APPROVAL SHEET

Title of Thesis: Real-time Realistic Rendering of Sunrise and Sunset on the Ocean

Name of Candidate: Yuping Zhang
Master of Science, 2015

Thesis and Abstract Approved: ________________________
Dr. Marc Olano
Associate Professor
Department of Computer Science and
Electrical Engineering

Date Approved: ______________________________

Curriculum Vitae

Name: Yuping Zhang.

Permanent Address: 4816 Grand Bend Dr, Catonsville, MD 21228

Degree and date to be conferred: M.S. of Computer Science, Feb 2015

Collegiate institutions attended:

University of Maryland Baltimore County, M.S. of Computer Science, 2015.

Major: Computer Science.

Professional publications:

Rendering Point Clouds with Feature Textures, Yuping Zhang, Marc Olano, Jonathan P. Dandois, Jian Chen, IEEE Visualization 2013.

Professional positions held:

Graduate Research Assistant, UMBC, Baltimore, MD. (Jan 2013 – Dec 2013).

ABSTRACT

Title of Thesis: Real-time Realistic Rendering of Sunrise and Sunset on the Ocean

Yuping Zhang, Master of Science, 2015

Thesis directed by: Dr. Marc Olano, Associate Professor
Department of Computer Science and
Electrical Engineering

The aim of this thesis is to study advanced real-time realistic rendering techniques for outdoor natural scenes. Much research has been done for rendering realistic outdoor scenes, such as sky, ocean, terrain, etc. But sunrise and sunset are rarely discussed. Interesting phenomena like the sun mirages during sunset and sunrise could create splendid visual effects.

This thesis presents a method to render sun mirages. It uses precomputed atmospheric refraction profiles for different atmosphere condition and performes ray tracing to render the sun deformation. Combined with other methods, it renders sunrise and sunset on the ocean in real time. It simulates realistic sun mirages, sky color, ocean waves and lighting, which can be useful for realistic scenes in movies, video games or for scientific simulators.

Real-time Realistic Rendering of Sunrise and Sunset on the Ocean

by

Yuping Zhang

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland, Baltimore County in partial fulfillment
of the requirements for the degree of
Master of Science
2015

© Copyright Yuping Zhang 2015

To Wendy when she was a littile girl.

ACKNOWLEDGMENTS

I would like to thank everyone who has supported me during the work with my thesis. I'm grateful to my advisor Dr. Marc Olano, for helping me shooting problems during planning the thesis and implementing the algorithms. I am thankful to my committee and all my friends in the VANGOGH lab and DAVINCI lab, for their time to share their knowledge with me. Finally I would like to thank my parents for always supporting me.

TABLE OF CONTENTS

DEDICATION . . . . **ii**

ACKNOWLEDGMENTS . . . . **iii**

LIST OF FIGURES . . . . **vi**

Chapter 1 INTRODUCTION **1**

1.1 Local illumination and global illumination 2
1.2 Physically based shading 4
1.3 Sunset and sunrise 5
1.4 Contribution 5

Chapter 2 BACKGROUND AND RELATED WORK **7**

2.1 Atmospheric scattering 7
2.2 Sky dome radiance 10
2.3 Atmospheric refraction and mirages 13
2.4 Ocean modeling and lighting 20
2.5 Tone mapping 21

Chapter 3 APPROACH **23**

3.1 The skylight model 23

3.1.1 Hosek-Wilkie skylight model 23

3.1.2 Implementation 24

3.2 Flattened Sun and mirages 26

3.2.1 Refraction calculation 26

3.2.2 Implementation 32

3.3 Ocean modeling, lighting and animation 32

3.3.1 Ocean modeling 33

3.3.2 Ocean lighting 36

3.3.3 Implementation 37

Chapter 4 RESULTS 39

4.1 Visual quility 39

4.2 Performance 43

4.3 Limitations 43

Chapter 5 CONCLUSION 44

LIST OF FIGURES

1.1 Images Rendered Using Parthenon, a GPU-Accelerated Renderer [Hachisuka, 2005] 3

1.2 A Scene with a GPU-Accelerated Radiosity Solution [Coombe & Harris, 2005] 3

2.1 **Ray path.** Ray path is shorter at noon and much longer during sunset and sunrise. 8

2.2 **Rayleigh scattering and Mie scattering.** Rayleigh scattering is caused by particles that much smaller than the wavelength. Mie scattering is caused by particles whose size is comparable to the wavelength. Rayleigh scattering is strongly wavelength dependent while Mie scattering is not. 9

2.3 **Flattened Sun.** Summer sunset image taken by [Cowley]. Atmosphere refraction lead to a flattened sun during sunset. (shown with permission) . . 13

2.4 **Sunrise mirages.** Sunrise mirage sequence captured over Casco Bay, Cape Elizabeth, Maine by John Stetson. (shown with permission) 14

2.5 **Omega Sunset at Cape Hatteras, North Carolina by Myers.** A classic Etruscan vase sunset as he looked westwards over Pamlico Sound towards the mainland. (shown with permission) 15

2.6 **Mock Sunset at Poinite de la Torche, France by Laveder.** The faint lines have been added to indicate layers of different temperature layers. The slices show descending erect images of the solar disk and ascending inverted images. (Images ®Laurent Laveder, shown with permission) . . . 16

2.7 **Simple Ray bending.** A simple ray bending leads to a flattened sun. 16

2.8 **Ray path of the Omega mirage.** The warm air layer above the Earth surface changes the ray direction dramatically. 17

2.9 **Refraction vs. Altitude by Bennett's formula** 19

3.1 An average temperature profile through the lower layers of the atmosphere. [Service, 2014] 28

3.2 **Standard Atmosphere.** True altitude vs. apparent altitude using the Bennett formula. 30

3.3 **Inferior Mirage temperature profile.** Temperature with a sharp increase near the sea level. Observe at 5m, lapse rate above 6.5m is 0.0065 K/m, between 6.5m and 5.5m is 0.4 K/m, below 5.5m is 1.3 K/m. 31

3.4 **Ray tracing.** In fragment shader, check if each ray hits the sun. 32

3.5 **Ocean Model.** Transitions from geometry to BRDF, filtered with a wavelength dependent weight. [Bruneton, Neyret, & Holzschuch, 2010] 34

4.1 **Sunset in a clean sky.** Sky turbidity as 2.0. When the sun goes down, the sky color changes from blue, voilet to red. As the scene turns dark, the reflection of the sun on the ocean turns to yellow/orange. 40

4.2 **Omega sun mirage.** Image sequence shows the sun deformation when it is close to the horizon using the inferior temperature profile. The sun intensity is decreased by a factor of the solar elevation. 42

Chapter 1

INTRODUCTION

Real-time realistic rendering is one of the vital topics of 3D computer graphics. It has been widely used in computer and video games, simulators, movies, etc. Realistic rendering usually uses physically based shading, including modeling the geometry of objects, simulating light effects and creating animation. Real-time rendering focuses on efficiently rendering complex objects or scenes. Methods like hierarchical representations and level-of-detail approaches are used to improve the performance.

Creating a realistic scene with visual complexity has many challenges. These can be divided into geometric or material complexity and lighting complexity. Geometric complexity is modeling a complex object with information such as meshes to describe the surface details. There are shading methods, texture mapping, bump mapping, etc. to describe the surface without complex geometry. Many materials can be modeled by the standard methods, but some need to be considered separately for their special properties. This is why many BRDFs (Bidirectional Reflectance Distribution Functions) are used for different materials. Modeling for more complicated materials such as skin and cloth have even become independent research topics.

Lighting is another essential part in realistic rendering. Lighting effects can be computed by local illumination or global illumination methods. Local illumination describes the effect of direct light. Global illumination describes how light interacts with objects and scene including shadows, inter-reflections, etc.

1.1 Local illumination and global illumination

Local illumination describes the effect when light sources shine directly onto the objects. The color is determined by the light sources and the objects themselves. It is hard to get a realistic scene with recognizable materials by only implementing local illumination. Global illumination dramatically improves the realism because it also takes indirect light into account. Given inputs of the geometry, the material and lighting, the rendering equation calculates the outgoing light at a particular point and to a certain direction as the sum of the emitted light and the reflected light. The reflected light is the integral of the incoming light from all directions, multiplied by the surface reflection and incoming angle. At time t, the rendering equation for a fixed wavelength is as follows:

$$L_0(x, \omega_0) = L_e(x, \omega_0) + \int_\Omega f_r(x, \omega_i, \omega_0) L_i(x, \omega_i)(\omega_i \cdot n) d\omega_i \tag{1.1}$$

where L_0 is the total spectral radiance directed outward along direction ω_0 from a particular position x, $L_e(\omega_0, t)$ is emitted spectral radiance, Ω is the unit hemisphere containing all possible values for ω_i, f_r is the bidirectional reflectance distribution function, the proportion of light reflected from ω_i to ω_0 at position x. L_i is spectral radiance coming inward toward x from direction ω_i. $\omega_i \cdot n$ is the weakening factor of inward irradiance due to incident angle, as the light flux is smeared across a surface whose area is larger than the projected area perpendicular to the ray.

There are many algorithms being used in global illumination, such as radiosity, ray tracing, path tracing, ambient occlusion, photon mapping, image based lighting, etc. These algorithms usually involve costly computation, but with powerful graphics hardware, people have been able to adapt CPU methods to the GPU to perform those computations in real time. Figures 1.1 and 1.2 show the result of some global illumination algorithms. Some soft shadows and color bleeding effects can be seen in those images.

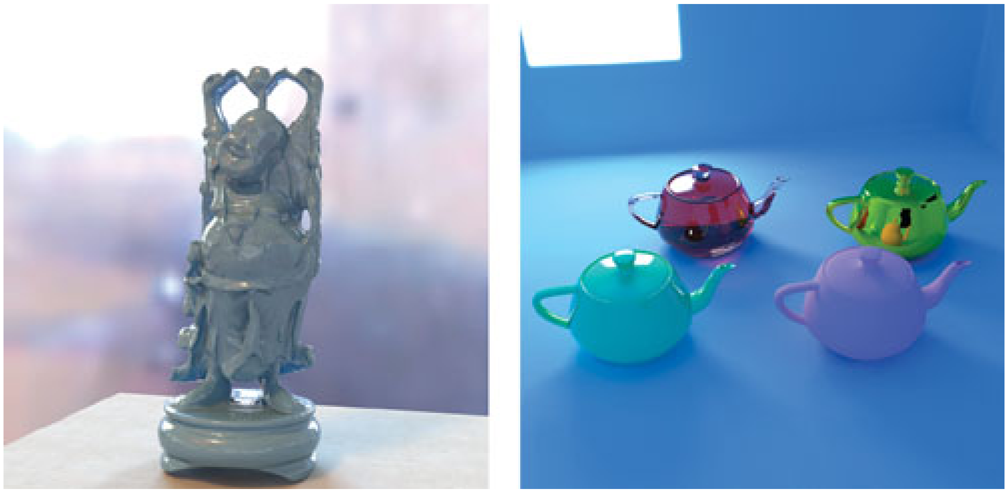

FIG. **1.1:** *Images Rendered Using Parthenon, a GPU-Accelerated Renderer [Hachisuka, 2005]*

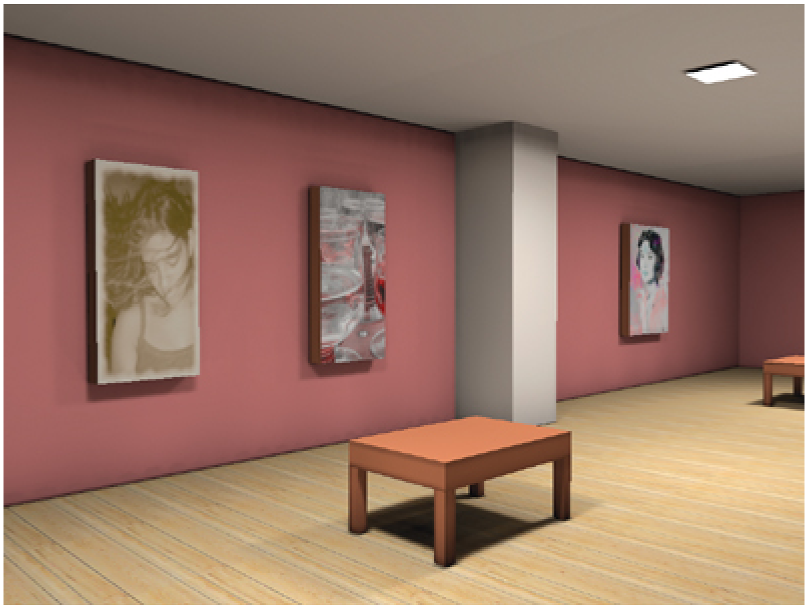

FIG. **1.2:** *A Scene with a GPU-Accelerated Radiosity Solution [Coombe & Harris, 2005]*

1.2 Physically based shading

Physically based shading describes how light interacts with materials based on physical rules. It enforces plausible parameters to model materials. The shading model would use complex BRDFs, Fresnel, normalization of specular highlights, energy conservation etc. for an accurate result.

