2014

ISSN : 0974 - 7435 Volume 10 Issue 24

BioTechnology
An Indian Journal

FULL PAPER

BTAIJ, 10(24), 2014 [15095-15100]

A wavelet based real-time rendering technology for indoor mixed reality

Fengquan Zhang*, Jiaojiao Guo, Jianfei Wan
North China University of Technology, Beijing,100141, (CHINA)
E-mail : fqzhang@ncut.edu.cn

ABSTRACT

One of the important research targets in the Augmented Reality system is the consistent illumination of virtual objects with respect to the real-world environment. It plays an important role in the mixing of virtual and real scene. After researching on the relevant issues, a real-time illumination system was developed to resolve the consistent illumination problems of the indoor mixed reality application, and the good effects have been obtained. To produce shade and shadow effect on the virtual object, the renderer utilizes pre-computed radiance transfer to calculate the wavelet coefficients, and uses these coefficients in vertex shader on GPU during rendering to achieve interactive frame rates.

KEYWORDS

Consistent illumination; PRT; Wavelet transform; Rendering; Real-time.

INTRODUCTION

Augmented Reality requires consistent illumination of virtual objects with respect to the real-world environment[1,2]. Usually, virtual light is applied for virtual objects rendering, taking no account of the real environment. Without ambient lighting information, consistent effects cannot be acquired even with reflection properties. Therefore, ambient lighting, such as real shadow, glow, reflections, is essential and add to the total rendering effects.

Real-time rendering with ambient lighting is a complex hemispheric integration including incident lighting from all direction. The model we present utilizes pre-computation and wavelet transform to reduce computation in run-time and achieves interactive change of ambient lighting and viewpoint. Compared with spherical harmonic transform, wavelet transform is rather simpler and preserve more high-frequency part in ambient lighting. Signals saved as wavelet primary function coefficient are minimal. Among other merits, they can be truncated or removed, and reconstruct original signal with little distortion[3,4].

ALGORITHM

Under ambient lighting[5,6], $I(x,\omega_0)$, radiation measurement of point x on the object surface to direction ω_0 can be expressed by integration of unit sphere Ω:

$$I(x,\omega_0) = \int_{\Omega_{2\pi}} l(x,\omega)\rho(x,\omega,\omega_0)V(x,\omega)(\omega\cdot n)d\omega \tag{1}$$

In (1), I means the object color under ambient lighting, and also the ultimate value we want to acquire. Defined by coordinate vertex x and emergent light direction ω_0, it is integration of illuminated episphere Ω against incident light L, BRDF ρ and visibility function V. Incident light L is a function of coordinate vertex X and incident light direction ω. BRDF ρ is a function of coordinate vertex x, incident light direction ω and emergent light direction ω0. Visibility function V is a function coordinate vertex x when incident light ω is 0 or 1.

We use diffuse reflection model for illumination rendering, in which BRDF is a constant expressed as$\rho(x,\omega,w_0) = Kd/\pi$. And $Kd\in[0,1]$ means superficial absorption. Besides, to simplify computation, weight cosine function is applied together with illumination when $L' = L(x,\omega)\bullet(\omega\cdot n)$. And (1) becomes:

$$\mathrm{I}(\mathrm{x},\omega_0) = \frac{\mathrm{Kd}}{\pi}\int_{\Omega_{2\pi}} \mathrm{L}'(\mathrm{x},\omega)\mathrm{V}'(\mathrm{x},\omega)(\omega\cdot\mathrm{n})\mathrm{d}\omega \tag{2}$$

Integration of two functions in (2) can be projected into appropriate orthogonal basis functions$\emptyset_{\mathrm{i}}(\omega)$, and we then have:

$$L' = \sum_i L_i\emptyset_i(\omega) \qquad V = \sum_j V_j\emptyset_j(\omega) \tag{3}$$

(2) can then be expressed as:

$$I = \frac{Kd}{\pi}\sum_i\sum_j L'_j V_j = \frac{Kd}{\pi}\sum_i L'_i V_i = \frac{Kd}{\pi}(L'\cdot V) \tag{4}$$

By now, the integration has been transformed into point multiplication of coefficient vector L'and V of two orthogonal basis functions. In this way, computation complexity can be greatly reduced.

Haar wavelet is the simplest wavelet and its primary function is generated by translation and scaling of a single mother wavelet, which is narrowed by the power of 2. Each narrowed wavelet is translated with its width as increment. In this way, one complete group of wavelets can cover the whole interval in any scale. Since mother wavelet is decreased by the power of 2, and its scale is increased by the power of $\sqrt{2}$ to secure orthonormality. All the above-mentioned factors constitute a group of orthornomal primary function.

