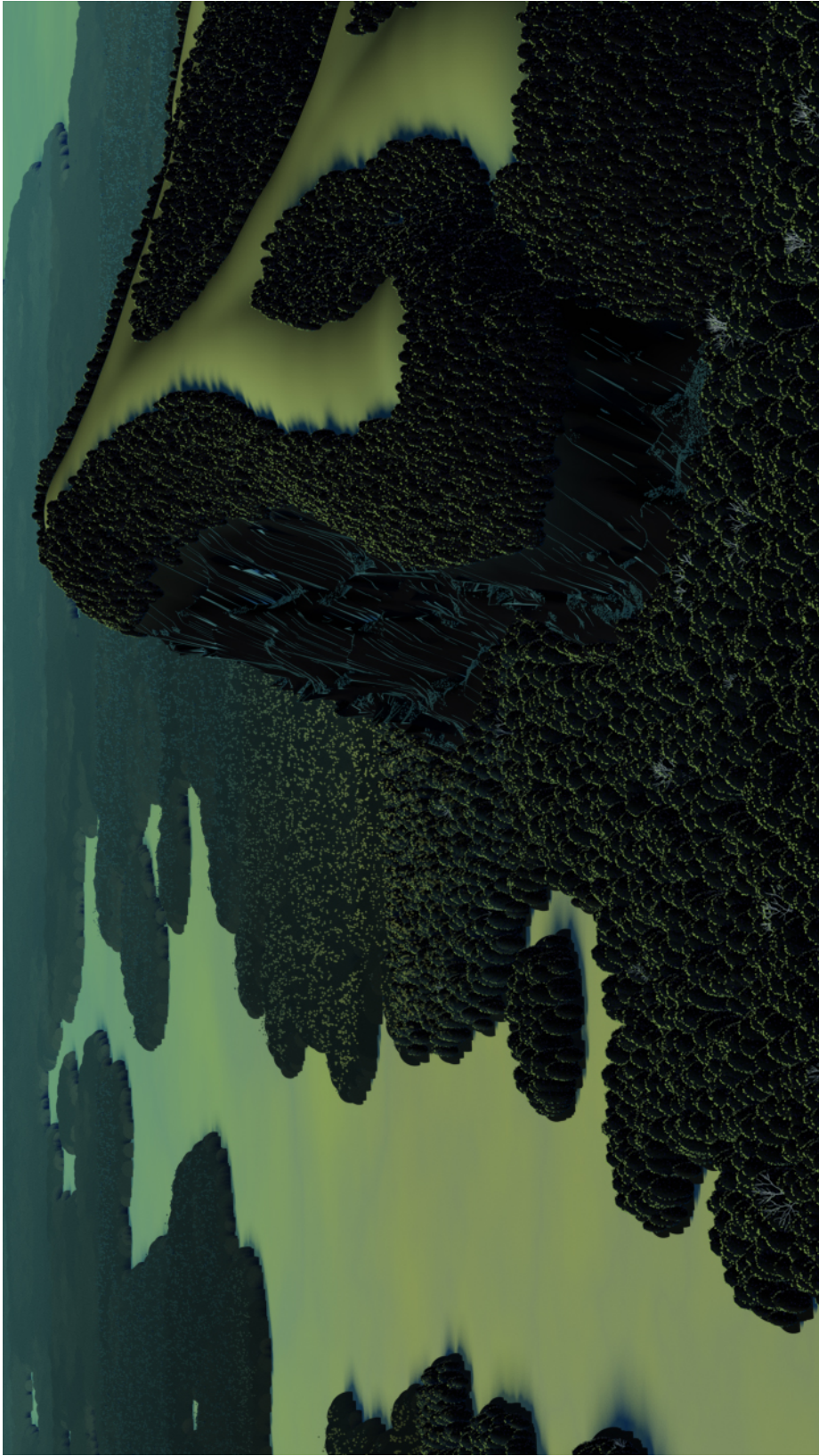


Figure 6.1: A cliff surrounded by forest.

