Real-time Hair Rendering using Irradiance Maps

Xiaoxiong Xing[1,a) Yoshinori Dobashi[1] Tsuyoshi Yamamoto[1]

Abstract: We present a real-time framework for rendering hair under environment lighting. We precompute the diffuse and specular irradiance environment maps for Kajiya-Kay model with a given environment map, thus reduce the runtime shading computation to simple texture lookups, and we simulate hair self-shadowing by using offline ambient occlusion. In addition, we further achieve runtime editing of the environment map by using spherical harmonic lighting to accelerate the computation of irradiance maps.

1. Introduction

Real-time hair rendering is a challenging topic in the graphics field due to the complexity of hair geometry and its complex scattering properties. Kajiya and Kay [2] proposed the first experimental hair shading model that has been commonly used in real-time applications due to its simplicity. Marschner et al. [5] proposed a more accurate model that decomposes the reflected light into three groups, R, TT and TRT. The two models aim at capturing the single scattering property of hair. To simulate multiple scattering of hair in real-time, Zinke et al. [11] proposed the dual-scattering algorithm for the approximation of multiple scattering.

Over the past few years, a lot of work has been done to interactive rendering under environment lighting. The usual way to represent environment light is to use an environment map such as a cube map that contains the radiance information from all directions. This kind of environment map is often used in rendering perfect reflection surfaces. Another kind of environment map is generated by embedding the radiance information together with the reflection function or simply the shading result. The environment map generated in this way is called an irradiance environment map. Sloan et al. [8] introduced the spherical harmonic lighting to an effective representation of environment light. The idea is to represent rendering equation with orthogonal basis functions which was extended to the irradiance map by Ramamoorthi and Hanrahan [6].

The intuitive way to render hair under environment light is to densely sample the environment map to obtain a large number of directional lights, render the hair model under the lights iteratively and then accumulate the results. For an environment map consisting of $16 \times 16 \times 6$ texles, the number of rendering passes required is a multiple of $16 \times 16 \times 6$ that is beyond the ability of current hardware to achieve real-time frame rates. Ren et al.[7] proposed to use a set of spherical radial basis functions (SRBFs) to approximate the environment light, thus reduce the number of lights being considered to the number of SRBF lights. In their single scattering framework, they accumulate the results rendered by all SRBF lights. However, their work relies on such heavy pre-computation of the SRBFs that cannot edit environment light at runtime.

Inspired by King [4] that uses irradiance maps for the Phong shading model, we compute diffuse and specular irradiance maps for Kajiya-Kay model at run-time, thus reduce the runtime computation to simple texture lookups. The irradiance maps are updated when the environment map is altered or edited. We further apply spherical harmonic lighting to the generation of irradiance maps making it possible to edit environment maps during runtime.

In the following sections, we first provide an overview of our method, and then describe the generation process of the irradiance map, followed by the offline computation process of transmittance for hair vertices. Next, we describe the method to represent the environment map with spherical harmonic basis functions to accelerate the computation of irradiance maps. At last, we conclude our paper with experiment results and future plan.

2. Overview

We use line segments to represent the hair strand. The outgoing radiance towards $\mathbf{v}$ at a point $\mathbf{x}$ on the hair strand is the integration over the sphere:

$$L_o(\mathbf{x}, \mathbf{v}) = \int T(\mathbf{x}, \mathbf{l}) f(\mathbf{t}, \mathbf{v}, \mathbf{l}) L_i(\mathbf{l}) d\mathbf{l} \tag{1}$$

where $\mathbf{t}$ is the tangent direction at $\mathbf{x}$, $T(\mathbf{x}, \mathbf{l})$ is the transmittance, and $L_i(\mathbf{l})$ is the incoming radiance.

