

# Master-Element Vector Irradiance for Large Tessellated Models

Grégory Lecot\*  
Project ALICE  
CNRS / VSP-Technology

Bruno Lévy†  
Project ALICE  
INRIA Lorraine

Laurent Alonso‡  
Project ALICE  
INRIA Lorraine

Jean-Claude Paul§  
Tsinghua University



Figure 1: Left: Master-Element global illumination applied to “David” (500K $\Delta$ ). Center: Isis model lighted with an area light (375K $\Delta$ ). Right: A scene with five data sets (1.5M $\Delta$ ). The Master-Element method decouples the representation of the lighting from the geometry. Sharp shadows are accurately captured by adaptative wavelets, represented in parameter-space.

## Abstract

We propose a new global light simulation method for diffuse (or moderately glossy) scenes comprising highly tessellated models with simple topology (e.g., scanned meshes). By using the topological coherence of the surface, we show how to extend a classic Finite Element method called the Master Element: We generalize this method to efficiently handle tessellated models by using mesh parameterization and mesh extrapolation techniques. In addition, we propose a high-order and hierarchical extension of the Master Element method. Our method computes a compact representation of vector irradiance, represented by high-order wavelet bases. For totally diffuse scenes, the so-computed vector irradiance maps can be transformed into light maps. For moderately glossy scenes, approximated view-dependent lighting can be computed and displayed in real-time by the GPU from the vector irradiance maps. Using our methods, view-dependent solutions for scenes with over one million polygons are computed in minutes and displayed in real time. As with clustering methods, the time complexity of the method is independent on the number of polygons. By efficiently capturing the lighting signal at a suitable scale, the method is made independent of the geometric discretization and solely depends on the lighting complexity. We demonstrate our method in various settings, with both sharp and soft shadows accurately represented by our hierarchical function basis.

**CR Categories:** I.3.3 [Computer Graphics]: Picture and Image Generation—; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—

**Keywords:** radiosity, global illumination, finite elements, master-element, vector-irradiance

\*e-mail: gregory.lecot@loria.fr

†e-mail: bruno.levy@loria.fr

‡e-mail: laurent.alonso@loria.fr

§e-mail: paul@tsinghua.edu.cn

## Introduction

Global illumination may be considered in terms of computing the interactions between the *lighting signal* and the *geometric signal* (i.e. the scene). These interactions occur at various *scales*. To deal with this issue, the *signal processing* family of approaches is a well-suited formalism. As such, the *multi-scale* approach is a natural choice, which dramatically improves performance. When computing an energy transfer, the main idea consists in considering the problem at a suitable scale. For large-scale variations of lighting, grouping primitives makes computations more efficient in uniform zones. For small-scale variations, adaptively subdividing primitives enables capturing lighting variations of high frequency.

In classic finite element approaches, a finite element is attached to each facet of the scene. As a consequence, the representation of the geometry and the lighting signal are tightly coupled. To overcome this limitation, *clustering* considers the problem at various scales, and computes lighting transfers between adaptively refined groups of primitives (called clusters). In this paper, we show an alternative method, well suited to highly tessellated models with topological coherence (e.g., scanned meshes). As shown further, recent advances in digital geometry processing make it possible to adapt the *master element* formalism to meshed models. Our method uses a parameterization of the meshed model and computes vector-irradiance represented by higher-order wavelet bases in parameter space.

The remainder of this paper is organized as follows. The next section reviews the previous work in Finite-Element based global illumination and clustering. Section 2 introduces our problem setting, and presents the generic vector-irradiance solution mechanism. In section 3, we present our Meshed Master Element representation that decouples the lighting from the geometry, and show how to instantiate the generic vector irradiance solution mechanism with our representation. Section 5 gives some results and statistics.

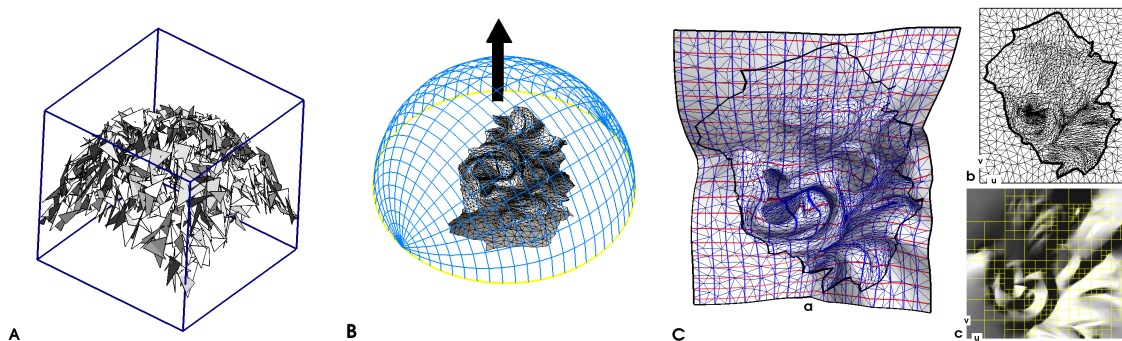


Figure 2: A: volume clustering is best suited to unstructured objects, such as plants. B: facet clustering keep more geometric information, by grouping triangles that approximatively share the same normal. C: our method may be thought of as an evolution of these clustering methods, that preserves all the geometrical information. Each chart (a) is provided with a parameterization (b), and illumination can be represented in parameter-space by high order wavelets (c). This decouples the lighting representation from the geometric support.

