An Exploration of Non-Photorealistic Rendering Techniques And Styles To Bridge Mood and Style Using Unreal Engine

Ethan Richards
ericha7@clemson.edu

Follow this and additional works at: https://tigerprints.clemson.edu/all_theses

Recommended Citation
Richards, Ethan, "An Exploration of Non-Photorealistic Rendering Techniques And Styles To Bridge Mood and Style Using Unreal Engine" (2024). All Theses. 4258.
https://tigerprints.clemson.edu/all_theses/4258

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.
AN EXPLORATION OF NON-PHOTOREALISTIC RENDERING TECHNIQUES AND STYLES TO BRIDGE MOOD AND STYLE USING UNREAL ENGINE

A Thesis
Presented to
the Graduate School of
Clemson University

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

by
Ethan C. Richards
May 2024

Accepted by:
Dr. Eric Patterson, Committee Chair
Prof. Anthony Summey
Dr. Daljit Singh Dhillon
Abstract

Each story is married to unique styles in which the story is told. The study completed in this thesis explores the relationship between a story and the visuals created to display that story both artistically and technically through the development of *A Fateful Night*, a collection of two non-photorealistic 3D art pieces. This project explores and integrates the themes, motifs, architecture, composition, and color of Cosmic Horror and Gothic Horror art alongside modern non-photorealistic styles through the development of materials, shaders, and post-processing shaders within the real time graphics engine, Unreal Engine. The project successfully led to the development of two stylized art pieces that wove in the desired themes and integrated the technical tools of deferred rendering.
Dedication

This paper is dedicated to my family and friends who have supported me through my academic endeavors. I would specifically like to thank my parents for everything they have done to enable me to achieve my goals.
Acknowledgments

I would like to acknowledge and thank my advisors, Dr. Eric Patterson, Professor Anthony Summey, and Dr. Daljit Singh Dhillon for being my guides through the development of this project and paper. Your guidance, support, and critique shaped the successes throughout this project.

I would also like to acknowledge all of those who assisted in critiques, markups, and suggestions for the project, I appreciate your words, energy, and assistance. This project would not be the same without you.
# Table of Contents

Title Page ................................................................. i
Abstract ................................................................. ii
Dedication ............................................................... iii
Acknowledgments ......................................................... iv
List of Figures ........................................................... vii

1 Introduction ........................................................... 1
   1.1 A Fateful Night ................................................. 1
   1.2 Artist Statement ............................................... 2

2 Story, Style, and Design References and Concept Development .......... 3
   2.1 Gothic and Cosmic Horror Story Reference ..................... 3
   2.2 Building Style References .................................... 4
   2.3 Developing a Story ............................................. 8
   2.4 Creating Concept Art ......................................... 9

3 Rendering Systems and Techniques .................................. 15
   3.1 Deferred Rendering ............................................ 15
   3.2 Virtual Textures ............................................... 16
   3.3 Technical Methods Used in the Painterly Stylization .......... 16
   3.4 Technical Methods Used in the Comic Stylization ............ 18

4 Scene Design and Development ..................................... 20
   4.1 Modeling ....................................................... 20
   4.2 Painterly Style UV Creation .................................. 20
   4.3 Comic Style UV Creation ...................................... 22
   4.4 Surfacing Techniques in Substance Painter ................... 22
   4.5 Scene Setup in Unreal Engine .................................. 23

5 Development and Results of the Painterly Style in Unreal Engine ....... 24
   5.1 Material Creation .............................................. 24
   5.2 Post Processing Shaders ....................................... 25
   5.3 Final Results .................................................. 28

6 Development and Results of the Comic Style in Unreal Engine .......... 32
   6.1 Material Creation .............................................. 32
   6.2 Post Processing Shaders ....................................... 33
   6.3 Final Results .................................................. 36
List of Figures

2.1 Architecture Study of Gothic Architecture done to loosely understand the shapes of Gothic Architecture .................................................. 5
2.2 Cover of *Hellboy: Seed of Destruction* Paperback Published By Dark Horse Comics in 1994 [15] .................................................. 6
2.3 Promotional and card art for *Arkham Horror The Card Game*’s expansion [4], *Carnevale of Horrors* .................................................. 7
2.4 Image From Netflix’s Television Show, painted by Tomas Osang Muir *Arcane* [17] ............................... 8
2.5 Image from the animated movie *Teenage Mutant Ninja Turtles: Mutant Mayhem* [18] ............. 8
2.6 Concept art for the animated movie *Spiderman: Across the Spider-Verse* [33] ................................. 9
2.7 Painting test created to experiment with the concept of “painting with light” ............................. 10
2.8 Final concept art design for the painterly stylization ................................................................. 12
2.9 Final concept art design for the comic stylization ................................................................. 13
2.10 Concept Shot Panel One .................................................................................................... 14
2.11 Concept Shot Panel Two .................................................................................................... 14
2.12 Concept Shot Panel Three .................................................................................................... 14
3.1 Kuwahara Filter regions to gather means and standard deviations ......................... 16
3.2 Visual effect of edge detection in Unreal Engine before (left) and after (right) .......... 17
3.3 Example of hatching used to describe form and shadow on a sphere. ......................... 19
4.1 Low (left) and High (right) Poly Model of Hero Building . .................................................. 21
4.2 UV Layout for the Painterly Stylization ............................................................................. 21
4.3 UV Layout for the Comic Stylization ................................................................................. 22
5.1 Scene without a filter (left) next to the scene with the Kuwahara filtering (right) .......... 26
5.2 Scene without a filter (left) next to the scene with the Edge Detection filtering (right) ..... 27
5.3 Scene without a filter (left) next to the scene with the Artistic Highlighting effect (right) .... 28
5.4 Image of all filters applied to the scene ............................................................................ 29
5.5 Frame 3 from Render ......................................................................................................... 30
5.6 Frame 40 from Render ......................................................................................................... 30
5.7 Frame 290 from Render ....................................................................................................... 31
6.1 Scene without a filter (left) next to the scenes with the edge detection effect and the line texturing effect (right) ....................................................... 33
6.2 Hatching Materials that were applied for each darkness range ............................................. 34
6.3 Scene with textured lines (left) next to the scene with the hatching effect added (right) .... 35
6.4 Scene with the hatching effect (left) next to the scene with the added smoke effect (right) ........................................................................................................... 36
6.5 Scene with the smoke effect (left) next to the scene with the added halftones effect (right) ........................................................................................................... 37
6.6 Image of all filters applied to the comic style scene ............................................................. 38
6.7 Frame 3 from Render ......................................................................................................... 39
Chapter 1

Introduction

When visualizing a story, a world of style opportunities makes itself known. Stylization is so broad that it encapsulates the works of Tim Burton’s *A Nightmare Before Christmas* [22] and Sergio Pablos’ *Klaus* [28]. These two films may share the common topic of Christmas, but they are extremely different. Each story is married to unique styles through which they are told. The marriage between each film’s style and the story they tell is imperative to building such beloved films with public and critical acclaim. The study completed in this thesis explores the relationship between a story and the visuals created to display it both artistically and technically.