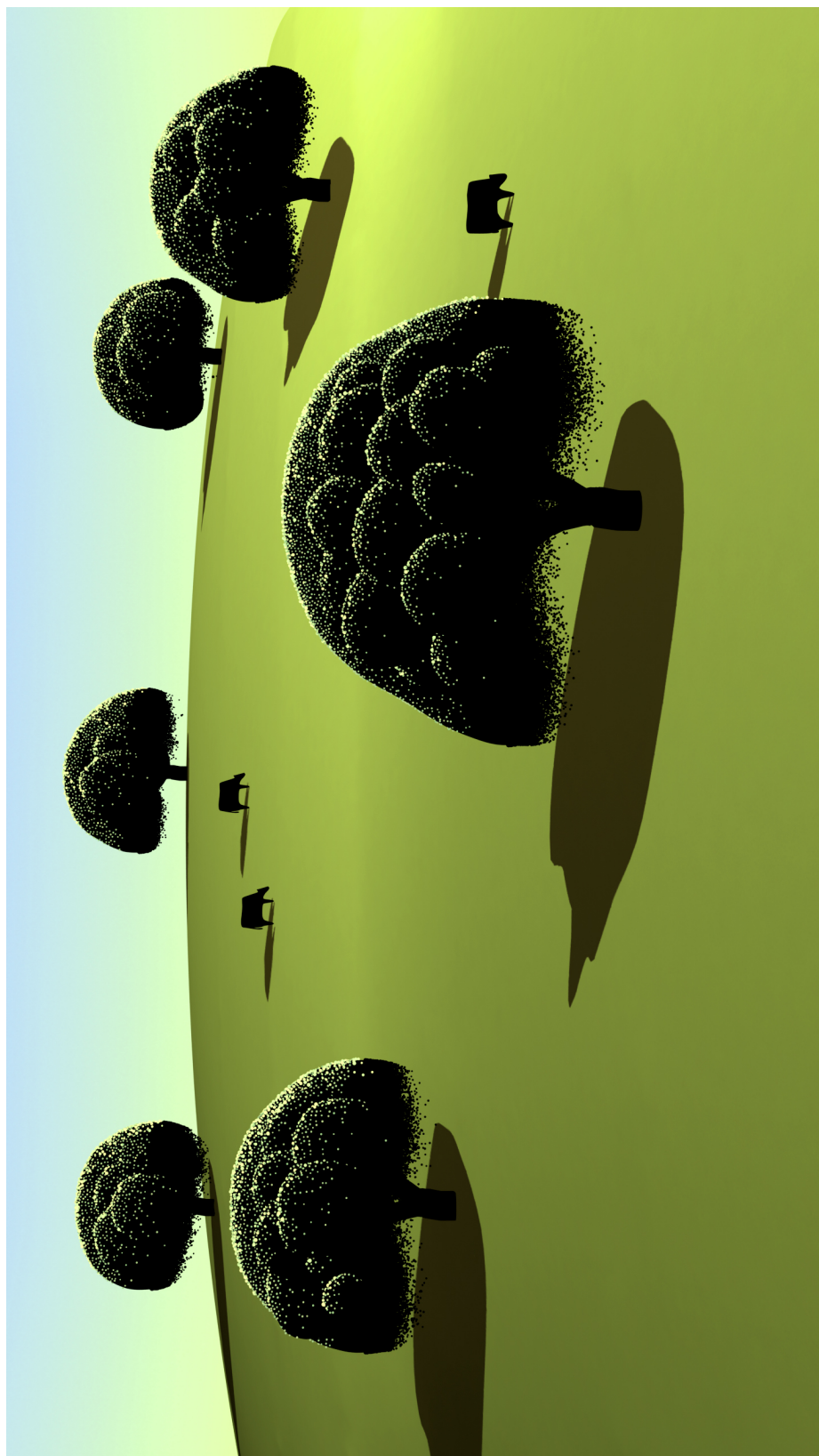


Figure 6.2: A hill with trees.

