

A BARYCENTRIC SHADING PROCESS TO CREATE TIME-LAPSE DYNAMIC PAINTINGS
IN THE CONTEMPORARY FAUVIST-EXPRESSIONIST STYLE WITH REFLECTIONS

A Thesis

by

JACK HARRISON CLIFFORD

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, Ergun Akleman
Committee Members, Felice House
Rodney C. Hill
Head of Department, Tim McLaughlin

December 2019

Major Subject: Visualization

Copyright 2019 Jack Harrison Clifford

ABSTRACT

In this work, we developed a Barycentric shading process for creating dynamic paintings in the contemporary Fauvist-Expressionist style. Both Fauvist and Expressionist movements are known for conveying emotion through intense, bold, and clashing colors along with distorted forms and perspectives. The exaggerated colors and shapes of this style make them difficult to emulate using processes based on common modeling and rendering approaches. However, Barycentric shading provides a powerful alternative to emulate such non-photorealistic artworks with a simple procedure.

To demonstrate the efficiency of our Barycentric shading process, we re-created two paintings of Anne Garney as time-lapse animated artworks. Garney is one of the leading contemporary Fauvist-Expressionist painters. Her work includes strong, vivid colors and bright outlines typical in Fauvist paintings as well as painterly reflections. These aspects of her paintings are critical in demonstrating the power of our rendering process. Our methods for re-creating Garney's artwork as dynamic scenes emulates her style, while providing animated illumination, shadows, and reflections. Additionally, we added new elements to her scene that are still consistent with the original style.

DEDICATION

To my family and friends who supported and guided me throughout this endeavor.

ACKNOWLEDGMENTS

I would like to thank my committee chair Ergun Akleman for his guidance and support throughout my research in preparation for this thesis. I also want to thank my other committee members Felice House and Rodney C. Hill for the encouragement and feedback they provided as well as the artist Anne Garney, who graciously let me use some of her artwork in order to conduct this research.

Next, I would like to express my gratitude towards the Texas A&M Department of Visualization for allowing to learn, adapt, and utilize a wide assortment of skills and knowledge throughout my undergraduate and graduate careers. Staci Dunn has been an integral part in assisting me with form submissions and guiding me on the path to graduation, and for all of that I thank her immensely.

Finally, I want to thank my Mom, Dad, family, roommates, and close friends for being an ever-flowing fountain of support. I could not have done this without you all.

CONTRIBUTORS AND FUNDING SOURCES

This work was supervised and supported by my thesis committee chair Ergun Akleman of the Department of Visualization. My other committee members Felice House of the Department of Visualization and Rodney C. Hill of the Department of Architecture were also valuable supporters. Staci Dunn assisted me in the submission of any forms or paperwork for my thesis research. Lastly, many students in the Department of Visualization gave me emotional support and encouragement throughout this endeavor. There were no outside funding sources for this thesis research.

TABLE OF CONTENTS

	Page
ABSTRACT	ii
DEDICATION	iii
ACKNOWLEDGMENTS	iv
CONTRIBUTORS AND FUNDING SOURCES	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	viii
1. INTRODUCTION AND MOTIVATION	1
1.1 Motivation	2
1.2 Non-Photorealistic Rendering	2
1.3 Fauvist and Expressionist Paintings	3
2. BACKGROUND AND LITERATURE REVIEW	5
2.1 Non-Photorealistic Rendering	5
2.2 Non-Photo-Realism in Texas A&M Visualization	8
2.2.1 Shape Matching and Anamorphic Bas-Reliefs	9
2.2.2 Color Matching and Barycentric Shading	10
3. PROCESS	12
3.1 Creation of Four Control Textures	14
3.2 Creation of Proxy Geometry	15
3.3 Projection of Control Textures	16
3.4 Rendering Four Control Animations	16
3.5 Rendering S(t) Images	17
3.6 Compositing	17
4. IMPLEMENTATION	20
4.1 Creation of Four Control Textures	20
4.2 Creation of Proxy Geometry	22
4.3 Projection of Control Textures	23
4.4 Rendering Four Control Animations	23

4.5	Rendering $S(t)$ Images	25
4.6	Compositing	26
5.	CONCLUSION AND FUTURE WORK	30
5.1	Conclusion.....	30
5.2	Future Work	30
	REFERENCES	33
	APPENDIX A. TIME-LAPSE DYNAMIC PAINTING VIDEOS	39

LIST OF FIGURES

FIGURE	Page
1.1 Some examples of Anne Garney’s work	1
1.2 Chinese brushwork non-photorealistic rendering. Reprinted from [49]	3
1.3 Some more examples of Anne Garney’s work	4
2.1 Non-photorealistic volume rendering using stippling techniques. Reprinted from [30]	5
2.2 Real-time charcoal rendering using contrast enhancement operators. Reprinted from [31]	7
2.3 Barycentric shading with painterly filtered layers: a process to obtain painterly portraits by emulating subsurface scattering. Reprinted from [6]	9
2.4 Barycentric shading with control images to create dynamic landscape paintings. Reprinted from [21]	10
3.1 Four control images used to compute Barycentric shading for the re-creation of "Le Dolce Vita' Burano, Italy."	14
3.2 Four control images used to compute Barycentric shading for the re-creation of "Fusion at Playa del Carmen."	14
3.3 Proxy geometry that was used for the re-creation of "Le Dolce Vita' Burano, Italy."	16
3.4 Shadow & shading images that demonstrate the need for computing reflections.	17
3.5 Example frames of animated re-creations of Anne Garney’s "Le Dolce Vita' Burano, Italy." These images demonstrate reflections in moving water as well as dynamic changes in illumination.	18
3.6 Example frames of animated re-creations of Anne Garney’s "Fusion At Playa del Carmen." In this case, we also added new objects, the ball and sailboat, into the original painting to demonstrate that we can obtain visually consistent results.	19
4.1 An example of the "Match Color" process being performed on a building element in "Le Dolce Vita" Burano, Italy." Here, we are color matching the shadow color of the pink building to the full illumination color in preparation for the T11 image... ..	21

4.2	An example of a node network for a constant shader in Autodesk Maya that is set up for texture projection.....	23
4.3	Two images that demonstrate the need for a reasonably good orientation for water surface. Note that the reflections of the boat is unusually large in the original reflection.	24
4.4	A comparison of the shadow image we utilized and a mock-up of a shadow image that was better shading. Note that there is still no shading on any of the people in the mock-up as it requires the implementation of normal maps to be correct.	26
4.5	A comparison between before the water has been filtered and after it has been filtered to look like brushstrokes. This process can be done in most compositing software.	27
4.6	A diagram showing the simplified process and math exhibited in Equation 4.1. Here, we are assuming that the parameter $F(t)=1$ (no change to sunset color) and that time is constant (no time-lapse) to simplify the overall process. The parameter $(1-S)$ denotes the usage of the shadow map, whereas S denotes the usage of an inverse shadow map. The end product is a single, composited image.	28
4.7	Another diagram showing the simplified process and math exhibited in Equation 4.1. Here, we are still assuming that the parameter $F(t)=1$ (no change to sunset color), but we have reintroduced the element of time. The end product is a dynamic version of the painting.	29
5.1	A comparison between the original paintings by Anne Garney and our time-lapse dynamic painting versions of her work. The paintings are "Fusion at Playa del Carmen" and "'Le Dolce Vita' Burano, Italy," respectively.	32

1. INTRODUCTION AND MOTIVATION

Many artists have begun or expressed interest to create dynamic paintings as an alternative way to convey stories and emotions found in the painting medium. However, many of the processes involved in accomplishing this sort of task can be difficult or time-consuming to learn and understand for many traditional artists. Therefore, there is a need for a simplified, accessible process to create these animated versions of paintings. In this work, we demonstrate a process of creating dynamic works of art for the Fauvist and Expressionist styles of painting based on a general Barycentric shading process. To demonstrate this procedure, I have created painterly dynamic environments based on two of the works of the artist Anne Garney.



(a) "Le Dolce Vita" Burano, Italy, Oil on canvas, 30 x 30 in. Anne Garney.



(b) New York Times Square, Oil on canvas, 30 x 30 in. Anne Garney.

Figure 1.1: Some examples of Anne Garney's work

Garney's artwork is generally classified under the representational and contemporary Fauvist-Expressionist styles. While representational simply means that her artwork represents something

(therefore giving it identity), the contemporary Fauvist-Expressionist label has a few more parts. Expressionism generally refers to the practice of conveying emotions and sensations through work. These emotions can often be split into two definitions, with one denoting the feelings of the artist during the creative process, and the other meaning the feelings of the viewer while observing the completed work [18]. The other two parts of her style, Contemporary and Fauvist, designate being modern and utilizing vivid, non-naturalistic use of color, respectively.