1.1 A Fateful Night

The goal in the creation of *A Fateful Night* is to evaluate and build the relationship between the story and world being presented and the style in which it is portrayed through custom Non-Photorealistic Rendering (NPR) tools. Through the themes and motifs of Cosmic Horror and Gothic Horror, *A Fateful Night* is built upon the basis of horror through the supernatural and unknown, integrating core influences of architecture, composition, and color. The development of *A Fateful Night* was completed through the use and exploration of stylization techniques involving material, shader, and post-processing shader work.

The utilization of real time graphics allowed for the variables within these systems to be tuned quickly and precisely for quality results. The two art pieces that make up *A Fateful Night* diverged early in the development process to take advantage of the flexibility and functionality
provided in the foundations of Unreal Engine 5. Modern stylization is vast and constantly pushing the boundaries of what is possible, drawing inspiration from media and mediums past, *A Fateful Night* also tackles the challenges of implementing stylized rendering techniques to reinforce the themes and motifs visually.

### 1.2 Artist Statement

As an artist, I have always been enthralled by unique and inspiring ways that worlds and stories have been portrayed. From the strong color contrast and thick black shapes in the *Hellboy* [16] comics from one of my favorite artists, Mike Mignola, to the unique techniques and painterly style used by Fortiche to build Piltover and Zaun in *Arcane* [21], both artists approach to visualizing stories are vast and diverse. Creations like *Hellboy* [16] and *Arcane* [21] along with the modern stylized films such as *Spiderman: Across the Spider-Verse* [23] and *Teenage Mutant Ninja Turtles: Mutant Mayhem* [18] and unique games such as *Hades* [26] are large inspirations for me as an artist. My love and passion for these stories, their worlds, and their respective mediums serve as a motivational foundation for *A Fateful Night*.

The foundations of this project are built on the development of a love and a passion for the mysterious and strange. Whether they are created through dark cinematics and fear inducing themes, or they are done through unsettling visuals and colors that induce such feelings, those tales are enthralling and provide amazing conflict. I desire to build a unity between the storytelling visuals and the story being told in unique ways. The story I am aiming to build the foundations for are inspired by the stories, themes and tropes of Gothic and Cosmic Horror, Urban Fantasy, and Steampunk. I want to communicate to the viewer the few moments before cataclysmic destruction because that moment contains the pinnacle of fear and excitement that the horrors create. In this moment, the antagonizing forces are deep at work, and the viewer knows what may come, creating the sense of anxiety and fear that looking forward can provide.
Chapter 2

Story, Style, and Design References and Concept Development

When building out the concept and style of the shots, it was important to gather references for world building and style, while also defining the themes and moods that will be expressed through the culmination of design and techniques. The basis of the world building and foundation of the concept comes from the desire to create a dark and unsettling world built upon Gothic and Cosmic Horror, while being designed through incorporating visual elements from mediums that have been influential in modern storytelling techniques.

2.1 Gothic and Cosmic Horror Story Reference

There is a tight relationship between a story and the way the story is told. A strong example of a genre that embodies this link is Gothic Horror [10]. Gothic Horror is the battle between man and evil, which can be caused by man or something supernatural in nature. Gothic Horror is a genre dating back to its notable foundation in the publication of Horace Walpole’s *The Castle of Otranto* [32], that has notably become the basis of the themes and motifs of omens, prophecies, and supernatural events [8].

Beyond the words of the page, it has been explored through celebrated movies such as Alfred Hitchcock’s *Psycho* [1] and M. Night Shyamalan’s *The Sixth Sense* [13]. While *Psycho* [1]
brilliantly uses the dark themes visually through silhouetting characters and unsettling angles, M. Night Shyamalan’s The Sixth Sense [13] delivers on the supernatural horror of ghosts and the faults of humanity. Gothic Horror has also seen other adaptations and developments through the creation of the iconic gothic hero, Hellboy. Mike Mignola himself stated that when he had read Dracula that he had “found his thing [30].”

Gothic Horror frequently runs hand in hand with another sub-genre of horror, Cosmic Horror. H.P. Lovecraft, the notable creator of Cosmic Horror, credits the work of the Gothic Horror author, Edgar Allen Poe, to be a turning point in horror history [14]. Lovecraft’s work is sub-genre defining, but from his early works collected within the Macabre Stories [12], it is clear that his stories come from the Gothic Horror genre of his time [12].

The major difference between Cosmic and Gothic Horror is the origin of where the source of fear originates. In Gothic Horror, the power to be feared is traditionally enabled by and caused by the actions of man, while in Cosmic Horror, the powers come from beings and monsters so great that it shatters the mind of man. Another unique trait of Cosmic Horror that differs from Gothic Horror is the prevalent era. Cosmic Horror often ties to the northeastern United States of America in the late 1800s through the early 1900s.

These foundational elements of Gothic and Cosmic Horror drove the world and story being told within A Fateful Night. It displays the surrounding Gothic urban street which drives the viewer directly to the windows of a tower where the summoning of a cosmic entity is afoot.

2.2 Building Style References

2.2.1 Gothic Architecture

The first place in which style references were to be found was in the real world. Gothic architecture is very important to establishing the visuals of the world, as it is foundational to Gothic Horror’s original base in castles from middle ages. To begin the exploration of Gothic architecture, references from iconic buildings like Notre Dame de Reims, Canterbury Cathedral, and Lincoln Cathedral is a must.

Gothic Architecture is known to have been built from the mid-12th century to the 16th century [3]. Much of Gothic architecture can be seen to have frequently repetitive elements, high vaults, flying buttresses, and a great deal of symmetry both across a center line and radially. Many
windows can be seen curving up to a point above.

Gothic Architecture was designed to allow the structure of the building to “paint with light” as a tool to connect to the divine. The windows told stories as the light shines through them. They were considered beautiful in the era they were built, but as they aged began to look darker and gaudy to those who reflected on the buildings.

Using some of the core elements, a loose study of Gothic architecture was done, and can be seen in Figure 2.1. The goal of this study was to pull in the architectural patterns of Gothic architecture, such as the symmetrical flourishes and windows, while exploring loose shapes that make up a Gothic building, such as the high vertical spires that repeat themselves frequently throughout the shape.

