Managing Geometry Complexity for Illumination Computation of cultural heritage scenes

Romain Pacanowski, Xavier Granier and Christophe Schlick
INRIA Bordeaux Sud-Ouest – LaBRI – Université de Bordeaux
Domaine Universitaire - 351, cours de la Libération
33405 Talence Cedex - France
{pacanows|granier|schlick@labri.fr
http://iparla.labri.fr/members/

Résumé

Pour l'étude du patrimoine, de plus en plus d'objets 3D sont acquis par le biais de scanners 3D [Levoy 2000]. Les objets ainsi acquis contiennent de nombreux détails et fournissent une très grande richesse visuelle. Mais pour les affichers, leur très grande complexité géométrique nécessite l'utilisation d'algorithmes spécifiques. Nous montrons ici comment simplifier ces objets par un maillage de failble résolution et une collection de cartes de normales [Boubekeur 2005] pour préserver les détails. Avec cette représentation, nous montrons comment il est possible de calculer un éclairement réaliste à l'aide d'une grille et de données vectorielles [Pacanowski 2005]. Cette grille permet de capturer efficacement les basses fréquences d'un éclairement indirect. Nous utilisons des textures 3D (pour des gros objets) et potentiellement des textures 2D (pour les objets quasi-plan) afin de stocker un nombre pré-déterminer de vecteurs d'irradiance. Ces grilles sont calculées au cours d'un pré-calcul par toute méthode stochastique de calcul d'éclairement global. Pour l'affichage, l'éclairement indirect due à la grille est interpolé au sein de la cellule associée à la position courante, forunissant ainsi une représentation continue. De plus, cette approche vectorielle permet une plus grande robustesse aux variations local des propriétés géométriques de la scène.

Abstract

For cultural heritage, more and more 3D objects are acquired using 3D scanners [Levoy 2000]. The resulting objects are very detailed with a large visual richness but their geometric complexity requires specific methods to render them. We first show how to simplify those objects using a low-resolution mesh with its associated normal maps [Boubekeur 2005] which encode details. Using this representation, we show how to add global illumination with a grid-based and vector-based representation [Pacanowski 2005]. This grid capture efficiently low-frequency indirect illumination. We use 3D textures (for large objects) and 2D textures (for quasi-planar objects) for storing a fixed set of irradiance vectors. These grids are built during a preprocessing step by using almost any existing stochastic global illumination approach. During the rendering step, the indirect illumination within a grid cell is interpolated from its associated irradiance vectors, resulting in a smooth everywhere representation. Furthermore, The vector-based representation offers additional robustness against local variations of geometric properties of a scene.

1 - Introduction

2 – Geometric simplification

Before any lighting computation, and in order to reduce the computational complexity to fit the 3D model with available memory, we need to reduce the number of polygons without loosing the visual richness of original models. For this purpose, we use a mixed representation, based on a simplified geometry for the definition of overall shape and topology, and normal maps for preserving their local variations on the surface. Since the lighting mostly depend on the normal, this approach preserves the objects' appearance.

The large size of 3D models is not only a problem for the lighting simulation. We have also to deals with the fact that the main memory would not be sufficient to load the complete model for the simplification process. We need to use out-of core solutions. We thus decomposed our algorithm in four steps (more detailed presentation of this approach can be found in the paper [Boubekeur 2005]):

1. We perform an out-of-core simplification of the huge model using a uniform resampling [Lindstrom 2000, Boubekeur 2005]. A non-uniform and adaptive solution [Boubekeur 2006] could be also used.
2. The resulting simplified point set is quickly converted into a triangle mesh [Boubekeur 2005, 2006] and organized in a bounding box tree-hierarchy.
3. Each leave of this tree hierarchy is associated with a quadrilateral texture. All the points of the original model are thus streamed through this tree and distributed to their corresponding leaves, where the point normal is projected into the associated texture. This streaming process is the key step of our technique, as it allows us to handle large models with limited memory. At the end of this step, each triangle of the low-resolution mesh is associated with a high-resolution (and possible sparse) normal map.
4. Since there is no guaranty that each texel of the normal map corresponds to an existing point and thus an existing normal, the resulting map can contains some holes. We thus use a diffusion algorithm, to get a continuous normal field, interpolating the original normals of the huge model.