These particular characteristics made Garney’s art a perfect candidate to evaluate and test our methodology. Through this work, we have developed a method to emulate Garney’s paintings, while demonstrating the power of Barycentric shading and certain aspects of non-photorealistic rendering. Furthermore, we were able to retain Garney’s style through the use of camera projections as well as computing dynamic lights, shadows, and reflections in each scene. The most important contribution to this particular process is the implementation of Barycentric shading as a Barycentric compositing process. This approach clearly differentiates the rendering and shading processes from prior research in this field. On the other hand, the second contribution is to include reflections into the Barycentric compositing process.

1.1 Motivation

Our goal in this work is to develop a simple, yet effective process to create animated versions of Fauvist and Expressionist paintings for further accentuating perception and immersion of the viewer. Unfortunately, most common physically based modeling and rendering approaches are not useful for emulating such non-realistic styles. Many NPR approaches require specialized solutions that can be hard for novice users. Barycentric shading provides a generic approach that can be directly applied to a wide variety of styles with simple modifications in a general modeling and rendering process [3].

1.2 Non-Photorealistic Rendering

The area of non-photorealistic rendering (NPR) in computer graphics has been well researched over the past couple of decades and a variety of systems have been developed to implement this

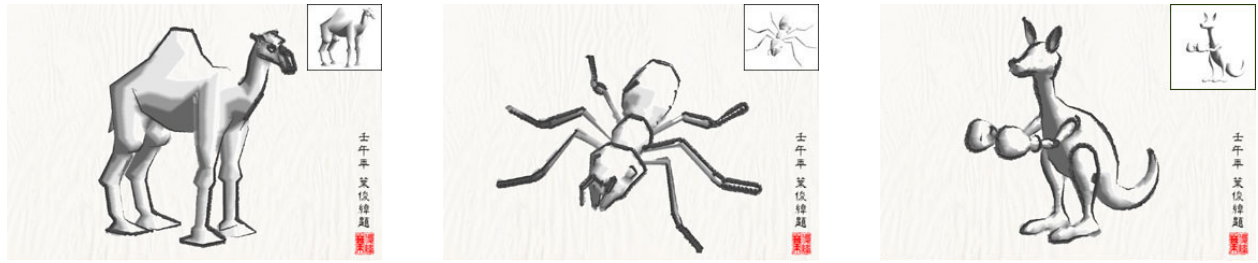


Figure 1.2: Chinese brushwork non-photorealistic rendering. Reprinted from [49]

type of rendering into both 2D and 3D software. It has been used to emulate a variety of traditional styles including Chinese brush paintings [49] and graphite pencil drawings [41]. Despite the varying results, all of the research poured into this realm of computer graphics expresses the desire to push the limits of the types of images we can create through computational methods. Moreover, these techniques have garnered interest amongst many artists and content creators as a means of telling stories and conveying information in new, unique ways.

Another key-point of information regarding NPR is that it is often compared with photo-realistic rendering methods. In general, photo-realistic rendering techniques are typically associated with computer graphics, and have proven to be useful for tasks like virtual reality simulations [36] and incorporating new elements into live-action film. NPR can be utilized for similar tasks, but is particularly well-suited for creating automated graphics that communicate and stylize narratives, ideas, and emotion [17]. In other words, it brings computerization and traditional art together, which has proven to sometimes be more complicated than generating photo-realistic imagery.

1.3 Fauvist and Expressionist Paintings

Fauvism and Expressionism were two of the earliest avant-garde art movements, centered on the expression of feeling through intense colors with distorted forms and perspectives. The Fauvist and Expressionist paintings were primarily made to convey ideas, places, or people through the manner of the artists' style, which consequently elicit a variety of reactions from the audience.



(a) Penn Valley Skyline View, Oil on canvas, 24 x 20 in. Anne Garney.



(b) Fusion at Playa del Carmen, Oil on canvas, 24 x 20 in. Anne Garney.

Figure 1.3: Some more examples of Anne Garney's work

Additionally, they tend to create distorted connections in the mind of the viewer, allowing them to perceive space and three-dimensional objects from an unusual point-of-view in a two-dimensional medium.

For this thesis, I directed my attention to Anne Garney's work, which falls under this style of artwork. Throughout her work, she typically focuses on landscape paintings of locales she has visited, although there are a couple instances of pieces that more closely resemble still life paintings. Furthermore, Garney makes use of vibrant color schemes, bright outlines, and unique brushwork. Some examples of her work include "Penn Valley Skyline" and "Fusion at Playa del Carmen." Altogether, these characteristics made her artwork a perfect candidate for testing out our methodology with her permission.

2. BACKGROUND AND LITERATURE REVIEW

2.1 Non-Photorealistic Rendering

Non-photorealistic rendering (NPR) has been a fairly well researched field of computer graphics, having a variety of different stylistic approaches researched and developed since the 1990s. As a whole, NPR consists of any computational or scientific method that deals with producing computer generated images that invoke the appearance of being created "by hand" [43]. This consequently lends itself to many possibilities for research in this area, and has led many to pursue this advancement of computer generated images. For instance, in 1999, Hertzmann surveyed an assort-

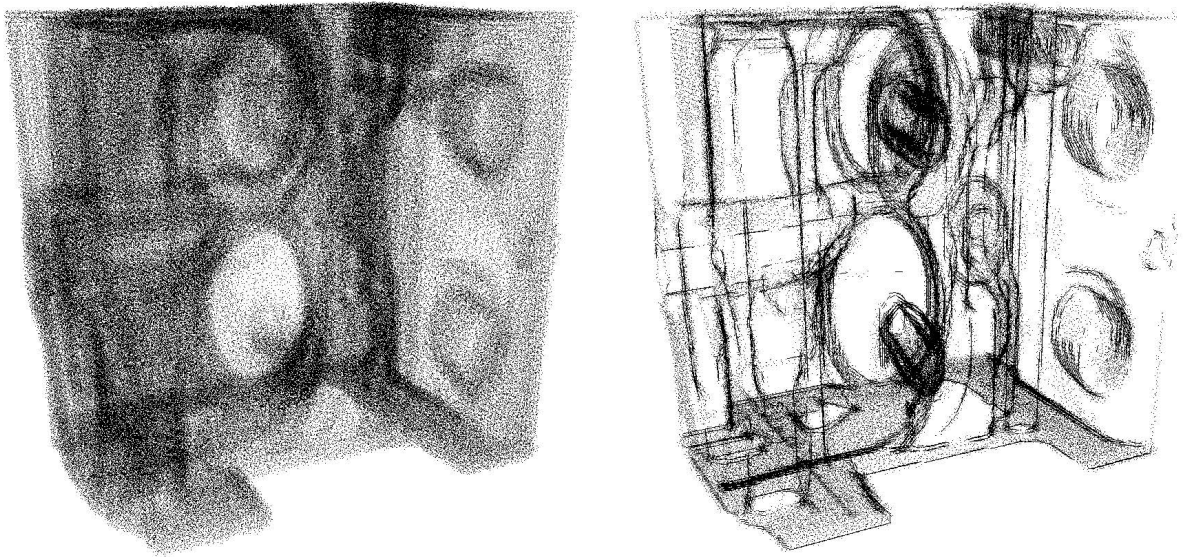


Figure 2.1: Non-photorealistic volume rendering using stippling techniques. Reprinted from [30]

ment of basic tools used for non-photorealistic rendering of 3D scenes, looking into algorithms for detecting outlines of object shapes that operate in a 2D image space and a 3D world space [20]. He showed results that had object shapes being rendered in a manner close to that of pencil sketches, with pencil drawing techniques like hatching and shading being utilized. In a similar vein, many others have researched analogous NPR methods, such as a lighting system that lets one generate

images that bear resemblance to technical illustrations found in textbooks [16], 3D tree models that render as pen and ink drawings [13], and emulation of the stippling technique of traditional drawing [30].

In addition to emulating specific styles of traditional art for NPR, some have been more interested in establishing techniques that allow for interactive generation of brush strokes to create painterly images. Sometimes referred to as Stroke based Rendering (SBR), this area of NPR makes use of algorithms to place brush strokes and the compositing of rendering marks (i.e. lines, brush strokes, etc.) on digital canvases [44]. A specific exploration of this was done by Hertzmann, who introduced a methodology for creating images with a painting-like appearance from photographs through utilization of spline brush strokes. These generated strokes could be of varying sizes and match colors found in the source image to create painterly renderings in many different styles [19].

Similar to Hertzmann's work, there are some painterly rendering methods that primarily focus on emulating specific physical properties of the painting process. For instance, in 1997, watercolor painting in NPR was examined when Curtis et al created a watercolor system that was based around sets of translucent glazes (a data structure they developed that holds washes of different pigments of various quantities), which are made via fluid simulation [10]. Similarly, Yeh and Ouhyoung created a two-part rendering process that generates both the outer strokes as well as shading within the stroke lines in an effort to emulate traditional Chinese paintings of various animals [49]. They also implemented different brush types into their system to allow users more ease and freedom to produce their own renderings [49].