![Figure 2.1: Architecture Study of Gothic Architecture done to loosely understand the shapes of Gothic Architecture](image)

Clear influence can be seen in my piece from the silhouette and design components with frequent repetitive vertical features along with symmetrical patterns across the face of the buildings.
2.2.2 Visual Representations of Gothic and Cosmic Horror

A cursory glance at the results of a Google Images search for “Gothic Horror” encapsulates the signature visual motifs of the genre. Dark values seep into almost pure silhouettes while being contrasted by brights as white as paper. When not in greyscale, the colors are primarily desaturated and coated in an ambient blue grey. Iconically, Gothic Horror can also be depicted with vibrant reds. This can be seen on the many covers of reprints of *Dracula*, such as the Prabhat Prakashan publication in 2017 [24] and the Maplewood Classics publication in 2023 [25]. This characteristic reigns true in the works of Mike Mignola, as can be seen on the cover of *Hellboy: Seed of Destruction* [15] shown in Figure 2.2.

![Figure 2.2: Cover of *Hellboy: Seed of Destruction* Paperback Published By Dark Horse Comics in 1994 [15]](image)

The visualization of Gothic Horror differs from that of Cosmic Horror primarily on the front of color. Cosmic horror is much more vibrant with unsettling colors from sea blues and greens to bright pinks and purples, while maintaining a sense of high contrast and darkness. A good example of Cosmic Horror art is the collection of works used for the Arkham Horror tabletop game and
trading card game. The vibrant use of color can be seen directly in the *Arkham Horror: The Card Game* expansion, *Carnevale of Horrors* [4], seen in Figure 2.3.

![Figure 2.3: Promotional and card art for Arkham Horror The Card Game’s expansion [4], Carnevale of Horrors](image)

### 2.2.3 Style References

Stylization of modern animation has entered a magnificent new era. CG films and shows are taking the 3D medium and making worlds look as if they are coming straight out of paintings and comic books. *Arcane* [21] pushed the bounds of animation through its painterly style, created through projected matte paintings, blended 2D hand painted VFX, and brushy textures and colors [5]. In Fortiche’s SIGGRAPH Production Session, they spoke about the projection paintings that they completed onto the 3D models, creating custom paintings for almost every shot. The beauty of these efforts can be seen by the works of Tomas Osang Muir’s matte painting in Figure 2.4.

Another painterly animation that reference was drawn from was *Teenage Mutant Ninja Turtles: Mutant Mayhem* [18], but a very different and unique element included the use of lines and vibrant colors for all of the lights as well as strong highlighting from the lights onto the characters and other environmental components. Figure 2.5 is a strong depiction of the action in the film, where the painterly aspects of the application of light and focus of detail are clear in the composition of the shot. The vibrancy of the light generated at the focal point is contrasted by the dark surroundings, and the mirroring color of red on the right side of the shot.

The final major film reference of styles that were used and pulled together for *A Fateful*...
Night was Spiderman: Across the Spider-Verse [23], specifically the development of work done with Mumbattan. The visual excitement of color comic book style inking, hatching, and halftone dots create a unique style to be brought into CG. The colors provide an inclination of light as they similarly align with value. All of these properties can be seen in Figure 2.6, concept art of Mumbattan.

2.3 Developing a Story

The story being told through these pieces is the beginning of the end of man. A dark cult leader summons all-consuming cosmic gods from beyond the veil in a dark tower. The perspective of
the viewer is intended to be that of a pedestrian who is witnessing the doom from the streets below, helpless in the face of the impending doom. This summoning is happening amidst an ages-old city that’s developing in the turning point of technology and the introduction of steam and industry.

2.4 Creating Concept Art

2.4.1 Developing a Composition

When creating the composition of A Fateful Night, the plan was to create a composition that could be integrated with several styles. Core design components include: a tall and canvas dominating building where the focal point sits high in the building’s windows and a core under-lighting that assists in providing the unsettling doom inspired by Gothic and Cosmic Horror. Architectural elements from Gothic architecture were used such as the tall, pointed windows, symmetrical design, and roofing flourishes. The composition of the piece helps emphasize the themes of Gothic and Cosmic Horror by creating a claustrophobic vertical framing that forces the viewer to face the horrors that sit within the focal building.

The use of color and selection of value was done per stylization as each helped drive the mood and enforce each stylization individually. Although, it should be noted that the vibrancy of color was planned due to the desire for color to align more with the styles within Cosmic Horror art.
2.4.2 Creating a Painterly Stylization

When creating the painterly stylization, the methodology was done through the concept of “painting with light.” Outside of the background, the painting was done through blocking out black silhouettes, and blocking in color with each paint stroke to build out the forms within the structures as a whole. This style is primarily lineless, although some lines were used specifically to reinforce the lighting and emphasize the forms. This concept can be attributed to an exploration of the lighting stylization done within the film, *Teenage Mutant Ninja Turtles: Mutant Mayhem* [18].

When working through the painterly style, a study of integrating the painting with light techniques was completed before creating the chosen style reference seen in Figure 2.7. The results were duller when looking at color vibrancy than the desired goal, and lacking details, although these lessons provided strong guidance when pushing the process further.

![Figure 2.7: Painting test created to experiment with the concept of “painting with light”](image)

The final results of the first stylization, as seen in Figure 2.8, were elevated through the lessons learned when creating painterly study. Color selections and an understanding of how to
carve out forms varying from the parallels of the silhouette to strengthen the forms of the building.

The painterly stylization incorporates the themes of Gothic and Cosmic Horror through the same concepts as what Gothic Architecture does. The piece takes advantage of the layering of light to paint the story of the divine, or in this case, the monstrous. The underglow and green lights tell the story through the windows, as is done with Gothic Architecture. The villains portrayed in this story are both the man and the monster, who are both the sources of conflict in Gothic and Cosmic Horror. The painterly effect was also created in a way that would emphasize the high contrast dark spaces hidden from the light, an important element of both the stories and the style used for the visualization of Gothic and Cosmic Horror.

2.4.3 Creating a Comic Stylization

The second stylization was heavily influenced by classic comic book styles and the creation of the film *Spiderman: Across the Spider-Verse* [23]. Core components of the style include, inking edge lines, hatching on shadows and surfaces, and variation in colors with a relationship between color and value. The inking was done with texture to add a sense of grittiness; a similar brush was used for the hatching. The hatching was done with a variation of thickness and density to express a difference in value beyond that of just color. In shadows and darker areas, the colors are purples and greens, while in the lighter areas, the colors push towards yellows and pinks. The concept also contains a halftone dots texture on the image. Halftone dots originally came from the printing process of comic books, but instead of trying to mimic the physical accuracy and use, here it is used artistically to provide the feel of a comic book. The core colors of the style were pulled from the sickly yellows and greens that are involved within Cosmic Horror. These elements can be seen in the final concept in Figure 2.9.