## 1 Previous work

### Finite-Element based global illumination

The finite-element radiosity method was first introduced in [Goral et al. 1984]. It was later analyzed in [Zatz 1993] in terms of the Galerkin method applied to the radiosity equation. The radiosity equation is projected onto a function basis, and the resulting linear system is solved by an iterative method (Southwell). To improve performance, a hierarchical version of the method was introduced in [Hanrahan et al. 1991]. It adaptively refines the surfaces where needed according to the radiosity function variations. The hierarchical method was then formalized using wavelet function bases in [Gortler et al. 1993] and [Schroeder 1994]. To compute the integrals corresponding to the coefficients of the projected equation, different quadrature rules can be used (see [Troutman and Max 1993]). The hierarchical method was also used in [Tobler et al. 1997] based on a stochastic calculation of the global illumination.

### Clustering

Clustering approaches have appeared as a promising avenue to simplify the geometry of space. The foundation of clustering approaches is the Fast Multipole Method (FMM) approach [Greengard. 1988], initially proposed as an efficient way of solving the N-Body physical problem. FMM uses a hierarchical decomposition of the computational domain as well as higher-order series of expansions to group sets of sufficiently distant particles into “super-particles”. *Volume clustering* approaches [Smits et al. 1994], [Sillion et al. 1995], [Sillion 1995], [Gibson and Hubbard 1996], [Christensen et al. 1997] exploit this idea, by grouping primitives into volumes (see Figure 2-A). However, since FMM volume cluster are meant to approximate “well separated” bodies, they fail to account for the topology of surfaces in a geometric 3D model. Consequently, volume clustering approaches require a costly final gather pass. The *face cluster* approach [Willmott et al. 1999], [Garland et al. 2001] preserves more geometric information, by grouping facets having approximatively the same normal (see Figure 2-B). Since the *field irradiance* is smoother than the radiosity function, clustering approaches use a representation of irradiance, and recast the radiosity algorithm in terms of irradiance. In [Gobbetti et al. 2003], face clustering was extended to higher-order functions and vector irradiance. Although they introduce significant improvements in terms of performance and accuracy, face cluster methods

may introduce artifacts (visible as discontinuities between the clusters) and cannot represent lighting details finer than the original mesh resolution. Clustering methods allow to used hierarchical algorithms on any kind of surfaces but at the cost of geometric approximations often consisting of a hierarchy of bounding box.

### Curved objects

Note that the hierarchical radiosity methods mentioned above are limited to planar objects. To extend their scopes to curved objects, several methods were proposed. [Schäfer 1997] propose a clustering method adapted to curved objects. They dynamically create the clusters by adaptively tessellating the input geometry. In [Stamminger et al. 1997], a similar method is developed, with the addition of error bounds directly computed from the input geometry. In [Christensen et al. 1996], the lighting is decoupled from the geometry by storing an irradiance field over a discretized hemi-sphere for each cluster. For the case of parametric surfaces, [Alonso et al. 2001] proposes an alternative approach. They store radiosity in parameter space, and proposes accurate expression of the push-pull coefficients.

Our method shares some similarities with [Alonso et al. 2001]. The main differences is that our method operates on tessellated models, and computes vector irradiance instead of radiosity. Since the vector irradiance function is smoother than the radiosity function, this both decreases storage requirements and improves computation times.

## 2 Vector irradiance

Before introducing our Master-Element representation, we give the basic definitions, present the problem setting and outline the general solution mechanism. In this section, the only assumption we make is that the used function basis is orthonormal. The next section shows how to define the function basis over a triangulated surface.

### Problem setting

The radiosity  $B(\mathbf{x})$  in a non-participating environment composed of emitters and diffuse reflectors is described by the following integral

equation:

$$\begin{aligned} B(\mathbf{x}) &= B^e(\mathbf{x}) + \rho(\mathbf{x}) \int_A B(\mathbf{y}) V(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) dA_{\mathbf{y}} \\ \text{where:} \\ G(\mathbf{x}, \mathbf{y}) &= ((\mathbf{y} - \mathbf{x}) \cdot \mathbf{n}_{\mathbf{x}})_+ ((\mathbf{x} - \mathbf{y}) \cdot \mathbf{n}_{\mathbf{y}})_+ / \pi \|\mathbf{x} - \mathbf{y}\|^4 \end{aligned} \quad (1)$$

where  $V(\mathbf{x}, \mathbf{y})$  is one if  $\mathbf{x}$  is visible from  $\mathbf{y}$  and zero otherwise,  $\rho(\mathbf{x})$  is the diffuse reflectivity at point  $\mathbf{x}$ , and  $G(\mathbf{x}, \mathbf{y})$  is the geometry factor. In operator form (see e.g. [Arvo 1995]), this equation can be rewritten as:

$$\begin{aligned} (I - L)B &= B^e \\ \text{where: } LB(\mathbf{x}) &= \rho(\mathbf{x}) \int_A B(\mathbf{y}) V(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) dA_{\mathbf{y}} \end{aligned} \quad (2)$$