2D Haar wavelet primary function is defined by Stollnitz as follows:

Scaling function is unit value defined on unit area : $\phi(\mathrm{x},\mathrm{y}) = 1$ for $(\mathrm{x},\mathrm{y}) \in [0,1]^2$

The 1st grade wavelet primary function is scaling and translation of 3 mother wavelet functions.

$$\psi_{\mathrm{M}}^{\mathrm{lij}}(\mathrm{x},\mathrm{y}) = 2^{\mathrm{l}}\psi_{\mathrm{M}}(2^{\mathrm{l}}\mathrm{x}-\mathrm{i}, 2^{\mathrm{l}}\mathrm{y}-\mathrm{j}) \tag{5}$$

In (5), i and j are integers valuing at $[0,2^{\mathrm{l}}]$, M is 01,10, or 11, corresponding to horizontal, vertical, and diagonal mother wavelet functions respectively.

Octahedron sampling in ambient lighting

Cube mapping is utilized to describe lighting in global coordinate system. For lighting in local coordinate system defined by vertex normal vectors, we use octahedron sampling. Parameterization of octahedron .

For one random point $P = (p_x, p_y, p_z)^T$ on the sphere with radius r, and $|p_x| + |p_y| + |p_z| = r$. Then its projection on octahedron surface can be expressed by

$$P' = \frac{p}{|p_x|+|p_y|+|p_z|} \tag{6}$$

It can then be projected to mapping coordinates on *XZ* planes by octahedrons or ogdohedrons. Octahedron projection of unit ball can then be presented as:

$$\mathrm{P}_\mathrm{r}'' = \begin{cases} (\mathrm{p}_\mathrm{x}', 0, \mathrm{p}_\mathrm{z}')^\mathrm{T} \quad \mathrm{p}_\mathrm{y}' \geq 0 \\ (\sigma(\mathrm{p}_\mathrm{x}')(1-\sigma(\mathrm{p}_\mathrm{z}'))\mathrm{p}_\mathrm{z}', 0, \sigma(\mathrm{p}_\mathrm{z}')(1-\sigma(\mathrm{p}_\mathrm{x}')) \quad \mathrm{p}_\mathrm{x}')^\mathrm{T} \quad \mathrm{p}_\mathrm{y}' < 0 \end{cases} \tag{7}$$

in which $\sigma(x)$ is a sign function. Octahedron projection of unit hemisphere can be presented as:

$$\mathrm{P}_\mathrm{r}'' = \begin{cases} (\mathrm{p}_\mathrm{x}' - \mathrm{p}_\mathrm{z}' - 1, 0, \mathrm{p}_\mathrm{x}' + \mathrm{p}_\mathrm{z}')^\mathrm{T} \quad \mathrm{p}_\mathrm{y}' \geq 0 \\ (\mathrm{p}_\mathrm{z}' - \mathrm{p}_\mathrm{x}' + 1, 0, \mathrm{p}_\mathrm{x}' + \mathrm{p}_\mathrm{z}')^\mathrm{T} \quad \mathrm{p}_\mathrm{y}' < 0 \end{cases} \tag{8}$$

In (8), mapping scale of the projection is $\mathrm{p}_\mathrm{r}'' \in [-2; 2] \times [-1; 1], \mathrm{p}_\mathrm{r}'' \in [-1; 1]^2$. So a simple linear transform is necessary for texture coordinates to the scale of $[0,1]^2$.

RENDERING

Figure is the work flow diagram of the proposed model, which consists of two primary parts, pre-computation and real-time rendering. The former can further parted into geometric model pre-processing and visibility pre-computation. The latter involves lighting information computation and model rendering. The basic modules include[7,8]:

(1) Octahedron sampling: ogdohedrons are sampled according to direction vector or vertex normal vector. A local sampling image can then be acquired corresponding to the sampling direction.

(2) 2D wavelet transform: function module of 2D wavelet transform is designed to process the local sampling image and store wavelet coefficient.

(3) As is explained, pre-computation involves pre-processing of geometric information, which are then stored in KD-Tree structure, ready for visibility computation.

(4) Real-time rendering involves simulation of specific real scene and model rendering in that scene.

(5) Specific real scene simulation with global illumination: HDR images are introduced to construct specific real scene and skybox rendering. Virtual object model are imported into the scene.

(6) Real-time model rendering: visible wavelet coefficients are extracted along the line of sight for each vertex. Lighting wavelet coefficient is acquired according to vertex normal vector. Ultimately, the color can be acquired by dot product.

