Realistic Real-Time Rendering of Human Face with Environment Lighting

Mengzhao Yang, Kuanquan Wang and Wangmeng Zuo
School of Computer Science and Technology
Harbin Institute of Technology
Harbin, China

Mengzhao Yang
School of Computer Information and Engineering
Heilongjiang University of Science and Technology
Harbin, China

***Abstract*—Instead of simple point or area lights for illuminating human face in most rendering methods, we aim to render the appearance of human face influenced by surrounding environment lighting in some real scenes. In this paper, we accurately approximate the specular reflection by using the Kelemen/Szirmay-Kalos reflectance model and ten varying specular parameters over the whole face. Also, subsurface scattering can be well simulated with linear sum of Gaussian blurring in real-time. Further, we use the stretch-map texture to correct UV distortion problem result from blurring in the texture-space. Considering the environment lighting influence to appearance of human face, we use spherical harmonics to represent environment lighting and a quick occlusion approach to obtain a much more realistic rendering of human face, which make the human face look much more natural according to surroundings.**

***Keywords*—human face; specular reflection; subsurface scattering; environment lighting; occlusion**

I. Introduction

Realistic human face rendering is a really important topic in games development and creation of special effects in movies. To obtain a realistic human face, we need to study the skin properties and how the light interacts with skin.

Skin is a highly translucent material and made up of multiple translucent layers. Each of these layers contributes unique subsurface scattering (SSS) properties that independently affect the appearance of skin [1]. SSS approximation is now the main mean to model light reflection from the skin, which can be described by diffusion theory by different researchers. Firstly, Stam [2] introduced diffusion equation to simulate multiple scattering to explain SSS effect. Then, Jensen [3] presented a simple, but very effective dipole model to approximate light diffusion based on [2], which was quickly adopted in the practical application [4]. Further, Donner and Jensen extended their dipole model to multiple dipoles in multiple layers skin [1] and obtained a very realistic rendering. However, it has to take a long time to render human face using methods above. Nowadays, with the increasing requirements of modern 3D technology, the main challenge of rendering is not only to approximate the complex SSS to give a realistic looking skin but also develop sufficiently efficient method to allow for real-time rendering and easy to implement so that it can be well integrated with existing pipelines for practical application.

This work is supported by the National Natural Science Foundation of China under Grant No.s 61173086 and 61179009, and Heilongjiang Province Department of Education Funds under Grant No. 11551435.

Recently, several real-time algorithms to simulate skin have already been introduced. Borshukov and Lewis [5] showed the feasibility of simulating diffusion in texture-space. Then, Gosselin et al. [6] approximated real-time skin rendering in texture-space diffusion, but their technique was not based on multi-layer translucent material and was only a very rough approximation of true scattering. More recent studies by d'Eon and Luebke [7] showed that they could use sum of Gaussian blurring to accurately match the three-layer model for skin given in [1] and got a realistic real-time skin rendering in texture-space. Compared with texture-space diffusion, Jimenez et al. [8] provided a solution to simulate SSS in screen-space at interactive frame rates, and their skin rendering technology had been applied in game development [9]. These methods above focus on the rendering using simple point or area lights for illuminating. In real scene, however, the appearance of human face can be influenced by environment lighting. In addition, the complexity to implement for occlusion should be considered in some practical applications for the real-time(e.g, 3D game developments).

To address these problems, we consider the environment lighting influence to human face in practical applications. Environment lighting can be represented by methods such as spherical harmonics (SH) [10] or wavelet [11], [12]. Spherical harmonics are effective for representing low frequency light but not for high frequency and wavelet can capture both low frequency and high frequency light in a compact manner. However, here we choose the SH based on these considerations: On the one hand, the rendering equation using SH can be rewritten as a simple dot-product, or a matrix-vector multiplication, which allows more easily real-time evaluation and combination with practical applications. This is the main reason why SH is so attractive. On the other hand, SH allows real-time dynamic lighting in arbitrary lighting environment. Furthermore, normally, many scenes are in the sunlight and which is merely low frequently changing, so we can use SH to represent the whole incoming light from the sky quite enough.

In this paper, we extend texture-space diffusion approach related by d'Eon [7] by using environment lighting and implements a real-time rendering of human face. We implement an accurate specular reflection approximation and subsurface

 978-1-4673-0199-2/12/$31.00

scattering approximation by adding the UV stretch correction for real-time rendering. Using environment lighting and occlusion approach for illuminating human face in real scene, much more realistic looking effect can be obtained by the proposed method.