We decouple the transmittance term from the integral and approximately represent the outgoing radiance by the following equation:

$$L_o(\mathbf{x}, \mathbf{v}) = A(\mathbf{x}) \int f(\mathbf{t}, \mathbf{v}, \mathbf{l}) L_i(\mathbf{l}) d\mathbf{l} = A(\mathbf{x})\, R(\mathbf{t}) \tag{2}$$

where $A(\mathbf{x})$ is the average transmittance and $R(\mathbf{t})$ indicates the irradiance map. $A(\mathbf{x})$ is expressed by the following equation:

$$A(\mathbf{x}) = \int \exp(-\Omega(\mathbf{l}, \mathbf{x})) d\mathbf{l}, \text{ where } \Omega = \int_0^l \rho(l') dl' \tag{3}$$

[1] Hokkaido University, Japan
[a)] xingxiaoxiong@ime.ist.hokudai.ac.jp

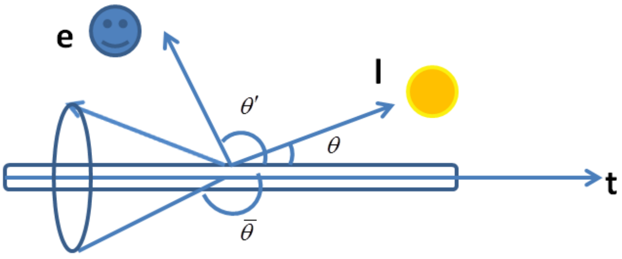

Fig. 1 Notation for scattering geometry.

In Equation 3, $\Omega(\mathbf{l}, \mathbf{x})$ is the opacity value at $\mathbf{x}$. l is the optical depth from the light in direction $\mathbf{l}$ to $\mathbf{x}$, ρ is an extinction function along the path.

3. Irradiance Environment Maps for Kajiya-Kay model

The diffuse component of Kajiya-Kay model [2] for a single light source is expressed by the following equation:

$$\Phi_{diffuse} = K_d \sin\theta \tag{4}$$

where θ is the angle between light direction $\mathbf{l}$ and tangent direction $\mathbf{t}$, and K_d is the diffuse reflectivity. The scattering geometry is shown is **Fig. 1**. If we treat the environment map as a collection of N directional light sources (each texel is a light source), then the color of a vertex with tangent vector $\mathbf{t}$ is determined by accumulating the contributions from all the light sources:

$$I_{diffuse} = K_d \sum_{i=1}^{N} (\sin\theta_i \, L_i) \tag{5}$$

where L_i is the intensity of the ith light source. The above formula indicates that vertices having the same tangent direction yield the same shading result. So we compute the above sum (leave out K_d which should be a vertex attribute) for any tangent directions and save them in their corresponding texels in the diffuse irradiance map. To evaluate the sum for a tangent direction, we iterate over all texels in the original environment map. For each texel i, use the intensity of that texel as L_i. When rendering a hair strand, we compute the tangent direction at the fragment and use it as index to look up the irradiance map.

The specular component of Kajiya-Kay model is expressed as:

$$\Phi_{specular} = K_s \cos^p(\bar{\theta} - \theta') \tag{6}$$

where $\bar{\theta}$ is the supplementary angle of θ, and θ' is the angle between tangent direction and view direction. Similar to the computation of the diffuse irradiance map, the value that would be stored in a texel of the specular irradiance map is the sum of the above singular form:

$$I_{specular} = K_s \sum_{i=1}^{N} \cos^p(\bar{\theta}_i - \theta')L_i \tag{7}$$

Because the right side of Equation 7 contains two angles whose information cannot be expressed by just one irradiance map, we decided to compute a set of specular irradiance maps for several different θ' values. Because θ' lies in the range $[0, \pi]$, we evenly divide the domain into M intervals and compute $M+1$ irradiance maps for θ' that equal the end points of these intervals. The generation process of each specular irradiance map is similar to that of the diffuse map except the difference in reflection functions. During runtime we find out the nearest two specular irradiance maps between which the current θ' lies in, and interpolate between two values from the two maps indexed by tangent direction.

In more concrete terms, if we generate a total of $M+1$ specular irradiance maps, the texel value in each map will be:

$$I_{specular} = K_s \sum_{i=1}^{N} \cos^p(\bar{\theta}_i - m\pi/M)L_i \tag{8}$$

where m is an integer from 0 to M.

Figure 2 shows the diffuse and the specular maps with $M = 4$. Clearly the rendering result differs when using different number of specular maps. Refer to **Fig. 3**(d) and Fig. 3(e) for the comparison of the specular component rendered by five and nine specular maps.

4. Ambient Occlusion for Hair

This section describes an offline approach for ambient occlusion.