The BRDF defines how much light reflected from the incoming direction to the outgong direction. At the position x, with a fixed wavelength, BRDF is defined by:

$$f_r(\omega_i, \omega_r) = \frac{dL_r(\omega_r)}{dE_i(\omega_i)} = \frac{dL_r}{L_i(\omega i)\cos\theta_i d\omega_i} \tag{1.2}$$

where L is radiance, E is irradiance, and θ_i is the angle between ω_i and the surface normal n.

Physically based BRDFs have some additional properties. First, the BRDF should be positive: $f_r(\omega_i, \omega_r) \geq 0$. Second, it should obey Helmholtz reciprocity, meaning the light reflected should be the same if the incoming direction and outgoing direction are reversed: $f_r(\omega_i, \omega_r) = f_r(\omega_r, \omega_i)$. Last but not least, total energy should be conserved so there won't be new energy created: $\forall\omega_i, \int_\Omega f_r(\omega_i, \omega_r)\cos\theta_r \mathrm{d}\omega_r \leq 0$.

The fresnel term is an important part of the BRDF. The fresnel function describes the transmissivity when light travels through media of differing refractive indices. When reflection and refraction happen at the same time, the Fresnel term is the specular reflection coefficient that describes how much light will be reflected. The formula is:

$$F = \frac{1}{2}\left(\frac{n_1\cos\theta_i - n_2\cos\theta_t}{n_1\cos\theta_i + n_2\cos\theta_t}\right)^2 + \frac{1}{2}\left(\frac{n_1\cos\theta_t - n_2\cos\theta_i}{n_1\cos\theta_t + n_2\cos\theta_i}\right)^2 \tag{1.3}$$

where n_1, n_2 are refractive indices, θ_i, θ_t are incident angle and refracted angle. Schlick's approximation is a common way to approximate the BRDF model and the Fresnel factor

Schlick [1994]. According to Schlick's model, the Fresnel equation can be approximated by:

$$F_\lambda(u) = f_r + (1 - f_\lambda)(1 - u)^5, f_\lambda = (\frac{n_1 - n_2}{n_1 + n_2})^2 \tag{1.4}$$

where u is $\cos(\theta)$, θ is the angle between the viewing direction and the half-angle direction. f_λ is the reflection coefficient for light incoming parallel to the normal. n_1, n_2 are indices of refraction of the two media.

1.3 Sunset and sunrise

Many previous works have been done to create realistic sky lighting. However, most of them focused on day and night sky. During sunset and sunrise, there are many interesting atmospheric phenomena such as sun mirages and green flashes. Those phenomena happen as results of atmospheric refraction, which is influenced by different temperature and pressure profiles. Adding those atmospheric effects can greatly increase the realism of scenes. It can be used in simulations for astronomical study, or for realistic rendering in movies and video games. Most previous works on rendering outdoor scenes do not cover the sun mirage or simply add some perturbation, which cannot give an accurate result. I want to introduce a way to simulate them based on precomputed atmosphere profiles.

1.4 Contribution

I present rendering of sunrise/sunset on the ocean, which is aimed for a real-time realistic rendering for accurate atmospheric optics phenomena. My main contribution is adding the sun mirage simulation and integrated the sun, sky and ocean model together. The simulation follows the physical rules to calculate the atmosphere scattering and refraction. It considered many the influential factors, including turbidity, albedo, temperature and atmosphere pressure. It can easily generate different atmosphere profiles under a wide range of air conditions for any desired result.

This thesis is organized as follows:

Chapter 2 provides a review of the background and some of the related research.

Chapter 3 describes my approaches to the different part of the scene.

Chapter 4 presents and evaluates the result.

Chapter 5 summarizes my project and discusses the future work.

Chapter 2

BACKGROUND AND RELATED WORK

In this section, I want to briefly introduce some background and related works. The content is divided into five parts, atmospheric scattering, sky dome radiance, sun flattening and mirages, ocean waves modeling and lighting, and tone mapping. To avoid ambiguity, sunrise and sunset in this thesis refer to the entire process of the sun crossing the horizon and its accompanying atmospheric effects.

2.1 Atmospheric scattering

The visible sunlight consists of a spectrum of colors from violet-blue to orange-red. The wavelengths vary from 0.47um to 0.64um. There are two types of scattering: Rayleigh scattering and Mie scattering. Rayleigh scattering happens when particles are significantly smaller than the wavelength of the radiation. Mie scattering happens when the particle size is comparable to the wavelength of visible light. In a clean sky, most air particles are much smaller than the wavelength. That is when Rayleigh scattering happens. The atmosphere scatters shorter wavelengths of visible light (violet and blue) more effectively than the longer wavelengths (orange and red). When we look up to the sky during daytime, more of the violet-blue light reaches to our eyes (Figure 2.1). Since human eyes are more sensitive to blue than violet, the sky looks blue in a cloudless day. During the sunrise/sunset, sunlight travels much longer to reach our eyes. Most visible light of shorter wavelength has been scattered away in the middle of the atmosphere, leaving the longer wavelengths in the end. That is why we see more orange and red color during the sunrise and sunset. When haze rises, small particles are suspended in the air and Mie scattering happens. So we see a whitish sky in a cloudy day.

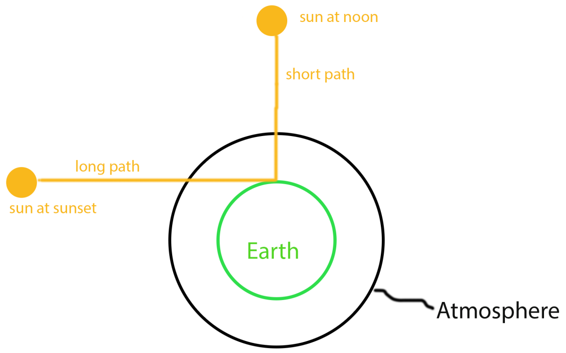

FIG. 2.1: ***Ray path.*** *Ray path is shorter at noon and much longer during sunset and sunrise.*

Rayleigh scattering and Mie scattering are solutions to Maxwell's equations, which describe the scattering of electromagnetic radiation by a sphere. Based on the electromagnetic theory, the atmosphere scattering and refraction are all due to the electrical polarization of the scatterer by the incident electromagnetic wave.

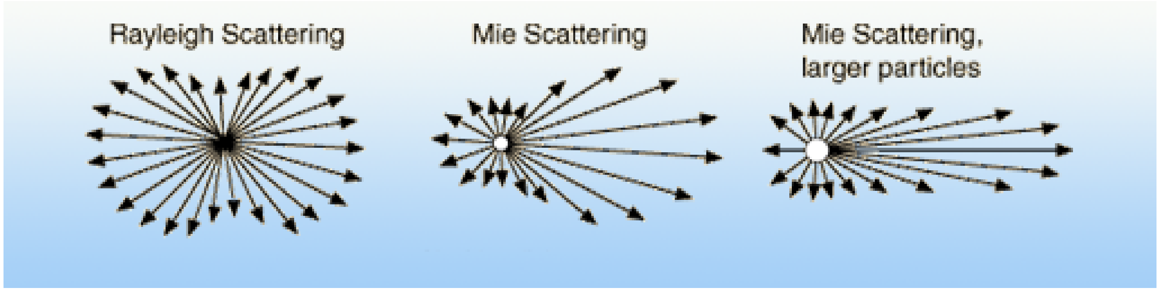

FIG. **2.2:** ***Rayleigh scattering and Mie scattering.*** *Rayleigh scattering is caused by particles that much smaller than the wavelength. Mie scattering is caused by particles whose size is comparable to the wavelength. Rayleigh scattering is strongly wavelength dependent while Mie scattering is not.*

Rayleigh scattering describes the scattering of light by spheres much smaller than the light wavelength. The macroscopic cross-section of Rayleigh scattering can be described by:

$$S_R macro = \frac{128\pi^5\alpha_0^2(6+3\rho_n)}{3\lambda^4(6-7\rho_n)} \tag{2.1}$$

The intensity of the scattered radiation is given by

$$I = I_0(\frac{1+\cos^2\theta}{2R^2})(\frac{2\pi}{\lambda})^4(\frac{n^2-1}{n^2+2})^2(\frac{d}{2})^6 \tag{2.2}$$

where I_0 is the light intensity before the interaction with the particle, R is the distance between the particle and the observer, θ is the scattering angle, n is the refractive index of the particle, and d is the diameter of the particle. Rayleigh scattering is elastic, meaning the total energy of the incident light will not be lost. However, the ray will be recast in new directions, as showed in Figure 2.2. The phase function describes how much light is scattered toward any certain direction. A phase function is modeled using the Henyey-

Greenstein analytic approximation [Cornette & Shanks, 1992]:

$$F_M(\theta) = \frac{3(1-g^2)}{2(2+g^2)} \cdot \frac{1 + cos^2\theta}{(1 + g^2 - 2g \cdot \cos\theta)^{\frac{3}{2}}} \tag{2.3}$$

where θ is the angle between incoming and outgoing light directions, g is an anisotropy factor that affects the symmetry of the scattering. Rayleigh scattering is almost symmetric, so g can be set to 0. The equation will become:

$$F_R(\theta) = \frac{3}{4}(1 + \cos^2\theta) \tag{2.4}$$

Mie scattering happens when the size of the particles is comparable to the wavelength, which can be dust or water droplets in atmosphere. To use the phase function for Mie scattering, depends on the droplet size. Value g are usually set between -0.75 and -0.999. Mie scattering's cross-section and intensity are not easy to describe by formulas. An infinite expansion series was presented in the book by Hulst & Van De Hulst [1957].

2.2 Sky dome radiance

Sunset and sunrise create unique atmospheric optical phenomena. Everyone has seen marvelous sunrise or sunset at one time and been amazed by the beautiful colors, however, it is hard to render them in real time. Many realistic rendering techniques have been developed to create a realistic sky model taking atmosphere scattering into consideration. Nishita *et al.* [1993] provided the very first atmospheric scattering algorithm to render colored sky. He considered single scattering in his model, but ignored the inter-reflections between the ground and the air. Then he added multiple scattering simulations in a revised model [Nishita, Dobashi, & Nakamae, 1996] using discrete pre-computed table values. Preetham, Shirley, & Smits [1999] presented a model for daylight which has been widely used since then. It is based on the model of Perez, Seals, & Michalsky [1993]. The formula is:

$$F_{Perez}(\theta, \gamma) = (1 + Ae^{\frac{B}{\cos\theta}})(1 + Ce^{D\gamma} + E\cos^2\gamma) \tag{2.5}$$

where the parameters A to E are used to tune the luminance distribution. The Perez model has tabulated values for the parameters, the Preetham model instead computed them analytically. First, they generate reference images by Nishita's model, then fit the Perez formulas to the reference images.