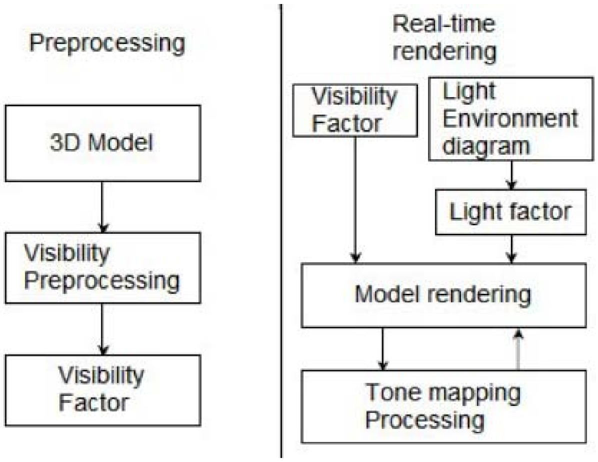


Figure 1 : Lighting rendering diagram.

We establish a K-D tree to categorize model triangles according to coordinate axis. In this way, triangle intersection verification can be reduced. KDNode, KDTree and other categorization are designed to establish the K-D tree. In the categorization of KDTree, Build member function is utilized for K-D tree construction.

Pre-computation of visibility, in other words, is to sole visibility function of its corresponding vertexes, and import the result into visibility mapping for wavelet transform. The vertex visibility is directly relevant to the generation of model shadows, which means that favorable visibility sampling produces favorable shadows.

The solving process can be seen as vertex projecting rays in the episphere where the vertex normal vector stays. If the projection ray is blocked in the space, then the corresponding sampling pixel is 0, otherwise is 1. After obtaining visibility sampling matrix, its corresponding transfer factor can be acquired by Haar wavelet transform, as is in Figure 2.

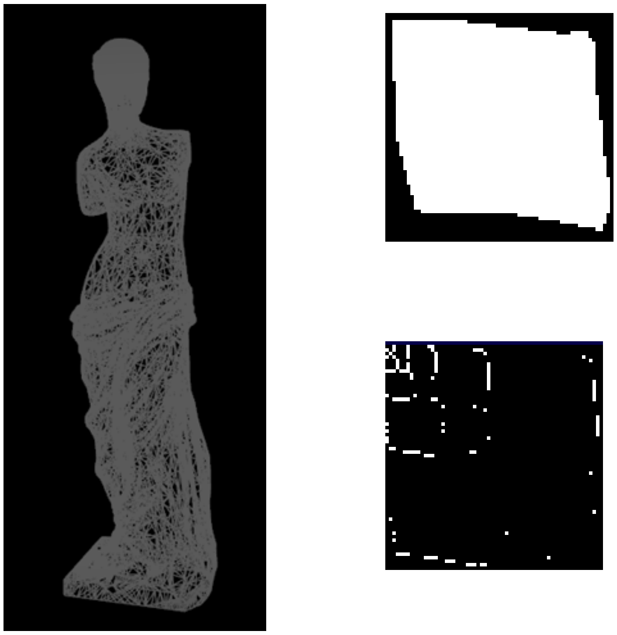


Figure 2 : Venus model(left), vertex visibility sampling image and wavelet transfer factor(right).

Illumination transfer factors are pre-computed and the result is read to compute color value of vertexes. The model can then be rendered in the real scene. The user can see the illuminated model from different angles through keyboard operation.

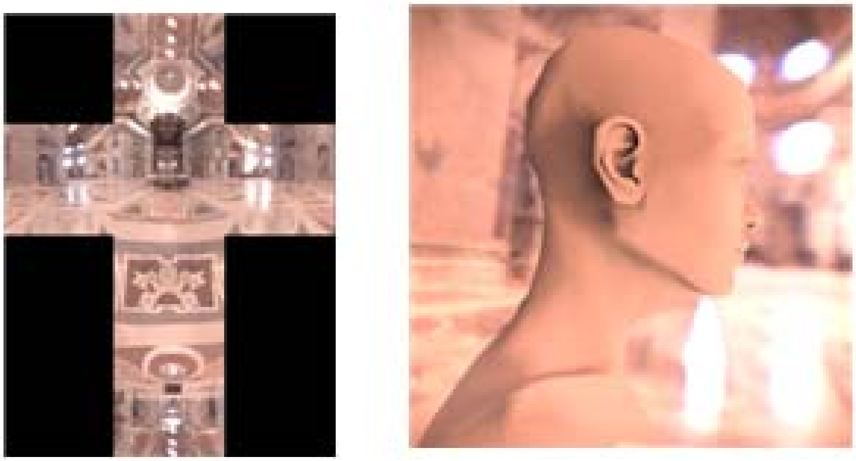


Figure 3: Indoor lighting(left) and illuminated head model (right).

