EUROGRAPHICS 2000 / M. Gross and F.R.A. Hopgood
(Guest Editors)

Volume 19 (2000), Number 3

A novel approach makes higher order wavelets really efficient for radiosity

François Cuny[†], Laurent Alonso[‡] and Nicolas Holzschuch[‡]

ISA research team
LORIA[§]
Campus Scientifique, BP 239
54506 Vandœuvre-les-Nancy CEDEX, France

Abstract

Since wavelets were introduced in the radiosity algorithm[5], surprisingly little research has been devoted to higher order wavelets and their use in radiosity algorithms. A previous study[13] has shown that wavelet radiosity, and especially higher order wavelet radiosity was not bringing significant improvements over hierarchical radiosity and was having a very important extra memory cost, thus prohibiting any effective computation. In this paper, we present a new implementation of wavelets in the radiosity algorithm, that is substantially different from previous implementations in several key areas (refinement oracle, link storage, resolution algorithm). We show that, with this implementation, higher order wavelets are actually bringing an improvement over standard hierarchical radiosity and lower order wavelets.

1. Introduction

Global illumination simulation is essential for realistic rendering of virtual scenes. In global illumination, we take the geometric definition of a virtual scene, we simulate the propagation of light throughout the scene, modelling its visual and physical effects, such as shadows and reflections. Global illumination simulation has applications in all the areas where a realistic rendering is interesting, such as architecture, archeology, urban planning and computer-aided design.

The radiosity method is one of the methods used in global illumination simulation. In the radiosity method, we model the exchanges of energy between the objects of the scene in order to compute the radiant energy per unit area (or *radiosity*) on all the surfaces of all the objects in the scene. The radiosity can be used directly to display the objects of the scene and the quality of the simulation is directly linked to the precision we have on the radiosity function.

The radiosity function is usually computed using finite element methods. The most efficient of these methods are hierarchical and use a multi-scale representation of the radiosity function[6] to reduce the algorithmic complexity of the computations. Hierarchical methods have been extended with wavelets[5]. The simplest wavelet base is piecewise constant (Haar wavelets), but many other wavelet bases can be used in radiosity computations.

In theory, higher order wavelets are providing a more compact representation of complex functions. Hence they use less memory and give a smoother representation of the function, that looks better on display. Higher order wavelets should be the ideal choice for radiosity computations.

In practice, the memory required to store the interactions between objects grows with the *fourth* power of the order of the wavelet base, prohibiting any real computation with complex wavelets. Furthermore, Haar wavelets allow many simplifications and optimisations that exploit their great simpleness. If these optimisations are kept with higher order wavelets, they can inhibit some of their properties. In one

† Institut National Polytechnique de Lorraine.
‡ INRIA Lorraine.
§ UMR n° 7503 LORIA, a joint research laboratory between CNRS, Institut National Polytechnique de Lorraine, INRIA, Université Henri Poincaré and Université Nancy 2.

experimental study[13] the practical problems of higher order wavelets were largely overcoming their theoretical benefits.

However, these practical problems are not inherent to higher order wavelets themselves, only to their implementation in the radiosity method. In this paper, we present a new approach to higher order wavelets, that is substantially different from previous implementations in several key areas, such as refinement oracle, link storage and resolution algorithm. Our approach has been developed by taking a complete look at higher order wavelets and at the way they should integrate with the radiosity method. With this implementation, we show that the theoretical advantages of higher order wavelets are overcoming the practical problems that have been encountered before. Higher order wavelets are now providing a better approximation of the radiosity function, with faster convergence to the solution. They also require *less* memory for storage.

Our paper is organised as follows: in section 2, we review the previous research on wavelet radiosity and higher order wavelets. Then in section 3, we present our implementation, concentrating on the areas where it is substantially different from previous implementations: the refinement oracle, not storing the interactions and the consequences it has on the resolution algorithm.