Hoffman & Preetham [2002] presented a rendering outdoor scattering model. They simplified Nishita's equations so that it can be implemented into a GPU shader. The assumptions include the viewer is close to the ground and atmosphere has a constant density at all altitudes.

Haber, Magnor, & Seidel [2005] provided a physically based model that can realistically simulate various atmospheric conditions for the twilight phenomena. The basic idea is dividing the atmosphere into blocks and computing the radiative transfer between them.

Another real time rendering model presented by Elek & Kmoch [2010] incorporates planetary atmosphere scattering and also light scattering in water. It considered all the light contributions in the environment, including direct illumination, indirect illumination by atmosphere scattering and interreflections. Their work was based on the methods presented by Bruneton & Neyret [2008], using pre-calculated scattering data as a lookup table in real-time rendering. They extend the previous work to more generalized conditions. Their model can render large aquatic bodies and also work with arbitrary density of atmosphere.

An analytic model for full spectral sky-dome radiance, provided by Hosek & Wilkie [2012] can obtain a genuinely spectral analytical model with separate fitting for different wavelengths. The Hosek-Wilkie model extended the useful range of turbidities then previous work to render the localized aureole. The inputs of the model are turbidity, albedo and sun elevation. The parameter turbidity is in the range of 1 to 10, which measures the cloudiness of the sky. The parameter albedo, the ground albedo in this case, is the reflection coefficient that describes the ratio of reflected radiation from the surface to incident radiation upon it. Radiance Distribution Parameters are computed with a combination of solar elevation, albedo and turbidity. Then the final formula is:

$$F(\theta, \gamma) = (1 + Ae^{\frac{B}{\cos\theta + 0.01}})(C + De^{E\gamma} + F\cos^2\gamma) + G\chi(H, \gamma) + I\cos^{\frac{1}{2}}\theta \tag{2.6}$$

$$\chi(g, \alpha) = \frac{1 + cos^2\alpha}{(1 + g^2 - 2g\cos\alpha)^{\frac{3}{2}}} \tag{2.7}$$

The Preetham model can handle the low turbidities very well, but presents a uniform orange-yellow sky at high turbidities during the sunset. That is because the Preetham formula is limited to a small turbidity range and the linear approximation for the parameters A to E is not accurate enough. Comparing the Preetham model with Hosek-Wilkie model, the Hosek-Wilkie model provides much more precise sky colors under a wide range of conditions. The Hosek-Wilkie model improved the Preetham model in mainly three ways:

First, it extends the support for a wide range of turbidity by adding an anisotropic term $\chi(g, \alpha)$. It is similar to the Mie scattering phase function, which can present more accuracy when the turbidity rises.

Second, it added a term $I\cos^{\frac{1}{2}}\theta$. In real life, at lower solar elevations, the aureole does not extend towards the zenith nearly as much as it does towards the horizon and sides. To capture that, this term is added to produce a smooth gradient around the zenith. Also, it replaced 1 by a variable C, produce the huge spikes in brightness around the sun.

Third, the previous formula in the Preetham model cannot define the radiance near the horizon. Because as the viewing direction neared the horizon, the value $\cos(\theta)$ diverged towards infinity, $\frac{B}{\cos\theta}$ becomes undefined. The Hosek-Wilkie model added a fudge factor to prevent divergence to infinity at zenith.

The Hosek-Wilkie model has been adapted in many recent works since it gives a pretty realistic result. Comparing with the Preetham model, the Hosek-Wilkie model has a deeper blue sky, while the sky color is more pinkish in the Preetham model during the day.

2.3 Atmospheric refraction and mirages

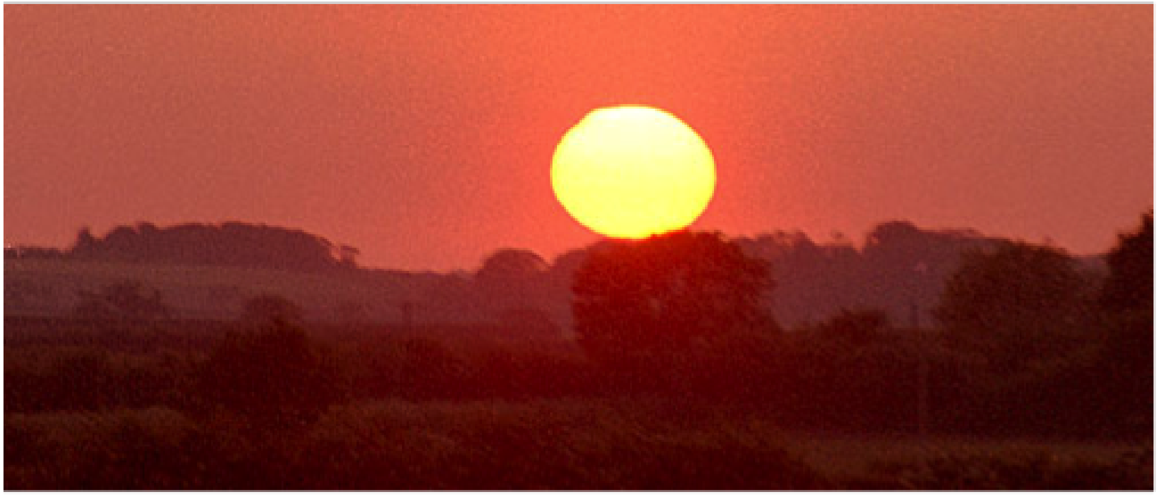

FIG. **2.3:** ***Flattened Sun.*** *Summer sunset image taken by [Cowley]. Atmosphere refraction lead to a flattened sun during sunset. (shown with permission)*

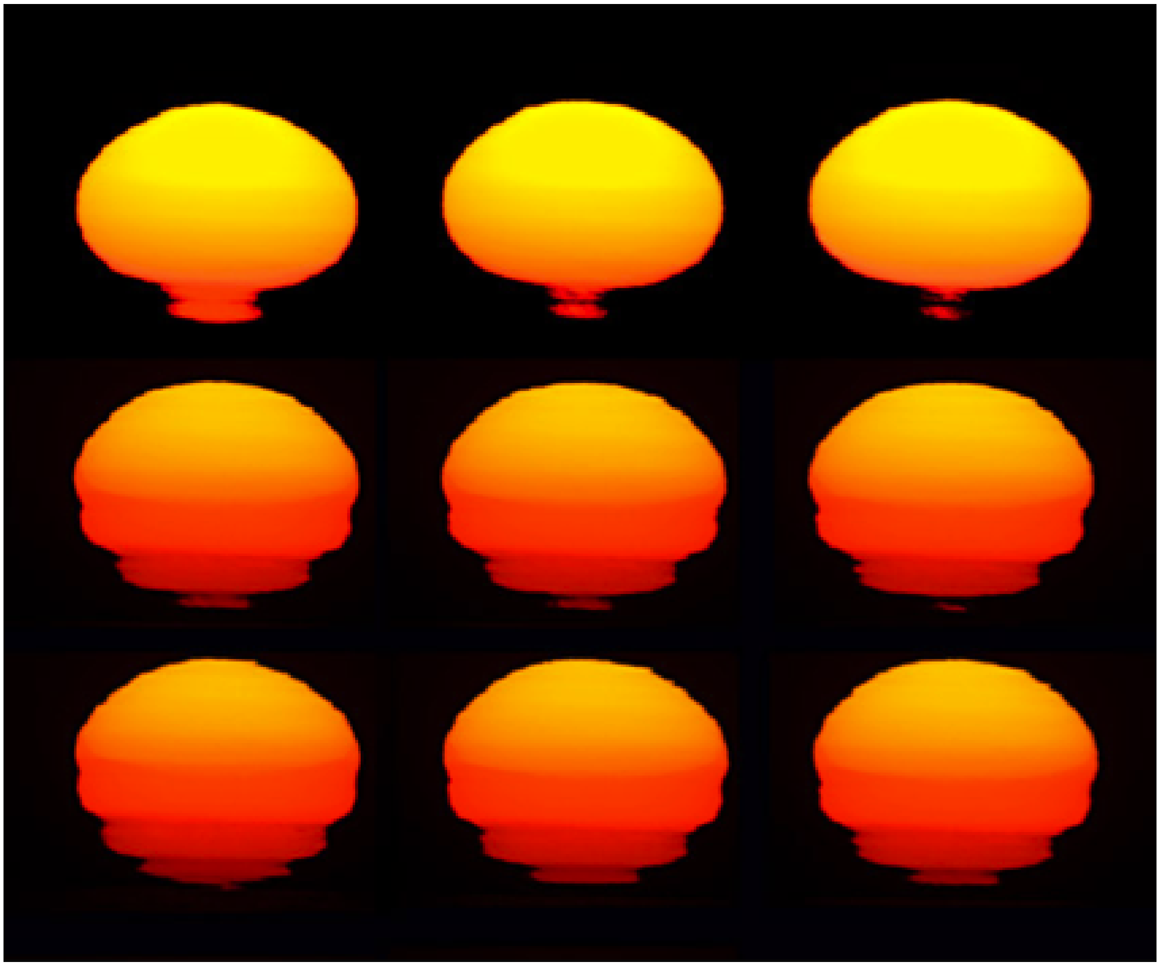

FIG. **2.4:** ***Sunrise mirages.*** *Sunrise mirage sequence captured over Casco Bay, Cape Elizabeth, Maine by John Stetson. (shown with permission)*

During sunset and sunrise, the sunlight travels a much longer path to reach the ground. The light ray goes through dense air and is bent by the atmosphere. The amount of refraction depends on the temperature, pressure and different air conditions. From these, various phenomena of atmospheric optics will be generated, such as sun flattening, sun mirage, free flashes and so on. The air density varies in altitude and the sun heats up the earth's surface, which leads to the atmosphere refraction. The air closer to the ground is warmer and less dense than the layers of air higher in the atmosphere. Usually the air temperature decreases smoothly with an ascending height. Light rays bend towards media with a higher refraction index, meaning higher pressure and lower temperature in this case. That gives us a flattened sun near the horizon. Sometimes the air temperature inversion happens, meaning the warm air overlays colder air, or ocean surface is heated and results in a steep temperature change. The rays will be bent unexpectedly and the sun mirage will occur. The common phenomena are mock mirage and omega mirage.

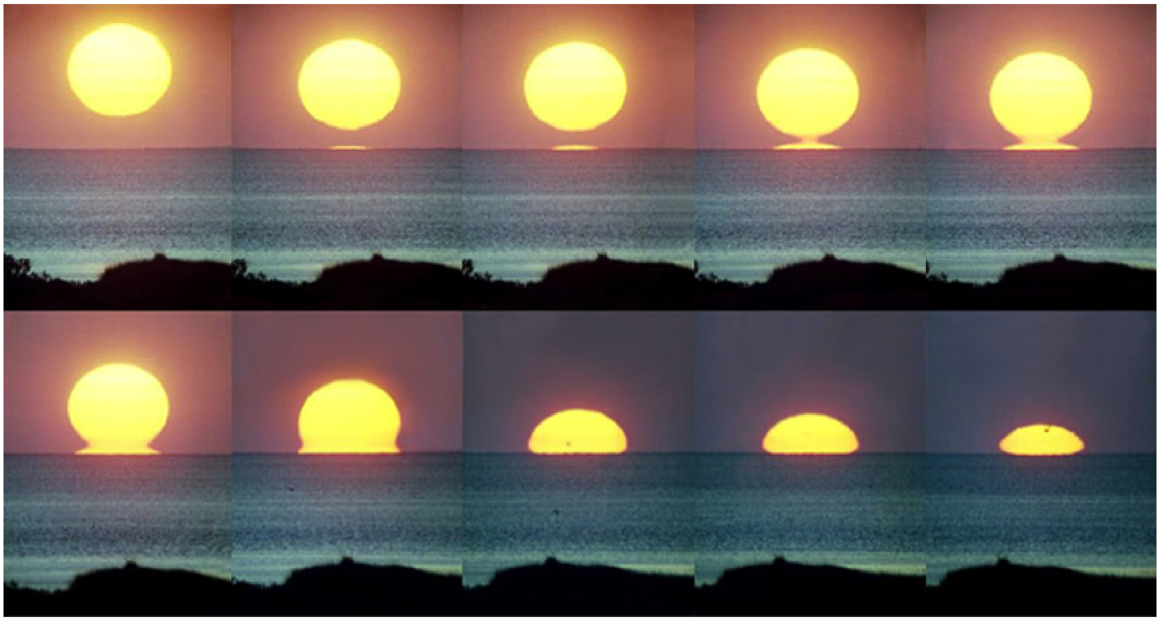

FIG. **2.5:** ***Omega Sunset at Cape Hatteras, North Carolina by Myers.*** *A classic Etruscan vase sunset as he looked westwards over Pamlico Sound towards the mainland. (shown with permission)*