Our offline method renders three deep opacity maps [10] in addition to a depth map. To render deep opacity maps, two rendering passes are needed. The first pass generates a depth map from the view of the light source, and the second pass divides the hair volume into several layers following the depth of the geometry as seen from the light source, and then assign the opacity contribution of each fragment to the layer it falls in and behind it. This layering method solves the artifacts that often occur in opacity shadow maps [3] and reduce the number of layers needed to get acceptable results.

In our offline ambient occlusion method, we keep an array to store the transmittance for all hair vertices. We iterate over the randomly sampled 100 points on the unit sphere. In each iteration we first render three layers of deep opacity map from the view of that sample, and then draw the transmittance of each hair vertex to a single fragment in the framebuffer, and then fetch the transmittance from the framebuffer and add it to the vertex's corresponding entry in the transmittance array. The data is saved in a file and loaded as vertex attribute in our real-time program. **Figure 4** shows the flowchart of our program. Refer to **Fig. 5**(c) for the result after applying ambient occlusion.

5. Spherical Harmonic Lighting

The irradiance maps in the above framework are generated before the rendering of hair. For an environment map consisting of N texels, the total number of operations for generating one channel of an irradiance map is $N \times N$, which is approximately 2.4 million in the case of a $16 \times 16 \times 6$ cube map. It is beyond the ability of current commercial hardware to compute the diffuse and specular maps while introducing unnoticeable delay. In order to achieve runtime editing of the environment map, we apply spherical harmonic lighting to accelerate the computation of irradiance maps. A detailed discussion on spherical harmonic lighting can be found in [1].

The environment light can be approximated by a set of spheri-

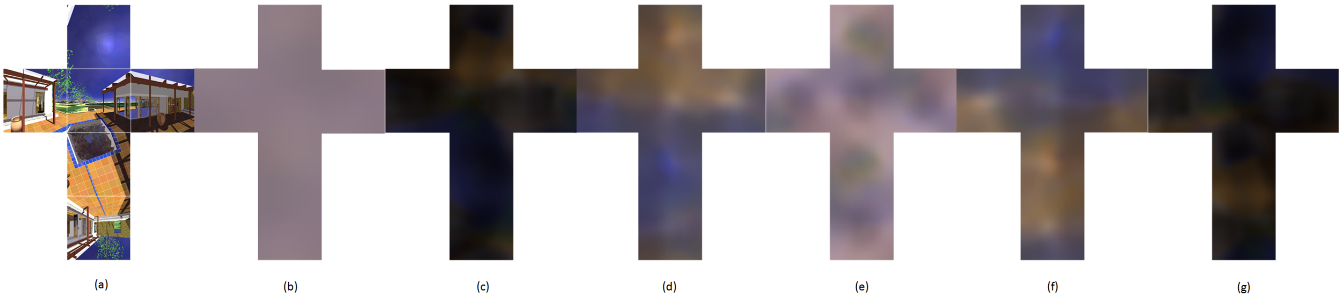

Fig. 2 (a) is the cube map. (b) is the diffuse irradiance map. (c) to (g) are the five specular irradiance maps.

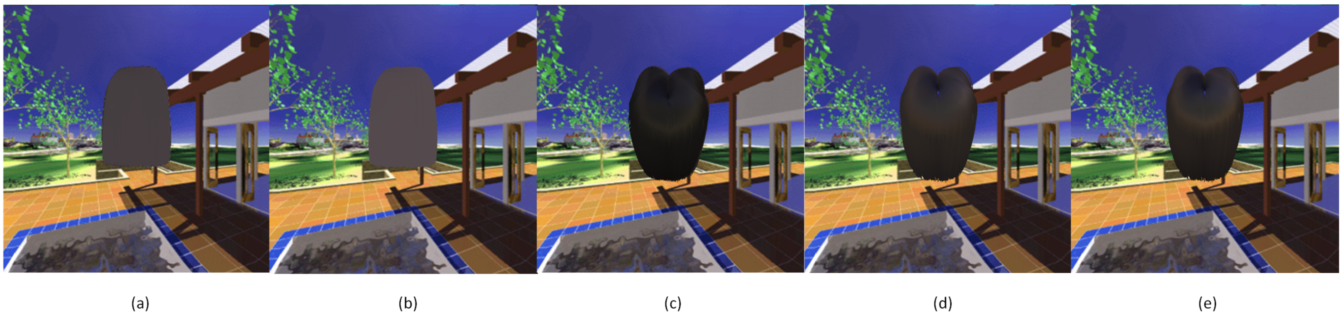

Fig. 3 (a) shows the diffuse component rendered with a brute-force way by accumulating results under all texel lights. (b) was rendered by diffuse irradiance map. (c), (d) and (e) show the comparison between using different number of specular maps. (c) was rendered with the brute-force way. (d) and (e) were rendered using five and nine specular irradiance maps respectively. (e) accurately captured the ring around the top of the head that does not apparent in (d).