The process of computationally converting image and video files to have a painterly look represents another major area of painterly NPR rendering. Litwinowicz described a method of transforming video sequences into animations that look similar to Impressionist paintings by manipulating temporal coherence in video files to design a filter that can automatically reproduce an Impressionist effect onto the images while retaining the sense of the original video [28]. Wang et al introduced another NPR video technique by producing brush strokes based on this semantic content [27]. Some more recent work for this sort of painterly rendering comes from Atelier des

Lumières, a group who developed large scale video projections of many of the famed artist Vincent van Gogh's best works. These projections consist of animated, painterly elements like water and stars in the night sky, and ultimately create a three-dimensional way of viewing the artwork [12].

Rendering non-photorealistic images or scenes in real time is another important problem in NPR. Markosian introduced a real time NPR method by focusing on speed instead of detail and accuracy in the simulation of various styles of simple line drawing [32]. Majumder and Gopi developed a real-time charcoal renderer using contrast enhancement operators (functions that modify image contrast) on a 3D model's colors and textures to generate images look as if they were traditional charcoal drawings [31]. These examples show how diverse and complex the techniques of the painterly rendering field in NPR can be.



Figure 2.2: Real-time charcoal rendering using contrast enhancement operators. Reprinted from [31]

Despite these advancements, NPR is not particularly suited for all tasks in computer graphics. Photo-realistic rendering has proven to be more useful when producing computer generated images that involve simulations or technical blueprints and descriptions [17]. On the other hand, NPR, which was originally developed in 1990s in response to photo-realistic rendering methods, is better suited for roles concerning the communication and stylization of an idea driven by human perception, just as many traditional works of art do [17]. Many have sought to push and develop these concepts of NPR further, with many of these creators recently looking towards painterly rendering techniques as a way of accomplishing this.

2.2 Non-Photo-Realism in Texas A&M Visualization

There has also been strong interest in non-photorealistic rendering in the Texas A&M Visualization program. For example, Meadows and Smith developed multi-camera rendering methods to obtain cubist paintings from 3D scenes [34, 33, 39, 38]. Similarly, Krawczyk developed a method to obtain Futuristic paintings and Sculptures [25] and Chan developed two methods to create Chinese painting [8, 7].

Additionally, there has also been work to design new visual aesthetics. One instance of this is found in the research of Song, who extended classical cartoon speed lines into 3D [40]. Other explorations of this area are displayed by Akleman, who introduced the concept of implicit painting [1, 2] and Lee, who introduced a concept of volume painting through extended line integral convolution to volumes [26]. Wang and Akleman later developed a method by turning photographs into a new type of abstract Expressionist painting [45]. Finally, Davis developed a system to create his own virtual 3D paints [11].

In recent years, there has been growing interest to emulate the style of specific artists instead of an entire painting style. This sort of research requires stronger study in matching color and shape. For instance, Murphy developed a non-photorealistic approach for matching shapes and colors of the artwork of famed Disney background painter Eyvind Earle. To achieve Earle's style in the 3D computer generated images, she employed two different modeling and shading methods. The first employed the generation of detailed geometry as a shader to emulate stippling and small lines, while the second utilized techniques from toon shaders to avoid having objects in the 3D environment react to light realistically [35]. The final renderings match Earle's style very well, creating many opportunities in animation and digital artwork for the system to be utilized in. However, this approach requires a significant amount of work to achieve the final results.

Cheng later explored the possibilities of emulating painterly reflections with a watercolor shader applied to 3D objects. Through her research, she found that she could faithfully replicate non-reflected areas in a watercolor style, but reflected regions would not have the same exact colors found in the reflections of the original work [9]. This issue demonstrates the overall difficulty of

matching colors when emulating another artist’s style.

Finally, one last area where emulating the style of a particular artist is also required is for the compositing of paintings and illustrations. Stanley and Kiker both demonstrated that it is possible to obtain convincing composites when utilizing illustrations [42] and paintings with unusual perspectives [24]. Stanley specifically investigated techniques to composite digital characters into traditional artwork by matching lighting, perspective, and style [42]. On the other hand, Kiker looked into procedures for incorporating digital graphics, from primitive objects to complex character models, into traditional artwork. His final results were able to produce composites that were able to retain the likeness of the original artwork while producing convincing visuals because he based his additions on aspects found in the original art [24].



Figure 2.3: Barycentric shading with painterly filtered layers: a process to obtain painterly portraits by emulating subsurface scattering. Reprinted from [6]

2.2.1 Shape Matching and Anamorphic Bas-Reliefs

One of the key problems when emulating distinct styles is creating geometry that can match non-realistic shapes in paintings and illustrations. Wang et al developed a way to represent non-realistic shapes using Mock-3D shapes, which are really anamorphic bas-reliefs [50, 46]. Furthermore, Xiong developed a web-based tool that allows artists to create 3D-looking, stylized images based on input 2D artwork using Mock-3D shapes. With this tool, one can control a variety of parameters of the resulting image including reflection, specular highlights, and refractions [47].

Lastly, Perumal developed a way to obtain subsurface scattering with a type of anamorphic bas-relief called depth maps [37]. In this work, I also use the concept of anamorphic bas-reliefs to replicate the shapes in the paintings.

2.2.2 Color Matching and Barycentric Shading

Barycentric shading was initially introduced to solve color matching problems using Barycentric equations that interpolate or approximate control images [3]. Barycentric shaders can be organized into a structure that allows one to ensure a stylistic decisions are consistent within any rendering or lighting system. In other words, shader operators are restricted to be of Barycentric form for all shading points, guaranteeing that results will remain within the framework specified by the control images. These principles of Barycentric shading can be applied to non-photorealistic rendering to achieve successful results.



Figure 2.4: Barycentric shading with control images to create dynamic landscape paintings. Reprinted from [21]

Over the past few years, Barycentric shaders have been used in emulating a wide variety of styles. Siran Liu utilized Barycentric shading to create painterly renders with reflections that resemble the style of renowned contemporary Chinese artist Yang Ming-Yi [29]. Yan was able to develop painterly ocean surface shading based on the artist Ivan Aivazovsky through the implementation of a Barycentric shading process and rendering architecture [48]. Kabeerdoss adapted Barycentric shaders to emulate a painting by Rachel Cunningham [23] through the use of real-time

ray tracing with reflections. Yuxiao Du also used similar Barycentric methods to create charcoal and crosshatching drawings with reflections [14, 15].

In 2017, Castaneda studied methods of extracting color data from photographs in order to develop Barycentric shaders that generate differently styled portrait paintings [6], emulating the appearance subsurface scattering. He ultimately created an entire "Expressive Art Exhibit" of animated portrait paintings [5] (see Figure 2.3). Finally, in 2018, Justice created dynamic time-lapse animations based on some of the works of Edgar Payne, using Barycentric shading as the core of his procedure [22, 21] (see Figure 2.4). In this thesis, I will demonstrate our methods to extend time-lapse animations with reflections in the Fauvist-Expressionist style.

3. PROCESS

Many artists have begun or expressed interest to create dynamic paintings as an alternative way to convey stories and emotions found in the painting medium. However, many of the processes involved in accomplishing this sort of task can be difficult or time-consuming to learn and understand for many traditional artists. Therefore, there is a need for a simplified, accessible process to create these animated versions of paintings. In this work, we demonstrate a process of creating dynamic works of art for the Fauvist and Expressionist styles of painting based on a general Barycentric shading process. To demonstrate this procedure, we have created painterly dynamic environments based on two of the works of the artist Anne Garney.

Before going in depth about our process, I will describe our Barycentric shading methods. To begin, within Barycentric shading, we can restrict any shader operators to be of the Barycentric form for every shading point. This consequently ensures that final results produced through this method will remain within the framework specified by the control images. In order to make time-lapse dynamic paintings that do not include any reflections or refractions, the utilization of a Bilinear Barycentric shader with two shader parameters that interpolate four control images is an acceptable procedure [22]. We will also make use of a Bilinear Barycentric shader for obtaining dynamic paintings in this work, but our four control images will be animated and rendered for each frame (essentially, they will become control image sequences). This will be essential for the inclusion of moving, reflective elements like water.

Next, I will be defining a variety of mathematical terms and variables that are crucial to understanding the Bilinear Barycentric shading procedure. First of all, we will let the term $I(t)$ stand for an image (or frame) of dynamic animation and $I(t, u, v)$ denote the color of a single pixel in $I(t)$. These u and v terms equate to the position of pixels, whereas t represents time, but we generally ignore these three variables to simplify the final equation. Additionally, since we are using a Bilinear Barycentric shader, we will be implementing two shading parameters: $S(t)$ and $F(t)$. The first one, $S(t)$, corresponds to the percentage of sunlight that can reach a particular shading point.