### Variational approximation

We apply Galerkin's method to Equation 2, using a function basis  $(\Phi_i)$ . In what follows, the functions  $(\Phi_i)$  are linear and quadratic wavelets. This yields the following linear system:

$$M\alpha = \mathbf{b} \quad \text{where:} \quad \begin{cases} a_{i,j} &= \langle (I - L)\Phi_i, \Phi_j \rangle \\ &= \|\Phi_i\|^2 - \langle L\Phi_i, \Phi_j \rangle \\ b_i &= \langle B^e, \Phi_i \rangle \end{cases} \quad (3)$$

where  $\langle f, g \rangle = \int_A f(\mathbf{x})g(\mathbf{x})dA_{\mathbf{x}}$  denotes the classic inner product in function space, and where  $\|f\|^2 = \langle f, f \rangle$ .

### Solution mechanism

Equation 3 is solved using the Southwell iterative method (also called *shooting*), as follows:

```

 $\alpha \leftarrow 0$  ;  $\mathbf{r} \leftarrow -\mathbf{b}$ 
while  $\|\mathbf{r}\| > \varepsilon$ 
   $i \leftarrow \text{argmax}(|r_i|)$ 
   $\alpha_i \leftarrow \alpha_i + r_i$ 
   $S \leftarrow r_i$ 
  for all  $j$ 
     $r_j \leftarrow r_j + S \langle L\Phi_j, \Phi_i \rangle$ 
  end//for
end//while

```

The hierarchical version with push-pull and adaptive subdivision criteria is explained in the next section.

Instead of solving for radiosity  $B = \sum \alpha_i \Phi_i$ , we use the same strategy as in clustering approaches (see e.g., [Gobbetti et al. 2003]). We solve for another set of variables, referred to as *vector irradiance*  $\mathbf{E}$ , and linked to the radiosity  $B$  by the following relation:

$$B(\mathbf{x}) = \rho(\mathbf{x}) (\mathbf{n}_{\mathbf{x}} \cdot \mathbf{E}(\mathbf{x}))_+$$

The vector irradiance  $\mathbf{E}$  is an approximation of the field irradiance. In contrast with the radiosity  $B$ , the expression of the vector irradiance  $\mathbf{E}$  does not contain the dot-product with the normal of the surface. As a consequence,  $\mathbf{E}$  is a smoother function, that requires less processing time to be computed than the radiosity  $B$ . This is confirmed by our statistics in Section 5.

We represent the vector irradiance  $\mathbf{E}$  in a function basis  $(\Psi_i)$ :

$$(E^x, E^y, E^z) = (\sum \beta_i^x \Psi_i, \sum \beta_i^y \Psi_i, \sum \beta_i^z \Psi_i)$$

Then, we proceed to recast the Southwell solution mechanism in terms of vector irradiance. In practice, this means finding the initial values and the updating schemes for the variables  $\beta_i^x, \beta_i^y, \beta_i^z$  and the residual  $r_i^x, r_i^y, r_i^z$  that reproduce the initial values and updating schemes of their radiosity counterparts  $\alpha_i$  and  $r_i$ :

```

 $\beta \leftarrow 0$ 
for all  $i$ 
   $\mathbf{r}_i \leftarrow \int_A B^e(\mathbf{x}) (\mathbf{n}_{\mathbf{x}} \cdot \Psi_i(\mathbf{x}))_+ / \rho(\mathbf{x}) dA_{\mathbf{x}}$ 
end//for
while  $\|\mathbf{r}\| > \varepsilon$ 
   $i \leftarrow \text{argmax}(|r_i|)$ 
   $S \leftarrow \mathbf{r}_i$ 
   $\beta_i \leftarrow \beta_i + \mathbf{r}_i$ 
  for all  $j$ 
     $\mathbf{r}_j \leftarrow \mathbf{r}_j + S \int_R \int_E \rho(\mathbf{y}) \mathbf{m}(\mathbf{y}, \mathbf{x}) \mathbf{n}_{\mathbf{y}} \Psi_j(\mathbf{y}) dA_{\mathbf{x}} dA_{\mathbf{y}}$ 
  end//for
end//while

```

where  $\mathbf{m}(\mathbf{y}, \mathbf{x})$  denotes the transport vector between  $\mathbf{x}$  and  $\mathbf{y}$ , defined by  $\mathbf{m}(\mathbf{y}, \mathbf{x}) = V(\mathbf{x}, \mathbf{y}) \frac{((\mathbf{x} - \mathbf{y}) \cdot \mathbf{n}_{\mathbf{y}})_+}{\pi \|\mathbf{y} - \mathbf{x}\|^4} (\mathbf{y} - \mathbf{x})$ .

Now that we have introduced the general problem setting and solution mechanism, we proceed to show how to define wavelet bases over charts grouping a large number of facets. Then we proceed to describe the numerical integration, push-pull and adaptive subdivision strategies.