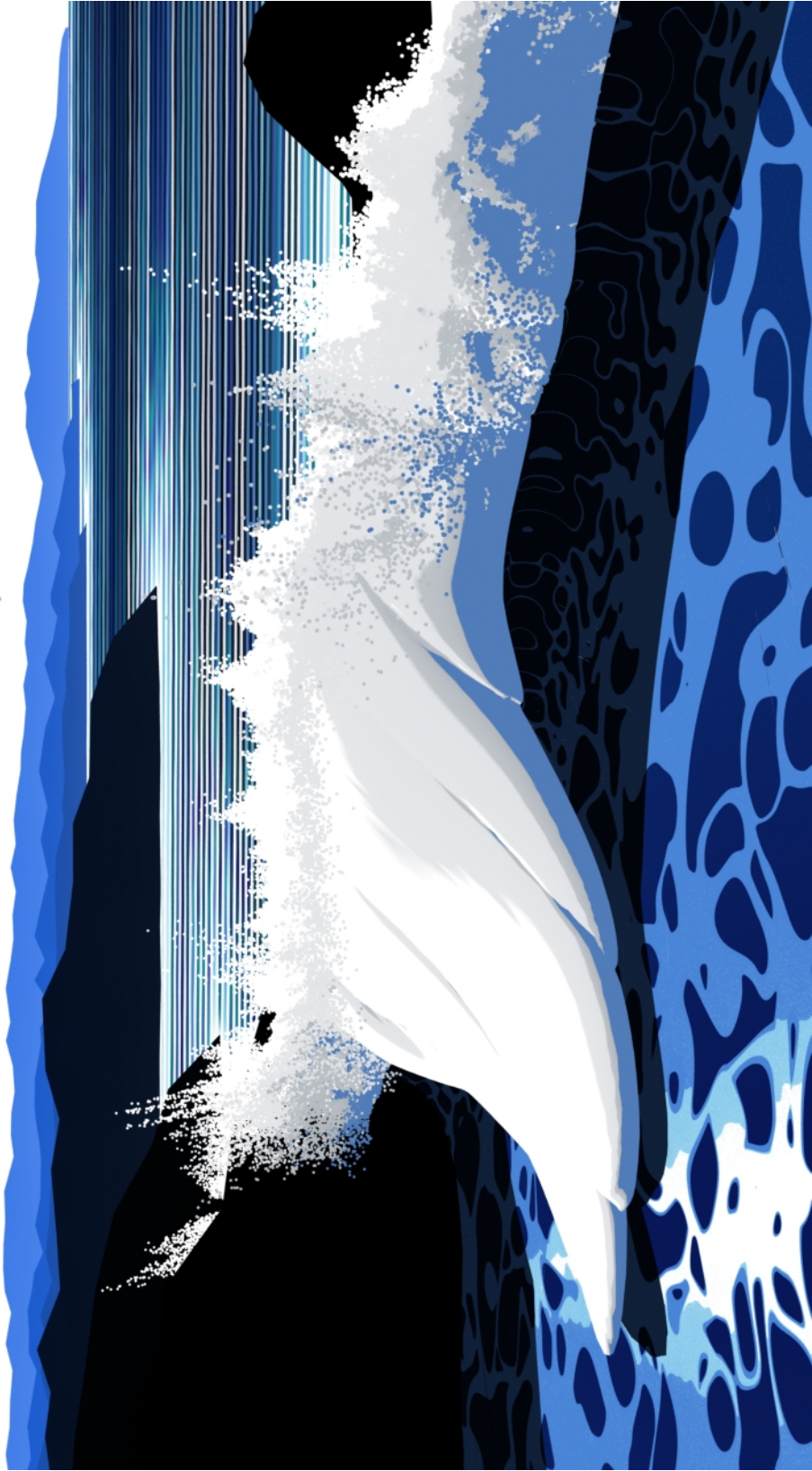


Figure 6.3: Coast scene with a wave.

similarity to the unique graphic style of Eyvind Earle’s work. The assets described in the previous chapter are combined into three final example scenes. Each is rendered as a video with a simple camera move. Notably, the hill with trees scene is based heavily on *Cachuma Ranch* (1999), and the coast scene is based heavily on *The Wave* (1990). The cliff scene is a more general interpretation of Eyvind’s compositions featuring a forest and deep atmosphere.

Additionally I chose a view of each scene to be rendered at a high resolution and printed on matte fine art paper. The high resolution of the printed stills gives viewers a similar experience to viewing one of Eyvind’s paintings or serigraphs in real life. Modern screens and standard digital viewing devices do not have a high enough resolution to display the vast amounts of intricate, minute details present in the physical copies of Eyvind’s work.

6.3 Problems

The default Mantra micropolygon renderer caused render artifacts in some of the assets. Figure 6.4 shows three examples. The issue was solved by switching to the Mantra Ray Tracer when necessary. The point sprite representation of the forest in the cliff scene rendered significantly differently. The Mantra Ray Tracer gives a softer quality to the many points used in the farther regions of the forest, while the Mantra Micropolygon Render makes the point sprites sharp and crisp with a high degree of contrast. In the coastal wave scene, the micropolygon render caused erroneous speckles in the seafoam shader and jagged boundaries between lines in the ocean lines shader.