In other words, when $S = 1$, the shading point is being fully illuminated by the sun and when $S = 0$, the shading point is receiving no illumination from the sun and is thus in-shadow. The second shading parameter $F(t)$ is the animation frames, which are normalized as a number from 1 to 0. In our case, $F = 1$ (a white image) corresponds to noon time and $F = 0$ (a black image) corresponds to sunset, the beginning and ending of our animations, respectively. Altogether, these shading parameters become images that are denoted as $I(t) = P(S(t), F(t))$, in which P is the Barycentric shader.

Based on these definitions, we can introduce a Bilinear Barycentric shader, which is given by the following formula:

$$\begin{aligned}
 P(S(t), F(t)) = & T_{00}(t)(1 - S(t))(1 - F(t)) + T_{11}(t) S(t) F(t) + \\
 & T_{01}(t)(1 - S(t))F(t) + T_{10}(t)S(t)(1 - F(t)).
 \end{aligned} \tag{3.1}$$

This bilinear equation $P(S(t), F(t))$ interpolates four control image sequences, $T_{00}(t)$, $T_{10}(t)$, $T_{01}(t)$, and $T_{11}(t)$, as its end-points. These control images would be completely static and not change with time if a re-created paintings did not have any dynamic, moving aspects [22]. However, because we will have animated objects and moving water that will reflect the environment, we will need to render these control images for any given time within our time-lapse using 3D proxy models. The shading parameters $S(t)$ and $F(t)$ will also need to be calculated in this fashion. Since these elements are rendered out separately, our bilinear formula 3.1 becomes a compositing equation that can be computed in the post-processing stage.

Additionally, there some extra advantages to utilizing this sort of system in our process. For one, some global illumination effects, such as sky illumination coming from scattered sunlight or ambient occlusion, will already be embedded within the control images. Furthermore, the changes in sky color are also embedded within the $F(t)$ parameter. Thus, we only need the sun's position along with proxy geometry to compute the image sequences that will be applied in $P(S(t), F(t))$.

Overall, this Barycentric shading framework to create dynamic paintings with moving reflec-



(a) $T_{11}(1)$: Full sun illumination at noon. (b) $T_{01}(1)$: No direct illumination at noon. (c) $T_{10}(0)$: Full sun illumination at sunset. (d) $T_{00}(0)$: No direct illumination at sunset.

Figure 3.1: Four control images used to compute Barycentric shading for the re-creation of "Le Dolce Vita' Burano, Italy."



(a) $T_{11}(1)$: Full sun illumination at noon. (b) $T_{01}(1)$: No direct illumination at noon. (c) $T_{10}(0)$: Full sun illumination at sunset. (d) $T_{00}(0)$: No direct illumination at sunset.

Figure 3.2: Four control images used to compute Barycentric shading for the re-creation of "Fusion at Playa del Carmen."

tive surfaces is not the same as the process described in [22], though we still make use of a Bilinear Barycentric shader. Therefore, the process of creating time-lapse animated paintings consists of six stages: (1) Creation of Four Control Textures; (2) Creation of Proxy Geometry; (3) Projection of Control Textures to Proxy Geometry; (4) Rendering Four Control Animations; (5) Rendering Shadow Images; and (6) Compositing. Any 2D artist with minor training in 3D modeling and animation can produce such time-lapse animated paintings using this process.

3.1 Creation of Four Control Textures

To begin, the first stage of our process involves the creation of four control textures. These control textures are static images that designate diffuse regions of the four control animations and

do not change with time. We will ultimately render four control animations using these static control textures. As they correspond to the parts of the control images that do not change over time, they can be created as in [22] from available paintings using color transfer techniques [4], which are common staples in most modern image-editing software. In our particular case, we first create two control paintings at noon, as shown in Figures 3.1a and 3.1b, by making one completely illuminated and another completely in shadow using color transfer from one part of the image to another part of the image. This sort of procedure can be executed in almost any image-editing program.

However, we also need to create two additional control textures during sunset, but we did not have a point of reference for any sunset colors found in any of the artist’s original work. Thus, to create sunset colors, we improvised by adopting sky colors from sunset reference images. We then estimated how much hue and value must be shifted to create sunset images and changed the color of each texture based on our estimation. Finally, we obtained two more control images as shown in Figures 3.1c and 3.1d. This provides us with four static, “un-illuminated” control textures from which we can obtain control image sequences.

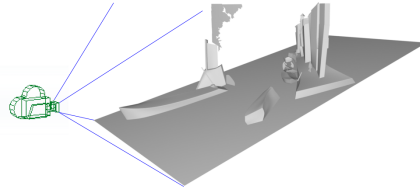
3.2 Creation of Proxy Geometry

For our next step, we wanted to simplify the overall process of generating proxy models. However, we still need the boundaries of our proxy geometry to match the object boundaries found in the original paintings, as any shape discontinuities can correspond to discontinuities in shadows. Thus, we model the proxy meshes within a single orthographic window to conserve simplicity, while allowing us to be exact with the vertices and edges of the geometry. Although in this example we build a polygonal model, the proxy geometry can even be a height field as shown in Figure 3.3a.

Similar to the creation of four control textures, this does not require any specific software. An artist can quickly create such 3D proxy geometry by following this general procedure. The only requirement is that depth must be a monotonically increasing function of the original depth [50]. In addition, one can also create and add new props or objects to the scene by making some extra



(a) Depth Map.



(b) Perspective View.

Figure 3.3: Proxy geometry that was used for the re-creation of "Le Dolce Vita' Burano, Italy."

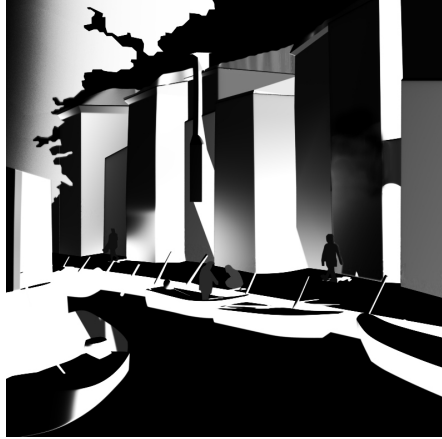
proxy models and control textures if they desire to.

3.3 Projection of Control Textures

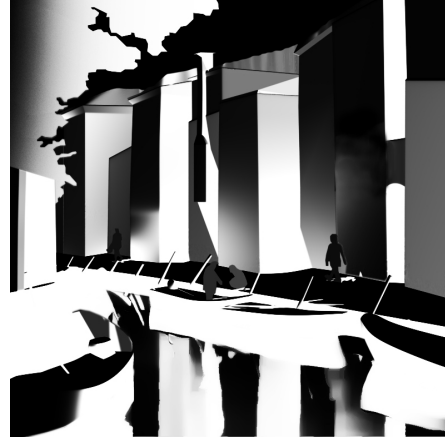
For the third stage, texture mapping the four control textures to proxy geometry can be accomplished with a simple camera projection. This is because the proxy geometry was already created to match the boundaries of the original work as shown in Figure 3.3b. Another important note is that in our time-lapse dynamic paintings, we did not animate the camera position. Yet, minor camera motion is still possible due to how the projection separates the 3D objects.

3.4 Rendering Four Control Animations

In our next step, we animated the light source, the water, and some of the objects in the scene. While adding motion to the light and props only requires some simple transformations, the water motion is a bit more complex. For our water, we add only subtle motion to its surface to obtain a similar visual appearance to the original painting, while also making sure that the reflections are physically correct. Put another way, for each of the four control textures, we need to find the corresponding reflected shading points and create four image sequences that include reflections. These four image sequences are then utilized in the Bilinear Barycentric formula given in Equation 3.1. This is also a simple operation that can be computed in any 3D or rendering software.



(a) Shadow image with no reflection.



(b) Shadow image with reflection.

Figure 3.4: Shadow & shading images that demonstrate the need for computing reflections.

3.5 Rendering $S(t)$ Images

Next, we render out our $S(t)$, or shadow and shading, images. For these sequences, we only need a light source position that is adequate enough to compute both shading and shadows for the objects throughout the scene. Yet, we also need to compute reflections of shading and shadows. As shown Figure 3.4, without reflections of shadow and shading, our bilinear equation will not provide shading and shadows in any reflected regions. I will be going over how to solve this particular issue in the next chapter. Nevertheless, this part of the process is also straightforward to implement by using any rendering software.

3.6 Compositing

Finally, once all animations have been rendered, creating the final time-lapse dynamic paintings is simply a compositing process. This can be done in any image manipulation software that allows for the compositing of animations. Figure 3.6 shows frames from another animated painting that was created through the same process. As demonstrated in Figures 3.5 and 3.6, this procedure provides creative control to obtain stylistically accurate time-lapse animated paintings.



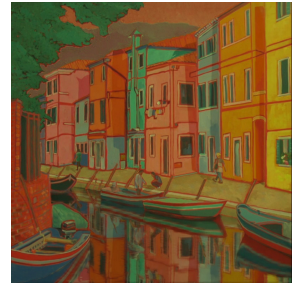
(a) Noon.



(b) Afternoon.