cal harmonic basis functions:

$$L(\mathbf{l}) = \sum_{i=1}^{k} w_i Y_i(\mathbf{l}) \tag{9}$$

where $L(\mathbf{l})$ represents the radiance coming from direction $\mathbf{l}$. In the cube map situation $\mathbf{l}$ is the vector from the world space origin to a texel center. $Y_i(\mathbf{l})$ is the ith spherical harmonic basis function, and w_i is the ith coefficient.

The spherical harmonic transformation is done by the following equation:

$$w_i = \int L(\mathbf{l})Y_i(\mathbf{l})d\mathbf{l} = \sum_{j=1}^{N} L(\mathbf{l}_j)Y_i(\mathbf{l}_j)d\omega \tag{10}$$

Since $Y_i(\mathbf{l})$ is independent of the environment light, we precompute Y_i for all texel centers and load them at the launch of our real-time program.

The computation for one texel in the irradiance map steps as follows:

$$I(\mathbf{t}) = \int L(\mathbf{l})f(\mathbf{t},\mathbf{l})d\mathbf{l} = \sum_{i=1}^{k} w_i(\sum_{j=1}^{N} Y_i(\mathbf{l}_j)f(\mathbf{t},\mathbf{l}_j)d\omega) \tag{11}$$

$I(\mathbf{t})$ is the intensity of the texel in direction $\mathbf{t}$. $f(\mathbf{t},\mathbf{l})$ is the reflection function. The sum $\Sigma_{j=1}^{N} Y_i(\mathbf{l}_j)f(\mathbf{t},\mathbf{l}_j)d\omega$ is independent of the environment light, so we can precompute it respectively for diffuse and specular components.

According to [4], 9 coefficients are sufficient for diffuse lighting, and $\sqrt{p}$ (p is the Phong exponent that equals 64 in our experiment) is sufficient for specular lighting. In our program, when the environment light changes, it first computes the spherical harmonic coefficients for the environment light using Equation 10 for the diffuse and specular component respectively, and then generate one diffuse irradiance map and nine specular irradiance maps. So the total operation is $9 \times N + 9 \times N$ for the diffuse map and $9 \times (64 \times N + 64 \times N)$ for the specular maps which is far smaller than the operation needed to compute irradiance maps regularly which is $(9 + 1) \times N \times N$. The computation time is less than one second using CPU; it can be further map to GPU in a similar way like in [4]. Figure 5(a) and Fig. 5(b) are the comparison of results by using regular irradiance maps and spherical harmonic lighting-generated irradiance maps.

6. Results

All the hair models used in our experiment were downloaded from [9]. We implemented our algorithm on a PC with Core i7-2600k CPU, 4GB memory, and an Nvidia GeForce GTX 580 graphics card. When the image size is 640 × 480, our method can render hair at 60 fps, and the frame rate does not drop with the increase of resolution and model complexity. Compared with the algorithm in [7] whose rendering speed is 10 fps on average,, our algorithm is much faster.

The effects of the hair self-shadowing in **Fig. 6** were generated with our offline ambient occlusion approach. Refer to **Table 1** for the offline computation time for the models.

In order to find out the appropriate number of sample directions used in the offline ambient occlusion computation, we rendered four results using different number of sample directions; they are 100, 64, 25, and 9. We computed the difference images between the result rendered by 100 sample directions and the other three as shown in **Fig. 7**. The offline computation time is shown in **Table 2** which almost increases linearly with respect to the number