These scenes are memory intensive to interact with and computationally expensive to generate. Generally for my system, only initial computation costs are expensive—when the entire node network is evaluated. Depending on the number of

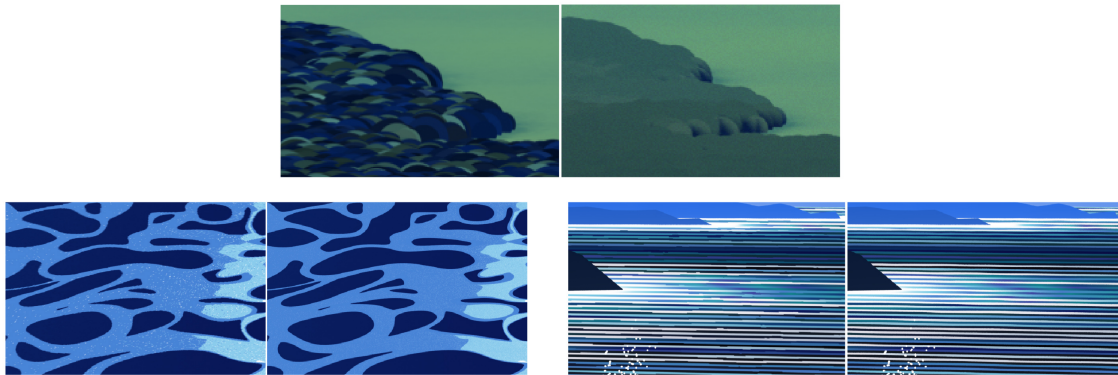


Figure 6.4: Render artifacts solved by switching from Mantra Micropolygon Render to Mantra Ray Tracing Render.

assets and the amount of detail geometry in the scene, a vast amount is produced and needs to be stored in memory. This was a problem when developing many aspects of the system; and it was necessary to block portions of the network from evaluation that weren't critical to the task at hand. The cliff scene was the worst memory-intensive scene. Despite running on the near top-of-the-line computer with 32GB of RAM, this was not enough. I had to develop, view, and render the scene in sections to avoid excessive delays. Research into better hardware and software for rendering the geometry and images is not worth pursuing in a project like this, as it is not the focus and best left to the professional computer engineers.

Another flaw is that much of the geometry is generated "to camera" – meaning with a preferential viewing direction. The look of the geometry breaks down with extreme camera angle changes. Theoretically this is a problem, but in reality for a rendered video or still, the scene is only viewed from certain directions. So, practically this is not a huge issue.

A final shortcoming is a lack of completeness due to the scope of this thesis. Most prominently, the ocean wave is missing the fractal-like edge on the crest of



Figure 6.5: Examples of similar shapes across different aspects of Eyvind Earle’s work.

the breaking wave. I settled for a smooth edge, which is good enough to obtain a successful scene image. Secondly, the level of detail range on the bushy trees is relatively shallow. My system only accounts for middle distance trees; it supports neither the closest nor the farthest examples I referenced from *Cachuma Ranch*.

6.4 Future Work

Implementing an artist’s style and techniques is a huge undertaking. While there are many aspects of Eyvind Earle’s paintings that I developed systems for, there are many more that I did not. He has so many paintings and graphic works, that it is not possible to cover all of them in a project like this. Even a project much larger in scope would still only cover a subset of available styles and landscape items because there is so much variability resulting from artistic design. Nonetheless, a more complete catalog of systems developed for the items would be desirable.