## 3 Master Element

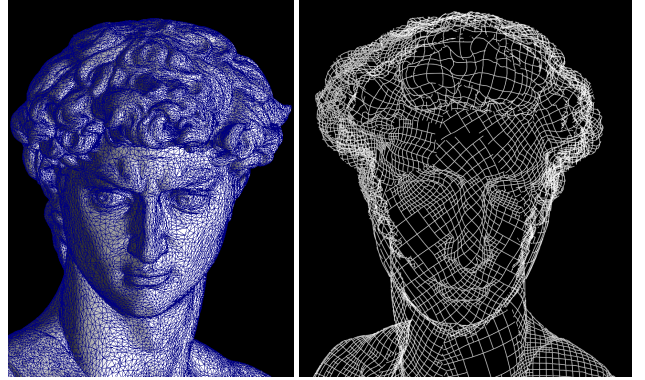


Figure 4: The mesh of the “David” dataset and the decoupled wavelet control mesh stored in parameter-space.

The main restriction in classical Finite-Element Methods is their restriction on regular elements (squares and triangles) as supports to compute polynomial approximations. However, FEM theory already gives a more general solution. Given a meshed model, made of triangular or quadrangular elements, it is possible to instantiate multiple times an idealized element, called the Master Element, residing in a normalized coordinate system [Zinkiewicz and Taylor 1989]. To our knowledge, the Master Element concept was first mentioned in the context of global illumination in [Cohen and Wallace 1993]. In this context, it was only used to represent simple geometric transforms, mapping flat surfaces onto the unit square. The *Virtual Mesh* [Alonso et al. 2001] generalizes the use of the master element method to define radiosity on quadrics using a parameterization of the quadric.



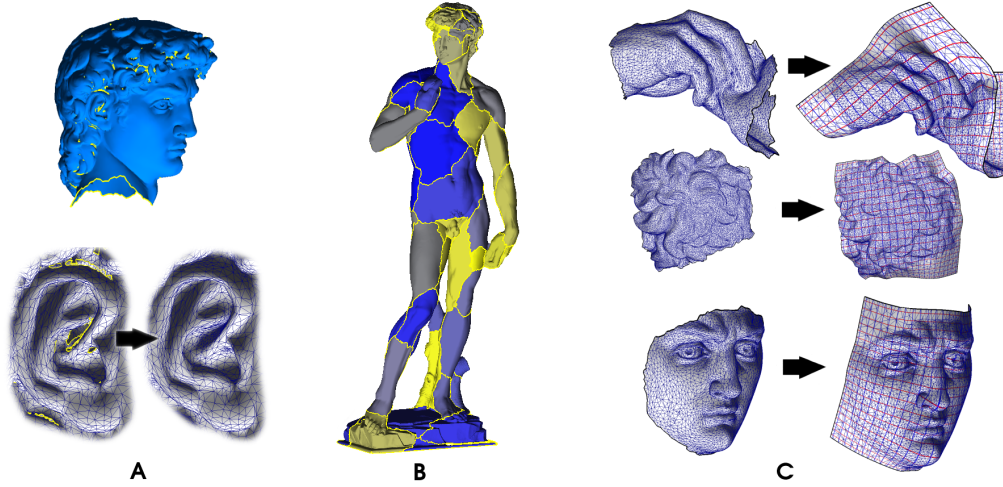


Figure 3: Master Element Construction. A: scanned meshes often have small holes and irregularities (yellow). They can be repaired using simple heuristics; B: the object is segmented into charts, using a discrete counterpart of centroidal Voronoi diagrams, driven by an estimate of the curvature; C: each cluster is parameterized and extrapolated. This puts it in correspondence with a square domain, where high-order wavelets are stored.

This is similar to our approach, with the difference that we consider tessellated surfaces and vector irradiance. In this context, instead of using a finite element per facet, we decompose the model into topological discs (or *charts*), and associate an element with each chart. Since each chart comprises a large number of facets, this representation dramatically improves performance. However, when considering a chart, in contrast with a parametric surface, no parameterization is known *a priori*. As shown in the next section, recent advances in mesh parameterization enable to deal with this issue, and instantiate the master element on a tessellated chart.

Note that the curvilinear  $u, v$  coordinates yielded by the parameterizations need to satisfy some differential properties, depending on the PDE to solve. In our context, to accurately manipulate the energy in canonical space, the Jacobian of the parameterization needs to be constant over each element. Moreover, it is well known that numerical computations are more stable if the parameterization approximates an isometric mapping. Ensuring this property has the following advantages. First, it gives a natural way of extending traditional FEM techniques to complex primitives; second, it simplifies the computation of push-pull coefficients on parametric surfaces. As shown in [Alonso et al. 2001], using an energy-preserving parameterization, they only involve a double sum of coefficients in the general case; and third, most importantly, the push-pull coefficients do not depend on the hierarchy level. They can thus be computed and stored at the top level of the hierarchy. Note that to define optimal quadratures and maintain coherence among the levels of the hierarchy, the parametric domain needs to be the unit square. The next section shows how to install a set of square master elements on a meshed model.