The comic stylization integrates the themes of Gothic and Cosmic Horror by integrating the vibrancy of color that are heavily tied to the visualization of Cosmic Horror. The style also reinforces the themes through emphasizing the composition components such as the verticality through the edge flow through the vertical lines. The Halftone effect also emphasizes the focal point by creating a sense of unity between everything in contrast to the focal point being the windows.
2.4.4 Creating the Shot

The final shot of this scene will bring the camera in towards the building, where it will pedestal upwards. It will settle on the windows and trolley towards them. This allows for the scene to build up the intensity of the scenario going on beyond the glass windows. Using Procreate Dreams, a rough shot test was made. Panels of the concept shot can be seen in Figures 2.10 - 2.12.
Figure 2.9: Final concept art design for the comic stylization
Figure 2.10: Concept Shot Panel One

Figure 2.11: Concept Shot Panel Two

Figure 2.12: Concept Shot Panel Three
Chapter 3

Rendering Systems and Techniques

3.1 Deferred Rendering

Deferred Rendering is the process of rendering used within Unreal Engine by default. When deferred rendering is used, the processing of the geometry is separated from the implementation of the shading. One of the largest benefits of deferred rendering is the minimizing of the impact on the performance. In the case of the development of stylized rendering, this was beneficial because it enabled the ability to quickly tune the values of the post process shaders. The process of deferred rendering separates the rendering stages into separate stages, the Geometry Stage, the Lighting Stage, the Post-Processing Stage, and the Final Stage [27].

In the Geometry Stage of the system, the base pass of the system is completed by gathering the objects and their material information then storing it into the G-Buffer. The existence of the G-Buffer assists in decreasing the light’s impact on the scene because its impact is now directly associated with the area of the screen space that it covers [34]. The G-Buffer is then accessed by the following parts of the rendering process.

The Lighting Stage uses the pixel shader to compute the direct and indirect lighting. It incorporates the information from the texture buffers in the screen space. The screen space is the 2D image space that is captured from the previous stages of the rendering process.

One of the disadvantages of deferred rendering that became a challenge when implementing stylized rendering effects was the issues with anti-aliasing. Anti-aliasing in deferred rendering has to be completed as a post processing because hardware anti-aliasing would not work correctly [19]. To
circumvent this, Unreal Engine is equipped with different post process techniques. The technique selected for this project was FXAA, or Fast Approximate Anti-aliasing. FXAA is known to reduce visible aliasing while retaining edge sharpness at a practical ms/frame cost [29]. The ms/frame cost is the milliseconds per frame it takes for the anti-aliasing to be completed.

3.2 Virtual Textures

Virtual Textures are a tool in Unreal Engine that allows for the implementation of U-Dimension (UDIM) texturing naming conventions. When each separate UV region is named in this method, Unreal Engine will allow for a collection of image files to be imported as a single Virtual Texture Asset[7]. The use of Virtual Textures was imperative to the creation of a singular material for each of the individual objects that were modeled to create the building.

3.3 Technical Methods Used in the Painterly Stylization

3.3.1 The Kuwahara Filter

The Kuwahara filter is a non-linear smoothing filter that preserves edges. Its origins come from work done by Michiyoshi Kuwahara who developed the process for medical imaging [11], but due to its painterly effect, it has been adopted for artistic purposes. The Kuwahara filter is implemented by using the mean and standard deviation of the surrounding regions of the center pixel. The region with the smallest standard deviation is the region in which defines the color of the center pixel. Figure 3.1 demonstrates how the regions are built around the center pixel. Each region has overlaps with each other region, and contain the center pixel.

![Figure 3.1: Kuwahara Filter regions to gather means and standard deviations](image)
The implementation of the Kuwahara filter in the development of *A Fateful Night* was done anisotropically. This means that the vertical and horizontal radii of the four surrounding regions were independently manipulated to be able to create directional strokes in specific directions.

### 3.3.2 Laplacian Edge Detection

The edge detection used in the painterly style was a Laplacian filter with a small kernel. The kernel uses the surrounding pixels to determine the strength and size that an edge will be based on how different the values are on the edge of the image. A drawback of Laplacian Edge detection is susceptible to noise without a blur filter being used first [20]. However, the implementation in Unreal Engine allows for weaker values to be easily filtered out of the masking, making a basic Laplacian filter suitable for this portion of the project. A visualization of Edge detection in engine can be seen in Figure 3.2.

![Figure 3.2: Visual effect of edge detection in Unreal Engine before (left) and after (right).](image)

A Laplacian filter is only usable for a single channel, or greyscale image, so when applied to the post process of a 3D render, it must be completed on a collection of images. This collection includes the scene depth mask and each channel of the world normals. The strongest value is what is used, and if none of the values are impactful, no edge would be detected, and no edge would be created on the new masked image.
3.3.3 UV Masking

Another technique used in the painterly stylization of A Fateful Night was the development of unique masks on the UVs. These masks are single channel masks that could be used for linear interpolation between two desired outputs depending on their location on the object. In the painterly design, this was extremely beneficial because it was heavily art direct-able. The art direct-ability of this was used for both the generation of the emissive color mask for the windows, and the non-photorealistic edge highlighting. The masks could be passed to post processing by using unnecessary rendering channels.

3.4 Technical Methods Used in the Comic Stylization

3.4.1 Laplacian Edge Detection

Edge Detection in the comic stylization needed to be tackled in a different way from the painterly stylization because its place is much more prominent in the visualization. The edge detection used in the comic stylization could still be implemented with a Laplacian Filter, but a larger kernel for better accuracy and a wider range of strength to allow for thicker line management. The Laplacian filter was also coupled with a nose that would apply grittiness to the lines.

3.4.2 Hatching

The hatching post process effect was created through the implementation of screen space texture mapping using image region masking. Hatching is a technique that uses ink lines in sets to indicate shadow or form on a surface. This can be seen in the visualization of a sphere in Figure 3.3. A variety of different types of hatching exist, and a blend of styles were used in this project.

Texture Objects in Unreal Engine are an expression used to provide a default texture for a texture function input [6]. These are used to bring in the texture hatching samples in order to apply the hatching using the World Aligned Textures node. These nodes allow for the Hatching to sit on the surface of the object statically, instead of flickering and moving whenever the camera moves.
3.4.3 Halftones

Developing the Halftone Textures was completed in a more simplified manner than other post processing effects because of the nature of the effect. The halftone texture is essentially a type of mask to mimic the dot texture that existed in the old techniques of printing comic books. This can be completed with fractal repeating dot texture, although in implementation, it would be even less intensive when implemented with a basic texture coordinate. This can be used by implementing the Red and Green Channels of a texCoord. The texCoord node has a zero to one gradient on the x-axis for the red channel, and a zero to one gradient on the y-axis for the green channel. When rotated 45 degrees, this create the diagonal dotted texture.
Chapter 4

Scene Design and Development

4.1 Modeling

Developing the model of the building was done by breaking down the components of the building into smaller shapes. This allowed for cleaner geometry in the development of each component for UVing. Individual objects that appear symmetrically on the building were modeled and UVed once, and duplicated to increase the pace at which the asset could be developed. The Modeling was done using Image Planes with the concept art with a camera placed to visualize and model the building to match.