The main result that we present in this paper is the experimental study we have conducted on higher order wavelets with our implementation. Section 4 is devoted to this experimentation and its results, namely that higher order wavelets are providing a faster convergence, a solution of better quality and require less memory for their computations. Finally, section 5 presents our conclusions and future areas of research.

2. Previous work

In this section we review the basis of the wavelet radiosity algorithm (section 2.1), then we present the implementation details of previous implementations for key areas of the algorithm (section 2.2): the refinement oracle, the visibility estimation and the memory problem. This review will help for the presentation of our own implementation of these areas, in section 3.

2.1. The wavelet radiosity algorithm

In the radiosity method, we try to solve the global illumination equation, restricted to diffuse surfaces with no participating media:

$$B(x) = E(x) + \rho(x) \int_S B(y)K(x,y)dy \qquad (1)$$

Eq. 1 expresses the fact that the radiosity at a given point x in the scene, $B(x)$, is equal to the radiosity emitted by x alone, $E(x)$, plus the radiosity reflected by x, coming from all the other objects in the scene. $K(x,y)$ is the kernel of the equation, and expresses the part of radiosity emitted by point y that reaches x.

To compute the radiosity function, we use finite element methods. The function we want to compute, $B(x)$, is first projected onto a finite set of basis functions ϕ_i:

$$\tilde{B}(x) = \sum_i \alpha_i \phi_i(x) \qquad (2)$$

Our goal is to compute the best approximation of the radiosity function, given the set of basis functions ϕ_i. We must also find the optimal set of basis functions. A possibility is to use wavelets. Wavelets are mathematical functions that provide a multi-resolution analysis. They allow a multi-scale representation of the radiosity function on every object. This multi-scale representation can be used in the resolution algorithm[6, 5], allowing us to switch between different representations of the radiosity function, depending on the degree of precision required. This multi-scale resolution results in a great reduction of the complexity of the algorithm[6].

There are two broad classes of resolution algorithm: *gathering* and *shooting*. In gathering, each patch updates its own radiosity function using the energy sent by all the other patches, whereas in shooting each patch sends energy into the scene, and all the other patches update their own radiosity. In both cases, the energy is carried along *links*, that are established by the wavelet radiosity algorithm, and used to store the information related to the interaction. A key element of the wavelet radiosity algorithm is the *refinement oracle*, that tells which levels of the different multi-scale representation of radiosity should interact.

Finally, before each energy propagation, we must update the multi-scale representation of radiosity, so that each level contains a representation of all the energy that has been received by the object at all the other levels. This is done during the *push-pull* phase.

2.2. Details of previous implementations

2.2.1. Refinement oracles

The refinement oracle is one of the most important parts in hierarchical radiosity algorithms. Since it tells at which level the interaction should be established, it has a strong influence on both the quality of the radiosity solution and the time spent doing the computations. A poor refinement oracle will give poor results, or will spend a lot of time doing unnecessary computations.

In theory, the decision whether or not to refine a given interaction could only be taken with the full knowledge of the complete solution. However, the refinement oracle must take the decision using only the information that is locally available: the energy to be sent, and the geometric configuration of the sender and the receiver.

Given two patches in the scene, let us consider their interaction: patch s, with its current approximation of the radiosity function $\tilde{B}_s(y)$, is sending light toward patch r. Using a combination of eq. 1 and eq. 2, we can express the contribution of patch s to the radiosity of patch r:

$$B_{s\to r}(x) = \rho \sum_i \alpha_i \int_s \phi_i(y) K(x,y) dy \tag{3}$$

For the interaction between the two patches we will use the relationship coefficients, C_{ij}:

$$\begin{aligned} B_{s\to r}(x) &= \sum_j \beta_j \phi_j(x) \\ \beta_j &= \int_r B_{s\to r}(x)\phi_j(x) \\ \beta_j &= \rho \sum_i \alpha_i \int_r \int_s \phi_i(y)\phi_j(x)K(x,y)dydx \\ \beta_j &= \rho \sum_i \alpha_i C_{ij} \end{aligned}$$

These C_{ij} coefficients express the relationship between the basis functions $\phi_j(x)$ and $\phi_i(y)$. Computing the C_{ij} requires the computation of a complex integral, which cannot be computed analytically and must be approximated, usually using quadratures.