### 3.1 Construction

This section presents a *fully automatic* procedure to convert a tessellated model into a set of master elements. The model is decomposed into *charts*, i.e. parts that are homeomorphic to discs. Each chart is then transformed into a master element, referred to as a Meshed Master Element (MME) in what follows. The method consists of four distinct steps, involving recent results in Digital Geometry Processing. Robust implementations of the required basic tools

are now available on the web [Fabri et al. 2000],[Graphite 2003].

1. **Model repairing** (Figure 3-A): It is well known that finite element methods require a clean geometry. However, scanned meshes are often “punctured” by tiny holes. For this reason, we propose here a simple method to “clean” the geometry before applying subsequent steps. To fill a hole, edges are iteratively inserted. Each new edge connects the pair of vertices  $(\mathbf{p}_i, \mathbf{p}_j)$  that maximizes the ratio  $(s_i - s_j)/d(\mathbf{p}_i, \mathbf{p}_j)$ , where  $d(\mathbf{p}_i, \mathbf{p}_j)$  denotes the 3D Euclidian distance between  $\mathbf{p}_i$  and  $\mathbf{p}_j$ , and where  $s$  denotes a curvilinear coordinate measured along the border.
2. **Segmentation** (Figure 3-B): We use an approach inspired by [Sander et al. 2003] to decompose the model into charts. To facilitate the parameterization step, our goal is to improve the curvature balance in the charts. The approach proposed in [Sander et al. 2003] may be thought of as a discrete version of Centroidal Voronoi Diagrams [Lloyd 1982]. In this context, balancing the curvature in the charts can be achieved by using a curvature estimator to drive the algorithm. We use the curvature estimator proposed in [Cohen-Steiner and Morvan 2003].
3. **Parameterization and Extrapolation** (Figure 3-C): To make the push-pull coefficient independent of the hierarchy level, our solution mechanism requires a *square*  $u, v$  parameter space (the used Master Element corresponds to the canonical  $[0, 1] \times [0, 1]$  square parametric domain). It is possible to construct the corresponding parameterization by constraining the chart boundaries on a square in parameter space, and applying the method proposed in [Sander et al. 2001]. However, in our case, it is better to use a *natural boundaries* parameterization, such as [Sheffer et al. 2005] (which we use in this paper) or [Levy et al. 2002], [Desbrun et al. 2002]. We then apply Dual Domain Extrapolation [Levy 2003] to obtain a square border. This adds a *virtual* part to the chart, taken into account as shown further. Applying this method to the segmentation obtained at step 2 creates a nearly *isometric* parameterization, suitable for the subsequent numerical computations (as explained in the previous section). At this step, the charts having a Jacobian deviation greater than a certain threshold (1



percent) are subdivided and recursively re-parameterized.

## 4 Using the master element

Now that we have defined our geometric support, we can instantiate the generic vector-irradiance algorithm presented in section 2. In our experiments, the function basis  $\Psi_i$  are constant, linear, quadric or cubic wavelets stored in parameter space (the influence of the order of the wavelets is shown in Section 5). We now explain the strategies for numerical integration, push-pull, adaptive subdivision and display.

### Numerical integration and Push-Pull

The integrals corresponding to the dot products are computed using numerical integration. We have experimented Galerkin and collocation [Troutman and Max 1993] methods. In our case, collocation gives the most visually compelling results, since it preserves continuity across the chart boundaries. Note that computing these integrals require efficient implementation of 2D queries in parameter-space. Since in most cases, the triangles of the surface have similar sizes, we use a regular 2D grid to optimize those queries. Each cell of the grid stores a list of pointers to the triangles intersecting the cell. Similarly, 3D queries are optimized by an octree, attached to each surface. Inter-chart visibility is optimized by a BSP. To implement energy transfers, as done in [Alonso et al. 2001], we use a different strategy for receivers and emitters.

**Receiver:** We compute the irradiance function on the entire virtual support. On the virtual part, we use a slightly modified visibility function. When estimating visibility at a point  $\mathbf{x}$  on the receiver,  $\mathbf{x}$  is replaced by  $\mathbf{x}'$  defined as follows: if  $\mathbf{x}$  is on the real part, then  $\mathbf{x}' = \mathbf{x}$ . If  $\mathbf{x}$  is on the virtual part, then  $\mathbf{x}'$  is the point of the real part nearest to  $\mathbf{x}$ . To implement these queries, [Alonso et al. 2001] uses a trapezoidal decomposition of parameter-space. In our case, since the number of primitives may be high, the trapezoidal decomposition algorithm may suffer from numerical instabilities. We prefer storing in each triangle its virtual flag, and reference to the nearest real triangle to extend the visibility function on the virtual support.

**Emitter:** First we need to avoid re-propagating the energy stored on the virtual part. The idea is to use smaller quadrature weights than the one used over real regions. Second, as shown in [Alonso et al. 2001], a perfect auralic mapping makes push-pull coefficients independent of the level in the hierarchy. In our case, we construct an approximatively *isometric* parameterization. The approximation is acceptable and does not produce any visible artifact since the approximation only concerns emitted energy (and not received energy). Note that the quadrature weights can be computed during the push-pull phase, which avoids re-propagating extra energy.

