

A PROCESS TO CREATE DYNAMIC LANDSCAPE PAINTINGS USING BARYCENTRIC  
SHADING WITH CONTROL PAINTINGS

A Thesis  
by  
MATTHEW SCOTT JUSTICE

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  
                                  Alan Freed  
Head of Department,      Tim McLaughlin

December 2018

Major Subject: Visualization

Copyright 2018 Matthew Scott Justice

## ABSTRACT

In this work, we present a process that uses a Barycentric shading method to create dynamic landscape paintings that change based on the time of day. Our process allows for the creation of dynamic paintings for any time of the day using simply a limited number of control paintings. To create a proof of concept, we have used landscape paintings of Edgar Payne, one of the leading landscape painters of the American West. His specific style of painting that blends Impressionism with the style of other painters of the American West is particularly appropriate for the demonstration of the power of our Barycentric shading method.

## DEDICATION

To my girlfriend Madison Kramer, Ergun Akleman, the employees of the Visualization Department, and my family.

## CONTRIBUTORS AND FUNDING SOURCES

I would like to thank my chair of my committee, Ergun Akleman. Without his help I would not have been able to finish this thesis. My committee member Felice House was extremely helpful with providing support and motivation for finishing as well. I would also like to thank Alan Freed for agreeing to be my outside member at the last minute. Staci Dunn has been very helpful with submitting paperwork and helping me through the process of graduating. The Department of Visualization was very generous with helping me as a student to succeed and I would like to thank everyone who works for them. There was no outside funding sources for this research.

## TABLE OF CONTENTS

	Page
ABSTRACT .....	ii
DEDICATION .....	iii
CONTRIBUTORS AND FUNDING SOURCES .....	iv
TABLE OF CONTENTS .....	v
LIST OF FIGURES .....	vi
1. INTRODUCTION AND MOTIVATION .....	1
1.1 Motivation .....	4
1.2 Principles of Painterly Rendering .....	4
1.3 Non-Photo Realistic Rendering .....	4
1.4 American Western Style Landscape Paintings .....	5
2. BACKGROUND AND LITERATURE REVIEW .....	7
3. METHODOLOGY* .....	12
3.1 Creation of Control Painting .....	13
3.2 Creation of Proxy Geometry .....	13
3.3 Projection of Control Images as Textures to Proxy Geometry .....	14
3.4 Animation of a light source that represents Sun .....	15
3.5 Shading & Rendering .....	15
4. IMPLEMENTATION .....	17
4.1 Creation of Control Painting .....	17
4.2 Creation of Proxy Geometry .....	17
4.3 Projection of Control Images as Textures to Proxy Geometry .....	19
4.4 Animation of a light source that represents Sun .....	20
4.5 Shading & Rendering .....	21
5. CONCLUSION AND FUTURE WORK .....	25
5.1 Conclusion .....	25
5.2 Future Work .....	25
REFERENCES .....	27

## LIST OF FIGURES

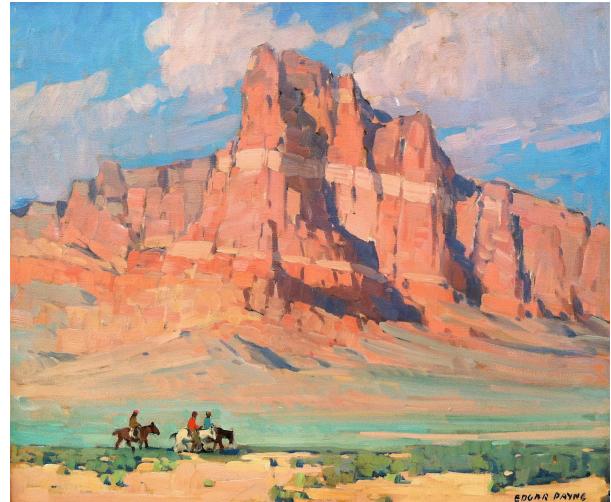
FIGURE	Page
1.1 Examples of Edgar Payne’s Western Paintings .....	1
1.2 Charcoal rendering technique .....	2
1.3 Before and after examples of NPR from Disney’s Paperman .....	3
1.4 Examples of Edgar Payne’s Western Paintings .....	5
2.1 What you see is what you get non-photorealistic rendering [14] .....	7
2.2 A Non-Photorealistic Lighting Model For Automatic Technical Illustration [11] .....	8
2.3 Barycentric shaders: art directed shading using control images .....	10
2.4 Hardware accelerated real time charcoal rendering [20] .....	10
3.1 Four texture images $T_{00}$ , $T_{10}$ , $T_{01}$ , and $T_{11}$ . ....	13
3.2 The proxy geometry that is used to compute illumination and shading. ....	14
3.3 Projection of painting to proxy geometry. ....	14
3.4 An example of artistically controlled shadows and highlights rendered as different times of the day. ....	16
4.1 Select by color range tool .....	18
4.2 Match color tool can copy hues and values from one layer to another .....	18
4.3 The proxy geometry that is used to compute illumination and shading. ....	19
4.4 Camera Projection node set up .....	20
4.5 Main Shader .....	22
4.6 Secondary shader that blends between Dusk and Daylight textures .....	23
4.7 Shader Network .....	24

## 1. INTRODUCTION AND MOTIVATION

The current modeling and rendering processes require too much knowledge to create hand-painted looking dynamic paintings for novice users such as traditional painters. Therefore, there is a need for the development of processes to obtain desired dynamic paintings with intuitive and artistic control. In this work, we present a simple process to create dynamic paintings that can change based on existing lighting conditions. To demonstrate the effectiveness of my process I created dynamic paintings starting from Edgar Payne's Western landscape paintings.



(a) Canyon De Chelly, Oil on canvas, 20 x 24 in. Edgar Payne



(b) Arizona Mesa, Oil on canvas, 25 x 30 in. Edgar Payne

Figure 1.1: Examples of Edgar Payne's Western Paintings.

This method uses modern ray traced based rendering techniques for simulating the style of painters of the American West. I provide a shader and process using off-the-shelf tools for artists to create dynamic paintings. The method also allows artists greater control for replicating a specific artist's painting style and allows artists to easily create new painterly renders of environments without much overhead. As an example, I turned two paintings of Edgar Payne into dynamic

structures.

I chose Payne since he understood how to make colors and landscapes have a luminous feel to them along with understanding how atmosphere plays into the way light interacts with the landscape (See Figure 1.1). Studying his artwork will allow computer scientists to make more visually pleasing images and push the field forward by learning from artists of the past. Furthermore, Payne's specific style of painting that blends Impressionism with the style of other painters of the American West was useful for this particular application since I was able to gather information from his other paintings.