In most current implementations, refinement oracle estimate the error on this approximation of the C_{ij}. This error is then multiplied by the energy of the sender, to avoid refining interactions that are not carrying significant energy. There are several ways to estimate the error on the C_{ij} coefficients: pure heuristics[6], sampling the C_{ij} at several sample points[5] and a conservative method giving an upper-bound on the propagation of the energy[10, 8].

A recurrent problem with current refinement oracles is that they concentrate on the C_{ij} coefficients. This provides a conservative analysis, but it can be too cautious, especially with higher order basis functions. The C_{ij} coefficients are usually bound with constant functions and hence so is the radiosity function. Such a binding does not take into account the capacity of higher order wavelets to model rapidly varying functions in a compact way. To take this into account, we need to move the radiosity function *inside* the refinement oracle. In section 3.1, we present a refinement oracle that addresses this problem.

2.2.2. Visibility estimations

Discontinuities of the radiosity function and its derivatives are only caused by changes in the visibility between objects[7]. Therefore, great care must be taken when adding visibility information to the radiosity algorithm.

As we have seen, we use a quadrature to compute the C_{ij} coefficients. This quadrature requires several estimates of the kernel function $K(x,y)$ and therefore of the visibility between points x and y. Computing a visibility sample is much costlier than computing a kernel sample without visibility. As a consequence, estimating the visibility between two patches is the most costly operation in wavelet radiosity[9]. Several methods have been developed in order to provide a quick estimate of visibility, sometimes at the expense of reliability.

The easiest method[6, 5] assumes a constant visibility between the patches. The constant is equal to 1 for fully visible patches, 0 for fully invisible patches, and is in $]0,1[$ for partially visible patches. It is estimated by computing several jittered visibility samples between the patches and averaging the results.

Another method computes exact visibility between the corners of the patches, and interpolates between these values for points located between the corners, using barycentric coordinates.

Shadow masks[16, 11] have also been used in wavelet radiosity computations. In theory, shadow masks allow the decoupling of visibility from radiosity transport, and therefore a better compression of the radiosity transport operator, thus reducing the memory cost.

All these methods attempt to approximate visibility by computing less visibility samples than kernel samples, in order to reduce the cost of visibility in wavelet radiosity. According to an experimental study of wavelet radiosity conducted by Willmott[13, 14] the result is a poor approximation of the radiosity function, especially near shadow boundaries.

Another method is to compute exactly one visibility sample for each kernel sample. It has been used at least by Gershbein[4], although it is not explicitely stated in his paper. According to our own experience, as well as Willmott extended study[14], this method gives better visual results. Furthermore, it gives more numerical precision. On the other hand, it can introduce some artefacts, because the visibility samples are forced to be in a regular pattern.

In our implementation, we used one visibility sample for each kernel sample, because we were looking for numerical accuracy, and because the artefacts are removed by our refinement oracle.

2.2.3. Memory usage

Since the computation of the C_{ij} coefficients can be rather long, they are usually stored once they have been computed, so that they can be reused. The storage is done on the link between s and r.

An important problem with previous wavelet radiosity implementations is the memory required for this storage. If we use wavelet bases of the order m, then we have m one dimensional functions in the wavelet base. For two dimensions, such as the surface of objects in our virtual scene, we have m^2 functions in the base. As a consequence, storing the interaction between two patches requires computing and storing m^4 C_{ij} coefficients.

Hence, the memory usage of wavelet radiosity grows with the fourth power of the wavelet base used. Wavelets of order 3 will have a memory usage almost two orders of magnitude higher than wavelets with 1 vanishing moment. In an experimental study of wavelet radiosity, Willmott[13] showed that this memory usage was effectively prohibiting any serious computation with higher order wavelets.