One convenient feature of Eyvind Earle’s work is that he uses similar shapes in many aspects of his paintings. Leveraging this fact would speed the development of further assets. Once a method of creating a certain shape is determined, it could be adapted to different objects. For example, the way the hills erode and slope is very

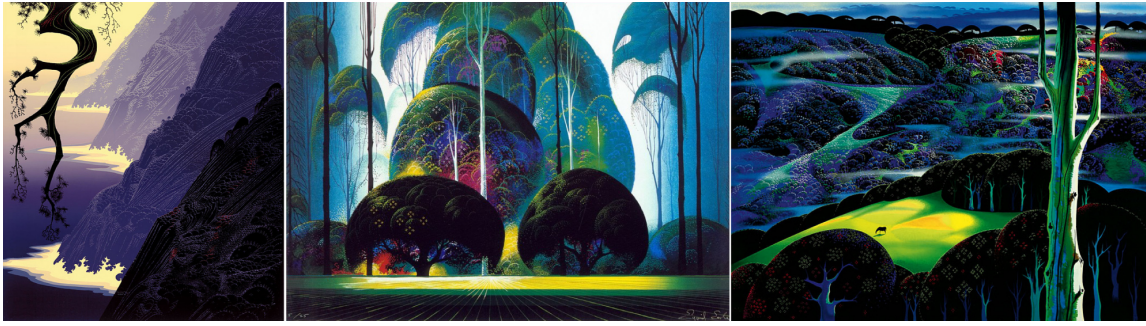


Figure 6.6: Three example landscape types in Eyvind Earle’s paintings. Left: *Purple Mist* (1989). Center: *Green Forest* (1989). Right: *A Touch of Magic* (1997).

similar to the look of a breaking wave, which is also similar to the snow piled on trees (see figure 6.5).

From this, a comprehensive scene generator could be developed. There would be several compositional presets like: coastal receding mountain ranges, a eucalyptus grove behind a field, or rolling hills with swooping clearings (see figure 6.6 for examples). This would also have controls for lighting and color palette, since these stylized landscapes do not strictly follow standard lighting models or conventional colorings.

Potentially an extension of these tools could be used in a large scale production environment, like a movie. More focus would need to go into improving the user interface for controlling details, rather than making compositional assumptions like in a scene generator. Arising from the strict design requirements of film, the system would need both global and local editing capabilities so that property changes in one area don’t necessarily affect surrounding areas (tree settings, lighting direction and color, etc.).

Dynamic level-of-detail would be an important improvement as well. In my implementation, each instance of an object is built given a particular distance from the

camera. Ideally, when the camera moves, the object should update its appearance to reflect the correct level-of-detail for the new depth. This would require developing schemes for each object to smoothly transition the amount of detail. Since the complexity of the features will agree with the depth from camera, by definition this increases the scene's robustness to camera movements (see section 6.3 for details on the problems with camera robustness). A potential solution could be similar to [24].

Finally, increasing the amount of available memory would generally allow for more possibilities and creativity when authoring scenes. More memory would allow the infusion of more details into the assets and a greater number of assets per scene. Small memory improvements could be made through out the generative system. Better hardware that will be available in the future would also be a solution.

REFERENCES

- [1] Roger Allers and Jill Culton. Behind the trees: Making “Open Season”. In *Open Season*. Sony Pictures Animation, 2006. DVD.
- [2] Roger Allers and Jill Culton. *Open Season*. Sony Pictures Animation, 2006.
- [3] Eyvind Earle. *The Disney Wikia*. Accessed 2/6/14. http://disney.wikia.com/wiki/Eyvind_Earle.
- [4] Eyvind Earle. *Horizon Bound on a Bicycle*. Earle and Bane, 1990.
- [5] Eyvind Earle. *Painting A Painting*. Eyvind Earle Publishing LLC, 1991. DVD.
- [6] Eyvind Earle and Ioan Szasz. *The Complete Graphics of Eyvind Earle*, volume I, 1940-1990. Eyvind Earle Publishing LLC, 1993.
- [7] Eyvind Earle and Ioan Szasz. *The Complete Graphics of Eyvind Earle*, volume II, 1991-2000. Eyvind Earle Publishing LLC, 2001.
- [8] Eyvind Earle: Artist of the Twenty-First Century. Accessed 9/2/13. <http://www.eyvindearle.com>.
- [9] Mike Gabriel and Eric Goldberg. Design: Still images. In *Pocahontas*. Disney, 1995. 10th Anniversary Edition, 2005. DVD Disk 2.
- [10] Mike Gabriel and Eric Goldberg. *Pocahontas*. Disney, 1995. 10th Anniversary Edition, 2005. DVD.
- [11] Philip Galanter. What is generative art? complexity theory as a context for art theory. In *In GA2003–6th Generative Art Conference*. Citeseer, 2003.