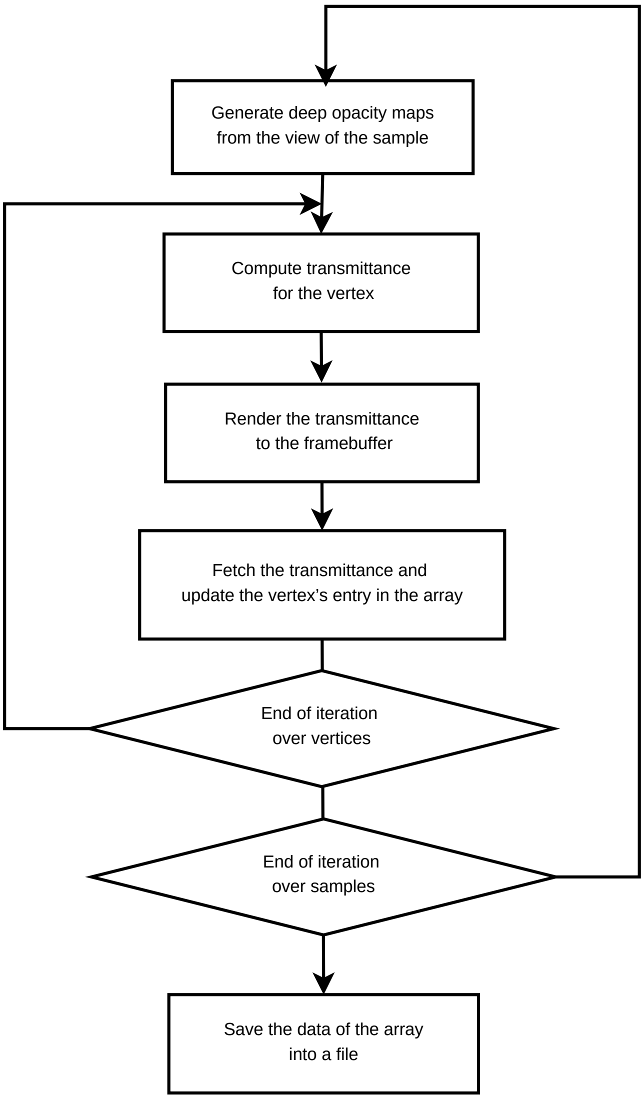
Generate deep opacity maps from the view of the sample
Compute transmittance for the vertex
Render the transmittance to the framebuffer
Fetch the transmittance and update the vertex's entry in the array
End of iteration over vertices
End of iteration over samples
Save the data of the array into a file

Fig. 4 Ambient occlusion flowchart.

Table 1 Computation time of the offline ambient occlusion approach for the models with different number of vertices.

	vertices	time(min)
Model 1	1,519,823	103
Model 2	687,532	46
Model 3	1,031,268	65

Table 2 Computation time for different numbers of sample directions. The hair model contains 872,756 vertices.

samples	time(min)
9	5.5
25	14.9
64	39.3
100	63.4

of sample directions.

We have experimented with different sizes of irradiance maps and found that irradiance maps of the size 16×16×6 are sufficient for our needs. The results rendered by different sizes of irradiance maps are shown in **Fig. 8**. We use one diffuse and nine specular irradiance maps. It takes about 32 ms to update irradiance maps of the size 16 × 16 × 6 using spherical harmonic lighting on the GPU.

7. Conclusion and Future Work

In this technical report, we proposed a real-time hair rendering framework that computes irradiance environment maps for Kajiya-Kay model in a preprocess to reduce runtime shading computation to texture lookups. By using spherical harmonic basis functions, we achieved runtime editing of environment maps. The main advantage of our environmental rendering framework is that it requires no scene-dependent precomputations, thus it allows the user to edit the environment light interactively . But the current implementation suffers from simple shading model and the use of precomputed transmittance that requires the hair strand not move much relative to each other.

In the future we would like to encode more advanced shading models into irradiance maps and add hair simulation to our current program to build a self-contained toolkit.

References

[1] Green, R.: Spherical Harmonic Lighting: The Gritty Details, Sony Computer Entertainment America (online), available from ⟨http://www.research.scea.com/gdc2003/spherical-harmonic-lighting.pdf⟩ (accessed 2012-05-16).

[2] Kajiya, J. T. and Kay, T. L.: Rendering fur with three dimensional textures, *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '89, New York, NY, USA, ACM, pp. 271–280 (online), DOI: 10.1145/74333.74361 (1989).

[3] Kim, T.-Y. and Neumann, U.: Opacity Shadow Maps, *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, London, UK, UK, Springer-Verlag, pp. 177–182 (online), available from ⟨http://dl.acm.org/citation.cfm?id=647653.732282⟩ (2001).