Computers are great at making perfect geometrically accurate renders of objects and the focus of most work in rendering algorithms and techniques recently has been to render objects as realistically as possible. Sometimes when a new medium comes out, such as computer graphics, the first adopters are the people who created the medium. Those scientists toil endlessly to make the render work and look to other sciences on which to base their formulas and programs. However, humans have already been rendering images for thousands of years using other mediums such as painting. There exists a vast knowledge on how to create images using traditional art techniques that should and must be transferred to computer graphics. By studying artists and painters and replicating their styles we can learn some important lessons that can be applied to computer graphics. Painters know how to capture light in a way that a camera or traditional render algorithm cannot replicate, but by making new tools we can push the quality of images produced by computer graphics forward and bridge the gap between the old and new. Non-photorealistic



Figure 1.2: Charcoal rendering technique



(a) Disney's Paperman, 2012



(b) Disney's Paperman, 2012

Figure 1.3: Before and after examples of NPR from Disney's Paperman

rendering (NPR) creates images that intend to replicate a specific artist's style or a genre of art that are historically not generated via a computer. An example is the charcoal rendering technique developed by Du (See Figure 1.2) [8]. It has also been used to blend computer generated elements with traditional hand drawn animation to allow for advanced effects or for lowering the cost of motion picture production (See Figure 1.3). The current focus for commercial renderers such as Renderman or Arnold has been physical accuracy and replicating real life. This creates a need for work in developing Non-photorealistic rendering techniques. Many animation productions desire to push the style of animated films beyond what has been achieved with these physically accurate renderers. Short films such as Disney's Paperman replicate the look and feel of hand drawn animation quite well and that is an example of a top tier production house seeing the need for Non-photorealistic rendering. As shown in Figure 1.3, the render on the left shows how traditional shading lacks the flat quality and that expressive lines that come from the 2D animation that the director wanted [25]. The simplification of forms that NPR allows makes it easier to tell a story. Traditional computer graphic push more and more towards the photo realistic which may not be compelling for the intended audience of cartoons: children.

Although there has been much work in the area of NPR, it is still not an easy task to achieve the desired look an art director wants or to mimic a specific artist's style. Shade trees rendering architecture developed by Cook [6] created a flexible tree-structured shading model that has been

used to allow users to create any look and style they desire. Recently a new approach using Barycentric algebra to control the color of light and shadows in renders has been developed by Akleman [1]. This approach has been used with success recently by [19] and [8]. I plan on expanding their work by replicating the style of Western paintings.

## 1.1 Motivation

The look and feel of a film refers to the artistic style defining the world and characters of a story and is one of the most important attributes defining one narrative from another [23]. My goal is to achieve a method of rendering images in a Western painterly style that can be used in a traditional animation or VFX pipeline. I want to make a shader and technique for creating texture maps that allows artists a quick and easy way to control the look of an image and match the style of various painters. I will develop a Barycentric shader to achieve this look along with a method to recreate brush strokes using projections and displacement mapping.

## 1.2 Principles of Painterly Rendering

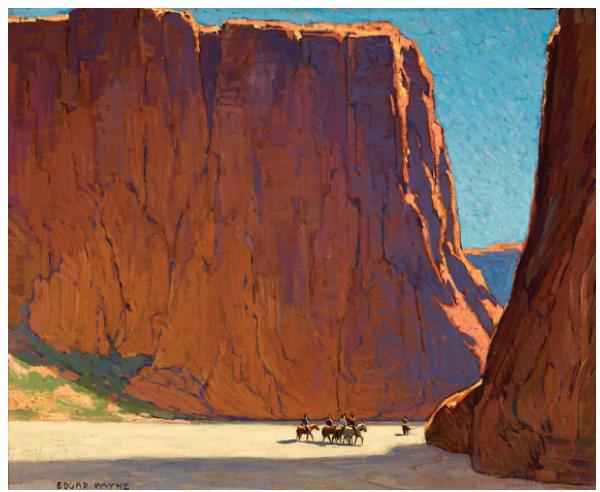
Painterly renders create the illusion that an image has been made with traditional painting techniques such as brush strokes that change in size depending on the desired amount of detail and look [17](See Figure 2.1). Some techniques try to convert 2D images to a painterly look while others try to achieve the style by using traditional VFX rendering techniques [14]. The overall goal is to replicate the look and feel of a painting with a computer. Painterly rendering is also be considered a part of non-photorealistic rendering, which can include any type of rendering that does not aim for photo realism [11].

## 1.3 Non-Photo Realistic Rendering

In the last three decades there has been much work in the field of Non-photorealistic rendering in a variety of styles that range from cartoon two dimensional work [28] to painterly rendering for animations [22] and beyond. Although techniques between methods differ, there has been a common goal: to push the look and feel of computer generated images beyond that of what is currently available. Content creators are interested in the development of these tools and methods as a way



(a) Desert Rain, 1925 - 1930, Oil on Canvas, 25 x 30 in. Edgar Payne



(b) Sunset, Canyon de Chelly, 1916. Oil on Canvas, 28 x 34 in.

Figure 1.4: Examples of Edgar Payne's Western Paintings.

to communicate information in new ways and to expand on narratives. Computer graphics has traditionally been focused on creating photo-realistic imagery which are well suited for documentation and simulation [10]. NPR is also suited for explanation, illustration, and storytelling roles. The field emerged in the 1990s as a response to photorealism. The goal was to make an image that was created by an artist's hand using traditional methods [3] such as watercolor [7], oil paints [12], or even traditional hand-drawn animation [29]. These methods mix automation with input from artists and attempt to mimic what traditional artists can do.

## 1.4 American Western Style Landscape Paintings

Paintings of the American West have a unique look and feel that captures the immense landscape (See Figure 1.4a) that lies west of the Mississippi. The Western United States is the largest region in the country and the most geographically diverse. It contains all the deserts in the United States, the highest mountain ranges, and the temperate rain forests of the Pacific Northwest. The wide range of landscapes has inspired artists to paint the area for decades. Edgar Payne is among the best known of those painters.

In this thesis, I will focus on Edgar Payne's landscape paintings. Author and painting instructor

Mitchel Albala says that Edgar Payne was among the best known of the California Impressionists of the early twentieth century [3]. He points out that many later landscape painters use his book of Composition of Outdoor Painting as a formative part of their training. Payne uses one of the most important lessons of impressionist colors, which is to keep colors luminous and to avoid overly dark and overly light values. Payne also uses radiant complementary colors to make landscapes appear as if they are glowing in sunlight. An example of this is seen in the painting Sunset, Canyon de Chelly (See Figure 1.4b), where the sky and highlights on the rocks are complements of each other. This technique could also be adapted easily for other artists and styles of painting and art.

## 2. BACKGROUND AND LITERATURE REVIEW

Painterly renders create the illusion that an image has been made with traditional painting techniques such as brush strokes that change in size depending on the desired amount of detail and look [17]. Some techniques try to convert 2d images to a painterly look while others try to achieve the style by using traditional VFX rendering techniques [14](See Figure 2.1). The overall goal is to replicate the look and feel of a painting with a computer. Painterly rendering is also be considered a part of Non-Photorealistic rendering, which can include any type of rendering that does not aim photo-realism [11]. In the last three decades there has been much work in the field of

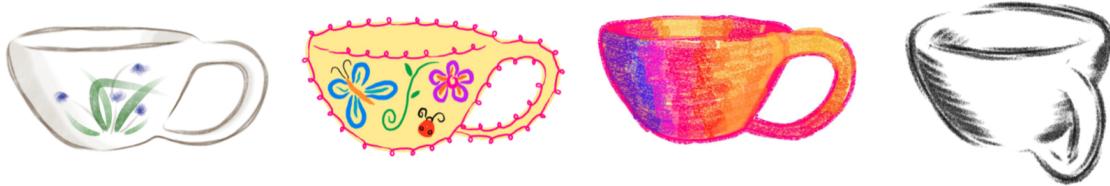


Figure 2.1: What you see is what you get non-photorealistic rendering [14]

Non-Photorealistic (NPR) rendering in a variety of styles that range from cartoon two dimensional work [28] to painterly rendering for animations [22] and beyond. Although techniques between methods differ there has been a common goal which is to push the look and feel of computer generated images beyond that of what is currently available. Content creators are interested in the development of these tools and methods as a way to communicate information in new ways and to expand on narratives. There are some roles that NPR are well suited for compared to photorealism. Computer graphics has traditionally at one point been focused on creating photorealistic imagery which are well suited for documentation and simulation [10]. NPR is well suited for explanation, illustration, and storytelling roles. The field emerged in the 1990s as a response to photorealism. The goal was to make an image that was created by an artists hand using traditional methods [3]

such as watercolor [7] or oil paints [12]. These methods were mostly automated and attempted to mimic what an artist illustrator might do.