II. Efficient Rendering of Human face

The skin appearance rendering consists of two approximation phases: specular reflection and subsurface scattering. First, we compute the amount of light which is reflected directly from skin surface without penetrating into the flesh, and can obtain accurate specular reflection by using the Kelemen/Szirmay-Kalos (KS) [13] model. Second, we use linear sum of Gaussian convolution and UV stretch-map correction to simulate subsurface scattering effectively. After these two main phases, we combine all light specular and diffusion results into final skin rendering shader.

A. Specular Reflection Approximation

The topmost layer of human skin is oil layer which makes some incoming light not be absorbed by the flesh and reflect directly from skin surface. In fact, the specular light in human skin only reflects about 9 percent of the whole light spectrum, and thus reflection light does not dominate the final face effect.

We can use a physically based Kelemen/Szirmay-Kalos (KS) model to describe the specular term and exploit the Schlick Fresnel approximation for a fairly efficient specular reflectance calculation that allows for the roughness parameter to vary over the object. This model can be described as in

$$KS_{spec} = \exp(-\tan^2(\alpha)/m^2)/(\pi m^2 \cos^4(\alpha)) \quad (1)$$

where, m is the roughness of the material, $\alpha = \arccos(N \cdot H)$. N is surface normal. H is the half-vector between incoming light direction and view direction.

Weyrich et al. [14] measured roughness and intensity for most distinct ten regions of the face across 149 faces and provided measured parameters for the specular BRDF model.

Since, the outermost oil layer of skin is dielectric materials that only reflects light without coloring it. Thus, a physically based skin shader should use a white specular color. Combing the KS model and Weyrich's measured parameters, we can obtain specular reflection of human face as shown in Fig. 1.

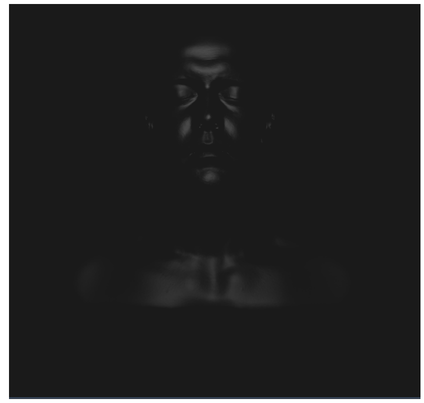

Figure 1. Specular reflection approximation of human face

B. Subsurface Scattering Approximation

To calculate the light diffusion process and reflection underneath the surface of skin, we apply a diffusion profile to approximate the process. The diffusion profile can model that how light scatters across a radial distance r and how much light emerges as a function of the angle and distance from center of its hit point. d'Eon et al. [7] found that three-layer skin diffusion profile can be approximated with linear sum of Gaussian function. To be more precise, six Gaussian blurring kernels could very closely fit the three-layer profile for skin. But, for some single layer materials, four Gaussians are enough to fit most profiles.

For each diffusion profile $R(r)$ from a dipole or multipoles-based model, we can use sum of k Gaussians with variances v_i and weights w_i to approximate the $R(r)$ as in

$$R(r) \approx \sum_{i=1}^{k} w_i G(v_i, r) \quad (2)$$

where we choose the following definition for the Gaussian of variance v as in

$$G(v, r) \equiv \frac{1}{2\pi v} \exp\left(-r^2/2v\right) \quad (3)$$

As mentioned in [7], diffusion profiles of skin can be approximated with a weighted sum of six Gaussian functions with the parameters and weights given in Fig. 2.