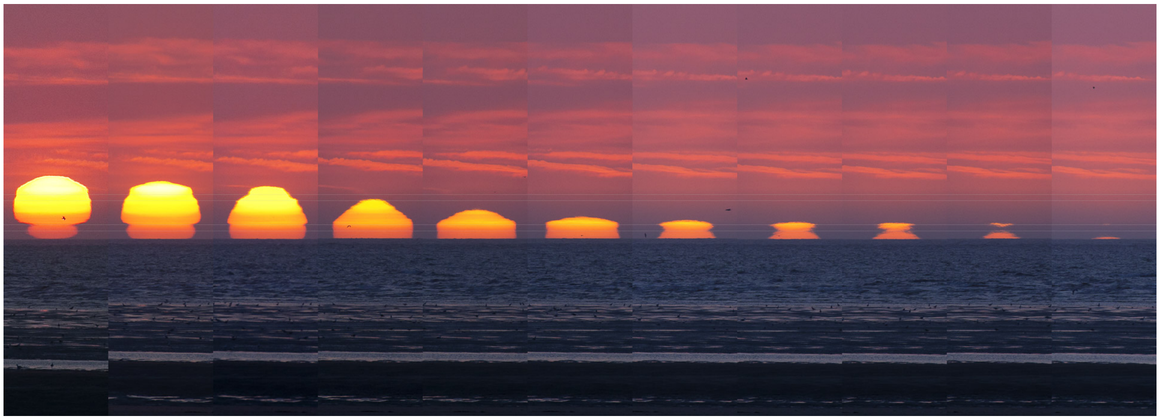

FIG. **2.6:** ***Mock Sunset at Poinite de la Torche, France by Laveder.*** *The faint lines have been added to indicate layers of different temperature layers. The slices show descending erect images of the solar disk and ascending inverted images. (Images ®Laurent Laveder, shown with permission)*

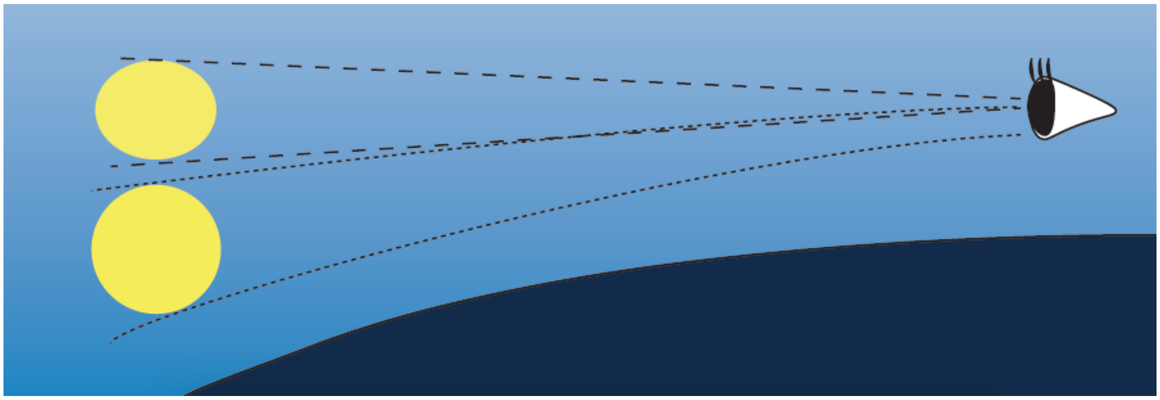

FIG. **2.7:** ***Simple Ray bending.*** *A simple ray bending leads to a flattened sun.*

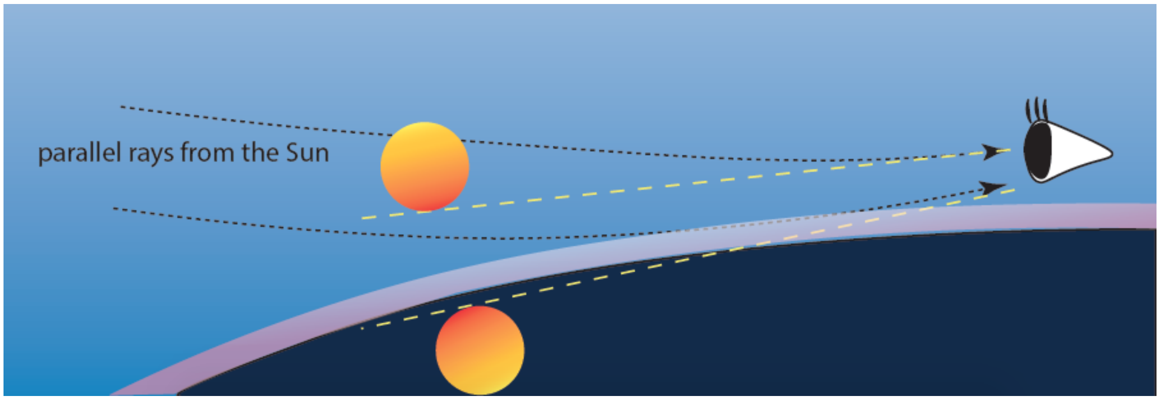

FIG. **2.8:** ***Ray path of the Omega mirage.*** *The warm air layer above the Earth surface changes the ray direction dramatically.*

The inferior mirage is similar to the mirage above a hot road surface. When the sun heats the ground, a layer of hot air of lower density will be generated. Incoming rays with a narrow angle bend into the denser air above. It looks like there is water on the ground reflecting the sky, but it is actually refraction. For a solar inferior mirage, the sun heats up the Earth surface and bends the rays upward in the air. This results in an inverted sun image below the apparent sun. The inferior sun mirage looks like an Omega or Etruscan Vase, so it is also called an Omega sun mirage.

The superior mirage is when the inverted image is overhead. It happens when the cold layers and hot layers of air overlay together near the surface of the Earth. The ray curve is bent even more then the curve of the Earth. Rays cross each other and pass through the atmosphere for a long way.

In more complex conditions, one or more temperature layers lead to a distorted image. The sun rays can be refracted by the different density layers and cross each other. The apparent sun looks like inverted horizontal slices connecting together. That is called a mock mirage.

Sunset and sunrise mirage is an interesting atmospheric optical phenomenon. There are some previous works for simulating the natural phenomena related to heat shimmering and mirage. A physical-based heat transfer model by Zhao *et al.* [2007] took conduction, convection and radiation into consideration, using a temperature texture to show the heat distribution from heat source objects to the ambient flow environment. There is a recent work by Kanazawa, Sakato, & Takahashi [2013] using a pencil tracing technique to generate an appearance of mirage without any complex thermodynamic simulation.

Sunset mirages happen when low slanting rays are unusually refracted between air layers of different temperature. Air temperature usually falls smoothly with height. The ocean surface heated by solar radiation can produce an abnormally hot air layer above it and generate mirage. Inferior mirage is the most common mirage during sunset/sunrise. Generally it lead to the vertical deformation of the sun like the Etruscan vase or a Greek letter omega Ω.

The main difference between sunrise and sunset, according to the research of earth's atmosphere, is the evening air typically contains more large particles than morning air, which makes the sunset color more vivid and brilliant.

Even though the phenomena vary, the calculation is similar. Refraction is defined by Snell's law. It states that the ratio of the sines of incident angle and refracted angle is equal to the ratio of the indices of refraction in the two media, as well as equal to the ratio of phase velocities or wavelengths in the two media. We set the relative index of refraction of the media as n, θ_1 and θ_2, are the incident angle and refracted angle from the surface normal. Then we have: $n \sin\theta_2 = \sin\theta_1$, so $\theta_1 = \arcsin(n \sin\theta_2)$. Near the zenith, the incident angle is very small. Use the small-angle approximation we have $\sin x \approx x$, we can deduce the refraction

$$r = \theta_2 - \theta_1 = \arcsin(n \sin\theta_1) - \theta_1 = n\theta_1 - \theta_1 = (n-1)\theta_1 \tag{2.8}$$

A simple approximation can be acheived by Bennett's empirical formula:

$$r = \frac{P}{101} \frac{283}{273 + T} \cot(h_a + \frac{7.31}{h_a + 4.4}) \tag{2.9}$$

where r is the refraction in arcminutes, h_a is the apparent altitude. The plot of refraction vs. altitude using this formula is as Figure 2.9.

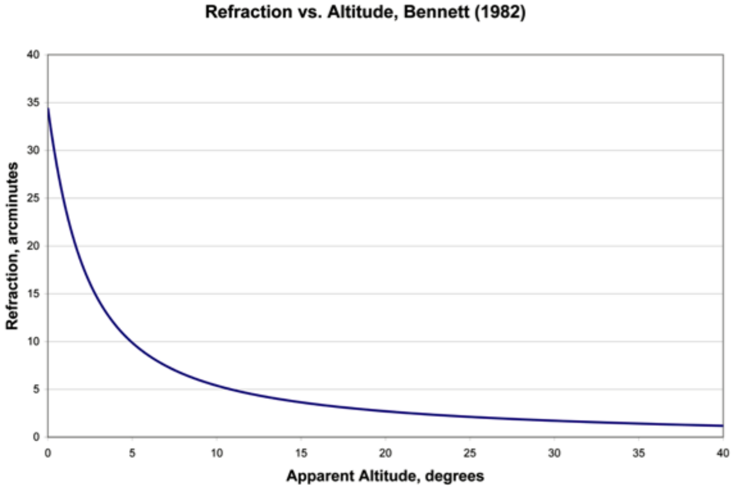

FIG. **2.9:** ***Refraction vs. Altitude by Bennett's formula***

For a more accurate result, especially when the sun is close to the horizon, an integral along the ray path is needed to be computed:

$$R = \int_{1}^{n_0} \frac{\tan(z)}{n} d(n) \tag{2.10}$$

where n_0 is the refractive index at the observer, z is the zenith angle. Auer & Standish [2000] provided a numerical method to solve this integral. To simulate sun mirage, variation of the pressure and temperature near the Earth surface must be modeled. Young [2006] discussed details about how the light ray is bent in different conditions. He simulated several kinds of mirage and green flashes by setting up different temperature variations.

2.4 Ocean modeling and lighting

Realistic animation and rendering of the ocean is a difficult problem because of the wave scales and dynamic lighting. An early ocean wave model was introduced by Fournier & Reeves [1986]. It focused on modeling wave deformation by the disturbing force from the wind and the restoring face from gravity.

One of the standard models, the Choppy Wave Model by [Tessendorf, 2001], considered the horizontal displacement of water surface and produces nice looking waves. It introduced the fundamental principle for realistic simulation of ocean waves, including a spectral method for modeling wave geometry and a wave lighting model.

Jensen *et al.* [2001] introduced a real time level-of-detail deep-water animation scheme. It uses many different proven water models, including an FFT-water algorithm, choppy waves, incompressible viscous fluid (described by the Navier-Stokes Equations), shallow water waves and surface waves.