(c) Evening.



(d) Sunset.

Figure 3.5: Example frames of animated re-creations of Anne Garney's "Le Dolce Vita' Burano, Italy." These images demonstrate reflections in moving water as well as dynamic changes in illumination.



(a) Noon.



(b) Afternoon.



(c) Evening.



(d) Sunset.

Figure 3.6: Example frames of animated re-creations of Anne Garney's "Fusion At Playa del Carmen." In this case, we also added new objects, the ball and sailboat, into the original painting to demonstrate that we can obtain visually consistent results.

4. IMPLEMENTATION

Through our research, we developed a process for the efficient creation of dynamic paintings based on the work of Anne Garney. I will now explain each step of our process in greater detail here. Once again, our six stages are as follows: (1) Creation of Four Control Textures; (2) Creation of Proxy Geometry; (3) Projection of Control Textures to Proxy Geometry; (4) Rendering Four Control Animations; (5) Rendering Shadow Images; and (6) Compositing. Any 2D artist with minor training in 3D modeling and animation can produce such time-lapse animated paintings using this process.

4.1 Creation of Four Control Textures

Initially, we chose to base our dynamic paintings off of Anne Garney's two works entitled "Fusion At Playa del Carmen" and "'Le Dolce Vita' Burano, Italy." We felt that these works in particular would work the best for testing at every aspect of our methodology. Then we began the first step of our process, which involved creating a full illumination at noon image (T_{11}) and a full shadow at noon image (T_{01}) within Adobe Photoshop. Additionally, a full illumination at sunset image (T_{10}) and a full shadow at sunset image (T_{00}) were created since we wanted our dynamic paintings to display a time-lapse into sunset. Each of these control images would later be implemented as a Barycentric shader.

These images were created quickly by putting each object of the original painting into their own layers (separating illuminated and in-shadow components). We accomplished this by using the "Polygonal Lasso" tool (or in rare cases the "Magic Wand" tool) to select each individual part of the painting scene. After organizing the layers by name, we would then utilize the "Match Color" feature to make illuminated elements the same color as their in shadow variant and vice-versa as seen in Figure 4.1. For anything that did not have an illuminated or shadow color present in the original work, we would simply adjust the brightness, saturation, or hue until it looked correct. Finally, all of the layers of fully illuminated objects would be merged to create the T_{11} image,

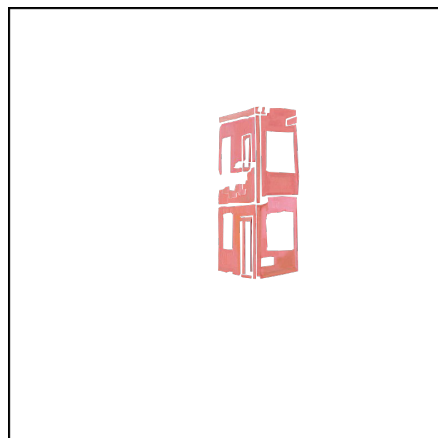
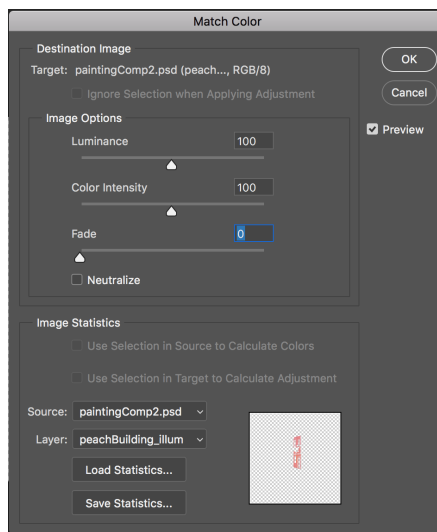
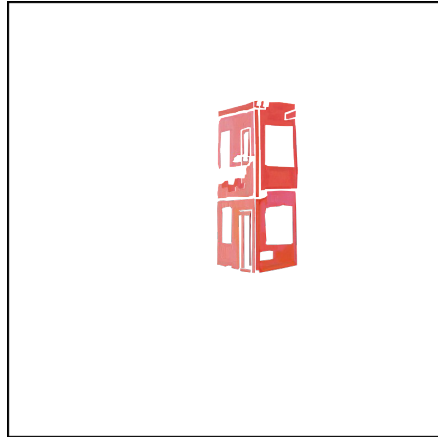


Figure 4.1: An example of the "Match Color" process being performed on a building element in "Le Dolce Vita" Burano, Italy." Here, we are color matching the shadow color of the pink building to the full illumination color in preparation for the T11 image.

while all of the layers of the in-shadow objects would be merged to make the T_{0l} image.

However, additional control images needed to be made for the sunset time-lapse. To create these sunset control images, we did not simply shift the hue of our noontime control images, but rather we shifted the hue of texture images of each individual object before merging all of them together. Despite this being a bit more time-consuming, it ultimately created a better looking final result as opposed to only changing the hue of the full illumination image.

4.2 Creation of Proxy Geometry

The next stage involved creating proxy geometry based on the scene of the original painting in order to generate dynamic light, shadows and reflections. Using Autodesk Maya, proxy models were efficiently created by modeling solely in the front orthographic viewport. This was done by first creating an image plane displaying an image of the original work and then creating primitive objects (we mostly used cubes due to their versatility) that roughly matched the size of each object. After adding more edges and vertices to these primitives, we would then move and drag their vertices so that the 3D proxy model would completely cover the area the object takes up in the painting. For any objects with hard edges, we made sure that its 3D proxy model counterpart would have hard edges in correct areas accordingly. Although these models did not look correct from other views, they still created shadows and reflections that were accurate to the original painting so long as the scene was rendered from the front orthographic view in which the proxy models were created.

In addition to creating this proxy geometry, using Maya also gave us the ability to insert objects not found in the original painting. Introducing a new object into the scene proved to be a fairly straightforward process since all it required was the addition of some simple proxy models that fit the general shape of new prop. For instance, in the case of our re-creation of "Fusion At Playa del Carmen," we added a beach ball in the foreground and a sailboat in the background, both of which were quickly modeled due their simple shapes. To create control image textures for these objects (as they were not in the original painting), we sampled brush strokes and colors from the original work and used these assets to create 2D versions of these objects that perfectly matched

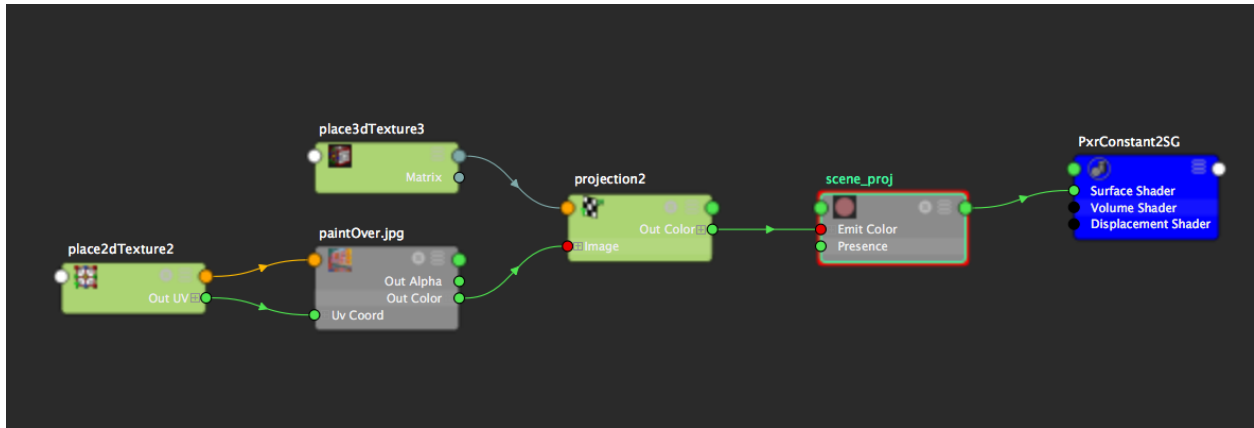


Figure 4.2: An example of a node network for a constant shader in Autodesk Maya that is set up for texture projection.

Garney’s style. As with our original control images, we created these textures in full illumination (T_{11}), complete shadow (T_{01}), and during the sunset (T_{10} and T_{00}).

4.3 Projection of Control Textures

When implementing textures for the objects in Maya, projection mapping the 2D control image onto a constant shader was the most efficient process (additional details about our projection mapping method can be found in [22]). For any dynamic objects or props in the scene, though, we found that it was necessary to give them their own individual texture projections that can move along with any animations they might have. As a side note, if the object was going to have any 3D motion (movement beyond shifting left and right or up and down), UV unwrapping and painted textures for these individual props would have to be utilized. We made our dynamic objects have fairly simple animations to conserve overall efficiency because of this factor.