- [12] Clyde Geronimi, Les Clark, Eric Larson, and Wolfgang Reitherman. Eyvind Earle: The man and his art. In *Sleeping Beauty*. Disney, 1959. Platinum Edition, 2008. Blu-ray Disk 2.
- [13] Clyde Geronimi, Les Clark, Eric Larson, and Wolfgang Reitherman. Picture perfect: The making of Sleeping Beauty. In *Sleeping Beauty*. Disney, 1959. Platinum Edition, 2008. Blu-ray Disk 2.
- [14] Clyde Geronimi, Les Clark, Eric Larson, and Wolfgang Reitherman. Sleeping beauty with audio commentary. In *Sleeping Beauty*. Disney, 1959. Platinum Edition, 2008. Blu-ray Disk 2.
- [15] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A non-photorealistic lighting model for automatic technical illustration. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 447–452, New York, NY, USA, 1998. ACM.
- [16] Doug Ikeler, Max Bruce, Darren Lurie, and Sean Mullen. The art of open season: Traditional 3d styling with today's bells and whistles. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06. ACM, 2006.
- [17] Tobias Isenberg, Petra Neumann, Sheelagh Carpendale, Mario Costa Sousa, and Joaquim A. Jorge. Non-photorealistic rendering in context: An observational study. In *Proceedings of the 4th International Symposium on Non-photorealistic Animation and Rendering*, NPAR '06, pages 115–126, New York, NY, USA, 2006. ACM.
- [18] Joan Earle. *The Herald, Monterey County*, March 18, 2005. Accessed 1/21/15. <http://www.legacy.com/obituaries/montereyherald/obituary.aspx?n=joan-earle&pid=3305591>.

- [19] Sung Ye Kim, Ross Maciejewski, Tobias Isenberg, William M. Andrews, Wei Chen, Mario Costa Sousa, and David S. Ebert. Stippling by example. In *Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering*, NPAR '09, pages 41–50, New York, NY, USA, 2009. ACM.
- [20] Matt Kuruc, Nancy Tsang, and Tom Nettleship. Art-directable canopies in pixar’s vegetation pipeline. In *ACM SIGGRAPH 2014 Talks*, SIGGRAPH '14, pages 69:1–69:1, New York, NY, USA, 2014. ACM.
- [21] Yotam Livny, Feilong Yan, Matt Olson, Baoquan Chen, Hao Zhang, and Jihad El-Sana. Automatic reconstruction of tree skeletal structures from point clouds. In *ACM SIGGRAPH Asia 2010 Papers*, SIGGRAPH ASIA '10, pages 151:1–151:8, New York, NY, USA, 2010. ACM.
- [22] Michael Losure. *A Non-Photorealistic Model for Procedural Painterly Rendered Trees in the Style of Carot*. Master’s thesis, Texas A&M University, 2008.
- [23] L-system surface node. *Houdini 12.5 Documentation*. Accessed 9/15/13. <http://www.sidefx.com/docs/houdini12.5/nodes/sop/lssystem>.
- [24] Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Loring S. Holden, J. D. Northrup, and John F. Hughes. Art-based rendering with continuous levels of detail. In *Proceedings of the 1st International Symposium on Non-photorealistic Animation and Rendering*, NPAR '00, pages 59–66, New York, NY, USA, 2000. ACM.
- [25] Barbara J. Meier. Painterly rendering for animation. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 477–484, New York, NY, USA, 1996. ACM.

- [26] Laura K. Murphy and Philip Galanter. Stylized trees and landscapes. In *ACM SIGGRAPH 2014 Posters*, SIGGRAPH '14, pages 22:1–22:1, New York, NY, USA, 2014. ACM.
- [27] Przemyslaw Prusinkiewicz and Aristid Lindenmayer. *The Algorithmic Beauty of Plants*. Springer - Verlag, 1990.
- [28] Stefan Schlechtweg, Tobias Germer, and Thomas Strothotte. Renderbots — multi-agent systems for direct image generation. *Computer Graphics Forum*, 24(2):137–148, 2005.
- [29] Adrian Secord. Weighted voronoi stippling. In *Proceedings of the 2Nd International Symposium on Non-photorealistic Animation and Rendering*, NPAR '02, pages 37–43, New York, NY, USA, 2002. ACM.
- [30] Arthur Shek, Dylan Lacewell, Andrew Selle, Daniel Teece, and Tom Thompson. Art-directing disney’s tangled procedural trees. In *ACM SIGGRAPH 2010 Talks*, SIGGRAPH '10, pages 53:1–53:1, New York, NY, USA, 2010. ACM.
- [31] Charles Solomon. *The Art of Frozen*. Chronicle Books, 2013.
- [32] Charles Solomon. *Once Upon a Dream: From Perrault’s Sleeping Beauty to Disney’s Maleficent*. Disney Editions, 2014.
- [33] Michael Sporn. 2 Beauties. In *Splog*. Michael Sporn Animation, Inc., 03 April 2006. Accessed 1/13/15. <http://www.michaelspornanimation.com/splog/?p=346>.
- [34] Michael Sporn. SB Pro & Con. In *Splog*. Michael Sporn Animation, Inc., 03 December 2008. Accessed 1/13/15. <http://www.michaelspornanimation.com/splog/?p=1678>.