In 1998, Stamminger[12] showed that it was possible to eliminate completely the storage of the interactions in hierarchical radiosity. His study was only made for hierarchical radiosity, but it could be extended to wavelet radiosity, and it would remove the worst problem of radiosity with higher order wavelets. In section 3.2, we review the consequences of not storing links on the wavelet radiosity algorithm.

3. A novel approach to higher order wavelets in the radiosity algorithm

Since experimental studies conducted with previous implementations of wavelet radiosity have shown that higher order wavelets are behaving more poorly than Haar wavelets, we need to review the key points of our implementation that differ from previous implementations: the refinement oracle and getting rid of interaction storage, along with the consequences it has on the algorithm.

All the elements described in this section have been implemented and tested thanks to our radiosity testbed software, Candela[15].

3.1. The refinement oracle

Instead of estimating the errors on the propagation coefficients, we estimate the error on the propagated energy directly. Our refinement oracle is quite similar to that of Bekaert[2, 3].

To estimate the errors on the radiosity function, we use control points on the receiver. These control points are located so that they provide meaningful information: they are different from quadrature points, and their number depends on the size of the receiver. Some of the control points are located on the boundary of the receiver, in order to ensure continuity with neighbouring patches.

Our refinement oracle is summed up in fig. 1. The radiosity values at the control points $B_{s\to P_i}$ are computed by direct integration of eq. 3 at point $x = P_i$, using a quadrature. To take the norm of the errors at the control points, we can use any norm, such as the L_1 norm, the L_2 norm, the L_∞ norm. We have found that all these norms are giving similar results for refinement.

3.2. Not using links and the consequences

In order to reduce the memory footprint of the radiosity algorithm, we have chosen not to store links, as in Stamminger[12].

for each interaction $s \to r$:
  compute the radiosity function on the receiver: $B_{s\to r}(x)$
  for each control point P_i
    compute the radiosity at this control point directly: $B_{s\to P_i}$
    compare with interpolated value,
    store the difference:
    $\delta_i = |B_{s\to r}(P_i) - B_{s\to P_i}|$
  end for
  take the L_n norm of the differences:
  $\delta_B = ||\delta_i||_n$
  compare with refinement threshold
end for

Figure 1: *Our refinement oracle*

Not storing links is some kind of a trade-off between memory and time: by not storing links, we are saving memory. However, the information that has not been stored will probably have to be recomputed at some stage in the algorithm, which will cost time.

Not storing links also has consequences on the structure of the algorithm itself. The main consequence is on the choice between gathering and shooting.

Gathering sends energy from all the patches to all the other patches at each iteration. All the links are used during a given iteration.

Shooting sends the unshot energy from one patch to all the other patches. At a given point in time, only the links from the shooting patch to all the patches are being used. The shooting patch is then send to the bottom of the shooting queue, and the links will not be re-used until it gets back to the top of the shooting queue.

Therefore, if we chose not to store links, it makes more sense to use shooting than to use gathering. But the reverse is also true: if you use shooting instead of gathering, it also makes more sense not to store links.

in gathering, the optimal link distribution for one iteration can be computed by refining the link distribution from the previous distribution, because the radiosity gathered at one point can only grow with subsequent iterations.

in shooting, the energy carried along the links is only the unshot energy at the shooting patch. Its distribution changes completely for each use of the patch. As a consequence, the optimal link distribution has no relation with the links computed for previous iterations.

4. Comparison of several wavelet bases

In this section, we present our experimental comparison of different wavelet bases. We start with a description of the experimentation protocol in section 4.1. We then present the results of our experiments in section 4.2. Discussion of these results and comparison with previous studies follows in section 4.3.