### Error Control and Adaptive Subdivision

We estimate errors on the radiosity function using control points on the receiver as in [Bekaert and Willems 1996]. In contrast with classic clustering method, we do not need to use bounded error control. With our method, it is instead possible to measure the error directly on the receiver by using the mapping function. Adaptive subdivision is triggered when the error measured on a grid of control points is too high.

## Display

To display the solution, we simply generate light-maps, by computing in each parameter space  $B(\mathbf{x}) = \rho(\mathbf{x})(\mathbf{n}_\mathbf{x} \cdot \mathbf{E}(\mathbf{x}))_+$ . As in [Gobbetti et al. 2003], it is also possible to compute approximate view-dependent lighting solutions for scenes with moderately glossy objects. In the algorithm, we replace the diffuse reflectance  $\rho$  by the average value of the BRDF over the hemisphere:  $\rho(\mathbf{x}) = 1/(2\pi)^2 \int \int f_r(\mathbf{x}, \mathbf{y}, \mathbf{z}) d\mathbf{y} d\mathbf{z}$ . Then, at display time, the radiance  $L(\mathbf{x}, \mathbf{z})$  at point  $\mathbf{x}$  towards viewing point  $\mathbf{z}$  is approximated by:

$$L(\mathbf{x}, \mathbf{z}) \simeq L^e(\mathbf{x}, \mathbf{z}) + \|\mathbf{E}(\mathbf{x})\| f_r(\mathbf{x}, \mathbf{x} + \mathbf{E}(\mathbf{x}), \mathbf{z}).$$

In our implementation, the approximated view-dependent lighting solution is displayed in real-time by the GPU. For the moderately diffuse surfaces, lightmaps are replaced with vector-irradiance textures. We use three textures to store the three components  $(x, y, z)$  in the three colors  $R, G, B$ . Examples are shown in Figures 5, 7.



Figure 5: Our method applied to a scene with diffuse and moderately glossy objects.

## 5 Results

Table 1 shows some timings obtained under various configurations. We have also implemented master-element based radiosity to compare with vector-irradiance. As can be seen from the table, using vector irradiance improves performance. This is due to the higher



Figure 6: Comparison between Face Cluster (left) and Meshed Master Element (right). Note the continuity of our solution.

|        | #<br>$\Delta$ | radio.<br>$\mathcal{M}_2$ | irr.<br>$\mathcal{M}_0$ | irr.<br>$\mathcal{M}_1$ | irr.<br>$\mathcal{M}_2$ | irr.<br>$\mathcal{M}_3$ |
|--------|---------------|---------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| Kiss   | 50K           | 260                       | 60                      | 35                      | 45                      | 80                      |
| Isis   | 375K          | 170                       | 40                      | 20                      | 25                      | 40                      |
| David  | 500K          | 230                       | 50                      | 30                      | 45                      | 70                      |
| Buddha | 1.1M          | 240                       | 55                      | 40                      | 60                      | 90                      |

Table 1: Timings in seconds of Master-Element global illumination with radiosity and vector irradiance applied to various data sets (PC Linux, 1.7GHz), using constant, linear, quadratic and cubic wavelets.

|        | # $\Delta$ | charts | $\mathcal{M}_0$ | $\mathcal{M}_1$ | $\mathcal{M}_2$ | $\mathcal{M}_3$ |
|--------|------------|--------|-----------------|-----------------|-----------------|-----------------|
| Kiss   | 50K        | 37     | 144             | 158             | 138             | 278             |
| Isis   | 375K       | 20     | 282             | 269             | 253             | 323             |
| David  | 500K       | 77     | 359             | 358             | 340             | 449             |
| Buddha | 1.1M       | 70     | 596             | 613             | 585             | 698             |

Table 2: Memory usage in Mbytes of MME global illumination with irradiance applied to various data sets (PC Linux, 1.7GHz), using constant, linear, quadratic and cubic wavelets.

precision required to capture the fine-scale variations of the radiosity due to normal variations on curved meshes. In addition, with irradiance, linear wavelets are sufficient to obtain visually compelling results, whereas radiosity requires quadratic wavelets in a similar setting. Figure 8 shows the wavelet bases and visual result obtained for both radiosity and irradiance. In the result, one can also observe that the Kiss data set (50K facets) requires more time than the Buddha data set (1.2M facets). As with clustering, the master-element method depends on the overall geometric complexity of the scene rather than on mesh discretization. Memory statistics are given in Table 2.

Figures 5, 6 and 7 show other examples. In the case of Figures 5 and 6, better results and similar timings to those reported in [Gobbetti et al. 2003] were obtained (Gobbetti reports results under two minutes with similar machine configuration to our). We have not used any post-processing or final gathering step. Note that discontinuities between charts are almost invisible. It is not due to any smoothing between charts (even if that could be done), but the solution is accurate enough to ensure continuity. In contrast, in [Gobbetti et al. 2003] discontinuities between clusters are important, and we can guess where the hierarchical subdivision has stopped.