In the late 1990s, researchers began to investigate dynamic NPR where the challenge was temporal coherence [18]. These methods create images that would have been incredibly time consuming to create by hand and traditional means and for animated films. Around the same time research into real time rendering algorithms were made possible by the advancement in computer hardware. These techniques were made for entertainment purposes[21] and also for technical renderings for engineering [10](See Figure 2.2). Xiao presents a method for NPR rendering for converting video to painterly image sequences [31]. Cartoon shading using strokes [14] and hatching techniques

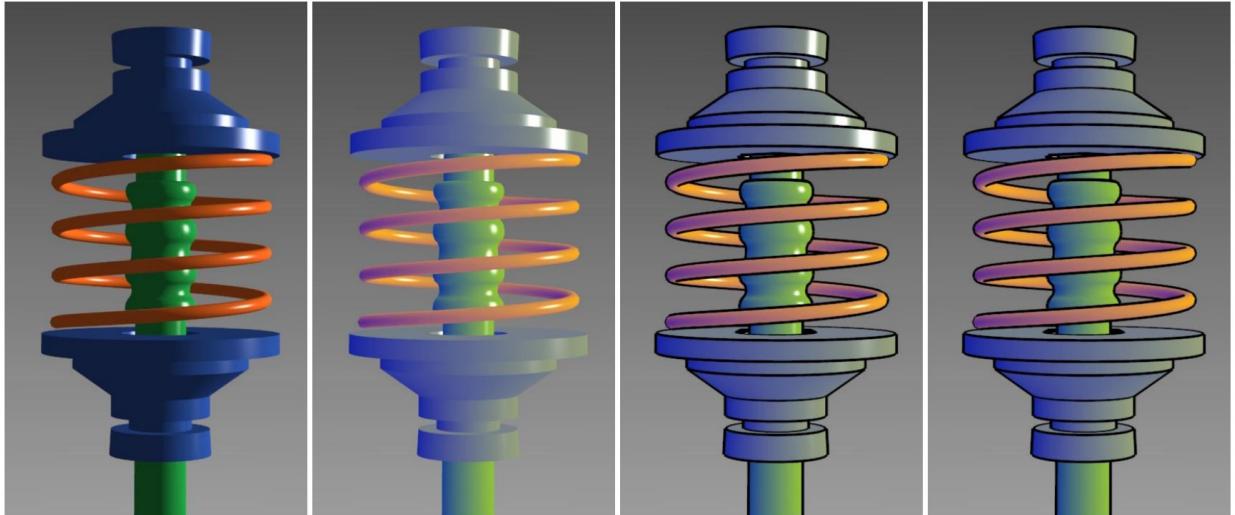


Figure 2.2: A Non-Photorealistic Lighting Model For Automatic Technical Illustration [11]

[24] are a popular use for NPR [28]. A single image can incorporate many different methods together to create a stylized image. Effects such as cartoon shading handle the diffuse color of a 3d object. On top of that, styled strokes are placed in occluded areas to represent shadows. A paper texture is then overlayed to add to the two dimensional look of a drawing. An outline is created using a filter and is applied to the image. All this compo-sited together gives the final look of an illustrated cartoon. An early important method in the area of stroke textures was by Winkenbach and

Salesin [30]. They vary the number of hatch marks to convey different contiguous tones. Drawing surfaces and models using hatch marks conveys material, tone, and form all at the same time so it is a useful technique to achieve. Winkenbach and Salesin wrote a paper describing the principles of traditional pen-and-ink illustration and shows how a number of them can be implemented as a part of an automated rendering system [30]. It introduces textures as strokes called stroke textures which can be used for achieving both texture and tone with line drawing. Hertzman and Zorin present a new algorithm for line art rendering of smooth surfaces in their paper Illustrating smooth surfaces [13]. These methods can be used to find line art in real time renderers. Lake presents real time methods for creating cartoon images from 3d objects in their paper [15]. Bousseau presents a study that reveals a compression of the range of representable gloss in stylized images so that shiny materials appear more accurate in cartoon renderings [5].

Painterly rendering is a fairly large area in the field of NPR. Vanderhaeghe and Collomosse present a way in which they turn photographs and two dimensional image data into painterly images by using simulate brush strokes using local greedy and global optimization approaches to stroke placement [27]. Lindemeir, Spicker, and Deussen made a technique for painterly renders that turn a canvas into a set of layers that reflect the spatial arrangement of the composition and the order in which the paintings are produced in a way the new strokes only minimally paint over existing ones [17]. The number of strokes and colors that are needed to represent an image are minimized by this effect. Lin, Zeng, Wang, Xu, and Zhu present an interactive video stylization system for transforming video interactively into a painterly animation [16]. Tsunematsu, Kawai, Sato, and Yokoya present a NPR technique that creates images with artistic styles of paintings that convert photographs to painterly styles by using paintings as a control image to transfer the style of that painting to the image [26]. Kalnins present a system that lets a designer directly annotate a 3d model with strokes which imparts their personal aesthetic to the non-photo real rendering of the object [14]. The artist chooses a brush style then draws strokes over the model from one or view points and then the system adapts the strokes appropriately to maintain the original look as the camera and object moves.



(a) Intuitive art-directed control of expressive style.



(b) A Chinese Painted 3D Scene

Figure 2.3: Barycentric shaders: art directed shading using control images

There are several different artistic styles or genres that NPR research has attempted to cover in the past. An approach by Akleman [1] using Barycentric shaders (See Figure 2.3) was made to create a framework that will guarantee that whatever style is intended will be obtained consistently regardless of the underlying rendering method and illumination model. This method has been used with success by Yuxiao [9] to create a charcoal rendering method with reflections. Liu also used the technique to create Chinese Ink and Brush Paintings with reflections with much success [19]. Real time rendering of NPR imagery has also been an area of research. An approach by Aditi

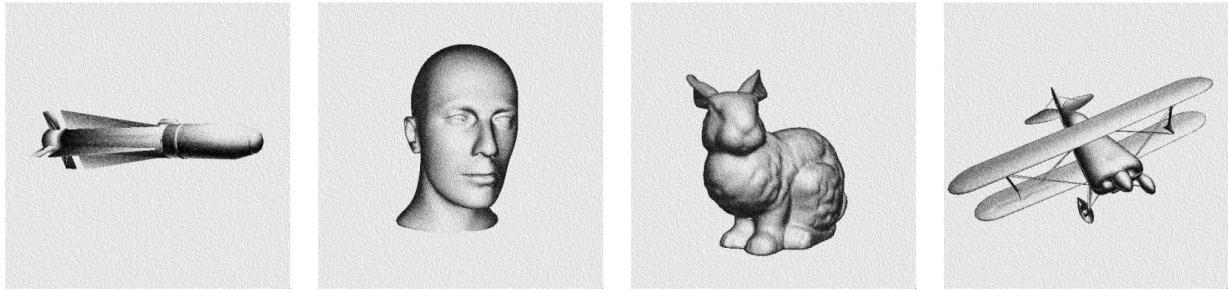


Figure 2.4: Hardware accelerated real time charcoal rendering [20]

Majumder and M. Gopi [20] (See Figure 2.4) created a way to render charcoal drawing images. They created a new algorithm to achieve the effects of charcoal rendering in real time using contrast

enhancement operators. Pruan developed a technique that replicates traditional hatching pencil strokes in real time [24]. Drawing surfaces with hatch marks conveys material, tone, and form all at once. Multitexturing hardware blends the hatch images onto geometry during run-time. Markosian created a NPR technique that modify Appel's hidden-line algorrithm with improvements based on the topology of singular maps of a surface into the plane [21]. This system has the potential to be used in any NPR method that requires a description of visible lines or surfaces.