Bruneton, Neyret, & Holzschuch [2010] presented a method of ocean rendering. It sums up trochoid waves at different scales by their weights and used a multi-resolution meshing scheme to represent a scalable ocean surface. It also modeled whitecaps based on the physical wave breaking. The ocean lighting covers sun reflections, sky reflections and local reflections. It is accurate at all distances due to seamless transitions from geometry to BRDF. Ross & Dion [2007] provided a physically-based anisotropic BRDF based on the surface slope variances, while the slopes and heights follow Gaussian distribution. This BRDF is chosen to be used in Bruneton's model. Then transitioning from geometry to BRDF is determined by a multiresolution reflectance model. I used this method in my work because it can run in real-time and produces highly realistic results.

2.5 Tone mapping

Tone mapping is a method to map the color of a high dynamic range image to something that can be shown in a certain medium, for example a display with restricted dynamic range. It can be useful for images containing strong contrast when the full range of luminance value cannot be displayed. There are many tone mapping operators. They divided into two types: local operators, which apply to each pixel according to the local features, and global operators, which apply the algorithm to every pixel in the image based on the luminance and other global variables.

The basic linear method does exposure adjustment and a gamma correction. It simply takes the input luminance and scales by a scale factor. This factor decides how bright or dark the scene is. Gamma correction is a nonlinear operation to code and decode luminance. It is usually defined by a power-law expression. In computer graphics, it is used for LCD displays for an accurate digital color reproduction. Colors are often stored with a gamma curve since our eyes perceive more data in the blacks and the gamma curve shifts more shades of color in the dark end.

To adjust aperture and contrast, interactive calibration method, also known as interval mapping, find interval based on the mean value of the image and map each pixel to new value by a chosen function [Matkovic & Neumann, 1996]. I used this one because it gives good results converting from the high dynamic range of sky radiance values.

Drago *et al.* [2003] provided a operator which is a good choose when a true tone result. The algorithm intended to simulate the human eye's response. It is based on logarithmic compression of luminance values. It uses the average luminance of the image and a external bias parameter to form a non-linear logarithmic function and apply it per pixel. This method is fast but not helpful to reveal more details in the image.

The operator invented by Durand & Dorsey [2000] has very realistic output. On the contrary to Drago's operator, Durand's operator is computationally heavy but reveals much more detail in the scene.

Reinhard *et al.* [2002] presented useful tone mapping operators. The simplest variation clamps the luminance value to the range from 0 to 1 by mapping x to $\frac{x}{1+x}$. This method also needs gamma correction after tone mapping.

Haarm-Pieter Duiker [Hable, 2010] came up with a filmic tone-mapping technique for games as a pioneer. Hejl and Burgess-Dawson provided an optimized formula to imitate the response curve of a kind of Kodak film [Hable, 2010]. It can produce nice highlights but saturates the black end.

Hable [2010] invented a tone mapping method called Uncharted 2 operator because it is used in the video game Uncharted 2. It has 7 parameters A, B, C, D, E, F and W, respectively are shoulder strength, linear strength, linear angle, toe strength, toe numerator, toe denominator and linear white point value. The linear color is computed by this formula:

$$F(x) = \frac{x(Ax + CB + DE)}{x(Ax + B + DF)} - \frac{E}{F}$$

and the final color is:

$$FinalColor = \frac{F(LinearColor)}{F(LinearWhite)}$$

Chapter 3

APPROACH

To render sunset and sunrise on the ocean, I integrated the sky model and ocean model with my method for the sun mirage and made several changes accordingly. Section 3.1 explains the Hosek-Wilkie sky model, and evaluates its advantages and limitations. Section 3.2 discusses the ocean modeling and lighting method. Section 3.3 describes the methods I used to generate atmosphere profile and render the sun mirages.

3.1 The skylight model

3.1.1 Hosek-Wilkie skylight model

Rendering a sky dome with correct hue is a crucial part of this project. I am using the Hosek-Wilkie skylight model to calculate the sky color [Hosek & Wilkie, 2012]. The Hosek-Wilkie model is an analytic model for full spectral sky-dome radiance. They modified the general approach from Perez, Seals, & Michalsky [1993] and Preetham, Shirley, & Smits [1999]. Their model improves the realism of skylight for sunsets, high atmospheric turbidity conditions and also provides a realistic circumsolar ring around the sun. Their model also takes ground albedo into account and handles each spectral component independently.

The Hosek-Wilkie model has many advantages over the Perez model and the Preetham model. Many improved areas of their formula were discussed in the previous sections. With separately fitting for each wavelength band, this spectral analytical model gives us a more accurate result. Another atmosphere rendering method is provided by [Bruneton & Neyret, 2008]. It considered Rayleigh and Mie multiple scattering and can generate many effects, including the daylight and twilight sky color, and even aerial perspective and light shafts. Their pre-computation is using a heightfield and a reflectance texture, and aimed for complex landform. It is hard to interactively change the turbidity and ground albedo for different air conditions. Since I am rendering ocean, a simple ground albedo is efficient enough to approximate ground contribution. I chose the Hosek-Wilkie model also because I want more accuracy for spectral luminance for various atmosphere conditions.

3.1.2 Implementation

I precomputed the radiance equation parameters A, B, C, $\ldots$ in the equation from the reference data set. I used this reference data directly in my implementation, but I will explain here what is this reference dataset, how the reference table is generated, and then I will talk about how to use them to calculate the parameters.

To create the reference data set, the Hosek-Wilkie model used a brute force path tracer to create reference images first. To generate the scattering events efficiently, they used a method for unbiased generation of free-light paths in a non-homogeneous environment from Raab, Seibert, & Keller [2008]. The free flight path describes how long a ray must travel before scattering. In the case of Rayleigh scattering, it is strongly dependent on wavelength. The phase function describes the distribution of directions in which light is scattered, as discussed the second chapter. The world geometry is simulated with a sphere representing the Earth and another sphere representing the upon boundary of the atmosphere. The camera is placed at 10 meters above the sea level and looking towards zenith. For different wavelength and solar elevation, 20 atmosphere conditions including turbidity in the range of 1 to 10, and albedo equals to 0 or 1, are considered respectively. Since the sky color changes dramatically during sunrise/sunset time but has minimal changes during the day, more samples are taken at lower solar elevations. 46 images were generated with solar elevation from 0 to 90 degree in total and the n-th image was placed at solar elevation $(n/45)^3 * 90$.

After the reference images are generated, the reference data set was fitted to the outputs of the reference path tracer using Levenberg-Marquardt non-linear least-square fitting in MATLAB.

During rendering, the radiance distribution parameters are calculated from the reference data set using a Brezier curve function. For each wavelength, the reference data set provides a 4-dimensional lookup table. For each of the 10 turbidities, two albedo values and nine distribution parameters, a set of six control points is given. These six control points define a quintic Bezier curve. During rendering, solar elevation would be applied to the Bezier polynomial as the interpolation parameter and the final color would be the interpolation along the Bezier curve:

$$\begin{aligned} v_p^\lambda = m^\lambda_{T,\alpha,p,1} \cdot (1-x)^5 + \\ m^\lambda_{T,\alpha,p,2} \cdot 5x(1-x)^4 + \\ m^\lambda_{T,\alpha,p,3} \cdot 10x^2(1-x)^3 + \\ m^\lambda_{T,\alpha,p,4} \cdot 10x^3(1-x)^2 + \\ m^\lambda_{T,\alpha,p,5} \cdot 5x^4(1-x) + \\ m^\lambda_{T,\alpha,p,6} \cdot x^5 \end{aligned} \tag{3.1}$$

By this formula, we can get a result of the 9-element parameter vector and plug them in the luminance equation. For non-integer turbidity and albedo, the parameters can be computed by the arrays of adjacent turbidities and albedos with a linearly interpolation.

This is one step from the final sky color because normalization is needed. In this model, it is calculated by:

$$L_\lambda = F(\theta, \gamma) \cdot L_{M\alpha} \tag{3.2}$$

where $L_{M\alpha}$ is the expected value of spectral radiance in a point randomly picked in the upper hemisphere with uniform distribution. A three-dimensional table is also included in the reference dataset. $L_{M\alpha}$ is calculated using Bezier interpolation too.

A limitation of the HosekWilkie model is they didn't extend the fitting process to the dusk condition. A reference image in their paper shows a solar elevation of -5° has additional dark crescent-shaped region as the Earth shadow. According to the authors, it is too complicated for the fitting process to include this case. So they excluded the situation when the sun goes below the horizon.

3.2 Flattened Sun and mirages

Flattened sun and mirages are phenomena of atmospheric optics that can be seen commonly during sunset and sunrise. The previous chapter explained atmospheric refraction and how it leads to those phenomena. In this section, I will discuss how to compute the corresponding atmospheric refraction given a certain refractivity profile including air pressure, temperature and some physical constant, for creating the flattened sun and mirage phenomenon. My work is based on the analysis of Young [2014] who studied mirage and green flashes from the astronomical perspective.

3.2.1 Refraction calculation

The flattened sun is most obvious at the horizon. When we see the sun disappear during sunset, the actual sun position is about one diameter or more below the horizon.

That is because the atmospheric refraction is $35.4'$ at the horizon and decreases quickly with rising altitude. It's about $18.4'$ at 2 degrees altitude and $9.9'$ at 5 degrees altitude.

The ray that hits the upper edge of the sun is bent less than the ray that hits the bottom edge of the sun, so it looks like being compressed. I modeled the flattened sun using the atmospheric refraction index calculated by the Bennett formula. This model uses the temperature gradient in the U. S. Standard Atmosphere, including a lapse rate (how much temperature decreases with the height).

As mentioned in the previous section, the atmospheric refraction is determined by the pressure and temperature profile. A mirage would need more severe temperature variations. There are thermal energy changes between boundary layers, which influence the whole temperature profile. The temperature through the atmosphere can be viewed as in the Figure 3.1.

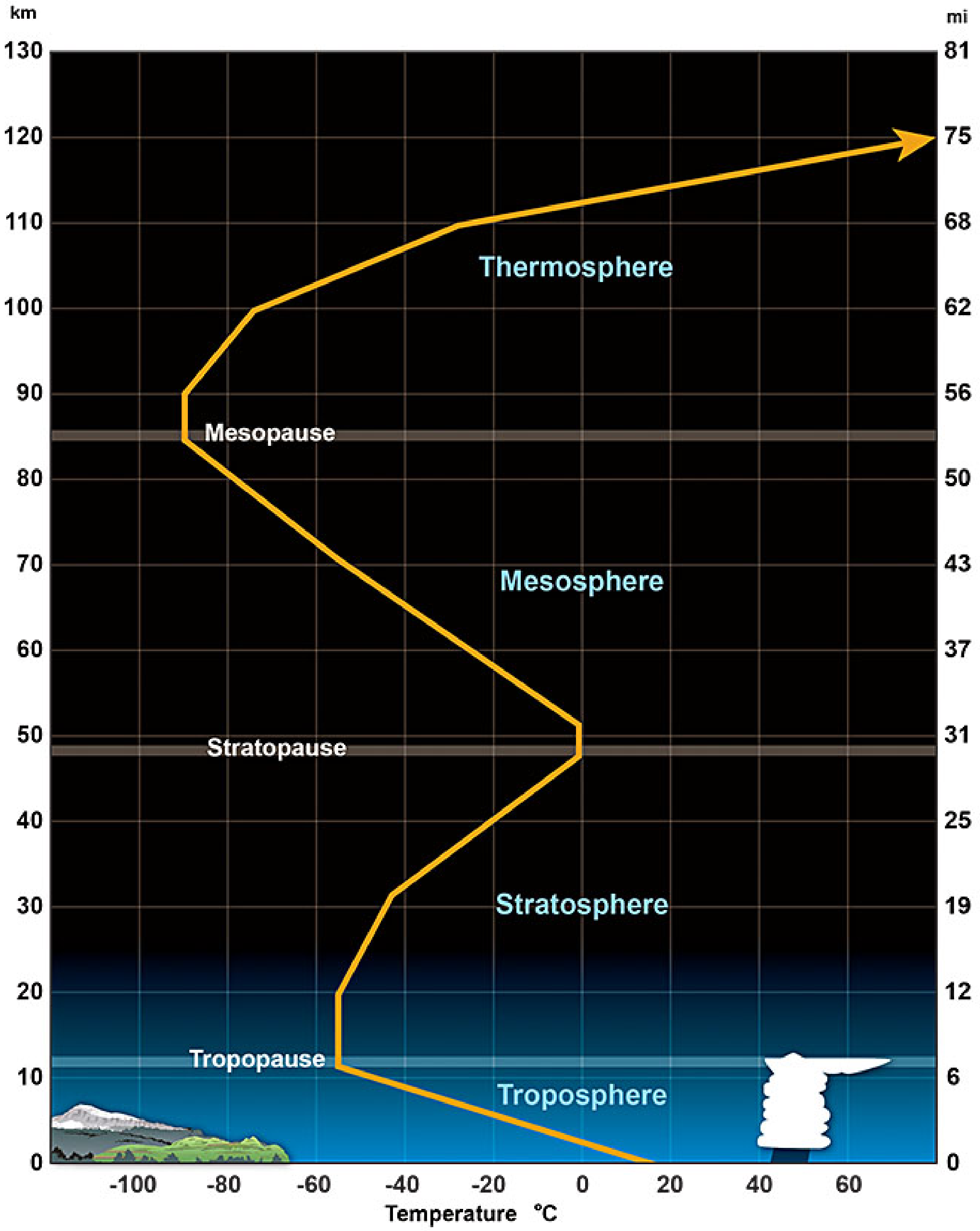

FIG. **3.1:** *An average temperature profile through the lower layers of the atmosphere. [Service, 2014]*