4.1. The experimentation protocol

4.1.1. The wavelet bases

We wanted to use our implementation of wavelet radiosity for a comparison of several wavelet bases. We have used the first three multi-wavelets bases : $\mathcal{M}_1$ (Haar), $\mathcal{M}_2$ and $\mathcal{M}_3$.

We use the $\mathcal{M}_n$ multi-wavelets as they were previously defined [1, 5]: the smoothing functions for $\mathcal{M}_n$ are defined by tensorial products of the first n Legendre polynomials.

We have not used flatlets bases ($\mathcal{F}_n$), because although they have n vanishing moments, they are only piecewise constant, and therefore do not provide a better approximation than Haar wavelets with further refinement.

4.1.2. The test scenes

Our tests have been conducted on several test scenes, ranging from simple scenes, such as the blocker (see fig. 2(a)) to moderately complex scenes, such as the class room (see fig. 2(d)). All our test scenes are depicted on fig. 2, with their number of input polygons.

4.1.3. Displaying the results

All the figures in this paper are depicting the exact results of the computations, without any post-processing of any kind: the radiosity function is displayed exactly as it has been computed. Specifically, there has been no attempt to ensure continuity of the radiosity function, except in the refinement oracle. Similarly, we haven't balanced or anchored the computed mesh. So, for example, in fig. 4(c), the continuity of the radiosity function is due only to the refinement oracle depicted in section 3.1.

$\mathcal{M}_3$ wavelets can result in quadrically varying functions, which can not be displayed on our graphics engines. To display these functions, we subdivide each patch into four subpatches, on which we compute four linearly varying functions approximating the quadrically varying radiosity function.

4.1.4. Computing the error

In order to compute the computational error, we have computed a reference solution, using $\mathcal{M}_2$ wavelets, with a very small refinement threshold. Furthermore, the minimal patch area in the reference solution was 16 times smaller than the minimal patch area in the computed solutions. We also checked that with all the wavelet bases, the computed solutions did converge to the reference solution.

We have measured the energetic difference between this reference solution and the computed solutions. In order to have comparable results on all our test scenes, this difference is divided by the total energy of the scene. It is this ratio of the energetic difference over the total energy that we call *global error*. Thus, a global error of 10^{-1} means there is an energetic difference of 10 % between the energetic distributions of the computed solution and the reference solution.

According to our experiments, this measure of global error is consistent, and gives comparable visual results on all the test scenes. For example, a global error of 10^{-1} will always give a poor result (see fig. 3(a)), a global error of 10^{-2} will give a better result, but still with visible artefacts at shadow boundaries (see fig. 3(b)), and a global error of 10^{-3} will always give a correct result (see fig. 3(c)). In our experience, (see fig. 3) the global error must be lower than 5.10^{-3} in order to get visually acceptable results.

As it has been pointed out[12], we have also found that this global error is closely correlated to the refinement threshold on each interaction (the *local error*).

4.1.5. Experimentation details

In all our experiments, we have used the same computer, a SGI Octane working at 225 MHz, with 256 Mb of RAM.

4.2. Results

4.2.1. Visual comparison of our three wavelet bases

The first test to conduct is whether higher order wavelets are giving a better visual impression. In previous tests[13], higher order wavelets were unable to provide a correct approximation of the radiosity function, especially near shadow boundaries. Shadow boundaries are very important because they have a large impact on the visual perception of the scene.

Our first experiment focuses solely on this problem. We have computed direct illumination from an area light source to a planar receiver, with an occluder partially blocking the exchange of light. All wavelet bases were used with the same computation time (66 s).