A challenge encountered during this section was defining camera settings and determining the distance from the model to provide a consistent scaling. This was important because the painting was not done using a 3D model block-out, so this information was not previously defined.

The high poly model was created in the ZBrush because the initial intent was to create details through sculpting, although after experimenting with surfacing and material techniques, it became evident that completing those details within the surfacing maps of the model would be the more advantageous method. The low and high poly models can be seen in Figure 4.1.

4.2 Painterly Style UV Creation

When UVing for the painterly style, it was planned to use the World Position values to assist in the variation of color, as the plan was not to create a true color map for the building. This
was to implement the “paint with light” concept that was used in the concept art development. This allowed for the UVs to be condensed by stacking the UVs of the repeated model components, such as the symmetrical pipes, windows, and door as seen in Figure 4.2.
4.3 Comic Style UV Creation

The UVs required separation when developing the Comic Style render. This was because a base color map was going to be used, and it was imperative that the colors were not symmetrical. This would mean that the UV tiles required re-evaluation. It was decided that each component would get a full UV tile to keep consistency in map scales across the two styles developed in this project as seen in Figure 4.3.

![UV Layout for the Comic Stylization](image)

Figure 4.3: UV Layout for the Comic Stylization

4.4 Surfacing Techniques in Substance Painter

4.4.1 Painterly Style

When using Substance Painter for the painterly style, there were not many tasks that were needed. The first task was to bake maps such as the normal map and the ambient occlusion. The second task that was completed was by creating a mask for the desired highlighting on the model. This was done as a painted mask because it provides notably strong artistic control over the highlights. The final task was to create an emissive mask to be used to generate the glowing windows effect that is desired.
4.4.2 Comic Style

The work in Substance Painter for the Comic style was more extensive. The model was hand painted using the colors picked from the concept art. Some hand painted inking was done directly onto the base colors of the building during the painting process to emphasize the additional detail lines desired in the style.

4.5 Scene Setup in Unreal Engine

When setting up each scene for each style, there were some consistencies in the process. When building the sky, for both styles, a skybox was used with a custom painted texture material. Additionally for each style, a collection of basic planes with a masked surface material was used to create the background buildings. These were made using the background panels from the concept art, where the layers could be exported independently and expanded upon through further paintings. This method allowed for quick results and easy artistic control over the shapes provided by the background, something that would have been much more challenging if it was modeled. By staggering the distance from the camera, the heights, and the angles, some variations on the background buildings became achievable.

For building the painterly style, the lighting was extremely important when building the scene, which most certainly reinforced the style choice to be “painting with light.” There are four core light sources in the painterly style. Two lights make up the red under glow light, one spotlight aimed directly at the building, and another point light providing more light. The other lights include a blue spotlight for the highlight from the upper right side and a green area light source to strengthen the impact of the emissive texture. These were set up as a foundation but required tuning throughout the development of each stage of the process.

When developing the comic style, the lighting was specifically used to control the locations of the hatching post processing effects. The light colors did not impact the image because the lighting was mostly replaced with the original base colors.
Chapter 5

Development and Results of the Painterly Style in Unreal Engine

5.1 Material Creation

Developing the surfacing material used in the painterly style required a mix of color and noise alongside the maps created in substance painter. The colors on the building are a gradient of color from red to cyan blue. The colors have noise based variance in value to assist the Kuwahara post processing filtering for getting variance in the “paint strokes.” To enforce the darker recesses of the building, the color is further darkened through the use of an ambient occlusion map and a custom mask.

The two other masks created in Substance Painter were also brought in and used on the material. The masks were created through hand painting edges and faces that were desirable for the different features of the drawing. The first mask used for artistically controlled highlighting used the metallic channel, as it was not going to be needed for anything in the style, so it could be used and overwritten with post processing effects. The second mask was a base mask for the emissive component of the texture. This was done as a mask so the emissive color could be controlled within the material instead of having to re-import the material each time it was adjusted.

The development of the material was done using the painting with light in mind, which is the method used in the concept painting. Setting up the blueprint involved using noise to create a dark
colored texture that matched the lighting as the building scales vertically. Then to emphasize the
darker areas, the ambient occlusion is applied to the color maps. The artistic highlighting painted
map is then applied to the metal channel to be used in the post processing filter, while the other
painted map is a mask for the green color as an emissive input to create the light in the windows.

5.2 Post Processing Shaders

The post processing materials carry a majority of the work in developing the styles for the
painterly style. These filters were built with a mixture of custom code blocks and built in blueprint
nodes. The most impacting post processing effect is the Kuwahara post processing filter. The post-
processing effects in the painterly stylization is a series of three post process effects combined to
create the final product.

5.2.1 Kuwahara Filtering

The Kuwahara filter was beneficial to the development of the painterly stylization due to
the ease of controls artistically. The system was easy to control through the inputs created.

The Kuwahara post processing filter is completed primarily with code implemented within
a Custom node as seen in Appendix A Section A.1. The custom code block has five main inputs,
where two are controllable through a Material Parameter Collection. The first input is the Scene
Texture for the PostProcessInput0. This is not actually used by name, but it is required to be
able to allow for the SceneTextureLookup function to be usable, otherwise it will be compiled out
[31]. The second input is the Scene UV. The last input that is not controllable with the material
parameter collection is the ViewSize. Each of these provides core components for the filter.

Controlling the filter’s strength and accuracy comes from the filter’s radius and sample
controls. The vertical and horizontal filter radiuses are controlled separately to provide maximum
control. Controlling the samples is necessary to manipulate the accuracy of the painterly effect
artistically. This control is essential because it has a higher impact on the performance of the engine
when running in real time.

The code within the custom block of the Kuwahara Filter is a combination of two processes.
The first stage is a sobel filter that assists in obtaining the gradient of the pixels with the luminance
of the pixel so they can be used to rotate the paint strokes and provide variance in stroke direction.
The second stage is the painterly Kuwahara component that takes advantage of the angle to offset the samples. The Kuwahara filter uses the mean color value and standard deviation to then create the painterly effect.

Lastly, the Kuwahara filtering is cleaned up with the use of the scene depth and the original image to repair any tearing that may happen from the filter, which was prevalent near light sources. The complete blueprint layout can be seen in Appendix B, Figures 1 - 6.