4.4 Rendering Four Control Animations

Next, we began to implement animated lighting and reflections as well as general animation of objects into the scene. Lighting proved to be straightforward as each painting we re-created only had one light source in the original work: the sun. Consequently, we utilized a distant light as our light source, and matched its direction to the original painting. After this point, we could properly

animate the light source to emulate a shift in time.

Including reflections into the scene re-creation proved to be a bit more complicated, however. We implemented reflections through the use of a Renderman mirror shader on a flat plane (reflections were only utilized for reflective water surfaces), taking up the space the water comprises in the original painting. Getting the angle of this plane in 3D space correct proved to be an issue, as having the plane at too high of angle made reflections look incorrect in comparison to the original work (a comparison of the reflections can be seen in Figure 4.3). However, we eventually found that the flatter this reflective plane is, the more realistic its reflections will look (we consequently needed to adjust a good portion of the proxy geometry because of this).

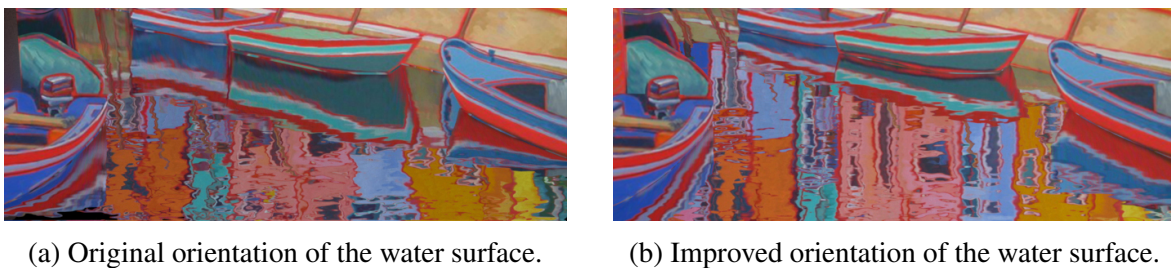


Figure 4.3: Two images that demonstrate the need for a reasonably good orientation for water surface. Note that the reflections of the boat is unusually large in the original reflection.

Additionally, we added an animated normal map to this mirror plane in order to simulate water ripples and waves. Getting the ripple frequency and amplitude correct proved to be somewhat difficult, so quite a few adjustments needed to be made to our normal map sequence. We finally settled with using a subtle, enlarged version of the original normal map with a slower ripple speed (which we decreased by time warping the animation to be slower overall) as it proved to yield the best results.

After finishing the lighting and reflections, we tested out how everything looked on a single frame. Once everything looked close to the original work, we moved on to the following stage of implementing animation into re-creation to make it dynamic. As we explained before, the

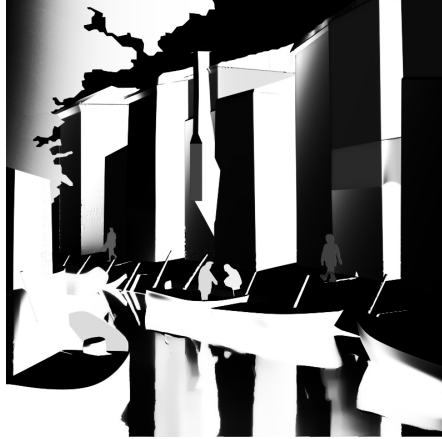
light source and water were given motion through simple animation and normal map sequences respectively. After this, however, we added animation to some of the various objects in the scene. In our re-creation of "'Le Dolce Vita' Burano, Italy," for instance, we added motion to the boats on the water. This proved to be fairly simple since they only needed to bob up and down with some slight rotation to achieve a fairly realistic movement.

One important extra note was that in order for these boat animations to look correct, we needed to do a slight paint over of certain areas in the control images. This was because we needed to create the illusion that there were objects and space behind the moving objects (which would not have been visible in the original work). This process was quickly accomplished by utilizing the "Clone Stamp" tool within Adobe Photoshop to copy parts of the painting close to the animated objects and paint over those areas. Occasionally, we also used the standard "Brush" tool to touch up certain areas of the paint over.

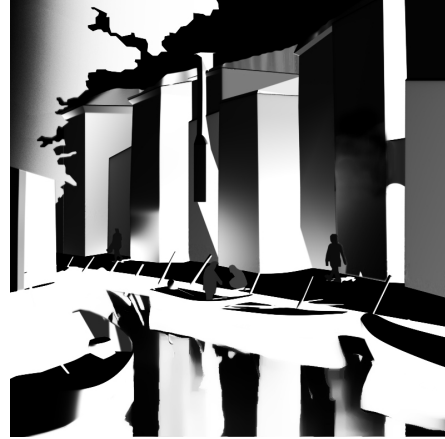
4.5 Rendering S(t) Images

The fifth stage of our system was concerned with rendering out the proper image sequences and compositing all of these renders together. Separate image sequences for each of our control images were rendered using the Renderman renderer, both of which executed quickly due to only utilizing constant shaders in both scenes. Shadow maps were rendered using Renderman's "Direct Lighting" integrator, allowing us to include proper shading and cast shadows when compositing (Note: the shadow pass for object reflections needs to be rendered separately using the "Path Tracer" integrator in order for the reflections to show up in the pass). Finally, we rendered out a mask for the water in the scene to make some steps of our compositing process easier.

One extra matter I would like to note is that the shadow maps that we ultimately produced with our methods do not yield the best results for object shading (cast shadows come out perfectly fine, though) as seen in Figure 4.4. This may have something to do with the rendering integrator or the light source we utilized, but did not greatly affect our final composites. In other words, these issues were not noticeable in the final dynamic paintings. A possible solution for implementing better shading in future endeavors could involve the use of normal maps based on the original



(a) A shadow map that is more akin to what we ended up using.



(b) A mock-up shadow map that shows better shading in general.

Figure 4.4: A comparison of the shadow image we utilized and a mock-up of a shadow image that was better shading. Note that there is still no shading on any of the people in the mock-up as it requires the implementation of normal maps to be correct.

artist's brushwork. An example of this is that it could be utilized on the people found in the scene to give them slightly more accurate shading from the sunlight. All in all, I believe this warrants some study for future endeavors in this field.

4.6 Compositing

Finally, in order to composite all of these image sequences together we used the equation I discussed earlier:

$$\begin{aligned}
 P(S(t), F(t)) = & T_{00}(t)(1 - S(t))(1 - F(t)) + T_{11}(t) S(t) F(t) + \\
 & T_{01}(t)(1 - S(t))F(t) + T_{10}(t)S(t)(1 - F(t)).
 \end{aligned}
 \tag{4.1}$$

Within this equation, $(1-S(t))$ denotes a shadow map, $S(t)$ is a black image (or no shadow map), $(1-F(t))$ denotes that the image is at sunset, and $F(t)$ means the image is at noon. $P(S(t),F(t))$ is essentially the final composited image sequence (a more detailed explanation of all of these parameters can be found in Chapter 3: Process). We then applied this equation to blending modes within Adobe AfterEffects, using the "Multiply" blend mode for multiplication and the "Linear

Dodge" blending mode for addition (a couple of diagrams depicting this in a simplified, literal fashion are displayed in Figures 4.6 and 4.7). We chose this method as it is a simple, visual way of employing this math.

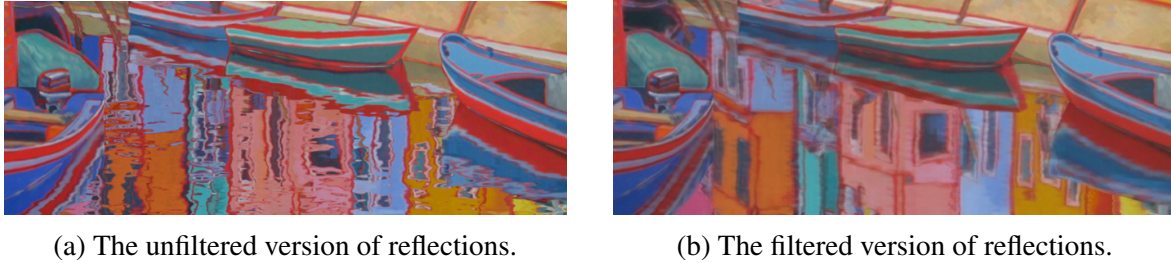


Figure 4.5: A comparison between before the water has been filtered and after it has been filtered to look like brushstrokes. This process can be done in most compositing software.

Additionally, AfterEffects allowed us to easily adjust brightness and saturation of certain parts of the sequence and implement a painterly brush filter to the water (this is one area where the water mask was utilized) so that it looked much closer to the original painting. As shown in Figure (insert figure here), the original renders were not producing the painterly reflections we desired, so applying this sort of filter was necessary to be faithful to the original work. Some other supplemental tweaks like the "Gaussian Blur" effect were also made to the shadow pass in order to make shadows slightly softer to ensure they looked even closer to the original painting. After these final adjustments, we rendered out of the video editor, which resulted in the final dynamic versions of each painting.