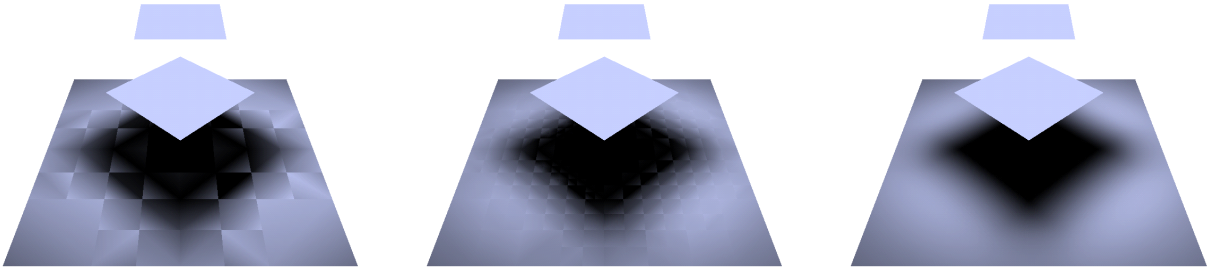
Fig. 4 shows the radiosity function computed for each wavelet base, along with the mesh used for the computation. Two elements appear clearly: higher order wavelets are providing a much more compact representation of the radiosity function, even near shadow boundaries, and the radiosity function computed with $\mathcal{M}_2$ and $\mathcal{M}_3$ wavelets is smoother than the function computed with Haar wavelets.

Haar wavelets are usually not displayed as such, but using some sort of post-processing, such as Gouraud shading. Fig. 5 shows the result of applying Gouraud shading to fig. 4(a). As you can see, although it can hide some of the discontinuities, Gouraud shading can also introduce some new artefacts.

Judging from fig. 4, higher order wavelets are better for radiosity computations than lower order wavelets. This is only a qualitative results and must be confirmed by quantitative studies; that is the object of the coming sections (4.2.2 and 4.2.3).

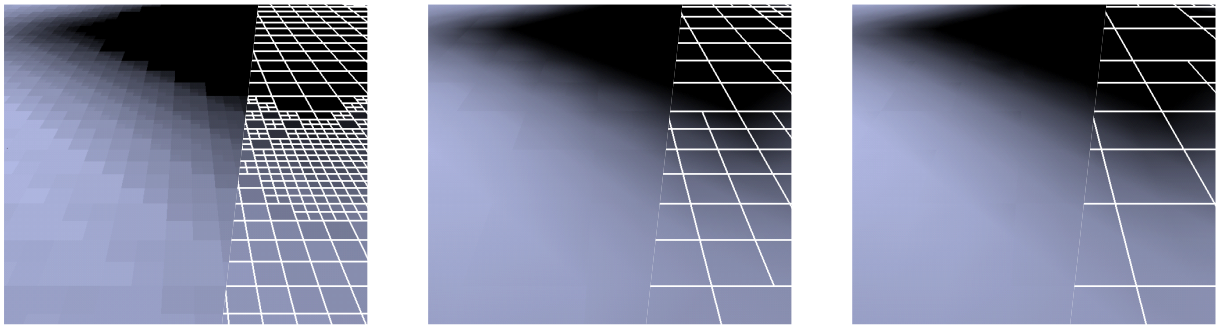
(a) Blocker (3) (b) Tube (5) (c) Dining room (402) (d) Classroom (3153)