The Kuwahara filter was also selected over bilateral filtering because in high contrast varying locations, such as the pipes and the spires, the Kuwahara filter preserved the shapes better. The implementation can be seen in Figure 5.1, where it is compared to the scene without any post-processing effects.

![Figure 5.1: Scene without a filter (left) next to the scene with the Kuwahara filtering (right)](image)

5.2.2 Edge Detection

When turned up to strong values to have extreme impacts, the Kuwahara filter would tend to lose the edges of the object. To battle this, the small kernel Laplacian edge detection filter was used where the scene normals and scene depths are used to get the edge extremes. The edges recovered in the edge detection re-emphasize the components of the composition and architectural elements that were lost in the Kuwahara filtering. The strongest variance provides a thicker edge,
and if there were overlapping values, the strongest value would be used.

The edge color is decided by the original color of the image, controlled by a material component brightness value, which controls the highlight. The two other important controls are the overall visibility of the edge filter, as it is not always needed to compensate for the losses on the Kuwahara filter, and the max distance to prevent the background and sky from being impacted. The complete blueprint structure can be seen in Appendix B in Figures 7 - 13. The visualization of the edge detection can be seen in Figure 5.2, where you can see the edge detection highlights compared to the scene with no post-processing effects.

![Figure 5.2: Scene without a filter (left) next to the scene with the Edge Detection filtering (right)](image)

5.2.3 Artistic Highlighting

The artistic highlighting post processing filter is a small but powerful filter. The filter uses the metallic scene texture to create a stronger highlight in desired locations by serving as a mask that is brought in from the metallic channel of the objects in the scene. The implementation of the effect can be seen in Appendix B, Figure 14. The artistic highlighting emphasizes the vibrancy of the window and the lighting to drive the visual intensity in color that is core to visualizing Gothic and Cosmic Horror. The artistic highlighting can be seen in Figure 5.3, where it is compared to the scene without post-processing effects.
The filter uses the mask and increases the strength of the colors in the locations in which the mask is visible. The strength is controllable through the material component collection.

5.3 Final Results

The development of the final shot of the style allowed the different controllable values on the post process effects to push the style’s capabilities that were much less possible in the original 2D concept shot. The filter controls could be used to create camera effects such as coming into focus.

The final results of the painterly style were successful at emulating the mood and themes of the original concept art. The painterly style and darker composition drive home the impending doom and horror themes that align with themes of Gothic and Cosmic Horror. The tall, Gothic influenced building assists in the creation of the mood by alerting the viewer to the setting of the story. The filters could be used and manipulated when creating the shot to allow for different camera locations to have different levels of focus, emphasizing the camera movements. The combining of all filters can be seen in Figure 5.4.

Although both the concept art and the NPR shot captured the essence and mood of Cosmic Horror, the NPR product does not quite hit the mark at matching the style of the concept. The softer edges within the style were not easily captured. The Kuwahara filtering provided a strong
painterly look, but it could benefit from smoother edges created in the geometry stages to work with. The systems used also had a noisier output than anticipated, due to how it interacted with the smaller, more detailed areas. More artistic control could be returned from the tools by allowing for materials to play a larger role in the painterly aspects rather than sticking as core to the concept of the technique “painting with light.” Figures 5.5 - 5.7 are frames from the final renders, showing how the filters look and are changed throughout the sequence to create the final visuals.

Figure 5.4: Image of all filters applied to the scene
Figure 5.5: Frame 3 from Render

Figure 5.6: Frame 40 from Render
Figure 5.7: Frame 290 from Render
Chapter 6

Development and Results of the Comic Style in Unreal Engine

6.1 Material Creation

The material creation for the comic style was less involved than the previous style because more work was done at the texture creation stages in Substance Painter painting. To control the colors a bit more in Unreal Engine, the base colors Texture Sample was converted to HSV that could be adjusted by multiplying each component by constants before being brought back together and converted back into RGB. Although, due to the bottom half of the building having a better base color, the original colors and the modified color controls are combined through a World Position Height Gradient to blend the two together.

The Material Created also uses the metallic channel to implement a mask that is used in the Hatching post process effect. This mask allows for the building to not have halftone dots on the focal point, creating a cleaner visual. The last few material components include the baked normal map.
6.2 Post Processing Shaders

The Process of developing the post process shaders for the comic style differed from the painterly style by being designed with each stage of the effects built atop each other. Outside of implementing the colors of the materials first, this process more similarly matches the artistic processes used in developing comic book pages digitally.

6.2.1 Edge Detection

Unlike the simple edge detection used in the painterly style, the edge detection in the comic style is much more prominent, so a higher fidelity filtering technique was used. The Laplacian Filter was implemented using custom code as seen in Appendix A Section A.2. This was done so the kernel size could be increased. There is only one input into the custom code block. This input is the Scene Texture for the PostProccessingInput0. This is not actually used by name, but it is required to be able to allow for the GetDefaultSceneTextureUV function to be usable. The code uses the normal and depth scene textures to determine the edges.

Figure 6.1: Scene without a filter (left) next to the scenes with the edge detection effect and the line texturing effect (right)
The process after the filtering is complete is the same as what is done in the painterly style. The maximum strength edge is used to define the edge, and a black line is placed on the edge with size variation. See Appendix C where Figures 15 - 17 demonstrate blueprint implementation.

A second post process shader was used to add texture to the lines. This shader used a world-aligned texture noise object with a weakened Constant Bias Scale as a UV input to the PostProcessInput0 Scene texture. This adds grittiness to the lines to mimic a rough ink brush. The implementation of this can be seen in Figures 18 - 20 in Appendix C. The comparison between the unfiltered scene and the effects of the two shaders can be seen in Figure 6.1.

6.2.2 Hatching

The hatching is created by creating a shadow mask and gathering darkness values to apply hand drawn hatching textures that can be seen in Figure 6.2. Obtaining a shadow mask was completed by inverting a light mask along a linear curve. Although, a light mask is not provided as a scene texture in Unreal Engine, so using the scene and base color, a light mask could be created after desaturation. Additionally, the Ambient Occlusion is removed from the mask to obtain those darker recesses [9].

Once the shadow mask is obtained, more curves can be used to obtain different value ranges of darkness. Using these ranges, different hatching Texture objects can be applied to different shadow depths on the mask. Three hatching densities were used in the shader, diagonal, cross, and cross overlapped with pluses.