	Variance (mm^2)	Red	Blur Weights Green	Blue
	0.0064	0.233	0.455	0.649
	0.0484	0.100	0.336	0.344
	0.187	0.118	0.198	0
	0.567	0.113	0.007	0.007
	1.99	0.358	0.004	0
	7.41	0.078	0	0

Figure 2. Six sum of Gaussian weights for a three-layered skin model

Therefore, we can obtain the SSS approximated result by convolving the irradiance texture with the weighted sum of Gaussian functions. Notice that we use a very nice property of convolution: the convolution of an irradiance texture I by a kernel that is a weighted sum of functions $G(v_i, r)$ is the same as a weighted sum of irradiance I, each of which is the original irradiance convolved by each of the functions:

$$I * R(r) = I * \left(\sum_{i=1}^{k} w_i G(v_i, r)\right) = \sum_{i=1}^{k} w_i I * G(v_i, r) \quad (4)$$

Because Gaussian functions are simultaneously separable and radially symmetric, Gaussian convolution at a wider stage can be computed from the result of a previous Gaussian convolution. Finally, the convolution of any two Gaussians is another Gaussian as in

$$G(v_1) * G(v_2) = G(v_1 + v_2) \tag{5}$$

Therefore, if the previous convolved irradiance texture contains the convolved version $I_1 = I * G(v_1, r)$ and we want to compute $I_2 = I * G(v_2, r)$ $(v_2 \geq v_1)$, we only need to convolve $I_2 = I_1 * G(v_2 - v_1), r)$.

Since this blur kernel will be applied on texture space coordinate, we also need to take care of UV stretch across our head mesh. This is mainly because if the 3D face mesh is a curved surface, the distance between two locations in the texture cannot correspond to the distance on the mesh due to UV distortion, thus the SSS would not be accurate. A simple solution to correct this problem is to compute at every frame and for each pixel a stretch value in both U and V direction and store it into a stretch-map texture. Then when performing convolution, we modulate the spacing of the convolution taps at each point on the surface in texture-space according to the value in the stretch-map texture. Fig. 3 shows our each blurring pass with UV stretch-map texture and the blurred texture-space rendering result.

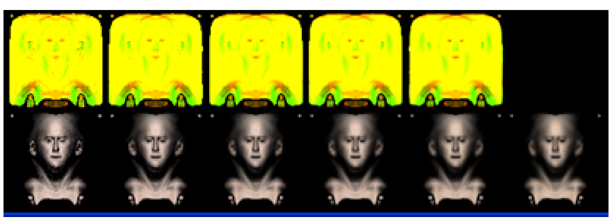

Figure 3. Top: five UV stretch-map correction textures for adding into between each two blurred textures. Bottom: six Gaussian blurred textures which have been added into UV stretch-map correction

III. Environment Lighting and Occlusion

In real world, instead of using simple point or area lights for illuminating a scene, it can be much more realistic to use environment lighting. A good way of representing environment lighting is by use of an environment map (a cube or a sphere) that contains the environment light coming from each direction around the object.

For human skin, diffusion light accounts for about 91 percent of the whole light spectrum, so diffusion light should be mostly considered. Ramamoorthi and Hanrahan [10] introduced a way to approximate diffusion lighting from an environment map with spherical harmonics. After pre-computation of the environment map and occlusion information, it can render human face using diffusion lighting from the environment map in real-time.

A. Definition of Spherical Harmonics

The spherical coordinates can be represented by the standard parameterization of points on the surface of a unit sphere S as in

$$S = (x, y, z) = (\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta) \tag{6}$$

The spherical harmonics basis functions $y_l^m(\theta, \varphi)$ are defined as in

$$y_l^m(\theta, \varphi) = \begin{cases} \sqrt{2}K_l^m \cos(m\varphi)P_l^m \cos\theta, (m > 0) \\ \sqrt{2}K_l^m \sin(-m\varphi)P_l^{-m} \cos\theta, (m < 0) \\ K_l^0 P_l^0 \cos\theta, (m = 0) \end{cases} \tag{7}$$

where P_l^m are the Legendre polynomials and K_l^m is a scaling factor as in

$$K_l^m = \sqrt{(2l+1)(l-|m|)!\backslash(4\pi(l+|m|)!)} \tag{8}$$

In the following, we will consider the spherical harmonics basis functions $y_i(\theta, \varphi)$ defined as in

$$y_i(\theta, \varphi) = y_l^m(\theta, \varphi) \tag{9}$$

where, $i = l(l+1) + m$ with $l \in R^+$ and $m \leq |l|$

Now, spherical harmonics coefficients c_i of one function f on the sphere S can be evaluated by integration as in

$$c_i = \int_S f(s) y_i(s) ds \tag{10}$$

Finally, these coefficients can be used to reconstruct an approximation of the function f as in

$$\hat{f}(s) = \sum_{i=0}^{l^2} c_i y_i(s) \tag{11}$$

Note that for a nth band approximation we need n^2 spherical harmonic coefficients. Here we only use a three band approximation $l = 3$ because it is enough to approximate the diffusion irradiance. Thus we only need 9 spherical harmonics coefficients. Note that each of the spherical harmonic coefficient is separated in the 3 color channels resulting in a total of 27 coefficients for a given environment map.