Figure 2: *Our test scenes, with their number of input polygons*



(a) 10^{-1} (b) 10^{-2} (c) 10^{-3}

Figure 3: *Visual comparison of results for different values of global error*



(a) Haar (b) $\mathcal{M}_2$ (c) $\mathcal{M}_3$

Figure 4: *Visual comparison of results for our three wavelet bases*

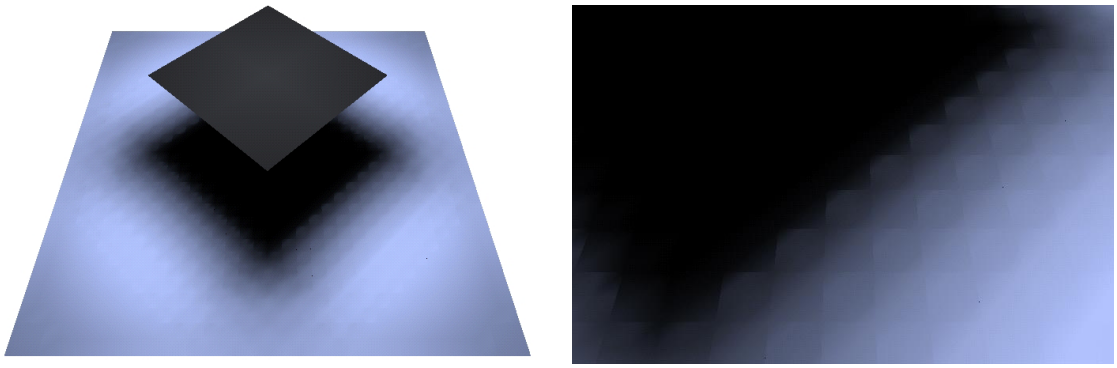
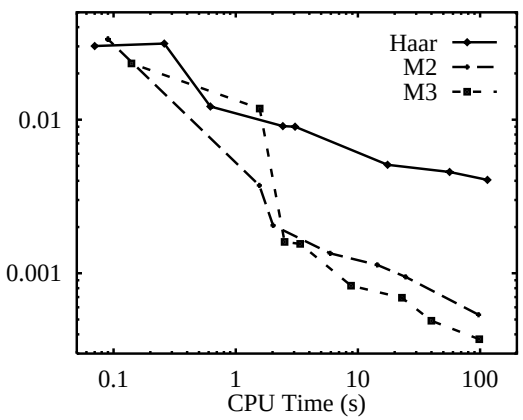
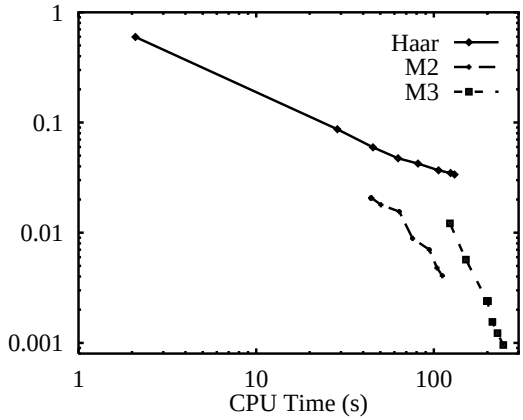


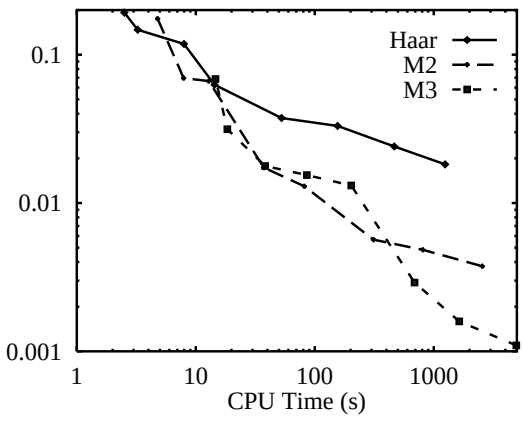
Figure 5: *Applying Gouraud shading to Haar wavelets*



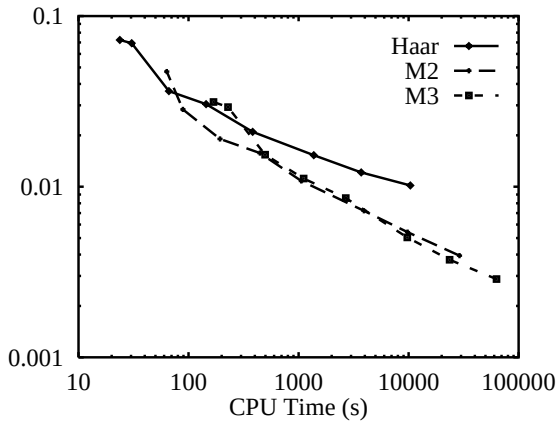
(a) Blocker



(b) Tube



(c) Dining room



(d) Classroom

Figure 6: *Global error with respect to computation time (in s)*

4.2.2. Computation time

Fig. 6 shows the relationship between global error and computation time for our four test scenes and our three wavelet bases.