In this research, we use panorama camera to acquire dynamic panorama image. More specifically, one scanning panorama camera is fixed on a rotating bracket to scan vertically in 360° of the real scene, which is then processed by HDRShop to generate dynamic cubic image. Figure 3 shows the illumination image under indoor light[9].

After octahedron sampling on each normal vector direction, each vector is also sampled, so we can have binormal vector and tangent vector with the sampled vector as coordinate axis. The binormal vector and tangent vector consist of local coordinate system. Local illumination sampling only takes account of episphere in the local coordinate system since only the ambient lighting of episphere contribute to the vertex color. The episphere in the local coordinate system is sampled by ogdohedron, with each sample representing a direction vector in the local coordinate system. After all sampled directions are transferred to the global coordinate system, the global illumination is sampled and multiply by weight factor ω•n. Then we have local ambient lighting corresponding to that normal vector. Figure 4 (top) shows the result of ogdohedron sampling on local ambient lighting. The sampling resolution is 64×64.

Then, the local illumination is processed by 2D wavelet transform to acquire wavelet coefficient. The result is shown in Figure 4 (bottom). Illumination information are computed when the user changes illumination environment. The acquired transfer factors are saved in internal storage for future model rendering.

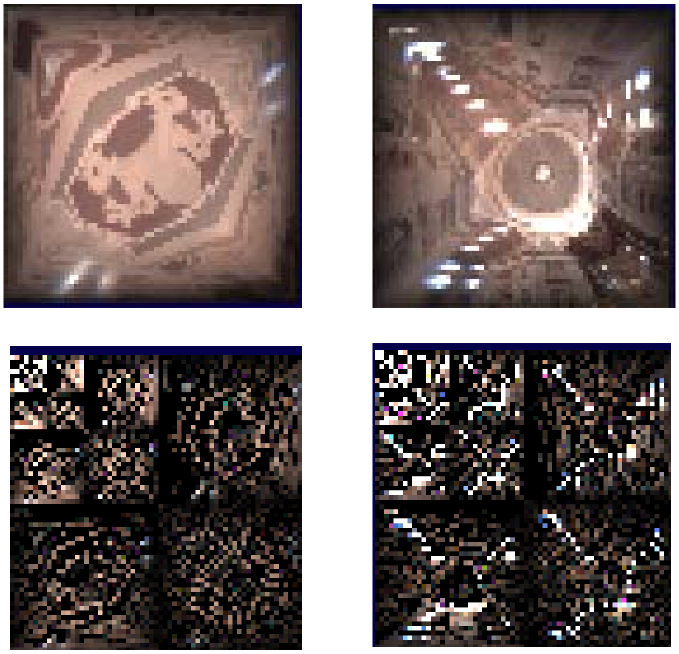


Figure 4: Illumination Pre-computation Result.

The pre-computed visibility wavelet coefficient and illumination wavelet coefficient acquired in Section 2 are loaded. In the read-in process, the vertex normal vector are mapped onto the normal vector texture mapping. Illumination wavelet coefficient can be read according to that coordinates. Visibility wavelet coefficients are indexed and each image can be read according to its corresponding vertex number.

Vertex normal vector is a vector in the global coordinate system. Otherwise it should be transferred into the global coordinate system. Its corresponding wavelet coefficient can be obtained through indexing local illumination wavelet coefficient. Since the vectors with pre-sampled wavelet coefficient are disperse, bilinear interpolation is applied on indexing results to generate wavelet coefficient of the vertex normal vector. On the other hand, visibility wavelet coefficient can be acquired through direct indexing of visibility wavelet coefficient according to the vertex number. Both illumination and visibility wavelet coefficient are 2D matrix, which can be seen as a floating number vector. The dot product of these two vectors multiply by the BRDF coefficient (previously defined as a constant) equals the vertex color value under the corresponding illumination. These values are imported into opengl for pipeline drawing and frame coordination mapping.

TEST RESULT

The algorithm is tested by real-time rendering of virtual objects under indoor global illumination. The details of testing models are summarized in Table 1.

TABLE 1 : Testing model data.

Model	No. of vertexes	No. of patches
Bunny	502	1000
Head	15943	30240
Venusm	19847	43357

With pre-computed results, we have real-time rendering of the models under illumination. Table 2 illustrates the rendering results of these models.