### 3. METHODOLOGY\*

The current modeling and rendering processes require too much knowledge to create hand-painted looking dynamic paintings for novice users such as traditional painters. Therefore, there is a need for the development of processes to obtain desired dynamic paintings with intuitive and artistic control. In this work, we present a simple process to create dynamic paintings that can change based on existing lighting conditions. Our process provides intuitive and artistic control to obtain the desired look. To demonstrate the effectiveness of our process we created dynamic paintings starting from Edgar Payne’s western landscape paintings.

In this particular example, we use a simple Barycentric shader (See [2] for formal description) with two variables  $s$  and  $t$ , where  $s$  represent what percentage of sunlight can reach a given point and  $t$  is the time of the day normalized as a number from 0 to 1. Let  $P(s, t)$  denote a dynamic painting for a given set of  $s$  and  $t$  values. Also let  $u$  and  $v$  in  $[0, 1]$  represent pixel positions and  $P(s, t, u, v)$  is the color of pixel  $u$  and  $v$  in time  $t$ . Note that  $s$  is the function of  $u$ ,  $v$ , and  $t$ .

Now let us assume that we know how we want our dynamic painting look like for a limited number of sample points, we then, can use these sample paintings as control points and compute  $P(u, v)$  as a weighted average of these control “paintings”. In this particular case, without loss of generality we provide four control paintings as texture maps as  $P(0, 0) = T_{00}$ ,  $P(1, 0) = T_{10}$ ,  $P(0, 1) = T_{01}$ , and  $P(1, 1) = T_{11}$  as shown in Figures 3.1a, 3.1b, 3.1c, and 3.1d respectively. Final dynamic paintings are obtained by using Bilinear interpolation for the given time  $t$  and computed  $s$  for each pixel  $u$  and  $v$  (See Equation 3.1). The advantage of this model, the effect of sky illumination coming from scattered sunlight can be embedded in texture maps using the time term  $t$  as shown in Figure 3.1. We only need the sun’s position along with some geometry information to compute the value of  $s$  per pixel  $u$  and  $v$ .

---

\*Reprinted with permission from “A process to create dynamic landscape paintings using barycentric shading with control paintings” by Matthew Justice, 2018. SIGGRAPH ’18 ACM SIGGRAPH 2018 Posters, Article No. 30, Copyright 2018 by ACM.

Based on this brief description of this particular Barycentric shader, our process requires only five simple stages to obtain dynamic paintings: (1) Creation of Control Paintings; (2) Creation of Proxy Geometry; (3) Projection of Control Paintings to Proxy Geometry; (4) Animation of a light source that represents Sun and (5) Rendering Animation with Barycentric Shaders. Each of these stages is simple and intuitive. Any 2D artist with minor training in 3D modeling and animation can produce a dynamic painting using this process.

### 3.1 Creation of Control Painting

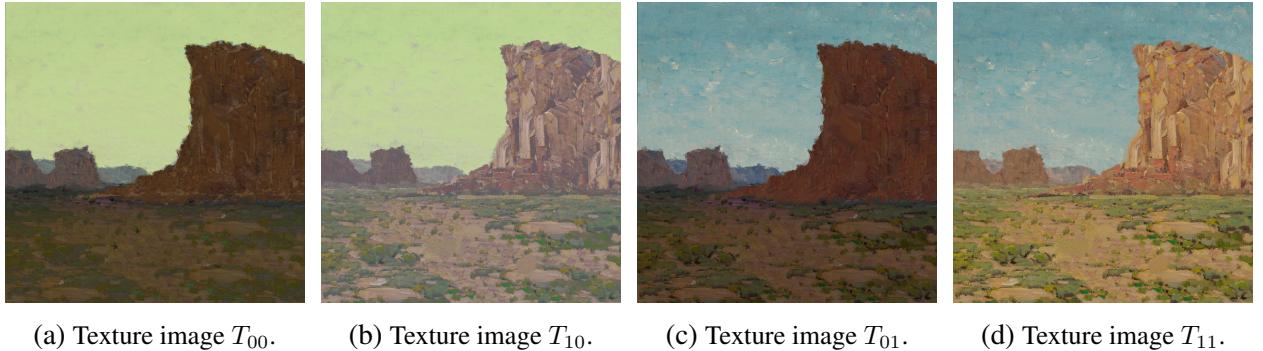
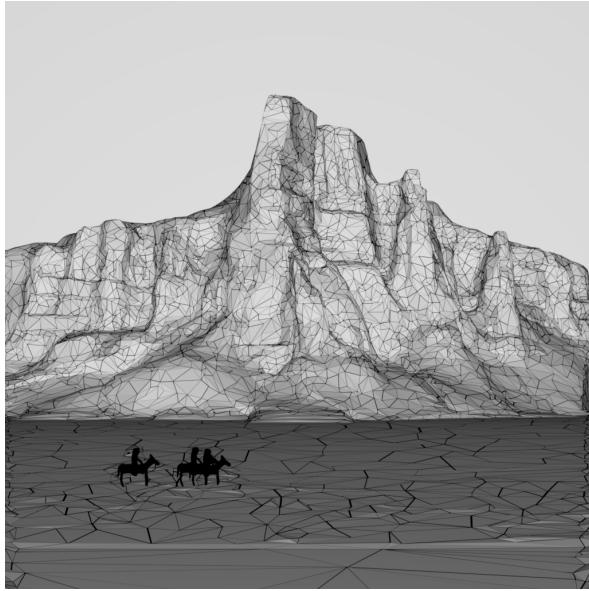


Figure 3.1: Four texture images  $T_{00}$ ,  $T_{10}$ ,  $T_{01}$ , and  $T_{11}$ .

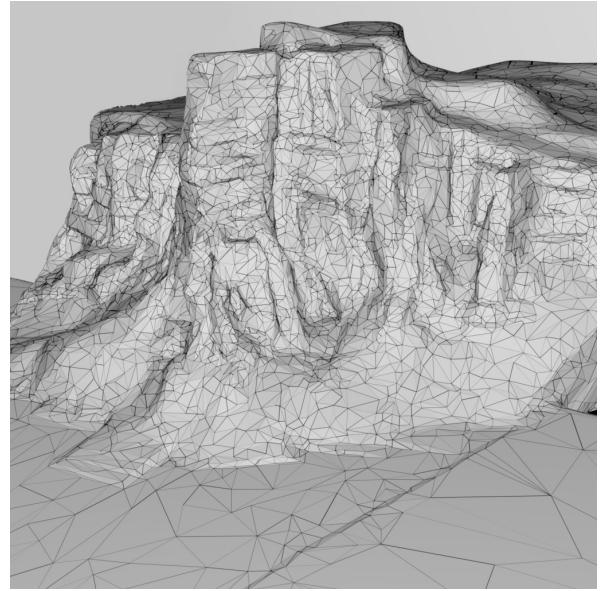
The first and the most critical stage is the creation of control images from available painting. We use a color transfer method to obtain control paintings [4]. We first choose a painting that can turn into a dynamic one. For this painting, we create two paintings such as the ones shown in Figures 3.1a and 3.1b by making one completely in shadow and another one completely illuminated. Using colors from another painting, we create two more images represent a different time of the day such as the ones shown in Figures 3.1c and 3.1d again using the color transfer. This provides us four “un-illuminated” control paintings that can be interpolated to obtain final paintings.