Only part of the stratosphere and troposphere layers need to be considered for atmospheric refraction. Above $11000m$, the atmosphere layers won't impact the refraction anymore. In the stratosphere below $11000m$, the temperature is constant. Within troposphere, temperature increases as altitude increases in a standard condition. A dry lapse rate is $10C^{\circ}/km(5.5F^{\circ}/1,000ft)$ for temperature changes in air not at 100% relative humidity. A wet lapse rate is $5.5C^{\circ}/km(3F^{\circ}/1,000ft)$ for temperature changes in air that is saturated. A proper atmospheric temperature lapse rate should be chosen within this range.

According to Young [2006], the Bennett formula is adequate for an altitude over 30°. Between 5° and 30°, a temperature gradient factor needs to be taken into account. When getting closer to the horizon, numerical integration is required to give more accurate results considering temperature gradient changes with height. Hohenkerk & Sinclair [1985] described an implementation of the numerical method Auer & Standish [2000] provided.

Here I use Bennett's formula with a lapse rate for altitude larger than 4.26° in all models and use the numerical method to pre-compute refraction tables for different atmospheric temperature profile for altitude less than 4.26°. Using z, the zenith angle as the independent variable, the integral of the bending light ray can be converted to:

$$R = -\int_{0}^{z_0} \frac{rdn/dr}{n + rdn/dr} d(z) \tag{3.3}$$

where r is the radial distance, n is the refractive index, z_0 is the value of z at the observer. The quadrature takes equal steps in z and for each step r, n and dn/dr is needed. By Snell's law, we have

$$nr\sin(z) = n_0 r_0 \sin(z_0) \tag{3.4}$$

r can be found by solving the equation:

$$F(r) = nr - \frac{n_0 r_0 \sin(z_0)}{\sin(z)} \tag{3.5}$$

We use Newton-Raphson iteration to find the root:

$$r_{i+1} = r_i - \frac{F(r_i)}{F'(r_i)} \tag{3.6}$$

The model of the atmosphere used has a discontinuity in the temperature gradient at the tropopause so that it needs to be computed seperately. The two parts of the integral were evaluated by Simpson's rule. If the temperature lapse rate changes, the integral for each segmentation will be calculated and summed up.

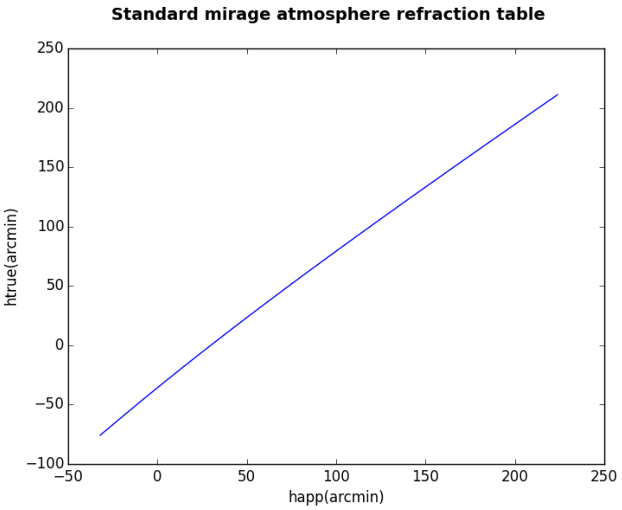

FIG. **3.2:** ***Standard Atmosphere.*** *True altitude vs. apparent altitude using the Bennett formula.*

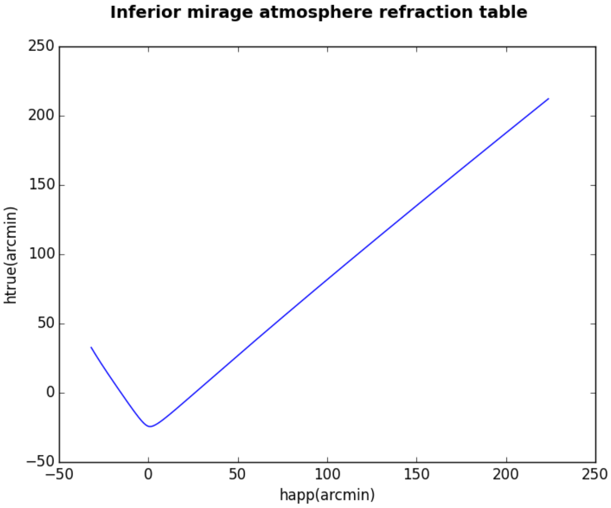

FIG. **3.3:** ***Inferior Mirage temperature profile.*** *Temperature with a sharp increase near the sea level. Observe at 5m, lapse rate above 6.5m is 0.0065 K/m, between 6.5m and 5.5m is 0.4 K/m, below 5.5m is 1.3 K/m.*

3.2.2 Implementation

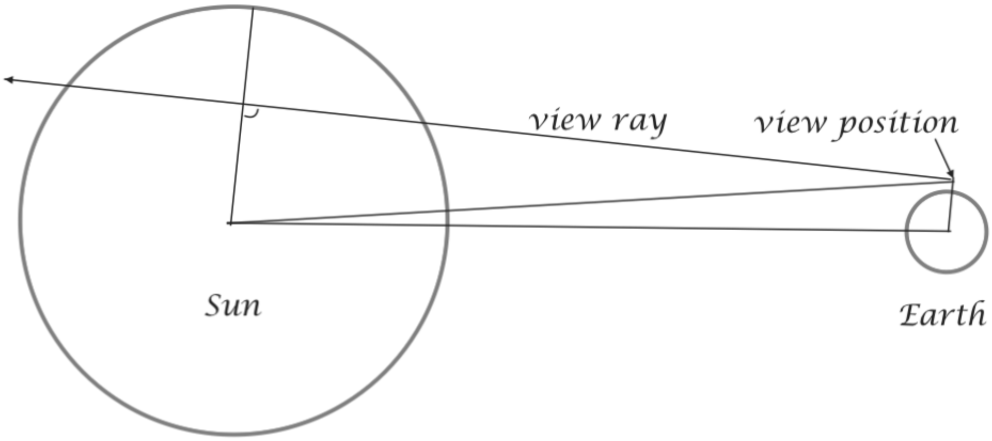

FIG. 3.4: ***Ray tracing.*** *In fragment shader, check if each ray hits the sun.*

The sun is rendered using ray tracing in the sky shader. It computes the view direction for the current pixel, which can give us the apparent altitude. The true altitude can be looked up from the pre-computed dataset. Performing a ray tracing we can determine if the view ray hits the sun. This gives us the shape of the sun. The sun color is determined by its radiance considering the optical depth the light ray goes through.

3.3 Ocean modeling, lighting and animation

The ocean simulation divides into two parts: surface modeling and lighting. The main technical problem of ocean modeling is combining waves of different scales and preventing aliasing when viewing from all distances. Calculating global illumination is challengeable because of the complexity of environmental lighting. I used the model provided by Bruneton, Neyret, & Holzschuch [2010]. The algorithm they presented can provide an accurate ocean model for all viewing distances while taking the sun, sky and refracted light into account. The core of Bruneton's ocean model is a hierarchical representation of the ocean, combining geometry, normal and $BRDF$.

3.3.1 Ocean modeling

The ocean wave model is based on the gravity wave theory. It synthesizes the ocean surface by summing wave trains in various wavelengths. First, a regular grid is generated in screen space, and then projected to the horizontal plane. The projected grid provides adaptive geometric resolution. The grid is then displaced by the time series waves and projected back to screen. The shape of waves is defined by the Gerstner's trochoidal wave model. It can form sharp crests compared to other wave models. A Gerstner wave is defined by

$$p = \begin{bmatrix} x + \sin(\omega t - kx) \\ h\cos(\omega t - kx) \end{bmatrix} \text{, where } \omega = \sqrt{gk} \tag{3.7}$$

Pierson & Moskowitz [1964] declared the energy distribution of gravity waves by a function of wave frequency. Hasselmann, Dunckel, & Ewing [1980] added a wave direction parameter to it. Wave trains in this model are defined by three parameters: amplitude, wavenumber and angular frequency from the Pierson-Moskowitz and Hasselmann spectra. The hierarchical representation of the ocean surface includes three models: an average position p, an average normal n and a BRDF.

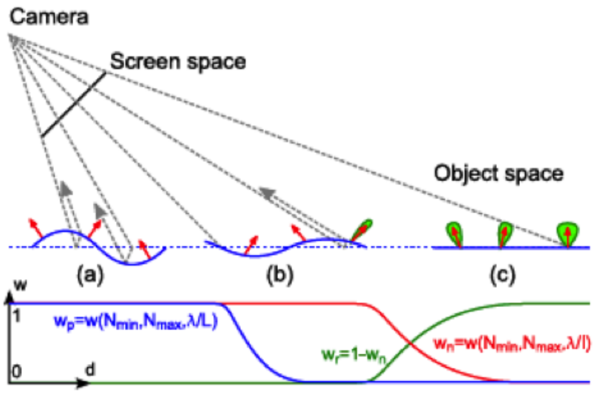

FIG. **3.5:** ***Ocean Model.*** *Transitions from geometry to BRDF, filtered with a wavelength dependent weight. [Bruneton, Neyret, & Holzschuch, 2010]*

Average positions

The average position inside a grid cell is computed by filtering the trochoids whose wavelength λ is less than N_{min} times the projected grid cell size L in object space. They scale each trochoid by

$$w_p = w(N_{min}, N_{max}, \lambda/L) \tag{3.8}$$

with $w(a, b, x) = 3\bar{x}^2 - 2\bar{x}^3$, $\bar{x} = clamp(\frac{x-a}{b-a}, 0, 1)$

$$p = \begin{bmatrix} X \\ 0 \end{bmatrix} + \sum_{1}^{n} w_{p,i} t_i, t_i = \begin{bmatrix} \frac{\mathbf{k}_i}{\|\mathbf{k}_i\|} h_i \sin(\omega t - k_i \cdot \mathbf{X}) \\ h_i \cos(\omega_i t - k_i \cdot \mathbf{X}) \end{bmatrix} \tag{3.9}$$

where $\mathbf{X} = [x \quad y]^T$ is the ocean surface at rest. According to the Nyquist-Shannon sampling theorem, the sampling rate should be larger than twice the occupied bandwidth of the signal to avoid aliasing, so N_{min} here should be larger than 2. However, in practice they found $N_{min} = 1.0$ and $N_{max} = 2.5$ give better results.