As is seen in Table 2, shadows projected by the model itself can be clearly seen. The illumination on the model is satisfying in the virtual reality. The illumination accuracy is relevant to the model accuracy. Figure 5 presents the shadows projected by the model itself.

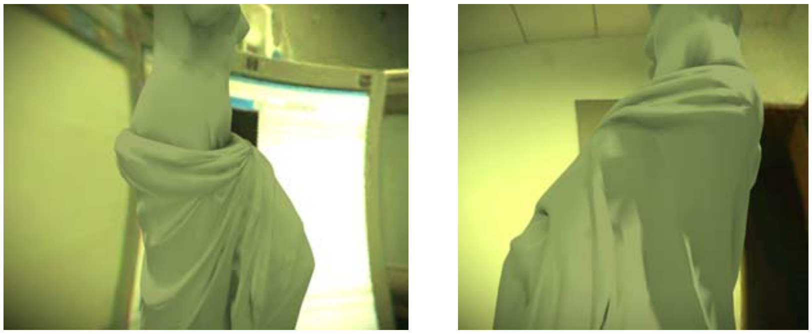
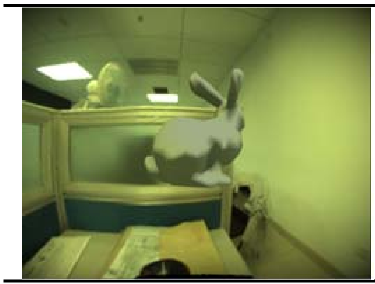
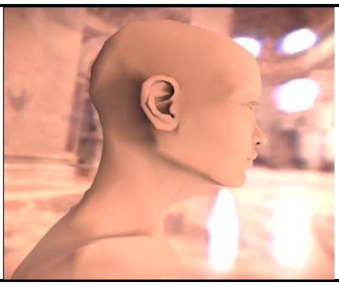
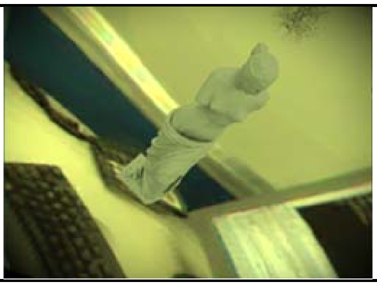


Figure 5 : Shadows projected by the model.

TABLE 1: Test result.

		
Pre-computation time: 10s	Pre-computation time: 180s	Pre-computation time: 220s
Real-time rendering speed: 70fps	Real-time rendering speed: 59fps	Real-time rendering speed: 45fps

CONCLUSIONS

The system we propose realizes consistent illumination between objects with diffuse reflection texture and the real scene. The shading effects are vividly expressed and during operation, the user can change illumination and view-point. There is still room for improvement in the system, such as rendering of smooth textures.

ACKNOWLEDGEMENT

This paper is supported by Foundation for Social Science of Ministry of Education (NO.14YJCZH200) and Foundation of PXM2014_014212_000097.

REFERENCES

[1] K. JACOBS, C. LOSCOS: Classification of illumination methods for mixed reality. Computer Graphics Forum, 25, 1 , 29–51.(2006)

[2] R. NG, R. RAMAMOORTHI, P. HANRAHAN: Triple Product Wavelet Integrals for All-Frequency Relighting. ACM Transactions on Graphics, 23, 3, 477-487. (2004)

[3] Y. XIAO, C.S. LEUNG, T. WONG: Concentric Spherical Representation for Omndirectional Soft Shadows. Computer Graphics Forum, 32, 6, 201-213. (2013)

[4] S. PESSOA, G. MOURA, J. LIMA, V. TEICHRIEB, K. JUDITH: Photorealistic Rendering for Augmented Reality: A Global Illumination and BRDF Solution. IEEE Virtual Reality Conference, 3-10. (2010)

[5] G. THOMAS, D. DANCH, A. STORK: Rendering Techniques for Mixed Reality. Journal of Real-Time Image Processing, 5, 2, 109–120. (2010)

[6] R. RAMAMOORTHI, P. HANRAHAN: All-frequency Shadows Using Non-linear Wavelet Lighting Approximation. ACM Transactions on Graphics, 22, 3, 376–381. (2003)

[7] T. HO, L. WAN C. LEUNG: Unicube for Dynamic Environment Mapping. IEEE Transactions on Visualization & Computer Graphics, 17, 1, 51–63. (2011)

[8] E. PRAUN, H. HOPPE: Spherical Parametrization and Remeshing. ACM Transactions on Graphics, 22, 3, 340-349. (2003)

[9] P. DEBEVEC: Image-based lighting. IEEE Computer Graphics and Applications, 22, 2, 26-34. (2002)