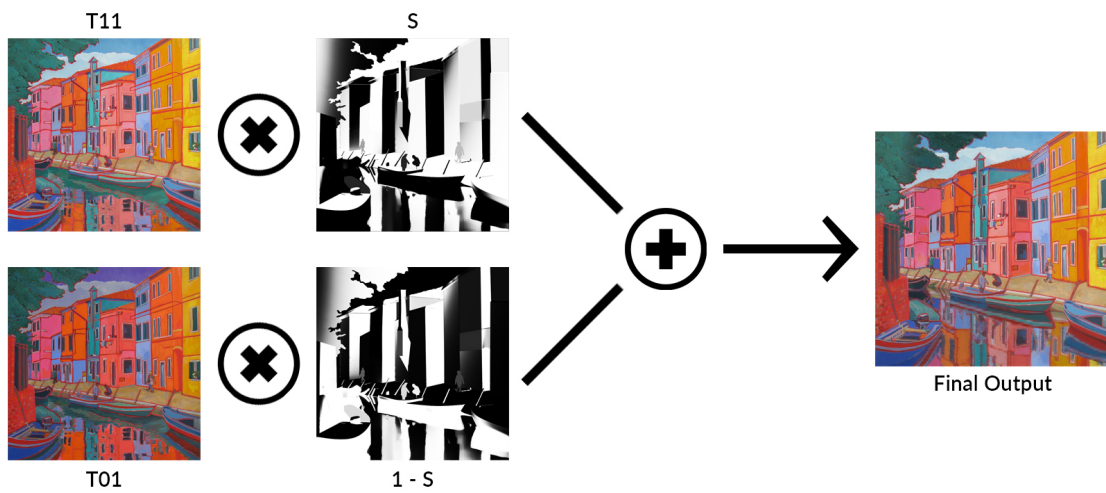


Figure 4.6: A diagram showing the simplified process and math exhibited in Equation 4.1. Here, we are assuming that the parameter $F(t)=1$ (no change to sunset color) and that time is constant (no time-lapse) to simplify the overall process. The parameter $(1-S)$ denotes the usage of the shadow map, whereas S denotes the usage of an inverse shadow map. The end product is a single, composited image.

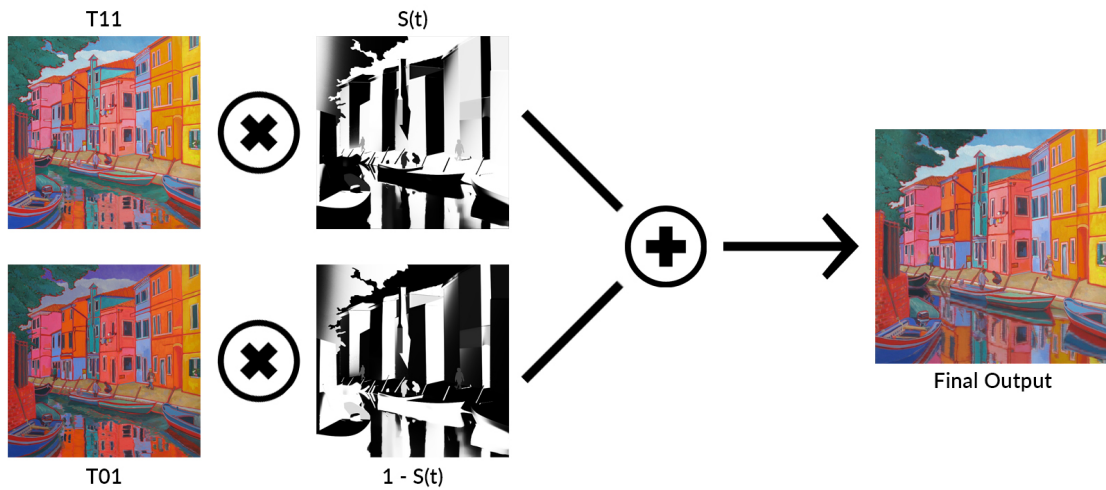


Figure 4.7: Another diagram showing the simplified process and math exhibited in Equation 4.1. Here, we are still assuming that the parameter $F(t)=1$ (no change to sunset color), but we have reintroduced the element of time. The end product is a dynamic version of the painting.

5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

We have displayed our process for efficiently re-creating and rendering time-lapse dynamic scenes with reflections based on the artwork of Anne Garney. With her unique style's visually pleasing colors, subtle shadows, and reflections, we believe her paintings are perfect candidates for testing the efficiency of rendering and shading processes to emulate a wide variety of styles. The process includes using Bilinear Barycentric shading in post-processing along with simple proxy geometry and lighting setups. Additionally, the procedure is able to incorporate adding new objects to the scene as well as reflections. We believe that this work provides additional evidence that the Barycentric shading approach can help to develop simple processes to create stylistic images and animations that are visually reminiscent of paintings.

5.2 Future Work

Some improvements for future work could include a few new features to the overall process. For one, properly implementing animation for any human or animal characters present in the painting would bring even more life and storytelling aspects to the time-lapse dynamic paintings. This would require some study into 2D animation (as that would be the most efficient way to animate people or creatures) as well as methods to properly employ this into a 3D scene for texture projections and obtaining shadow maps. An additional field I would like to propose for further exploration is implementing weather changes into the time-lapse dynamic artwork. This could be another visually interesting way of demonstrating a time shift within a painting. It would most likely require more research into an artist's library of work or possibly some guesswork as to how they might paint a stormy sky or snowy day. Perhaps looking into an entire seasonal time-lapse for dynamic paintings could serve as a foundation for research in the future.

Another area of future research could be looking into a greater utilization of normal or height maps based on the texture and surface quality of the brushwork found in the artists' original works.

These maps could even be applied to the shadow images in order to get more accurate, visually-interesting shading on props or human characters. Furthermore, one could even incorporate certain insights of traditional painting such as the fact that skies are generally painted as a lighter hue than the surface of water. Accomplishing these tasks would bring the results of our time-lapse dynamic painting process even closer to producing something akin to a real-life work of art.

Lastly, a couple of final subjects for future study in this area of computer graphics would be the development of software or programs that increase the overall efficiency and ease of our Barycentric procedure. For one, being able to directly plug in our Bilinear Barycentric equation into any compositing software, like Adobe AfterEffects or Nuke, would remove the need to translate this math to something akin to blending modes during the compositing process. Other software development to improve productivity would be creating a program that allows one to completely bypass any of our methods in Autodesk Maya. In other words, this would be software that can generate proxy geometry automatically, and allow for simple interpolation of these "bad" models. Consequently, this would remove the need for the person creating a dynamic painting with our procedure to have knowledge of 3D programs. Overall, I believe that any of these proposals would be attainable through more work and examination.



(a) Original work.



(b) Our time-lapse dynamic painting.



(c) Original work.



(d) Our time-lapse dynamic painting.

Figure 5.1: A comparison between the original paintings by Anne Garney and our time-lapse dynamic painting versions of her work. The paintings are "Fusion at Playa del Carmen" and "Le Dolce Vita' Burano, Italy," respectively.

REFERENCES

- [1] Ergun Akleman. Implicit painting of CSG solids. In *Proc. of CSG*, volume 98, pages 99–113, 1998.
- [2] Ergun Akleman. Implicit surface painting. In *Proc. of Implicit Surfaces' 98*, pages 63–68, 1998.
- [3] Ergun Akleman, S Liu, and Donald House. Barycentric shaders: Art directed shading using control images. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*, pages 39–49, Geneva, Switzerland, 2016. Eurographics Association.
- [4] Xiaobo An and Fabio Pellacini. User-controllable color transfer. *Computer Graphics Forum*, 29(2):263–271, 2010.
- [5] Saif Castaneda and Ergun Akleman. Shading with painterly filtered layers: A technique to obtain painterly portrait animations. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, page 8, New York City, NY, 2017. ACM.
- [6] Saif Aymen Castaneda. Shading with painterly filtered layers: A process to obtain painterly portraits. Master's thesis, Texas A&M University, College Station, TX, 2017. Retrieved from <http://oaktrust.library.tamu.edu/handle/1969.1/165954>.
- [7] Ching Chan, Ergun Akleman, and Jianer Chen. Two methods for creating Chinese painting. In *10th Pacific Conference on Computer Graphics and Applications*, pages 403–412. IEEE, 2002.
- [8] Ching (Clara) Chan. Two computational approaches for creating Chinese painting. Master's thesis, Texas A&M University, College Station, TX, 2000. Retrieved from <http://hdl.handle.net/1969.1/2643>.