### 3.2 Creation of Proxy Geometry

In the second stage, we create a proxy geometry based on these control images. The most critical issue is that boundaries of proxy geometry have to match shape boundaries in the images.



(a) The proxy geometry from point of view of the painting.



(b) The proxy geometry from another view.

Figure 3.2: The proxy geometry that is used to compute illumination and shading.

The internal shape of the proxy geometry is not that critical. However, we still need to preserve the internal boundaries, which will correspond to discontinuities in shadows. Although in this example we build a polygonal mesh, the proxy geometry can simply be a height field.

### 3.3 Projection of Control Images as Textures to Proxy Geometry

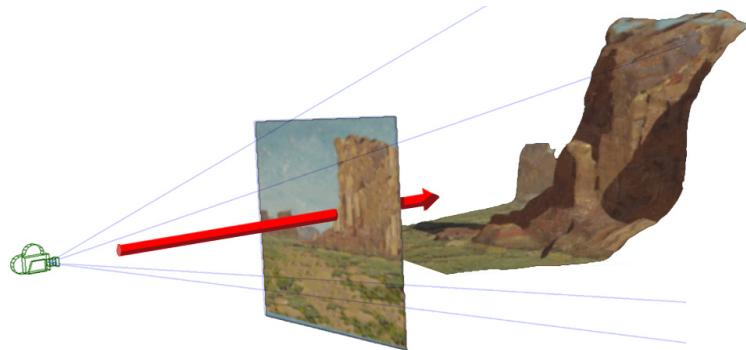


Figure 3.3: Projection of painting to proxy geometry.

The third stage is texture mapping four control images to proxy geometry. This process is simply a projection as shown in Figure 3.3. To animate the camera position, the texture images may need to be separated. In fact, in this case, we have separated the sky, ground, and rocks into three different textures. To simplify the presentation, we omit the details.

### 3.4 Animation of a light source that represents Sun

The fourth stage is to provide an animation of the light that represents the Sun. We simply need the position of the light to make a decision about shading and illumination. Illumination is more critical to obtaining 3D looking results since it can clearly provide occlusion information through shadows.

### 3.5 Shading & Rendering

In the shading and rendering stage, the key is the computation of  $s$  value. For  $s$  value, we still use the standard diffuse term, which is calculated as the dot product between the unit vector of a point in space representing the light in the scene and the shading point and the surface normal at that shading point. But, the key feature to obtain the 3D look comes from raytraced shadows. To compute shadows, we simply fire a ray from the shading point at the light using the trace function built into open shading language to identify if shading point is in shadow or not. The diffuse and shadow terms are then mixed to obtain an  $s$  value that provides a weight the highlight and shadow colors for that shading point and pixel  $u$  and  $v$ . The Bilinear Barycentric shader blends between four texture files using  $s$  and  $t$  values as follows:

$$\begin{aligned} P(s, t, u, v) = & T_{00}(u, v)(1 - s)(1 - t) + T_{01}(u, v)(1 - s) \ t + \\ & T_{10}(u, v) \ s \ (1 - t) + T_{11}(u, v) \ s \ t. \end{aligned} \quad (3.1)$$

As demonstrated in Figures 3.4, this simple process allows creating dynamic paintings that can change based on existing lighting conditions. In this work, we have not included global effects, but the method can be extended to handle such global effects.



Figure 3.4: An example of artistically controlled shadows and highlights rendered as different times of the day.

## 4. IMPLEMENTATION

Here, I explain the implementation of the methodology again in four stages: (1) Creation of Control Paintings; (2) Creation of Proxy Geometry; (3) Projection of Control Paintings to Proxy Geometry; (4) Animation of a light source that represents Sun and (5) Rendering Animation with Barycentric Shaders. Each of these stages is simple and intuitive. Any 2D artist with minor training in 3D modeling and animation can produce a dynamic painting using this process.

### 4.1 Creation of Control Painting

In order to create the control paintings we use to project onto the geometry, we first select a painting we wish to replicate. The goal is to make two control paintings, one of only the **highlights** and one of only the **shadows**. To achieve this we use Photoshop. Then we open the photo of the painting in Photoshop and duplicate it so the same image is in two layers. Using the select color range tool (See 4.1), select all the colors that are in shadow copy those to a new layer. Now you should have a layer of only the highlights selected. Invert this selection and copy those into a different layer as well and those should be only the highlights. With the shadow layer selected, use the match color tool (See 4.2) to copy the hues and values of the highlights layer and merge those layers together in order to create a image of just highlights. Repeat for the shadow control painting.

After we have the two control paintings from the painting we are basing the geometry off of we can then create control paintings (See 3.3) for different times of day. We have to search for paintings from other times of day which we then load into Photoshop. Then we can use the match color tool to copy the hues and values from the secondary painting to the highlights and shadow control paintings.

### 4.2 Creation of Proxy Geometry

To create the proxy geometry we first load the base painting we are basing our work off of into Autodesk's Maya as a image plane attached to the camera. The goal now is to create a rough