[4] King, G.: Real-Time Computation of Dynamic Irradiance Environment Maps, NVIDIA Corporation (online), available from ⟨http.developer.nvidia.com/GPUGems2/gpugems2chapter10.html⟩ (accessed 2012-05-16).

[5] Marschner, S. R., Jensen, H. W., Cammarano, M., Worley, S. and Hanrahan, P.: Light scattering from human hair fibers, *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, New York, NY, USA, ACM, pp. 780–791 (online), DOI: 10.1145/1201775.882345 (2003).

[6] Ramamoorthi, R. and Hanrahan, P.: An efficient representation for irradiance environment maps, *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, New York, NY, USA, ACM, pp. 497–500 (online), DOI: 10.1145/383259.383317 (2001).

[7] Ren, Z., Zhou, K., Li, T., Hua, W. and Guo, B.: Interactive hair rendering under environment lighting, *ACM SIGGRAPH 2010 papers*, SIGGRAPH '10, New York, NY, USA, ACM, pp. 55:1–55:8 (online), DOI: 10.1145/1833349.1778792 (2010).

[8] Sloan, P.-P., Kautz, J. and Snyder, J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments, *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '02, New York, NY, USA, ACM, pp. 527–536 (online), DOI: 10.1145/566570.566612 (2002).

[9] Yuksel, C.: HAIR Model Files, (online), available from ⟨http://www.cemyuksel.com/research/hairmodels/⟩ (accessed 2012-5-28).

[10] Yuksel, C. and Keyser, J.: Deep Opacity Maps, *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2008)*, Vol. 27, No. 2, pp. 675–680 (online), DOI: 10.1111/j.1467-8659.2008.01165.x (2008).

[11] Zinke, A., Yuksel, C., Weber, A. and Keyser, J.: Dual scattering approximation for fast multiple scattering in hair, *ACM Trans. Graph.*, Vol. 27, No. 3, pp. 32:1–32:10 (online), DOI: 10.1145/1360612.1360631 (2008).

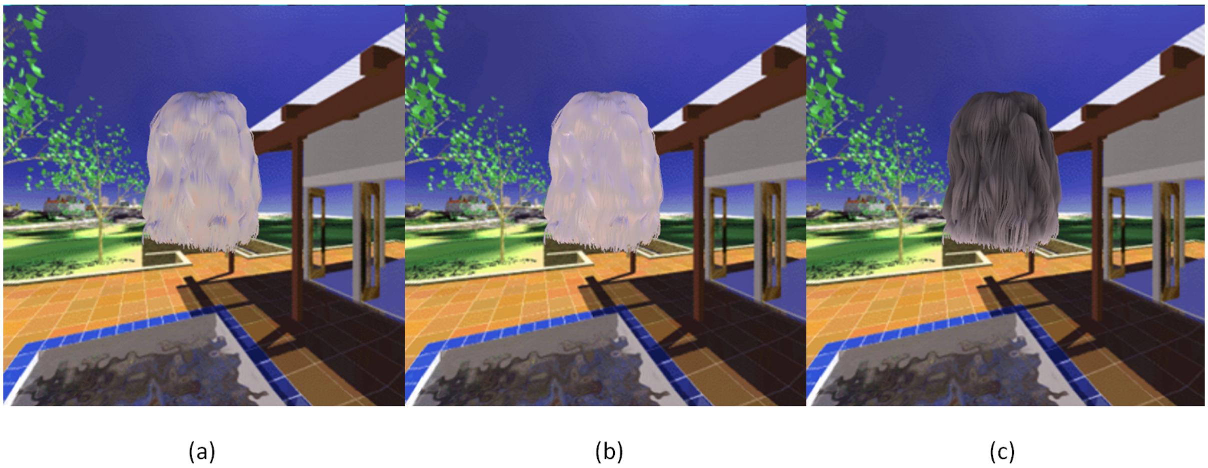

Fig. 5 Images from left to right were rendered by regular irradiance maps, irradiance maps generated by using spherical harmonics and regular irradiance maps with offline ambient occlusion.