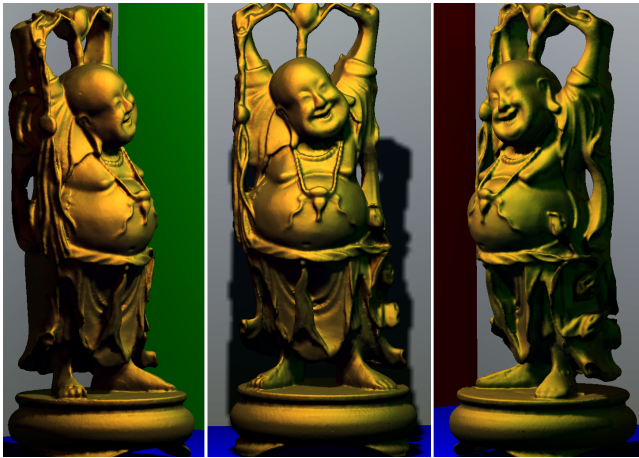


Figure 7: Approximated view-dependent lighting.

## 6 Conclusion and Future Work

In this paper, we have proposed an alternative to clustering for computing light simulation on scene comprising highly tessellated models. We obtain similar timings as face clustering. But with better results thanks to a finer error control. The representation of illumination in parameter space lets us envision several extensions, such as combining discontinuity meshing as done in [Holzschuch and Alonso 2004]. As a result, a compact representation of lighting will be obtained, even for highly tessellated models with sharp shadows.

## Acknowledgements

Thanks to the EU Network of Excellence AIM@Shape (IST NoE No 506766) for supporting B.Lévy. Thanks to Stanford University and Digital Michelangelo project for the models.

## References

- ALONSO, L., CUNY, F., PETITJEAN, S., PAUL, J.-C., LAZARD, S., AND WIES, E. 2001. The virtual mesh: A geometric abstraction for efficiently computing radiosity. *ACM TOG* 20, 3.
- ARVO, J. 1995. The role of functional analysis in global illumination. In *Rendering Techniques '95*, P. M. Hanrahan and W. Purgathofer, Eds. Springer-Verlag, New York, 115–126.
- BEKAERT, P., AND WILLEMS, Y. D. 1996. Error control for radiosity. In *Proceedings of the eurographics workshop on Rendering techniques '96*, Springer-Verlag, 153–ff.
- CHRISTENSEN, P., STOLLNITZ, E., SALESIN, D., AND DEROSE, T. 1996. Global illumination of glossy environments using wavelets and importance. *ACM TOG* 15, 1.
- CHRISTENSEN, P., LISCHINSKI, D., STOLLNITZ, E., AND SALESIN, D. 1997. Clustering for glossy global illumination. *ACM TOG* 16, 1.
- COHEN, M. F., AND WALLACE, J. R. 1993. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, Boston, MA.
- COHEN-STEINER, D., AND MORVAN, J.-M. 2003. Restricted delaunay triangulations and normal cycle. In *SOCG conf. proc.*
- DESBRUN, M., MEYER, M., AND ALLIEZ, P. 2002. Intrinsic parameterizations of surface meshes. In *Proceedings of Eurographics*, 209–218.
- FABRI, A., GIEZEMAN, G.-J., KETTNER, L., SCHIRRA, S., AND SCHÖNHERR, S. 2000. On the Design of CGAL. *Softw. – Pract. Exp.* 30, 11, 1167–1202. [www.cgal.org](http://www.cgal.org).
- GARLAND, M., WILLMOTT, A., AND HECKBERT, P. 2001. Hierarchical face clustering on polygonal surfaces. In *Symp. on Interactive 3D Graphics*, ACM.
- GIBSON, S., AND HUBBOLD, R. 1996. Efficient hierarchical refinement and clustering for radiosity in complex environments. In *Computer Graphics Forum*, Eurographics.
- GOBBETTI, E., SPANO, L., AND AGUS, M. 2003. Hierarchical higher order face cluster radiosity for global illumination walkthroughs of complex non-diffuse environments. In *Computer Graphics Forum*, Eurographics.
- GORAL, C. M., TORRANCE, K. E., GREENBERG, D. P., AND BATTAILLE, B. 1984. Modelling the Interaction of Light Between Diffuse Surfaces. In *Computer Graphics (ACM SIGGRAPH '84 Proceedings)*, vol. 18, 212–222.
- GORTLER, S., SCHRODER, P., COHEN, M., AND HANRAHAN, P. 1993. Wavelet radiosity. In *Computer Graphics (Siggraph)*, ACM.
- GRAPHITE, 2003. <http://www.loria.fr/~levy/Graphite/index.html>.
- GREENGARD, L. 1988. *The rapid evaluation of potential fields in particle systems*. MIT Press, Cambridge, Massachusetts.
- HANRAHAN, P., SALZMAN, D., AND AUPPERLE, L. 1991. A rapid hierarchical radiosity algorithm. In *Computer Graphics (Siggraph)*, ACM.