B. Directional Occlusion Computation

Directional occlusion is important to enhance the reality of the scene. Hoberock [15] presented a robust algorithm for computing high-quality ambient occlusion and pointed out that subsurface scattering may be computed with the same hierarchical integration method. Considering that some practical applications mostly emphasize faster computing and easier to implement, in our implementation, we use a simple technique [16] to compute the occlusion.

At each point on the surface of the object we want to compute the diffusion irradiance E which is given as in

$$E = \int_S L(\omega) V(\omega) (N \cdot \omega) d\omega \tag{12}$$

we can rewrite this as in

$$E = \int_S L(\omega) g(\omega) d\omega = \sum_{i=0}^{n} L_i g_i \tag{13}$$

where $g(\omega) = V(\omega)(N \cdot \omega)$ and g_i is a function that only depends on the geometry of the mesh. The coefficients L_i comes from the projection of the environment lighting function into spherical harnomics basis functions. Both of them are the same 9 coefficients that can be pre-computed when given an environment map.

To evaluate the g_i, the visibility term V has to be pre-computed. We adopt the method [16] which consists of placing the camera at the vertex position and then renders the mesh into a cube map, and make that the cube map is cleared in white and the mesh is rendered in black. Thus, we have a simple way to evaluate the visibility term V. So we only have to lookup in the cube map in the direction ω and if the cube map texel in this direction is white, it means the vertex is not occluded in this direction, or if it is black, the vertex is occluded by another part of the mesh.

IV. RESULTS

We have implemented environment lighting to render appearance of human skin in real-time. Based on the hardware platform: AMD Athlon II X4 Four Cores and NVIDIA GeForce GT430 (2GB), and using 4096x4096 resolution of irradiance texture and environment maps of our experiment from Humus [17], we have achieved frame rates of approximately 8 to 10 frames per second and have obtained much more realistic rendering of human face according to surroundings.

Besides the hardware configuration, the FPS of real-time rendering is mostly decided by resolution of irradiance texture, the rendering algorithm and the pre-computation of texture map. Penner and Borshukov [18] have provided a so-called pre-integrated skin shading and obtained a higher FPS in real-time skin rendering which stores all the needed data in vertices/texture maps, and uses light gradients and pre-integration scattering.

Fig. 4 shows the main window of the project, and we can render a human face in an environment map but no environment lighting influence added. For interacting and debugging operation easily, we can select corresponding option to view glow effect, environment lighting when adding the sky box or debug mode containing the depth map, six Gaussian blur pass results and five UV stretch results and so on. Also when one click right button on the rendering window, some menus will appear for us to adjust the exposure, blurring amount and width, and toggle test texture to view different rendering effect of human face, and all of these can be seen in the Fig. 5.

To clearly show the environment lighting influence in our implementation, we use a pure red color to illuminate a single face and can obtain much more realistic result near the ears, the eyes and around the nose as shown in Fig. 6 since occlusion approach is considered and implemented here.

Instead of using point or area lights for illuminating human face, we make the face look more realistic by using environment lighting represented by spherical harmonics. As mentioned in Section III, spherical harmonics can be easily realized in practical application and many scenes are in the sunlight which is merely low frequently changing. Therefore, we can use the spherical harmonics to represent these scenes quite accurately. With gamma correction, the result of environment lighting with and without directional light occlusion can been seen in Fig. 7 and Fig. 8. Compared with Fig. 7, with directional light occlusion we can attain a better realistic face in the regions around the ears, the eyes, the neck and the lip, etc as shown in Fig. 8. In summary, much more photo-realistic effect of the whole human face can be achieved using our environment lighting and occlusion approach.

