

A Process to Create Dynamic Landscape Paintings Using Barycentric Shading with Control Paintings

Matthew Justice

Department of Visualization, Texas A&M University
College Station, Texas 77831
mattscottjustice@tamu.edu

Ergun Akleman

Departments of Visualization & Computer Science and
Engineering, Texas A&M University
College Station, Texas 77831
ergun.akleman@gmail.com



Figure 1: An example of artistically controlled shadows and highlights rendered as different times of the day.

ABSTRACT

In this work, we present 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. 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.

CCS CONCEPTS

- Computing methodologies → Non-Photorealistic Rendering;

KEYWORDS

Rendering, Non-Photorealistic Rendering, Western Painting

ACM Reference format:

Matthew Justice and Ergun Akleman . 2018. A Process to Create Dynamic Landscape Paintings Using Barycentric Shading with Control Paintings. In *Proceedings of SIGGRAPH '18 Posters, Vancouver, BC, Canada, August 12 - August 16, 2018*, 2 pages.
<https://doi.org/10.1145/3230744.3230803>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGGRAPH '18 Posters, August 12 - August 16, 2018, Vancouver, BC, Canada

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5015-0/17/07...\$15.00

<https://doi.org/10.1145/3230744.3230803>

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. 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 [Akleman et al. 2016] for formal description) with two variables s and t , where s represents 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 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 2a, 2b, 2c, and 2d 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 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 2. We only need

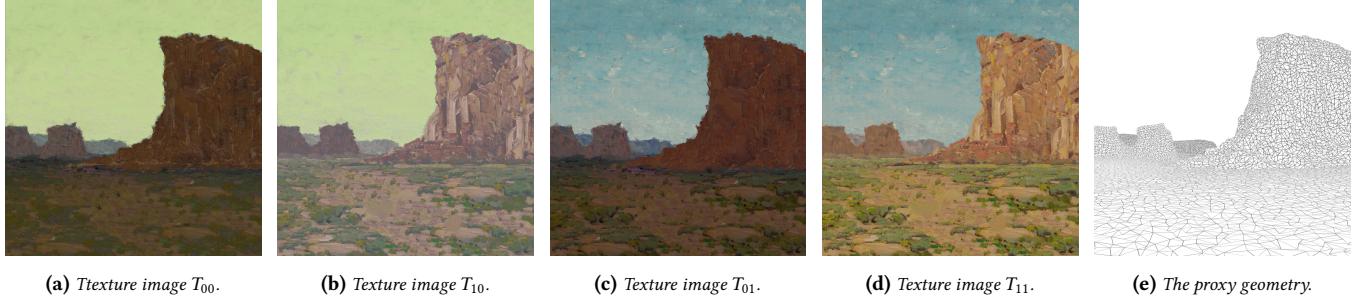


Figure 2: Four texture images T_{00} , T_{10} , T_{01} , and T_{11} and the proxy geometry that is used to compute illumination and shading.

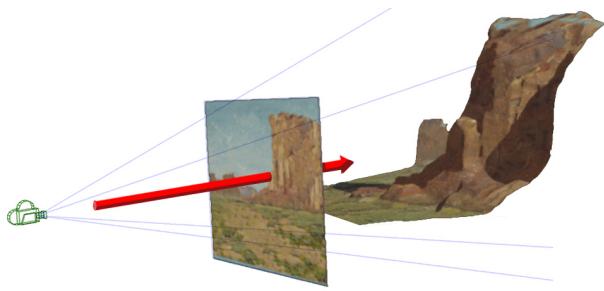


Figure 3: Projection of painting to proxy geometry.

sun's position along with some geometry information to compute the value of s per pixel u and v .

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.

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 [An and Pellacini 2010]. 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 2a and 2b 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 2c and 2d again using the color transfer. This provides us four "un-illuminated" control paintings that can be interpolated to obtain final paintings.

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

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

The next stage is to provide an animation of the light that represents Sun. We simply need the position of the light to make decision about shading and illumination. Illumination is more critical to obtain 3D looking results since it can clearly provide occlusion information through shadows. In this stage, the key is the computation of s value. Our shader to compute s is written in Open Shading Language and being rendered in Maya using the Arnold renderer. 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 3D look comes from ray traced shadows. This is simple with Open Shading Language and Arnold renderer. 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:

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

As demonstrated in Figures 1, this simple process allows to create 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.

REFERENCES

- Ergun Akleman, S Liu, and Donald House. 2016. Barycentric shaders: Art directed shading using control images. In *Proceedings of Expressive'2016*. Eurographics Association, Lisbon, Portugal, 39–49.
 Xiaobo An and Fabio Pellacini. 2010. User-Controllable Color Transfer. *Computer Graphics Forum* 29, 2 (2010), 263–271.