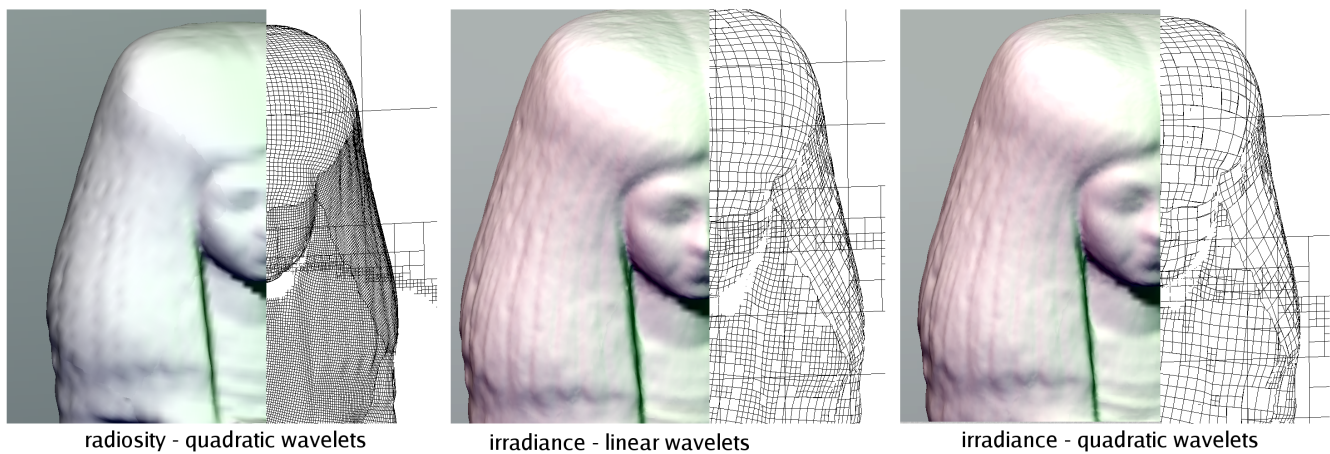


Figure 8: Wavelet mesh obtained with radiosity and quadratic wavelets (left), irradiance and linear wavelets (center), irradiance and quadratic wavelets (right).

- HOLZSCHUCH, N., AND ALONSO, L. 2004. Combining higher-order wavelets and discontinuity meshing: a compact representation for radiosity. In *Rendering Techniques 2004 (Eurographics Symposium on Rendering)*.
- LEVY, B., PETITJEAN, S., RAY, N., AND MAILLOT, J. 2002. Least Squares Conformal Maps for Automatic Texture Atlas Generation. *ACM TOG (SIGGRAPH conf. proc.)*, 362–371.
- LEVY, B. 2003. Dual Domain Extrapolation. *ACM TOG (SIGGRAPH conf. proc.)*.
- LLOYD, S. 1982. Least square quantization in PCM. *IEEE Trans. Inform. Theory* 28, 129–137.
- SANDER, P., SNYDER, J., GORTLER, S., AND HOPPE, H. 2001. Texture mapping progressive meshes. In *Computer Graphics (Siggraph)*, ACM.
- SANDER, P., WOOD, Z., GORTLER, S., SNYDER, J., AND HOPPE, H. 2003. Multi-chart geometry images. In *Symposium on Geometry Processing*.
- SCHÄFER, S. 1997. Hierarchical radiosity on curved surfaces. In *Rendering Techniques*.
- SCHROEDER, P. 1994. *Wavelet Methods for Global Illumination*. PhD thesis, Princeton University.
- SHEFFER, A., LEVY, B., MOGILNITSKY, M., AND BOGOMYAKOV, A. 2005. Abf++ : Fast and robust angle based flattening. *ACM Transactions on Graphics - In print*.
- SILLION, F., DRETTAKIS, G., AND SOLER, C. 1995. A clustering algorithm for radiance calculation in general environments. In *Proceedings of the 6th Workshop on Rendering*, Eurographics.
- SILLION, F. 1995. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE trans. on Visualization and Computer Graphics* 1, 3.
- SMITS, B., ARVO, J., AND GREENBERG, D. 1994. A clustering algorithm for radiosity in complex environments. In *Proceedings of the 21st conf. on Computer Graphics and Interactive Techniques*.
- STAMMINGER, M., SLUSALLEK, P., AND SEIDEL, H.-P. 1997. Bounded radiosity - illumination on general surfaces and clusters. *Comput. Graph. Forum* 16, 3, 309–318.
- TOBLER, R. F., WILKIE, A., FEDA, M., AND PURGATHOFER, W. 1997. A hierarchical subdivision algorithm for stochastic radiosity methods. In *Rendering Techniques*, Springer Wien, New York, NY, J. Dorsey and P. Slusallek, Eds., 193–204. ISBN 3-211-83001-4.
- TROUTMAN, R., AND MAX, N. L. 1993. Radiosity algorithms using higher order finite element methods. In *Proc. of SIGGRAPH-93: Computer Graphics*, 209–212.
- WILLMOTT, A., HECKBERT, P., AND GARLAND, M. 1999. Face cluster radiosity. In *Rendering Techniques*, Eurographics.
- ZATZ, H. 1993. Galerkin radiosity: a higher order solution method for global illumination. In *Computer Graphics (Siggraph)*, ACM.
- ZINKIEWICZ, AND TAYLOR. 1989. *The Finite Element Method*. Mc Graw Hill.