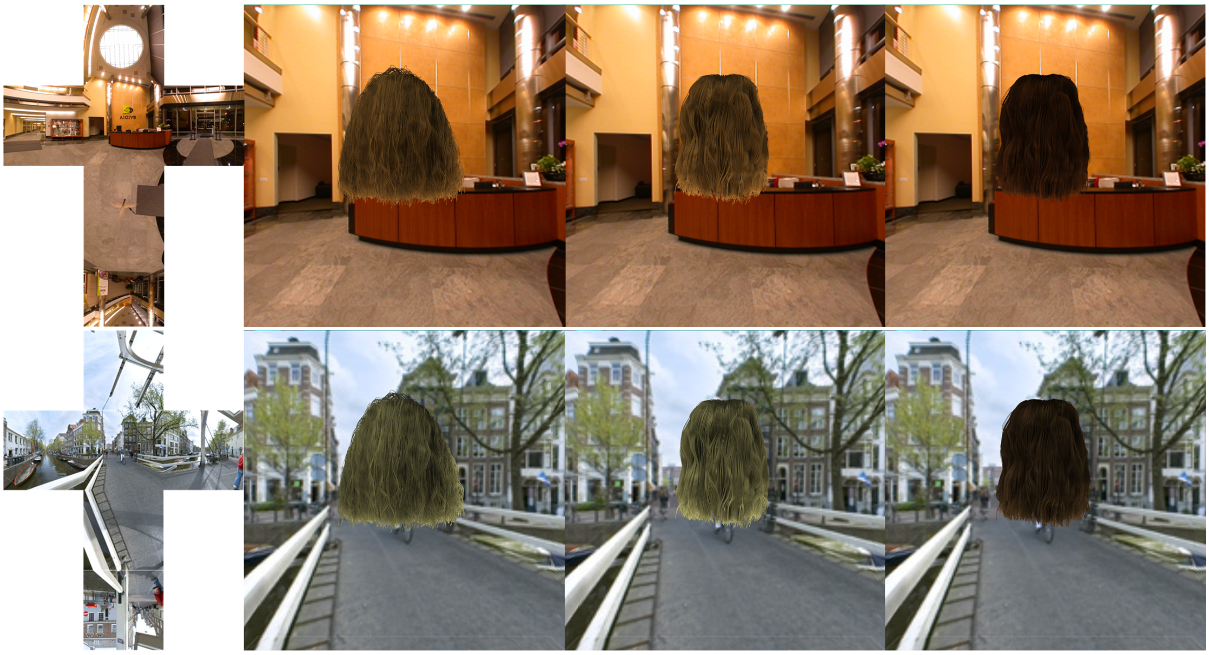

Fig. 6

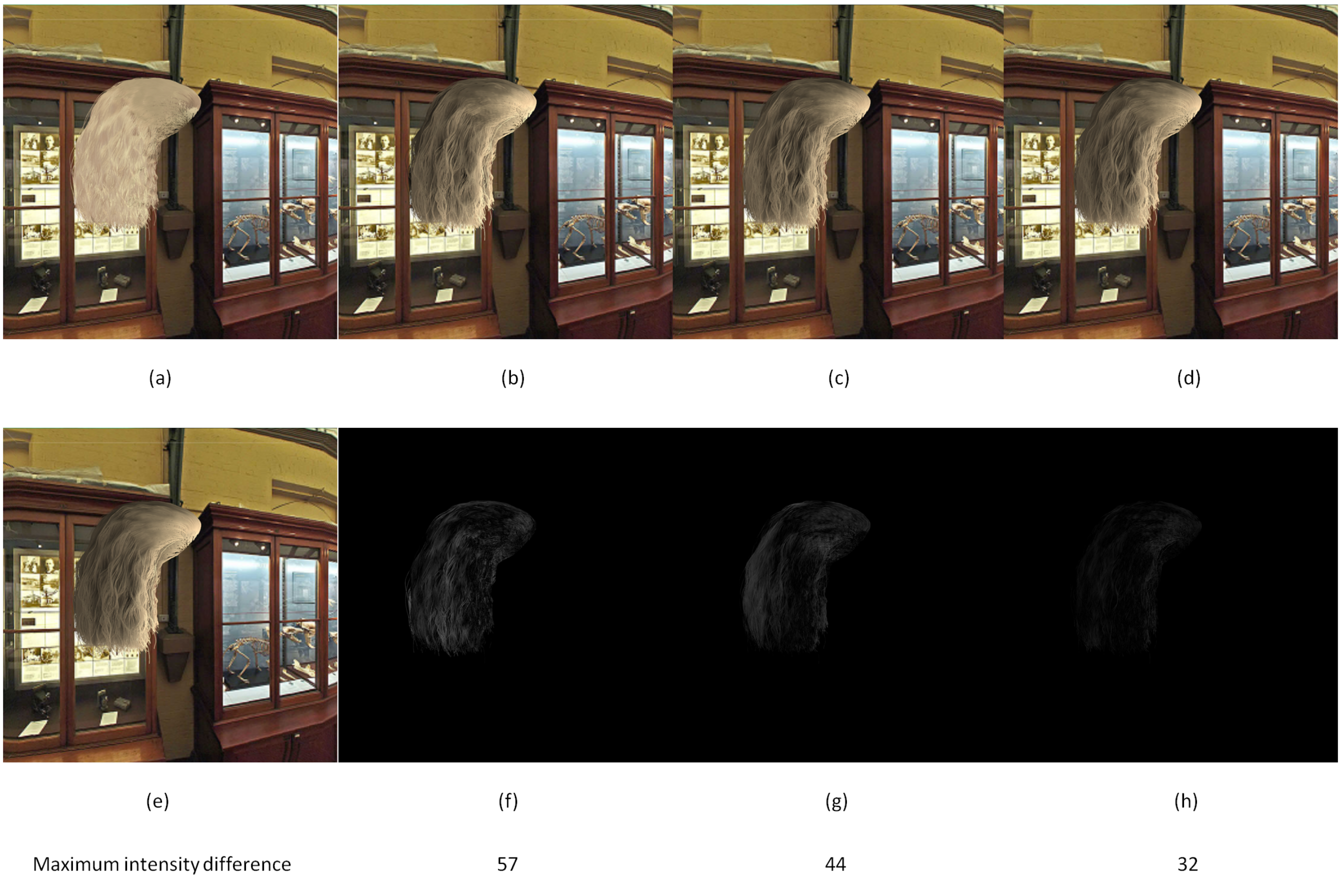

Fig. 7 (a) is the result before adding shadows. The shadows in (b), (c) and (d) were rendered by 9, 25 and 64 sample directions respectively. The shadow in (e) was rendered by 100 sample directions. (f), (g) and (h) are the difference images between (e) and (b), (c), (d). The maximum intensity in the difference images are shown in the bottom.