- [35] Michael Sporn. Eyvind Earle – recap. In *Splog*. Michael Sporn Animation, Inc., 25 September 2012. Accessed 1/13/15. <http://www.michaelspornanimation.com/splog/?p=3151>.
- [36] Michael Sporn. Egos, Books, and Michael Ocelot. In *Splog*. Michael Sporn Animation, Inc., 29 September 2012. Accessed 1/13/15. <http://www.michaelspornanimation.com/splog/?p=3157>.
- [37] Carey Villegas, David Seager, and Kelly Port. Inside the magical world of Disney’s Maleficent. In *SIGGRAPH 2014 Computer Animation Festival - Production Sessions*. ACM, 2014.
- [38] Eyvind Earle. *WikiArt Visual Art Encyclopedia*. Accessed 9/2/13. <http://www.wikiart.org/en/eyvind-earle/>.

APPENDIX A

L-SYSTEM INFORMATION

A.1 L-Systems in Houdini

The information in this section can be found in the L-system SOP help page [23].

Houdini L-system rules adhere to the following format. Each term is described in table A.1. I am omitting the context sensitive terms.

$$\textit{symbol}[: \textit{condition}] = \textit{replacement}[: \textit{probability}]$$

For example take the rule:

$$F : t < 5 = FF : .5$$

This rule means that while t (the number of iterations) is below 5, each “F” will be replaced by “FF” 50% of the time. Table A.2 details a selection of available operations, and table A.3 details the default variables and constants.

symbol	The symbol to replace. For example, if the symbol is A, occurrences of A in the initial string will be replaced with replacement (if this rule matches).
condition	An optional expression that must be true for this rule to match.
replacement	The string that will replace the symbol (if this rule matches).
probability	The optional chance (between 0 and 1) that this rule will be executed. For example, using 0.8 means this rule will execute 80% of the time.

Table A.1: Houdini L-system rule format terms [23]

F(l,w,s,d)	Move forward (creating geometry) distance l of width w using s cross sections of d divisions each.
H(l,w,s,d)	Move forward half the length (creating geometry) distance l of width w using s cross sections of d divisions each.
T(g)	Apply tropism vector (gravity). This angles the turtle towards the negative Y axis. The amount of change is governed by g. The default change is to use the Gravity parameter.
+(a)	Turn right a degrees. Default Angle.
-(a)	Turn left a degrees (minus sign). Default Angle.
&(a)	Pitch up a degrees. Default Angle.
^(a)	Pitch down a degrees. Default Angle.
\(a)	Roll clockwise a degrees. Default Angle.
/ (a)	Roll counter-clockwise a degrees. Default Angle.
	Turn 180 degrees
*	Roll 180 degrees
~(a)	Pitch / Roll / Turn random amount up to a degrees. Default 180.
”(s)	Multiply current length by s. Default Step Size Scale.
!(s)	Multiply current thickness by s. Default Thickness Scale.
;(s)	Multiply current angle by s. Default Angle Scale.
_ (s)	Divide current length (underscore) by s. Default Step Size Scale.
?(s)	Divides current width by s. Default Thickness Scale.
@(s)	Divide current angle by s. Default Angle Scale.
’(u)	Increment color index U by u. Default UV Increment’s first parameter.
#(v)	Increment color index V by v. Default UV Increment’s second parameter.
%	Cut off remainder of branch
\$(x,y,z)	Rotates the turtle so the up vector is (0,1,0). Points the turtle in the direction of the point (x,y,z). Default behavior is only to orient and not to change the direction.
[Push turtle state (start a branch)
]	Pop turtle state (end a branch)