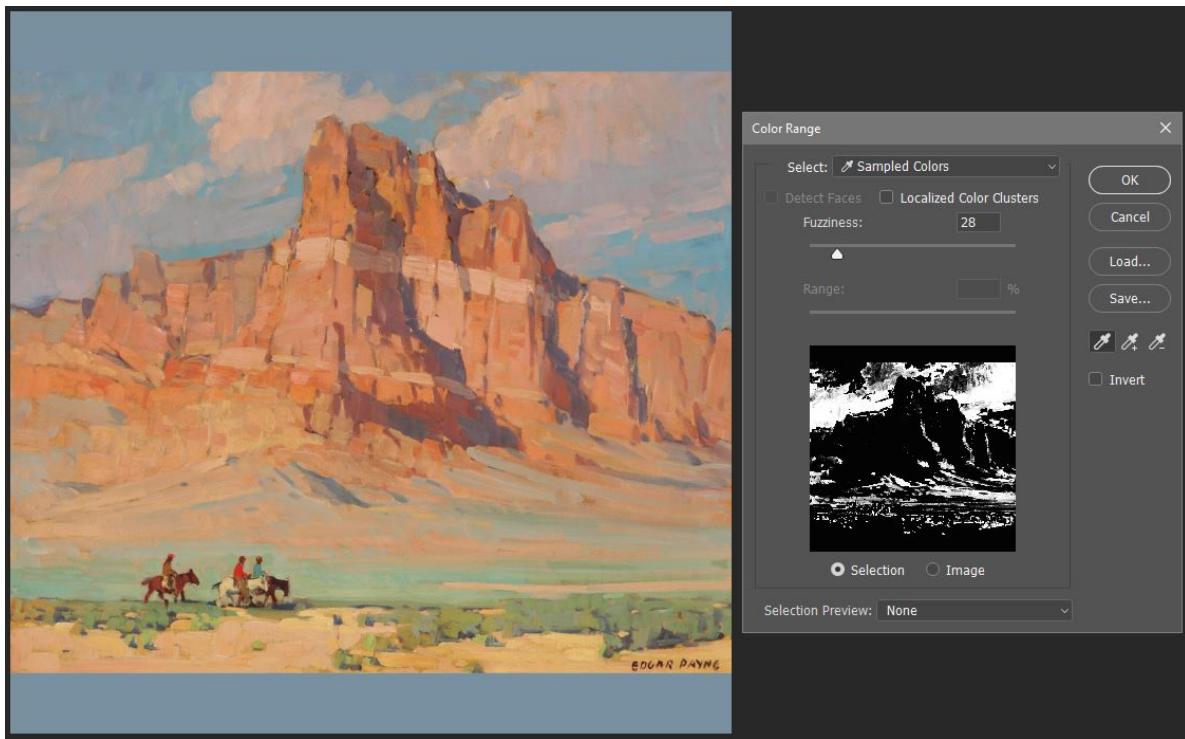


Figure 4.1: Select by color range tool

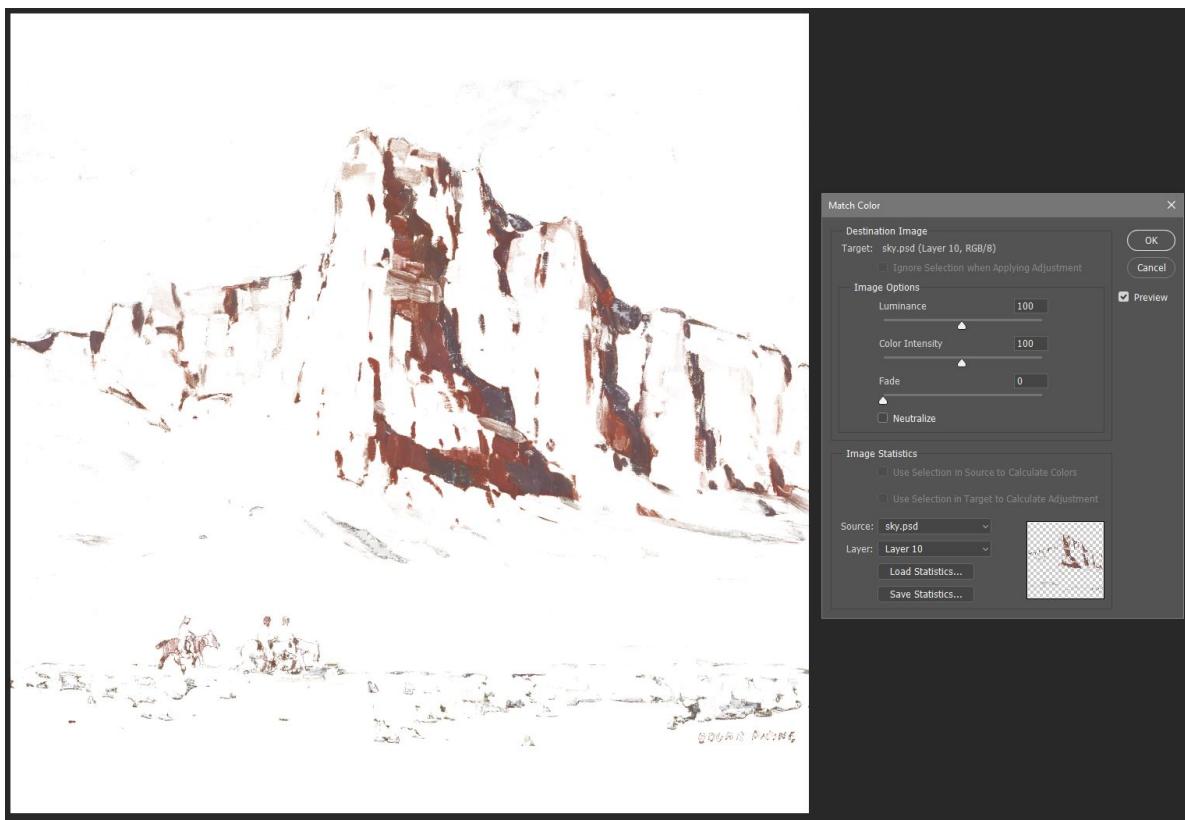
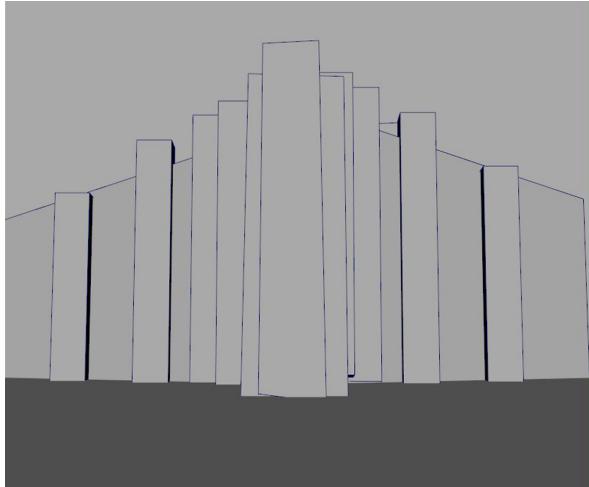


Figure 4.2: Match color tool can copy hues and values from one layer to another



(a) Proxy geometry made with cubes



(b) Rendered Proxy geometry

Figure 4.3: The proxy geometry that is used to compute illumination and shading.

version of the environment by placing simple cubes that have been transformed and translated into the forms that make up the painting. The topology of the meshes is not important at the moment, what is important is trying to match the scale and depth of the environment as best as possible, an example is shown in figure 4.3. What else is very important is matching the edge boundaries of the mesh when compared to the edges of the objects in the painting. After roughing in the environment, the job now is to merge all the geometry together. This is unnecessary but can be used to remove unused geometry. To do this we use Zbrush as it has powerful re-topology tools. The geometry is exported from Maya as an object, .obj, file and then imported into Zbrush. The Dynamesh tool inside Zbrush is then used to convert all the different pieces of geometry into one water tight object. That object is then exported from Zbrush and re imported into Maya. The object is then cleaned up inside Maya using the Cleanup tool to remove any issues the geometry may have such as Lamina faces. Once back in Maya the artist can use the sculpt geometry tools to easily grab the vertices of the environment geometry and drag them to better represent the paintings landscape.

### 4.3 Projection of Control Images as Textures to Proxy Geometry

One of the main problems with a lot of NPR methods that try to simulate paintings is that the objects still retain their 3d look. A lot of that I believe has to do with the way textures are mapped

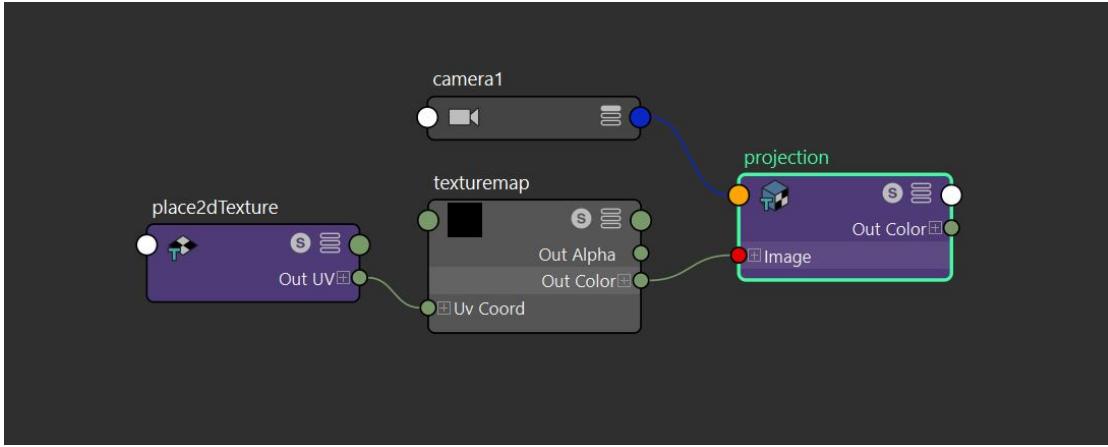


Figure 4.4: Camera Projection node set up

to the surfaces, usually using UV space. Paintings are 2d representations of the real world and have 2d qualities to them such as brush stroke and texture that can be lost when mapped to an object using UVs. As objects rotate or are angled away from the camera view, those details are skewed or transformed from what they would be in a painting. They lose the flat quality they have. Brush strokes in a painting are a quality of the surface of the painting itself and I think cannot be successfully mapped to a surface in anyway other than a camera projection. **A camera projection is a way of mapping a 2d image from the view of a camera onto 3d topology.** This allows us to preserve the scale of the brushstrokes and texture that makes up the surface of a painting throughout the 3d scene. The downside to this technique is that from every camera angle that is intended to be rendered, a new set of texture maps must be made.