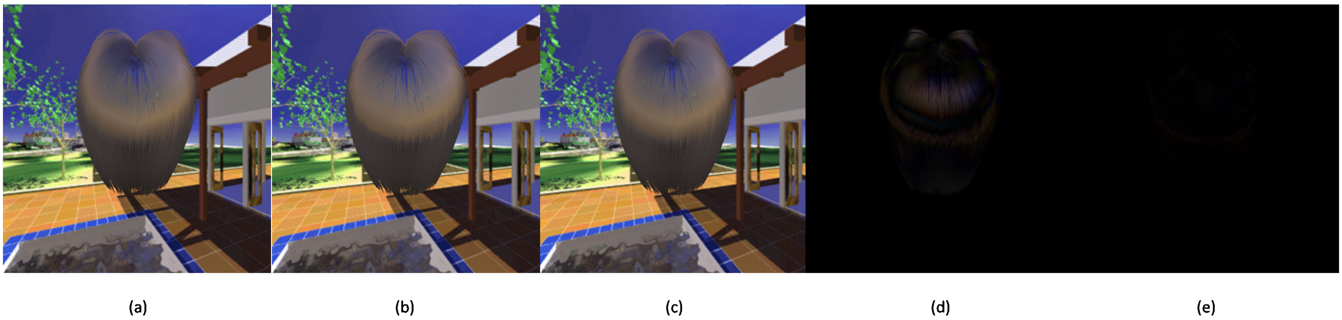

Fig. 8 (a) shows the specular component rendered by specular maps of the size $16 \times 16 \times 6$. (b) and (c) were rendered by specular maps of the size $4\times4\times6$ and $8\times8\times6$, respectively. (d) is the difference image between (a) and (b), and (e) is the difference image between (a) and (c). Both the intensity of (d) and (e) were enhanced for better view.