Table A.2: Houdini L-system commands [23]

a	The value of the Angle parameter.
b	The value of the b parameter.
c	The value of the c parameter.
d	The value of the d parameter.
g	The age of the current rule, initially 0.
i	The offset into the current L-system string where the rule is being applied.
t	The iteration count, initially 0.
x, y, z	Current turtle position in space.
A	Arclength from the root of the tree to the current point.
L	Current length increment at the point.
T	The value of the Gravity parameter.
U	Color map U value.
V	Color map V value.
W	Width at the current point.

Table A.3: Houdini L-system variables [23]

A.2 Rule Sets

Axiom	FFFA
Generations	9
Thickness	0.493
Thickness Scale	0.688
Step Size	0.25
Step Size Scale	0.9
Angle	28
Angle Scale	0.7
Rules	A="'[B]////[B]////B B=&FFFA

Table A.4: Bushy tree trunk L-system rules. This rule set also happens to be one of the presets on the L-system SOP.

Axiom	FVFVFVFVFVFVFVVVB
Generations	10
Thickness	0.306
Thickness Scale	0.602
Step Size	0.1
Step Size Scale	0.779
Angle	36.8
Angle Scale	0.7
Gravity (Tropism)	-5
Rules	B="(.95)!(.9)[QA]VVVBVB A=++(2)~(10)CCCC[;(9)-!VVVVVV"S] C=~(1)FC:.7 C=F:.3 V=~(1)FTV:.8 V=~(1)FT:.2 S=Q[+(2)C[;(9)-!VVVVVV"S]]:.5 S=Q[+(2)C[;(9)-!VVVVVV"S]][- (2)C[;(9)+!VVVVVV"S]]:.5 Q=/(1037)Q:.95 Q=/.05

Table A.5: Angular tree L-system rules

Axiom	FFFF-(1)FFFFFFFFF-(1)FFFFFFFFFFFFFFFFFFFFFFFFF-(1) FFFFFFFFV
Generations	200
Thickness	0.3
Thickness Scale	0.602
Step Size	0.1
Step Size Scale	0.99
c (rotation angle)	18 to 60
Gravity (Tropism)	2
Rules	V="!(.995)/(rand(x*453+y*23434+z*976875)*c)F!(.3)B]V B=[+(110)R] R=C:.5 R=FF:.5 C=T(5)-(1)F[L]C:.1 C=C:.85 C=TF[L]F[L]F[L]F[L]C:.05 L=[^(30)!(.95)D]:.5 L=[&(30)!(.95)D]:.5 D=TFD:.1 D=D:.9

Table A.6: Spruce tree L-system rules

Premise	VVVX
Generations	12 (short) to 100 (tall)
Thickness	0.158
Thickness Scale	0.9
Step Size	0.1
Step Size Scale	0.636
Angle	25
b	generations
c	thickness
Gravity (Tropism)	30
Rules	$X:t < b-7 = V[+(25)T((1.0-t/b)*100)B]!(.991)X$ $X:t = b-7 = F[+B]-/(180)B$ $B=!!!!Y(5)$ $V=F(.1*\text{rand}(x*664545+y*5346546+z*546663+564))$ $\quad /(\text{rand}(x*.6545+y*.3465+z*.663+64)*180)$ $Y(i):i > 0 = !!".75)U[+Y(i-1)][-Y(i-1)]$ $U=WW:.7$ $U=WWW:.3$ $W:t < .6*b = T((1.0-.6)*T*W/c)\sim(5)F$ $W:t > .6*b = T((1.0-t/b)*T*W/c)\sim(5)F$

Table A.7: Leafless bushy tree L-system rules

Axiom	X(3)
Generations	15
Step Size	1
Step Size Scale	0.95
Angle	10
Rules	$X(i):i>0 = "VX(i-1):.2$ $X(i):i>0 = "/(180)VX(i-1):.2$ $X(i):i>0 = "[VX(i-1)]["/(180)VX(\min(i,2))]:.3$ $X(i):i>0 = "[/(180)VX(i-1)]["VX(\min(i,2))]:.3$ $V = F-F-F-F'(.0002)\#(.0001)$

Table A.8: Hill shape L-system rules