Average normals

The average normal inside a pixel is computed by

$$n = \left(\begin{bmatrix} \frac{\partial \mathbf{X}}{\partial x} \\ 0 \end{bmatrix} + \sum_{1}^{n} w_{n,i} \frac{\partial t_i}{\partial x} \right) \wedge \left(\begin{bmatrix} \frac{\partial \mathbf{X}}{\partial y} \\ 0 \end{bmatrix} + \sum_{1}^{n} w_{n,i} \frac{\partial t_i}{\partial y} \right) \tag{3.10}$$

where the filter weight is $w_n = w(N_{min}, N_{max}, \lambda/l)$. So when $\lambda/l - N_{min} \leq 0$, meaning that wavelengths are less than N_{min} times the projected size of a pixel l, their weight is 0.

BRDF

The subpixel surface is described by slope variance. The surface slopes have a Gaussian distribution and slope variance is the sum of the variance of each trochoid slope distribution.

$$\begin{bmatrix} \sigma_x^2 \\ \sigma_y^2 \end{bmatrix} = \sum_{1}^{n} \frac{[k_{i,x}^2 k_{i,y}^2]^T}{\|\mathbf{k}_i\|^2} (1 - \sqrt{1 - \|\mathbf{k}_i\|^2 w_r^2 h_i^2}) \tag{3.11}$$

where σ_x^2 and σ_y^2 are the slope variances along the x and y axis, $w_r = 1 - w_n$. The BRDF is based on the model of Ross, Dion, & Potvin [2005].

$$brdf(v, l) = \frac{q_{vn}(\zeta_h, v, l) F(v, h)}{4 h_z^3 \cos\theta_l v \cdot h} \tag{3.12}$$

where q_{vn} is a normalized visibility probability distribution, defined as:

$$q_{vn}(\zeta, v, l) = \frac{p(\zeta) max(v \cdot f, 0) H(l \cdot f)}{(1 + \Lambda(a_v) + \Lambda(a_l)) f_z \cos\theta_v} \tag{3.13}$$

f is the microfacet normal and p is the Gaussian distribution of the slopes, Λ comes from Smith shadowing factors, H is the Heaviside function which limits the BRDF to directions above the surface.

3.3.2 Ocean lighting

Sun light Sun light reflected at an average position inside of a grid cell is the integration of the BRDF with the sun radiance over the sun solid angle. To get this result, the model computes the slope distribution in the tangent space with a Gaussian of the same variances as in world space. The tangent space is defined by the average normal n and the wind direction. The BRDF is considered constant over the sun solid angle Ω_{sun} and uses Schlick's approximate for the Fresnel term. The final equation is:

$$I_{sun} \approx L_{sun}\Omega_{sun}p(\zeta_h)\frac{R+(1+R)(1-v\cdot h)^5}{4h_z^4\cos\theta_v(1+\Lambda(a_v)+\Lambda(a_l))}\} \tag{3.14}$$

Sky light The sky light contribution at a position is the integration of the BRDF with the sky radiance over a hemisphere. As one of the contributions of Bruneton, Neyret, & Holzschuch [2010], they provided a fast approximate method to compute the illumination reflected by a glossy BRDF from a hemispherical environment map. They computed this integral as the product of average Fresnel reflectance and average sky radiance:

$$I_{sky} = \bar{F}\bar{L} \tag{3.15}$$

The average Fresnel reflectance is approximated by the following fitting function:

$$\bar{F}(v) \approx R + (1-R)\frac{(1-\cos\theta_v)^(5\exp(-2.69\sigma_v))}{1+22.7\sigma_v^1.5} \tag{3.16}$$

The average sky radiance is approximated with an anisotropic texture fetch from the hemispherical sky texture:

$$\bar{L} = tex2D(\mathcal{L}, u(v,0), 2\sigma_x\frac{\partial u}{\partial\zeta_x}(v,0), 2\sigma_y\frac{\partial u}{\partial\zeta_y}(v,0)) \tag{3.17}$$

where $u(v, \zeta) = u(\mathcal{R}(r(v, \zeta)))$ is the function mapping microfacets slopes ζ to texture coordinates u, via reflected view vectors r in tangent space, then transformed from the tangent space to world space by the matrix $\mathcal{R}$. The $tex2D$ function relies on the hardware anisotropic filtering, in which $u(v, 0)$ defines the center and $2\sigma_x \frac{\partial u}{\partial \zeta_x}(v, 0), 2\sigma_y \frac{\partial u}{\partial \zeta_y}(v, 0)$ are the major and minor axis.

Refracted light Multiple scattering happens in the deep ocean water after the sun and the sky light refract into it. The light can be refracted to the viewer again. In Bruneton, Neyret, & Holzschuch [2010], they made an assumption that the radiance L_{sea} coming out of ocean surface is diffuse. In the previous section, $\bar{F}$ is defined as the portion of light being reflected, so the amount of light being refracted is $1 - \bar{F}$. The refracted light can be calculated as:

$$I_{sea} \approx L_{sea}(1 - \bar{F}) \tag{3.18}$$

where in practice L_{sea} is determined by a given sea color multiplying by sky irradiance divided by π.

3.3.3 Implementation

Most of the computation for simulating the ocean is done in vertex and fragment shaders on the GPU. The wave spectrum sample is created first on the CPU and stored on a texture. The vertex shader takes the screen space grid, projects to world space and displaces it, and then projects back. The wave generation and animation is using the Cooley-Tukey FFT algorithm on the GPU. The fragment shader computes the per-pixel normal, takes the spectrum sample to compute the slope variances, and calculates the sun, sky and refracted light. They used the method from Bruneton & Neyret [2008] to estimate the environment light. I modified this part to use the hemispherical sky texture from my sky dome rendering.

Chapter 4

RESULTS

The rendering is implemented in C++ using OpenGL and GLSL shaders. By adapting to different atmospheric settings, this program renders very realistic results. In this chapter, I present results of low turbidity sky sunset/sunrise, high turbidity sky sunset/sunrise, omega sun mirage, mock sun mirage and sky color changes for the turbidity gradient. All images in this chapter is rendered with a observation height of 5 meters above the sea level.

4.1 Visual quility

The rendering images show the variation of sky color during the day time until sunset. Figure 4.1 is rendered with a turbidty 2.0. We can see the sky is nearly pure blue when the sun elevation is high. As the sun goes down, the blue goes dark and more violet red reveals. When the sun is close to the horizon, there is a distinct orange/red area around the sun.

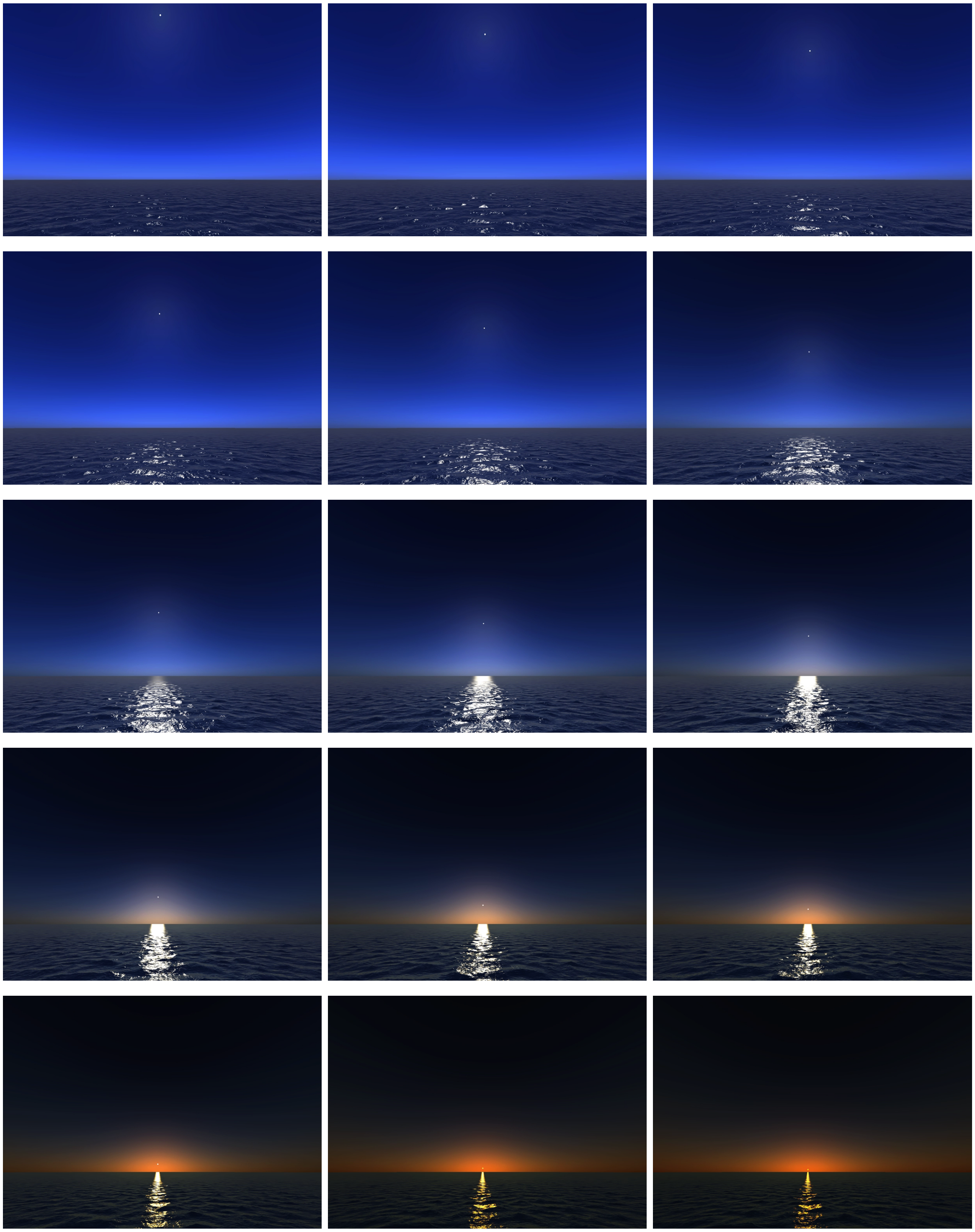

FIG. **4.1:** ***Sunset in a clean sky.*** *Sky turbidity as 2.0. When the sun goes down, the sky color changes from blue, voilet to red. As the scene turns dark, the reflection of the sun on the ocean turns to yellow/orange.*

With a closer look, we can see the deformation of the sun during sunset and sunrise. Figure 4.2 show the sequence of images of omega sun mirage. During the sunset, when the sun comes close to the horizon, the inverse of its image comes up and connect to the main image gradually. The “foot” of Omega shape can be easily noticed.