Thanks to this out of core simplification, the memory requirement is largely reduced while preserving most of the appearance details (see Figure X). Lighting can be computed on such a representation, but an adapted approach has to be developed.

Figure X: Original models and their simplified versions. Note the largely reduce size and the preserved appearance.

3 – Illumination Computation for Complex Geometry

In computer graphics, illumination computation has been extendily studied. Although direct illumination of an object is easily computed by suming the contribution of every light source seen from the object, the indirect lighting is a more challenging task. Indeed, when considering indirect illumination, every objects in the scene plays potentially the role of a light source. Therefore, many algorithms (cf. [Dutré 2006]) have been developed over the last decay to perform indirect illumination computation. Among these algorithms, some of them use precomputed specific structures that cache the indirect illumination. As pointed out by Tabellion and Lamorlette [Tabellion 2004], caching efficiency generally diminishes with the increasing geometric complexity of a 3D scene.

Main previous techniques are :

Therefore, we focus in this article only on structures that store the indirect lighting when using complex geometry.

In order to reduce this problem, this paper introduces a volumetric representation for indirect lighting based on irradiance vectors~\cite{Arvo1994}. From the irradiance vector, our representation inherits robustness against local variations of both photometric properties (diffuse component of reflectance) and geometric properties (surface normal vectors). From the volumetric structure and the associated interpolation scheme presented in this paper, we guarantee a smooth reconstruction everywhere in the 3D scene, making the cache representation directly accessible for the final rendering. Finally, our overall structure has low memory requirements and thus increases the scalability of the method. Furthermore, these qualities make this approach suitable for interactive hardware rendering.

3-1 Irradiance Vector Grid (IVG)

Our structure is based on an axis-aligned uniform rectangular 3D grid, divided into $N_i\times N_j \times N_k$ voxels. %This grid can be considered as the highest level. At each vertex $\bm{v}^{ijk}$ of the grid (where $i \in [0,N_i],\, j \in [0,N_j],\, k \in [0,N_k])$, six irradiance vectors are stored, one for each main direction ($\pm\bm{x}|\pm\bm{y}|\pm\bm{z}$). Note that we actually store an irradiance matrix, as one vector is used for each color channel. In the remaining of this paper, we will note $\bm{I}^{ijk}_{\bm{\delta}}$, the irradiance vector stored at vertex $\bm{v}^{ijk}$ in the direction $\bm{\delta}$ where $\bm{\delta} = \pm\bm{x}|\pm\bm{y}|\pm \bm{z}$.

3-2 Irradiance VECTOR

For a given wavelength, the \emph{irradiance vector}

$\bm{I}_{\bm{n}}(\bm{p})$, as introduced by Arvo~\cite{Arvo1994}, is

defined for a point $\bm{p}$ with normal $\bm{n}$ as

%

\begin{equation*}

\bm{I}_{\bm{n}}(\bm{p}) = \int_{\Omega_{\bm{n}}}\,L(\bm{p} \leftarrow \bm{\omega}_{i})\,\bm{\omega}_{i}\,d\bm{\omega}_{i}\,

\end{equation*}

%

where $L(\bm{p} \leftarrow \bm{\omega}_{i})$ represents the incident

radiance at $\bm{p}$ from direction $\bm{\omega}_{i}$, $d\bm{\omega}_{i}$

the differential solid angle sustained by $\bm{\omega}_{i}$ and

$\Omega_{\bm{n}}$ the hemisphere centered at $\bm{p}$ oriented toward

$\bm{n}$. The irradiance vector stores a radiometric and geometric

information and is directly related to the diffusely reflected

radiance:

\begin{equation}

L_{r}(\bm{p} \rightarrow \bm{\omega}_{o}) = \frac{\rho_{D}(\bm{p})}{\pi}\,\langle\bm{I}_{\bm{n}}(\bm{p}),\bm{n}\rangle

\label{reflRadEq}

\end{equation}

where ρ_{D} is the diffuse BRDF and $\langle,\rangle$ denotes a

dot product. The main benefits of irradiance vectors compared to

irradiance is that for a local variation of the normal, the reflected

radiance can be adjusted, making this representation more

geometrically robust. Irradiance Vectors are precomputed using any classical algorithm such as Photon-Tracing or Monte-Carlo sampling.