To project the image in Maya we use the projection node and link it to the render camera. See 4.4 . These nodes are then plugged into the rest of the node network 4.7. Using projections also allows us the to skip the task of uv unwrapping which will simplify the process for artists.

#### 4.4 Animation of a light source that represents Sun

In order for us to make it easier to animate the position of the light vector (sun), we use a locator. A locator is a small icon like an x-y-z axis that marks a point in space and that doesn't get rendered with the scene. It is useful because it gives the artist a visual representation of the light

position and allows an easy way to set keys to animate the light. The  $y$  value of the light is what we use to control the mix between the dusk and sunlight control paintings and allows us to have different times in our renders.

## 4.5 Shading & Rendering

Our shader to compute  $s$  is written in Open Shading Language, see 4.5, and it is rendered in Maya using the Arnold renderer. The computation is simple using Open Shading Language and Arnold renderer. The shader is complied using the OSLC compiler provided with Arnold and Maya 2018. The shader uses three float inputs for the location of the light vector. There are two color inputs, one for the highlight images and one for the shadow images. The shader finds cosinetheta by finding the dot product between the normalized light vector and the surface normal. It then interpolates that value between zero and one by using the smoothstep command and sets that value to  $s$ . The trace command is what we use to shoot a ray from the shading point,  $P$ , and the light location. If the ray hits a piece of geometry,  $hit$  returns the value of 1 and we set  $s$  to zero which makes the point being shaded be set to the shadow texture.

To blend between the dusk and daylight paintings to change the lighting depending on the time of day, we use a secondary OSL shader. The secondary shader takes in the results from two of the main shaders and mixes them depending on the  $y$  value of the light position; see 4.6. A total of four control paintings are used for this.

These shaders and control images are connected together in a node network inside Maya. See 4.7 After the shader is connected to a piece of geometry the scene can be rendered.

```

shader thesis()

//three float inputs for the location of the light vector
float diff_x = 0.0,
float diff_y = 0.0,
float diff_z = 0.0,

//color input for shadow images
color t0 = color(0,0,0),

//color input for highlight images
color t1 = color(1,1,1),

//variable used to blend the two color maps
float s = 0,

//result RGB values
output color resultRGB = color(0),
)

{

//makes the light location into a vector
vector A = vector(diff_x,diff_y,diff_z); // Locator for diffuse

//normalizes the vector between the light location and shading point
vector L = normalize(A-P);

//calculates the angle between the light vector and the surface normal
float cosinetheta = dot(N,L);

//interpolates cosine theta between 0 and 1
s = smoothstep(-1,1,cosinetheta);

//Retrieves data about the object hit by the last trace call made
int hit = trace(P, normalize(A));

//'"hit" int Zero if the ray hit nothing, 1 if it hit
if(hit ){

    //if a ray hits geometry set s to 0 to set in shadow
    s = 0;
}

//use s to mix the two color inputs
resultRGB = mix(t0,t1, s);
}

```

Figure 4.5: Main Shader

```

shader t (
    //dusk image
    color T0 = color(0,0,0),
    //daylight image
    color T1 = color(1,1,1),
    //output color values
    output color resultRGB = color(0)
    //y value of light position
    float diff_y = 0.0,
    //output variable used in main shader
    output float t = 0,
)
{

    //normalize t and clamp values to 0 -1
    t = diff_y / 100;
    if( t > 1){ t = 1;}
    if( t < 0){ t = 0; }

    //use t to blend between dusk and daylight images
    resultRGB = mix(T0,T1, t);
}