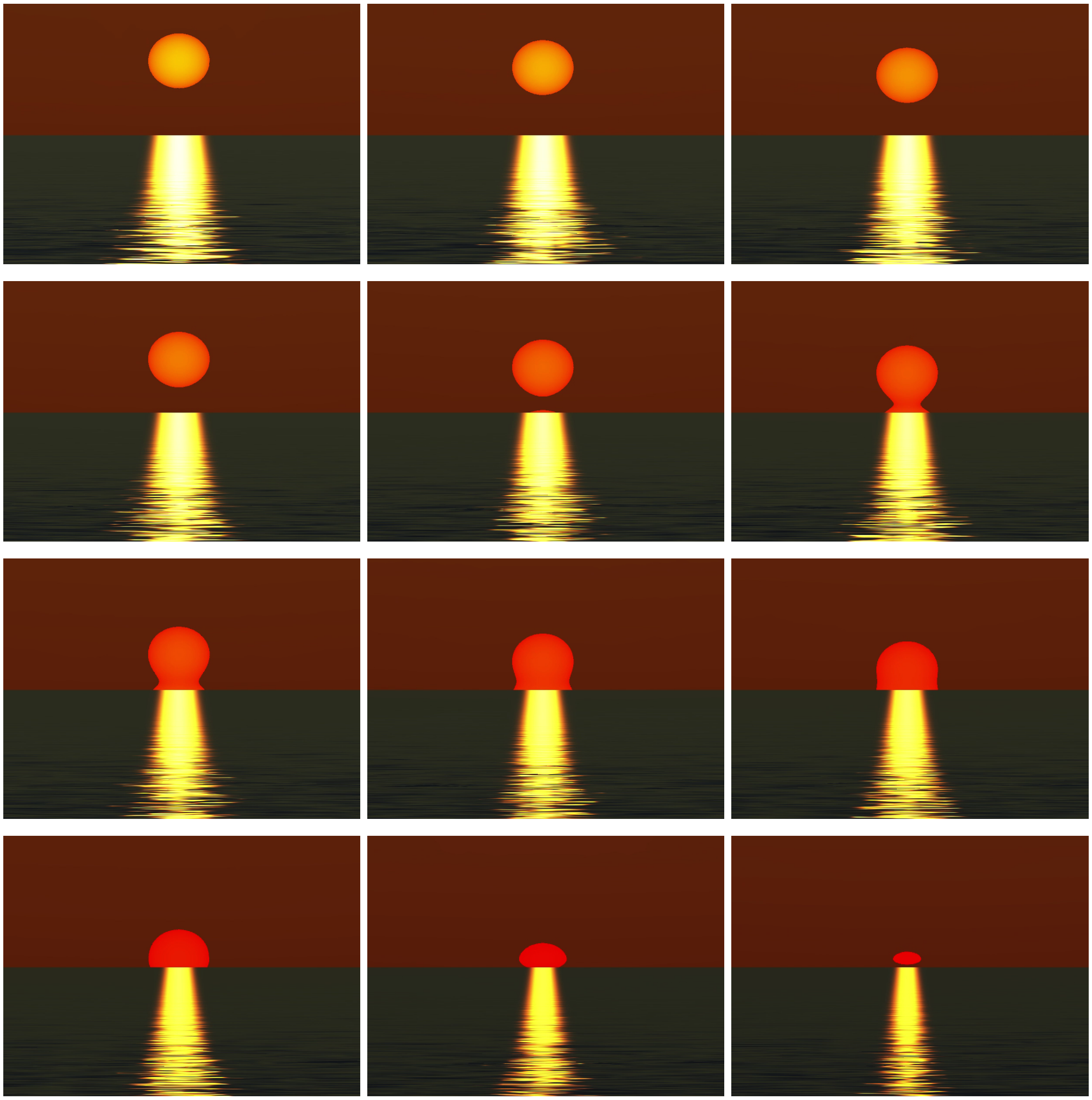

FIG. **4.2:** ***Omega sun mirage.*** *Image sequence shows the sun deformation when it is close to the horizon using the inferior temperature profile. The sun intensity is decreased by a factor of the solar elevation.*

4.2 Performance

The performance is measured on a MacBook Pro (Retina, Mid 2012), with OS X 10.10.1, Processor 2.6 GHz Intel Core i7, Memory 8 GB 1600 MHz DDR3, Graphics: NVIDIA GeForce GT 650M 1024 MB.

Frame rate	30fps
Ocean rendering	28.5 ms/frame
Sky and sun rendering	2.5 ms/frame

4.3 Limitations

There are several limitations of this program. First of all, the sun is not simulated as an area light. Area light problem has been an open topic in computer graphics for years. No good solution has been found yet. To treat it as a point light, we can not get the glow of the sun. Moreover, with a very close look during sunset or sunrise, the reflection of the sun light on the ocean is not very precise to match the mirage shape.

Second, due to the atmospheric refraction, the ocean horizon is higher than the actually astronomical horizon. If not consider that, the sun would be visible when it should have disappeared under horizon. To calculate the displacement of each vertex by refraction angle would add more computation and affect the performance. To adjust it in a simple way, I rotated the ocean plane from the sea level by a refractive angle for the horizonal position, which cause a small offset in the opposite direction. If any simulation can optimize the performance, it will be good to take atmospheric refraction into account for the ocean model.

Chapter 5

CONCLUSION

Comparing the rendering scene with the real photograph, the simulation results are proved to be accurate and realistic. This program can also simulate the different air conditions by adjusting the turbidity, albedo and change the pressure and temperature profiles.

I didn't include more complex situation such as ducting mirage. But it is easy to compute a refraction table by a given atmosphere profile and use it to render. There are many related works that can be added on. The Hosek-Wilkie sky model cannot cover the situation when the sun fell below the horizon. It would be good to have a model that can be adaptive to the dusk and dawn. Twilight happens when the sun itself is not directly visible, but the sunlight scatters in the upper atmosphere and illuminates the lower atmosphere. It is a very special case that should be handled separately. As we may notice in the real world, the sun disk is not a pure color. For more accurate Sun color, the solar radiance has to be calculated very precisely. To add a solar-radiance function to simulate the gradient color of the sun disk can give a more realistic look [Hosek & Wilkie, 2013]. There are many other kinds of interesting phenomena happening during the sunset and sunrise, such as green flash, solar pillar, etc. The dispersion of the refractivity is the basic cause of green flashes, which requires computation for each band of wavelengths. Moreover, a good area light model for the Sun may provide more plausible result for the Sun light contribution.

Bibliography

Auer, L. H., and Standish, E. M. 2000. Astronomical refraction: computational method for all zenith angles. *The Astronomical Journal* 119(5):2472.

Bruneton, E., and Neyret, F. 2008. Precomputed atmospheric scattering. In *Proceedings of the Nineteenth Eurographics Conference on Rendering*, EGSR '08, 1079–1086. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association.

Bruneton, E.; Neyret, F.; and Holzschuch, N. 2010. Real-time realistic ocean lighting using seamless transitions from geometry to BRDF. In *Computer Graphics Forum*, volume 29, 487–496. Wiley Online Library.

Coombe, G., and Harris, M. 2005. Global illumination using progressive refinement radiosity. *GPU gems* 2:635–647.

Cornette, W. M., and Shanks, J. G. 1992. Physically reasonable analytic expression for the single-scattering phase function. *Applied optics* 31(16):3152–3160.

Cowley, L. Atmospheric optics. `http://www.atoptics.co.uk/atoptics/sunflat.htm`. Accessed: 2015-02-12.

Drago, F.; Myszkowski, K.; Annen, T.; and Chiba, N. 2003. Adaptive logarithmic mapping for displaying high contrast scenes. In *Computer Graphics Forum*, volume 22, 419–426. Wiley Online Library.

Durand, F., and Dorsey, J. 2000. *Interactive tone mapping*. Springer.

Elek, O., and Kmoch, P. 2010. Real-time spectral scattering in large-scale natural participating media. In *Proceedings of the 26th Spring Conference on Computer Graphics*, SCCG '10, 77–84. New York, NY, USA: ACM.

Fournier, A., and Reeves, W. T. 1986. A simple model of ocean waves. In *ACM Siggraph Computer Graphics*, volume 20, 75–84. ACM.

Haber, J.; Magnor, M.; and Seidel, H.-P. 2005. Physically-based simulation of twilight phenomena. *ACM Trans. Graph.* 24(4):1353–1373.

Hable, J. 2010. Uncharted 2: HDR lighting. GDC 2010 presentation.

Hachisuka, T. 2005. High-quality global illumination rendering using rasterization. *GPU Gems* 2:615–633.

Hasselmann, D.; Dunckel, M.; and Ewing, J. 1980. Directional wave spectra observed during JONSWAP 1973. *Journal of physical oceanography* 10(8):1264–1280.

Hoffman, N., and Preetham, A. J. 2002. Rendering outdoor light scattering in real time. In *Proceedings of Game Developer Conference*, volume 2002, 337–352.

Hohenkerk, C. Y., and Sinclair, A. T. 1985. *The Computation of an Angular Atmospheric Refraction at Large Zenith Angles*. HM Stationery Office.

Hosek, L., and Wilkie, A. 2012. An analytic model for full spectral sky-dome radiance. *ACM Trans. Graph.* 31(4):95:1–95:9.

Hosek, L., and Wilkie, A. 2013. Adding a solar-radiance function to the hosek-wilkie skylight model. *IEEE Comput. Graph. Appl.* 33(3):44–52.

Hulst, H. C., and Van De Hulst, H. 1957. *Light scattering by small particles*. Courier Dover Publications.

Jensen, H. W.; Durand, F.; Dorsey, J.; Stark, M. M.; Shirley, P.; and Premože, S. 2001. A physically-based night sky model. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, 399–408. New York, NY, USA: ACM.

Kanazawa, K.; Sakato, Y.; and Takahashi, T. 2013. Pencil tracing mirage: Principle and its evaluation. In *ACM SIGGRAPH 2013 Talks*, SIGGRAPH '13, 41:1–41:1. New York, NY, USA: ACM.

Laveder, L. Laurent laveder photographe. `http://www.laurentlaveder.com/`. Accessed: 2015-02-12.

Matkovic, K., and Neumann, L. 1996. Interactive calibration of the mapping of global illumination values to display devices. In *Proceedings of the Twelfth Spring Conference on Computer Graphics*. Citeseer.

Myers, M. Award-winning websites by michael myers. `http://home.comcast.net/~mhmyers/`. Accessed: 2015-02-12.

Nishita, T.; Sirai, T.; Tadamura, K.; and Nakamae, E. 1993. Display of the earth taking into account atmospheric scattering. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, 175–182. New York, NY, USA: ACM.

Nishita, T.; Dobashi, Y.; and Nakamae, E. 1996. Display of clouds taking into account multiple anisotropic scattering and sky light. In *Proceedings of the 23rd Annual Con-*

ference on Computer Graphics and Interactive Techniques, SIGGRAPH '96, 379–386. New York, NY, USA: ACM.

Perez, R.; Seals, R.; and Michalsky, J. 1993. All-weather model for sky luminance distribution-preliminary configuration and validation. *Solar energy* 50(3):235–245.

Pierson, W. J., and Moskowitz, L. 1964. A proposed spectral form for fully developed wind seas based on the similarity theory of SA Kitaigorodskii. *Journal of geophysical research* 69(24):5181–5190.

Preetham, A. J.; Shirley, P.; and Smits, B. 1999. A practical analytic model for daylight. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, 91–100. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.

Raab, M.; Seibert, D.; and Keller, A. 2008. Unbiased global illumination with participating media. In *Monte Carlo and Quasi-Monte Carlo Methods 2006*. Springer. 591–605.

Reinhard, E.; Stark, M.; Shirley, P.; and Ferwerda, J. 2002. Photographic tone reproduction for digital images. *ACM Trans. Graph.* 21(3):267–276.

Ross, V., and Dion, D. 2007. Sea surface slope statistics derived from sun glint radiance measurements and their apparent dependence on sensor elevation. *Journal of Geophysical Research: Oceans (1978–2012)* 112(C9).

Ross, V.; Dion, D.; and Potvin, G. 2005. Detailed analytical approach to the gaussian surface bidirectional reflectance distribution function specular component applied to the sea surface. *JOSA A* 22(11):2442–2453.

Schlick, C. 1994. An inexpensive brdf model for physically-based rendering. *Computer Graphics Forum* 13(3):233–246.

Service, N. W. 2014. Standardized temperature profile. `http://www.srh.noaa.gov/srh/jetstream/atmos/atmprofile.htm`. Accessed: 2015-02-12.

Tessendorf, J. 2001. Simulating ocean water. *Simulating Nature: Realistic and Interactive Techniques. SIGGRAPH* 1(2):5.

Young, A. T. 2006. Understanding astronomical refraction. *The Observatory* 126:82–115.

Young, A. T. 2014. Astronomical refraction. `http://mintaka.sdsu.edu/GF/explain/atmos_refr/astr_refr.html`. Accessed: 2015-02-12.

Zhao, Y.; Han, Y.; Fan, Z.; Qiu, F.; Kuo, Y.-C.; Kaufman, A. E.; and Mueller, K. 2007. Visual simulation of heat shimmering and mirage. *IEEE Transactions on Visualization and Computer Graphics* 13(1):179–189.