- [9] Ji Cheng. Art directed watercolor shader for non-photorealistic rendering with a focus on reflections. Master's thesis, Texas A&M University, College Station, TX, 2017. Retrieved from <http://oaktrust.library.tamu.edu/handle/1969.1/165773>.
- [10] Cassidy J Curtis, Sean E Anderson, Joshua E Seims, Kurt W Fleischer, and David H Salesin. Computer-generated watercolor. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pages 421–430. ACM Press/Addison-Wesley Publishing Co., 1997.
- [11] Charles Barrett Davis. Landscapes of color. Master's thesis, Texas A&M University, College Station, TX, 2011. Retrieved from <http://oaktrust.library.tamu.edu/handle/1969.1/ETD-TAMU-2011-12-10352>.
- [12] Connaissance des Arts. *Van Gogh, La Nuit Étoilée: Atelier des Lumières*. Société Française de Promotion Artistique, Paris, 2019.
- [13] Oliver Deussen and Thomas Strothotte. Computer-generated pen-and-ink illustration of trees. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 13–18. ACM Press/Addison-Wesley Publishing Co., 2000.
- [14] Yuxiao Du and Ergun Akleman. Charcoal rendering and shading with reflections. In *ACM SIGGRAPH 2016 Posters*, page 32. ACM, 2016.
- [15] Yuxiao Du and Ergun Akleman. Designing look-and-feel using generalized crosshatching. In *ACM SIGGRAPH 2017 Talks*, page 40. ACM, 2017.
- [16] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A non-photorealistic lighting model for automatic technical illustration. In *SIGGRAPH*, volume 98, pages 447–452, 1998.
- [17] Bruce Gooch and Amy Gooch. *Non-Photorealistic Rendering*. AK Peters/CRC Press, 2001.
- [18] Donald E. Gordon. On the origin of the word expressionism. *Journal of the Warburg and Courtauld Institutes*, 29:368–385, 1966.

- [19] Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pages 453–460. ACM, 1998.
- [20] Aaron Hertzmann. Introduction to 3D non-photorealistic rendering: Silhouettes and outlines. *Non-Photorealistic Rendering. SIGGRAPH 99 Course Notes*, 1999.
- [21] Matthew Justice. A process to create dynamic landscape paintings using barycentric shading with control paintings. Master’s thesis, Texas A&M University, College Station, TX, 2018. Retrieved from <https://oaktrust.library.tamu.edu/handle/1969.1/174338>.
- [22] Matthew Justice and Ergun Akleman. A process to create dynamic landscape paintings using barycentric shading with control paintings. In *ACM SIGGRAPH 2018 Posters*, page 30, New York City, NY, 2018. ACM.
- [23] Chethna Kabeerdoss. Art directed shader for real time rendering - interactive 3D painting. Master’s thesis, Texas A&M University, College Station, TX, 2016. Retrieved from <http://oaktrust.library.tamu.edu/handle/1969.1/165773>.
- [24] Jonathan Kiker. Using local information for compositing CG into traditional art. Master’s thesis, Texas A&M University, College Station, TX, 2009. Retrieved from <http://hdl.handle.net/1969.1/ETD-TAMU-2009-05-325>.
- [25] Piotr Krawczyk. Futurist sculpting: Modeling movement in 3D. Master’s thesis, Texas A&M University, College Station, TX, 2006. Retrieved from <http://hdl.handle.net/1969.1/5014>.
- [26] Jaewook Lee. Volume painting: Incorporating volumetric rendering with line integral convolution (LIC). Master’s thesis, Texas A&M University, College Station, TX, 2005. Retrieved from <http://hdl.handle.net/1969.1/2642>.
- [27] Liang Lin, Kun Zeng, Yizhou Wang, Ying-Qing Xu, and Song-Chun Zhu. Video stylization: Painterly rendering and optimization with content extraction. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(4):577–590, 2012.

- [28] Peter Litwinowicz. Processing images and video for an impressionist effect. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pages 407–414. Citeseer, 1997.
- [29] Siran Liu and Ergun Akleman. Chinese ink and brush painting with reflections. In *ACM SIGGRAPH 2015 Posters*, page 46. ACM, 2015.
- [30] Aidong Lu, Christopher J Morris, David S Ebert, Penny Rheingans, and Charles Hansen. Non-photorealistic volume rendering using stippling techniques. In *Proceedings of the Conference on Visualization '02*, pages 211–218. IEEE Computer Society, 2002.
- [31] Aditi Majumder and M Gopi. Real time charcoal rendering using contrast enhancement operators. In *Proceedings of Symposium of Non Photorealistic Animation and Rendering*, 2002.
- [32] Lee Markosian, Michael A Kowalski, Daniel Goldstein, Samuel J Trychin, John F Hughes, and Lubomir D Bourdev. Real-time nonphotorealistic rendering. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pages 415–420. ACM Press/Addison-Wesley Publishing Co., 1997.
- [33] S. Meadows and E. Akleman. Abstract digital paintings created with painting camera technique. *Proc. D'ART 2000 / Information Visualization 2000*, 2000.
- [34] Scott Harrison Meadows. The painting camera: An abstract rendering system with camera-control images. Master's thesis, Texas A&M University, College Station, TX, 2000. Retrieved from <http://oaktrust.library.tamu.edu/handle/1969.1/ETD-TAMU-2000-THESIS-M42>.
- [35] Laura Kristine Murphy. Procedural generation and rendering of trees and landscapes in the style of Eyvind Earle. Master's thesis, Texas A&M University, College Station, TX, 2015. Retrieved from <https://oaktrust.library.tamu.edu/handle/1969.1/155331>.

- [36] George Papagiannakis, Gregoire L'Hoste, Alessandro Foni, and Nadia Magnenat-Thalmann. Real-time photo realistic simulation of complex heritage edifices. In *Proceedings Seventh International Conference on Virtual Systems and Multimedia*, pages 218–227. IEEE, 2001.
- [37] Fermi Niveditha Pramananda Perumal. An integrated model for direct illumination, subsurface scattering, specular highlights and shadow computation. Master's thesis, Texas A&M University, College Station, TX, 2017. Retrieved from <http://hdl.handle.net/1969.1/165958>.
- [38] Jeffrey Smith, Ergun Akleman, Richard Davison, John Keyser, et al. Multicam: A system for interactive rendering of abstract digital images. In *Bridges: Mathematical Connections in Art, Music, and Science*, pages 265–272. Bridges Conference, 2004.
- [39] Jeffrey Statler Smith. Multi-camera: Interactive rendering of abstract digital images. Master's thesis, Texas A&M University, College Station, TX, 2003. Retrieved from <http://oaktrust.library.tamu.edu/handle/1969.1/341>.
- [40] Won Chan Song. Speed-line for 3D animation. Master's thesis, Texas A&M University, College Station, TX, 2005. Retrieved from <http://hdl.handle.net/1969.1/4730>.
- [41] Mario Costa Sousa and John W. Buchanan. Computer-generated graphite pencil rendering of 3D polygonal models. *Computer Graphics Forum*, 18(3):195–208, 1999.
- [42] Michael Leighton Stanley. Digital compositing with traditional artwork. Master's thesis, Texas A&M University, College Station, Texas, 2005. Retrieved from <http://hdl.handle.net/1969.1/2643>.
- [43] Thomas Strothotte and Stefan Schlechtweg. *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation*. Morgan Kaufmann, 2002.
- [44] David Vanderhaeghe and John Collomosse. Stroke based painterly rendering. In *Image and Video-Based Artistic Stylisation*, pages 3–21. Springer, 2013.
- [45] Youyou Wang and Ergun Akleman. Turning photographs into abstract expressionist paintings. In *ACM SIGGRAPH 2012 Posters*, page 66. ACM, 2012.

- [46] Youyou Wang, Ozgur Gonen, and Ergun Akleman. Global illumination for 2D artworks with vector field rendering. In *ACM SIGGRAPH 2014 Posters*, page 95. ACM, 2014.
- [47] Yinan Xiong. Mock-3D web application: Interactive lighting, rendering and shading for 2D artwork. Master's thesis, Texas A&M University, College Station, TX, 2016. Retrieved from <http://oaktrust.library.tamu.edu/handle/1969.1/15794>.
- [48] Zhao Yan. Painterly shading ocean surface. Master's thesis, Texas A&M University, College Station, TX, 2015. Retrieved from <http://oaktrust.library.tamu.edu/handle/1969.1/15638>.
- [49] Jun-Wei Yeh and Ming Ouhyoung. Non-photorealistic rendering in Chinese painting of animals. *Journal of System Simulation*, 14(6):1220–1224, 2002.
- [50] Wang Youyou. *Qualitative Global Illumination of Mock-3D Scenes*. PhD thesis, Texas A&M University, College Station, TX, 2014. Retrieved from <http://oaktrust.library.tamu.edu/handle/1969.1/157921>.

APPENDIX A

TIME-LAPSE DYNAMIC PAINTING VIDEOS

To view the time-lapse dynamic paintings in video format, please look at the supplementary files associated with this thesis. The files are titled "fusionAtPlayaDelCarmen_recreation.mp4" and "leDolceVitaBuranoItaly_recreation.mp4." These video files are the best and most proper method of viewing the final results of our research.