![Hatching Materials](image)

Figure 6.2: Hatching Materials that were applied for each darkness range

The shadow ranges on the curves used are ranging from .4 - 1, with overlapping regions to blend across the different hatching Texture Objects. These three hatching masks are then combined
6.2.3 Smoke

The smoke on the bottom of the concept art needed to be replicated in the NPR version of the style. It was implemented in the same methods that it would be done on paper. To do so, the color of the smoke is implemented with a post processing shader that is controlled by the world height and uses the SmoothStep node functionality to create a soft gradient. The blueprint implementation can be seen in Figure 26 in Appendix C. The impact of the smoke filter can be seen in Figure 6.4 where it is compared to the previous stage of post-processing.

6.2.4 Halftone Texturing

The Halftone texturing is implemented in the project using a basic texture coordinate red and green channels to create a mask. Initially, it is controlled by the camera movement on the vertical axis, this is so that the dots look as if they move up and down as the camera moves up and down. It is rotated 45 degrees so the halftones are shifted each layer, aligning every other layer.
When the mask is made for the dots, it is combined with the metallic mask to remove the dots from the focal point of the scene. The halftones are then applied to the render. The colors vary from the top to the bottom of the page to allow them to be more or less apparent in artistically chosen places. The final results of the halftone filtering can be seen in Figure 6.5, where it is compared to the previous stage of post-processing.

The halftone textures are designed to have controls that allow for the size and density to be controlled using a material component collection. This allowed for the dots to be manipulated during the sequence for the final render. The implementation of the Halftone Texturing can be seen further in Figures 27 - 33 in Appendix C.

The halftone texture minimizes some of the contrasting elements outside of the focal point through unifying the other compositional elements. This also increases the use of contrast, underscoring the themes of Gothic and Cosmic Horror.

### 6.3 Final Results

The final design for the comic style both matched the style and the mood of the original piece well. The original took the colors, styles, and techniques from a variety of sources, so it was a challenge to balance them all. The vibrancy and color choice created the unsettling feeling that something evil was happening, but it provided a different feeling from that of the painterly style.
The final design for the painterly style can be seen in Figure 6.6.

The camera actions of the comic style much more accurately matched that of the concept shot created in Procreate Dreams because the flatter visuals emphasized the panel like movement, which further aligned with the style of the parallax of a moving comic page. Frames from the final render can be seen in Figures 6.7 - 6.10.

The technical methods used in the comic book style were more robust to the challenges involved in post processing. The flickering that can occur with thin lines was less prevalent because the edge detection used thicker lines, and the hatching was done with texture objects instead of lines developed by the post process effect.

A stylistic technical change that may provide better results is changing the halftone texture so that it is static to the 3D world rather than the 2D screen space. This would provide a more consistent movement of the post processing effect as the camera was moved rather than basing it off of the camera movements.

Overall, this style matched better between the concept art and the NPR scene, but the scene’s explicit connection to the themes of Cosmic Horror were not as strong. Because the composition of the scene was overall brighter, the impact of the smaller focal point in the window of the
building is less. A change that could be made to emphasize the themes that would connect with the style of this version would be to have the building already wrapped in the tentacles of the monster.

Making major compositional changes to the scene would be rather easy because the system is designed in a way that is more dependent on the asset development process than being catered to work with just the assets involved in the shot.
Figure 6.7: Frame 3 from Render

Figure 6.8: Frame 40 from Render
Figure 6.9: Frame 160 from Render

Figure 6.10: Frame 229 from Render
Chapter 7

Conclusion

7.1 Artistic Conclusions and Discussions

The visual results of the two styles did successfully apply the stylistic choices that were aimed for in the artist statement and goals. Both NPR scenes represented the concept art pieces and were able to capture the composition and mood. The overall composition used for both styles successfully captured the design and influence of Gothic architecture and storytelling. The composition of the comic book design could be adjusted to include a larger, more overarching focal point of the cosmic entity attacking, to enforce the themes of horror and doom more.

Both styles married the themes and style references well, creating two ways to tell one story. Each style emphasizes different aspects of the themes. The painterly style was more successful at taking the themes of horror, atmosphere of suspense, and impending doom than the comic style, but the comic style was much stronger at capturing the color and vibrancy that comes with Cosmic Horror. Though, both successfully highlighted the supernatural element in the window.

Each of the pieces collect the send of horror and excitement for what is happening in the windows, capturing the moment before cataclysmic destruction is going to occur. The colors, lighting, and environment provide the sense of danger and fear the emphasize that feeling when the monster is seen in the window. The filter techniques help drive the sense of danger and align with the more supernatural and uncertainty aspects of the world by being fluid and unreal in their own way.

The materials and post processing work were a strength of the project. The understanding
of the systems benefited the ability to cohesively connect the components of the work across each stage of the visual development. A weakness of the piece is the lack of detail on the modeling side of the project. With more models and higher detailed models, the systems could be tuned further and more accurately.

7.2 Technical Conclusions

The technical systems within this project were designed to be controllable and layered. In most of the project, the post processing systems successfully completed the tasks that they needed to. The use of real time graphics and a deferred rendering system allowed for quick adjustments and enabled development and experimentation quickly.

The technical development on the project was successful, but it could still use improvements or adjustments. The edge detection post processing provided a major challenge due to its flickering tendencies when creating small edges. A method that could be used to combat this issue would be to complete the edge detection at an earlier stage in asset development and generate masks that could be used for edge visualization. This process is similar to the method used for the artistic highlighting, which did not struggle from the same challenges as the edge detection filtering did.

7.3 Future Work

Future evolutions of this project involve recreating similar techniques in other software applications and modifying systems to work with forward rendering. It would be an interesting and unique challenge to apply further exploration of the painterly style using other applications such as Houdini and Rebelle 7. Furthermore, it would be extremely interesting to develop the comic book style further using tools like grease pencil in Blender.
Appendices
Appendix A  Code For Post Process Development

The first code block is two filters combined. The first filter is a sobel filter for edge detection. These values are then used to find the offset angle to angle the Kuwahara Filter, in the second part of the code block. The second code block is a Laplacian edge detection filter using the normal and depth map scene textures.