Figure 4. Main window of our rendering project

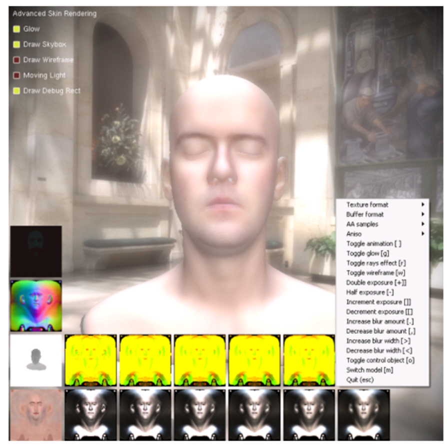

Figure 5. Glow effect, debug mode and menus to adjust parameters

Figure 6. Environment lighting with a pure red color on a single face. Left: without occlusion. Right: with directional light occlusion

Figure 7. Realistic human face rendering with environment lighting without occlusion

Figure 8. Realistic human face rendering with environment lighting with occlusion

V. Conclusion

This paper implements a novel method to achieve more realistic rendering of human face based on texture-space diffusion and environment lighting. Although the occlusion approach is simple, it can be easily implemented and obtain a realistic effect. Experimental results show some realistic effects of human face and verify the validity of the proposed approach. These methods can assist 3D game development and movie-making to get virtual character for requiring realistic scene.

However, the proposed method also has some limitations, e.g., it has no ability to capture the high-frequently light in some special scenes. In our future work, we will study the ways to approximate the environment map in a non-linear wavelet basis and achieve real-time human face rendering in the all frequency space.

Acknowledgment

We would like to thank our colleagues Wei Yang and Lei Zhang for providing valuable suggestions. Also, we thank IR-LTD for their scanned head model.

References

[1] C. Donner and H. Jensen, "Light diffusion in multi-layered translucent materials," in *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 1032–1039.

[2] J. Stam, "An illumination model for a skin layer bounded by rough surfaces," in *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*. Springer-Verlag, 2001, pp. 39–52.

[3] H. Jensen, S. Marschner, M. Levoy, and P. Hanrahan, "A practical model for subsurface light transport," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 511–518.

[4] H. Jensen and J. Buhler, "A rapid hierarchical rendering technique for translucent materials," in *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3. ACM, 2002, pp. 576–581.

[5] G. Borshukov and J. Lewis, "Realistic human face rendering for the matrix reloaded," in *ACM SIGGRAPH 2003 Sketches and Applications*. ACM, 2003.

[6] D. Gosselin, "Real time skin rendering," in *Game Developer Conference, D3D Tutorial*, vol. 9, 2004.

[7] E. d'Eon and D. Luebke, "Advanced techniques for realistic real-time skin rendering," *GPU Gems 3*, vol. 3, pp. 293–347, 2007.

[8] J. Jimenez, V. Sundstedt, and D. Gutierrez, "Screen-space perceptual rendering of human skin," in *ACM Transactions on Applied Perception (TAP)*, vol. 6, no. 4. ACM, 2009, p. 3.

[9] M. Mittring and B. Dudash, "The Technology Behind the DirectX 11 Unreal Engine "Samaritan" Demo," in *Game Developer Conference (GDC)*, Mar. 2011.

[10] R. Ramamoorthi and P. Hanrahan, "An efficient representation for irradiance environment maps," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 497–500.

[11] R. Ng, R. Ramamoorthi, and P. Hanrahan, "All-frequency shadows using non-linear wavelet lighting approximation," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 376–381, 2003.

[12] R. Wang, J. Tran, and D. Luebke, "All-frequency interactive relighting of translucent objects with single and multiple scattering," in *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 1202–1207.

[13] C. Kelemen and L. Szirmay-Kalos, "A microfacet based coupled specular-matte brdf model with importance sampling," in *Eurographics Short Presentations*, vol. 25, 2001, p. 34.

[14] T. Weyrich, W. Matusik, H. Pfister, B. Bickel, C. Donner, C. Tu, J. McAndless, J. Lee, A. Ngan, H. Jensen, *et al.*, "Analysis of human faces using a measurement-based skin reflectance model," in *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 1013–1024.

[15] J. Hoberock and Y. Jia, "High-quality ambient occlusion," *GPU gems 3*, vol. 3, pp. 257–274, 2007.

[16] "Spherical harmonics lighting," available at http://sebastien.hillaire.free.fr/demos/sh/.

[17] "Environment maps," available at http://www.humus.name.

[18] E. Penner and G. Borshukov, "Pre-integrated skin shading," *Gpu Pro 2*, vol. 2, p. 41, 2011.