3-3 IRRADIANCE VECTOR INTERPOLATION

In order to compute smooth indirect illumination, we

interpolate an irradiance vector for each point $\bm{p}$ with normal $\bm{n}$

that needs to be shaded. This interpolation

is performed in two successive steps: a spatial interpolation according to $\bm{p}$

and then a directional interpolation according to $\bm{n}$.

In the first step, the irradiance vector $\bm{I}_{\bm{\delta}}(\bm{p})$

is obtained by spatial interpolation of the irradiance vectors

$\bm{I}^{ijk}_{\bm{\delta}}$ stored at

the grid vertices surrounding point $\bm{p}$. The interpolation is

only done for three out of the six possible directions of

$\bm{\delta}$. The choice between $\bm{\pm x}$ (resp. $\bm{\pm y}$ and

$\bm{\pm z}$) is done according to the sign of n_x

(resp. n_y and n_z). Trilinear or tricubic interpolation approximate

satisfactory smooth results for spatial interpolation.

In the second step, the final interpolated irradiance vector

$\bm{I}_{\bm{n}}(\bm{p})$ is obtained by remapping the three spatially interpolated irradiance vectors according to the normal direction $\bm{n}$ at point $\bm{p}$:

$$
\bm{I}_{\bm{n}}(\bm{p}) =
\bm{I}_{\bm{x}}(\bm{p})\,n_{x}^{2}
+\bm{I}_{\bm{y}}(\bm{p})\,n_{y}^{2}
+\bm{I}_{\bm{z}}(\bm{p})\,n_{z}^{2}\,.
$$

3-4 Hardware Implementation

The great benefit of using a 3D regular grid is that the data structure can be straightforwardly uploaded on GPU as a 3D texture. In our case, the interpolated irradiance vectors are simply used by the fragment shader as additional light sources that are meant to encode indirect illumination.. Remember that each grid vertex holds one irradiance vector $\bm{I}_{\lambda}$ per color channel for each of the six $\bm{\delta}$ directions, i.e., $3 \times 3 \times 6 = 54$ floating point numbers.To reduce the number of texture fetches which may be costly in current graphics hardware, we compress the 3 irradiance vectors as one color and one direction:

$$r = \| \bm{I}_R \| \hspace*{0.15cm}
g = \| \bm{I}_G \| \hspace*{0.15cm}
b = \| \bm{I}_B \|\hspace*{0.5cm}
\bm{d} = \frac{\bm{I}_R + \bm{I}_G + \bm{I}_B}
{\| \bm{I}_R + \bm{I}_G + \bm{I}_B \|}.$$

These two vectors are encoded in two 16 bit 3D textures,and therefore the information for the six $\bm \delta$ directions require 12 3D textures.

4 – Results

To illustrate the geometric robustness of our structure we compare the indirec illumination obtained when using our structure precomputed with a geometry at full resolution (cf. Figure) vs a geometry at low resolution (cf. Figure). As illustrated in Figure the perceptive difference is very low with a Lab of only . Figure shows a complete scene where the indirect illumination is stored with our IVG.

Conclusion and Future Works

Références

Boubekeur T., Duguet F., and Schlick C. 2005: Volume-Surface Trees. In *EUROGRAPHICS VAST 2005*, 75-82.

Boubekeur T., Heidrich W., Granier X. and Schlick C. 2006: Rapid visualization of large point-based surfaces. In *Computer Graphics Forum (Proc. EUROGRAPHICS 2006)*, 25(3), 399-406.

Dutré P., Bala K. and Bekaert P. 2006: Advanced Global Illumination. A.K. Peters, Ltd.

Levoy M., Pulli K., Curless B., Rusinkiewicz S., Koller D., Pereira L., Ginzton M., Anderson S., Davis J., Ginsb., Shade J., and Fulk D. 2000: The digital Michelangelo project: 3D scanning of large statues. In *ACM SIGGRAPH '00*, 131–144.

Lindstrom P. 2000: Out-of-Core Simplification of Large Polygonal Models. In *ACM SIGGRAPH '00*, 259-262.

Pacanowski R., Granier X., and Schlick C. 2005: Nouvelle représentation directionnelle pour l'éclairage global. In *Actes des 18ièmes Journées de l'Association Française d'Informatique Graphique (AFIG).*

Tabellion E. and Lamorlette A. 2004: An Approximate Global Illumination System for Computer Generated Films. In *ACM SIGGRAPH '04,* 469--476.