A.1 Kuwahara Custom Node Code

```cpp
float3 mean[4] = { {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0} };  
float3 sigma[4] = { {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0} }; 
float2 offsets[4] = { {-RADIUS.x, -RADIUS.y}, {-RADIUS.x, 0}, {0, -RADIUS.y}, {0, 0} };  
float2 pos;  
float3 col;  
float gradientX = 0.0f;  
float gradientY = 0.0f; 
float sobelX[9] = {-1, -2, -1, 0, 0, 1, 2, 1};  
float sobelY[9] = {-1, 0, 1, -2, 0, 2, -1, 0, 1};  
int index = 0;  
float2 texelSize = 1.0/VIEWSIZE; 

for(int x = -1; x <= 1; x++) {  
for(int y = -1; y <= 1; y++) {  
if(index == 4) {  
index++;  
```
continue;

}
float2 offset = float2(x, y) * texelSize;
float3 pxCol = SceneTextureLookup(UV + offset, 14, false).xyz;
float pxLum = dot(pxCol, float3(0.2126f, 0.7152f, 0.0722f));

gradientX += pxLum * sobelX[index];
gradientY += pxLum * sobelY[index];

index++;
}
}
float angle = 0;
if(abs(gradientX) > 0.001) {
angle = atan(gradientY / gradientX);
}

float s = sin(angle);
float c = cos(angle);

for(int i = 0; i < Samples; i++){
    for(int j = 0; j <= RADIUS.x; j++){
        for(int k = 0; k <= RADIUS.y; k++){
            pos = float2(j, k) + offsets[i];
            float2 offs = pos * texelSize;
            offs = float2(offs.x * c - offs.y * s, offs.x * s + offs.y * c);
            float2 uvpos = UV + offs;
            col = SceneTextureLookup(uvpos, 14, false);
            mean[i] += col;
        }
    }
}
\[ \text{sigma}[i] += \text{col} \times \text{col}; \]

\}
\}
\}

float \text{n} = (\text{RADIUS}.x + 1) \times (\text{RADIUS}.y + 1);
float \text{sigma}_f;

float \text{min} = 1;

for(int \text{i} = 0; \text{i} < \text{Samples}; \text{i}++){
    \text{mean}[\text{i}] /= \text{n};
    \text{sigma}[\text{i}] = \text{abs}(\text{sigma}[\text{i}] / \text{n} - (\text{mean}[\text{i}] \times \text{mean}[\text{i}]));
    \text{sigma}_f = \text{sigma}[\text{i}].r + \text{sigma}[\text{i}].g + \text{sigma}[\text{i}].b;

    if(\text{sigma}_f < \text{min}) {
        \text{min} = \text{sigma}_f;
        \text{col} = \text{mean}[\text{i}];
    }
}

return \text{col};

[A.2 Comic Stylization Edge Detection Custom Node Code]

float2 \text{KernelUVs} = \text{GetDefaultSceneTextureUV}(\text{Parameters}, 1);
float2 \text{TexelSize} = \text{GetSceneTextureViewSize}(1).zw;
float2 \text{PixelUVs};

float \text{KERNEL\_SIZE} = 15.0;
float HALF_KERNEL_SIZE = floor(KERNEL_SIZE/2.0);
float HALF_KERNEL_SIZE_SQ = KERNEL_SIZE*KERNEL_SIZE/4.0;

float3 LaplacianFilter_Normal = float3(0.0, 0.0, 0.0);
float LaplacianFilter_Depth = 0.0;
float CenterWeight = 0.0;

for(int y = -HALF_KERNEL_SIZE; y <= HALF_KERNEL_SIZE; y++) {
    for(int x = -HALF_KERNEL_SIZE; x <= HALF_KERNEL_SIZE; x++) {
        if(x*x+y*y > HALF_KERNEL_SIZE_SQ) {
            continue;
        }

        CenterWeight++;

        PixelUVs = KernelUVs + TexelSize*float2(x,y);

        LaplacianFilter_Normal -= SceneTextureLookup(PixelUVs, 8, false).rgb;
        LaplacianFilter_Depth -= SceneTextureLookup(PixelUVs, 1, false).r;
    }
}

LaplacianFilter_Normal += SceneTextureLookup(PixelUVs, 8, false).rgb * CenterWeight;
LaplacianFilter_Depth += SceneTextureLookup(PixelUVs, 1, false).r * CenterWeight;

CenterWeight--;

CenterWeight = 1.0 / CenterWeight;
LaplacianFilter_Normal *= CenterWeight;
LaplacianFilter_Depth *= CenterWeight;
return float4(LaplacianFilter_Normal, LaplacianFilter_Depth);

[2]
Appendix B  Blueprint Setups for the Painterly Stylization

The blueprints in this appendix are organized by the filter type. Each filter is lead by an overarching blueprint architecture followed by each component. In order the filters are organized by: Kuwahara, Edge Detection, and then Artistic Highlighting.

![Figure 1: Kuwahara Filter: Blueprint Structure](image)

Figure 1: Kuwahara Filter: Blueprint Structure
Figure 2: Kuwahara Filter: Controllable Inputs

Figure 3: Kuwahara Filter: Custom Node Integration (Note: Code in Appendix A)
Figure 4: Kuwahara Filter: Original Post Process Input nodes

Figure 5: Kuwahara Filter: Scene Depth Mask nodes

Figure 6: Kuwahara Filter: Nodes to combine using Scene Depth and color correct
Figure 7: Edge Detection Filter: Blueprint Structure

Figure 8: Edge Detection Filter: Kernel Structure
Figure 9: Edge Detection Filter: Applying Kernel
Figure 10: Edge Detection Filter: Get Strongest Edge
Use Edge Detection for Increasing Brightness of Edges based on Material Collection Parameter

Figure 11: Edge Detection Filter: Set Edge Brightness

Save Background

Figure 12: Edge Detection Filter: Preserve Background
Figure 13: Edge Detection Filter: Overall Visibility Control

Figure 14: Artistic Highlighting: Blueprint Structure
Appendix C  Blueprint Setups for the Comic Stylization

The blueprints in this appendix are organized by the filter type. Each filter is lead by an overarching blueprint architecture followed by each component. In order the filters are organized by: Edge Detection, Edge Texture, Hatching Texture, Smoke, and then Halftone Texture.

Figure 15: Comic Edge Detection: Blueprint Structure

Figure 16: Comic Edge Detection: Laplacian Edge Detection (Note: Code in Appendix A)
Figure 17: Comic Edge Detection: Apply Line and Preserve the Background

Figure 18: Comic Edge Material: Blueprint Structure

Figure 19: Comic Edge Material: Setup Noise
Figure 20: Comic Edge Material: Apply Noise

Figure 21: Hatching Material: Blueprint Structure
Figure 22: Hatching Material: Shadow Mask
Figure 23: Hatching Material: Hatching Type Selection
Figure 24: Hatching Material: Apply to Base Color

Figure 25: Hatching Material: Preserve Background
Figure 26: Smoke Material: Blueprint

Figure 27: Halftone Material: Blueprint Structure
Figure 28: Halftone Material: Rotate Texture Coordinates

Figure 29: Halftone Material: Create Repeating Mask of Dots
Use Metallic UV Mask to remove tones from object

Figure 30: Halftone Material: Object Mask

Varies the Color by the height to make it brighter green in sky

Figure 31: Halftone Material: Color
Figure 32: Halftone Material: Original Image

Figure 33: Halftone Material: Apply Halftone
Bibliography