```

Figure 4.6: Secondary shader that blends between Dusk and Daylight textures

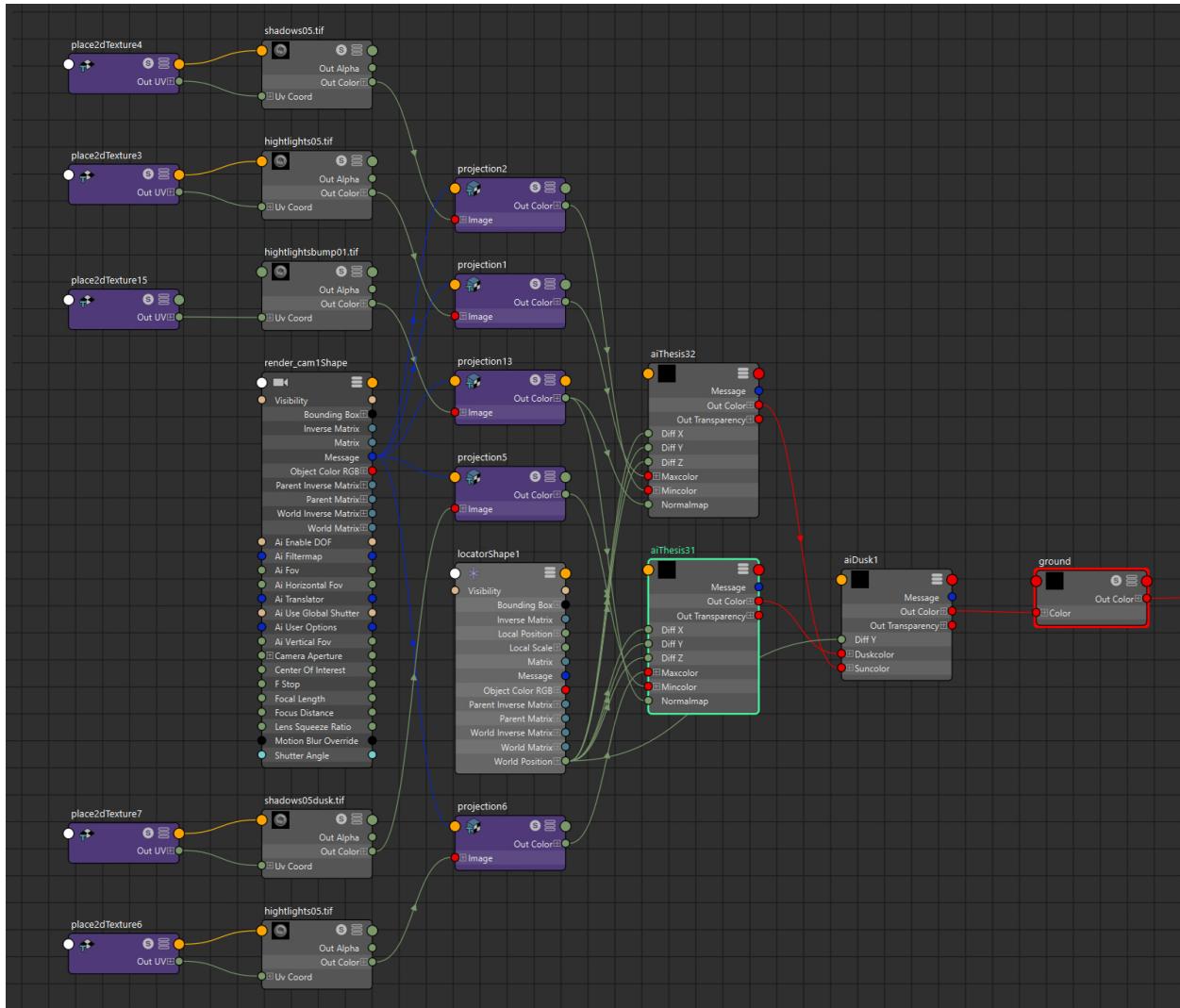


Figure 4.7: Shader Network

## 5. CONCLUSION AND FUTURE WORK

### 5.1 Conclusion

In this work, we presented a process that uses Barycentric shading method to create dynamic landscape paintings that change based on time of the day. Our process can allow creating dynamic paintings for any time of the day using simply a limited number of control paintings. His specific style of painting that blends Impressionism with the style of other painters of the American West is particularly appropriate for the demonstration of the power of our Barycentric shading method. We believe that this method is easy and simple to use and can be implemented by artists everywhere who have simple training in computer graphics.

One way to improve the results I believe is to have an artist paint the control paintings from scratch with the intention of the paintings being used in this process. The artist would have a 3d model as reference and then paint the highlights and shadow paintings from that. This would allow the artist to have more control of the outcome than altering the paintings in Photoshop.

### 5.2 Future Work

In the future I think this technique could be adapted to create dynamic background paintings for wallpapers on computers and smart phones and devices. Imagine having a painting as the background on your smart phone that dynamically changes throughout the day depending on time, location, and direction one is facing. With the advent of graphics cards being made with the sole intention of raytracing, I believe the technology will trickle down to consumer devices which will allow this technique to run on everyday devices in real time.

Another area to explore is the use of height maps gathered from satellite imagery that could allow the user to render anywhere in the world as if it was painted in the style of a specific artist. The control paintings would have to be procedurally generated and projected onto the geometry in a separate fashion than what we are using but I think that it would work and have good results.

This process can also be combined with other NPR techniques to create a more complete NPR

process. NPR reflections, refraction, and temporal techniques that allow camera movement could allow for better results.

Another area to explore is how to add surface qualities of "paintings" to the render. I am referring to the texture of paint strokes and the canvas and how that adds to the sculptural quality some paintings have. Artists like Vincent Van Gogh use a large amount of paint applied to canvases that give a depth to a painting that sometimes only comes across in person. I believe that this could be achieved easily through the use of normal maps and a simple shader added to the end of the node network.

## REFERENCES

- [1] Ergun Akleman, S Liu, and Donald House. Barycentric shaders: art directed shading using control images. In *Proceedings of Expressive'2016*, pages 39–49. Eurographics Association, 2016.
- [2] Ergun Akleman, S Liu, and Donald House. **Barycentric shaders: Art directed shading using control images.** In *Proceedings of Expressive'2016*, pages 39–49, Lizbon, Portugal, 2016. Eurographics Association.
- [3] Mitchell Albala. *Landscape Painting : Essential Concepts and Techniques for Plein Air and Studio Practice*. Watson-Guptill Publications, 2009.
- [4] Xiaobo An and Fabio Pellacini. User-controllable color transfer. *Computer Graphics Forum*, 29(2):263–271, 2010.
- [5] Adrien Bousseau, James P O’shea, Frédo Durand, Ravi Ramamoorthi, and Maneesh Agrawala. Gloss perception in painterly and cartoon rendering. *ACM Transactions on Graphics (TOG)*, 32(2):18, 2013.
- [6] Robert L Cook. Shade trees. *ACM Siggraph Computer Graphics*, 18(3):223–231, 1984.
- [7] 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.
- [8] Yuxiao Du and Ergun Akleman. Charcoal rendering and shading with reflections. In *ACM SIGGRAPH 2016 Posters*, SIGGRAPH ’16, pages 32:1–32:2. ACM, 2016.
- [9] Yuxiao Du and Ergun Akleman. Charcoal rendering and shading with reflections. In *ACM SIGGRAPH 2016 Posters*, page 32. ACM, 2016.

- [10] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A non-photorealistic lighting model for automatic technical illustration. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 447–452. ACM, 1998.
- [11] Bruce Gooch and Amy Gooch. *Non-photorealistic rendering*. AK Peters, Ltd., 2001.
- [12] 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.
- [13] Aaron Hertzmann and Denis Zorin. Illustrating smooth surfaces. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 517–526. ACM Press/Addison-Wesley Publishing Co., 2000.
- [14] Robert D Kalnins, Lee Markosian, Barbara J Meier, Michael A Kowalski, Joseph C Lee, Philip L Davidson, Matthew Webb, John F Hughes, and Adam Finkelstein. Wysiwyg npr: Drawing strokes directly on 3d models. *ACM Transactions on Graphics (TOG)*, 21(3):755–762, 2002.
- [15] Adam Lake, Carl Marshall, Mark Harris, and Marc Blackstein. Stylized rendering techniques for scalable real-time 3d animation. In *Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 13–20. ACM, 2000.
- [16] 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, 2013.
- [17] T. Lindemeier, M. Spicker, and O. Deussen. Artistic composition for painterly rendering. In *Proceedings of the Conference on Vision, Modeling and Visualization*, VMV '16, pages 119–126, Goslar Germany, Germany, 2016. Eurographics Association.

- [18] 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. ACM Press/Addison-Wesley Publishing Co., 1997.
- [19] Siran Liu and Ergun Akleman. Chinese ink and brush painting with reflections. In *SIGGRAPH 2015: Studio*, SIGGRAPH ’15, pages 8:1–8:1, New York, NY, USA, 2015. ACM.
- [20] Aditi Majumder and M Gopi. Real time charcoal rendering using contrast enhancement operators. In *Proceedings of Symposium of Non Photorealistic Animation and Rendering*, 2002.
- [21] 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.
- [22] Barbara J Meier. Painterly rendering for animation. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 477–484. ACM, 1996.
- [23] Robert Olson. *Art direction for film and video*. CRC Press, 1998.
- [24] Emil Praun, Hugues Hoppe, Matthew Webb, and Adam Finkelstein. Real-time hatching. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, page 581. ACM, 2001.
- [25] Cristina Radish. Director john kahrs talks paperman, how the idea for the short came about, the lack of dialogue, blending traditional and cg animation and more, Nov 2012.
- [26] Yuta Tsunematsu, Norihiko Kawai, Tomokazu Sato, and Naokazu Yokoya. Texture transfer based on energy minimization for painterly rendering. *Journal of Information Processing*, 24(6):897–907, 2016.

- [27] David Vanderhaeghe and John Collomosse. Stroke based painterly rendering. In *Image and Video-Based Artistic Stylisation*, pages 3–21. Springer, 2013.
- [28] 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.
- [29] Brian Whited, Eric Daniels, Michael Kaschalk, Patrick Osborne, and Kyle Odermatt. Computer-assisted animation of line and paint in disney’s paperman. In *ACM SIGGRAPH 2012 Talks*, page 19. ACM, 2012.
- [30] Georges Winkenbach and David H Salesin. Computer-generated pen-and-ink illustration. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 91–100. ACM, 1994.
- [31] Lu Xiao, Xiao-Xuan Huang, and Yi-Chang Lu. Non-photorealistic rendering from real video sequences with discontinuity reduction using fast video segmentation. In *SoC Design Conference (ISOCC), 2016 International*, pages 327–328. IEEE, 2016.