The most important point that can be extracted from these experimental data is that with our implementation, higher order wavelets are performing *better* than lower order wavelets. They obtain results of higher quality, and they are faster: to get a visually acceptable result on the classroom scene (global error below 5.10^{-3}), $\mathcal{M}_3$ wavelets use 10^4 s (see fig. 6(d)). In the same computation time, Haar wavelets only reach a global error level of 10^{-2}. This test scene is our hardest test scene, with lots of shadow boundaries. It is on such test scenes that higher order wavelets were behaving poorly with previous experimentations[13].

The advantage of higher order wavelets is more significant on high precision computations and on complex scenes. The more precision you need on your computations, the faster they are, compared to lower order wavelets.

On the contrary, for quick approximations, $\mathcal{M}_2$ wavelets are performing better than $\mathcal{M}_3$ wavelets. The same applies to Haar wavelets compared to $\mathcal{M}_2$ wavelets, for very quick and crude approximations.

Each wavelet base has an *area of competence*, where it outperforms all the other wavelet bases: Haar wavelets are the most efficient base for global error above 10^{-1} — which corresponds to a simulation with many artefacts still visible (see fig. 3(a)). $\mathcal{M}_2$ wavelets are better than all the other bases for global error between 10^{-1} and (roughly) 5.10^{-3}, and $\mathcal{M}_3$ wavelets are the best for global error below 5.10^{-3}.

4.2.3. Memory use

The key problem with higher order wavelets in previous studies[13] was their high memory use, that effectively prohibited any real computation. We have computed the memory footprint of our implementation of wavelets for our four test scenes and our three wavelet bases. Fig. 7 shows the memory used by the algorithm as a function of the global error.

As you can see, for high precision computations (global error below 5.10^{-3}), higher order wavelets actually have a *lower* memory use than low order wavelets. The effect is even more obvious on our more complex scenes (see fig. 7(c) and 7(d)).

On the other hand, for low precision computations, this hierarchy is reversed, and Haar and $\mathcal{M}_2$ wavelets have a lower memory use. Once again, each wavelet base has an area of competence, where it outperforms all the other wavelet bases. For very crude approximations, Haar wavelets are the most efficient with respect to memory use, then, for moderately good approximations, $\mathcal{M}_2$ wavelets are the most efficient, until $\mathcal{M}_3$ takes over for really good approximations.

A very impressive result is the way the memory cost of a given wavelet base degrades quickly if we try to bring the global error level below a certain threshold. This effect appears very clearly on fig. 7(c) and 7(d). There seems to be a maximum degree of precision for each wavelet base, and the wavelet base can only conduct global illumination simulations below this degree. Be aware, however, that the degradation is made more impressive on fig. 7 by the fact that we are using a logarithmic scale for global error and a non-logarithmic scale for memory use. Furthermore, the degradation is quite small when it is compared to the total memory used: between 10 % and 20 %. Since the effect appears in a similar way for all the wavelet bases used in the test we think it could be a general effect, and apply to all wavelet bases.

Please note that the fact that higher order wavelets have a lower memory use than lower order wavelets is actually quite logical. Higher order wavelets are providing a more powerful tool for approximating complex functions, with a higher dimensional space for the approximation. Furthermore, they have more vanishing moments, so their representation of a given complex function is more compact and requires less coefficients. Our experiments are therefore bringing practical results in connection with theoretical expectations.

The fact that lower order wavelets are more compact for low precision computations was also to be expected from theory. Low precision computations are, by nature, not taking into account all the complexity of the radiosity function. As a consequence, they provide a very simple function, that is also easy to approximate, especially for simple wavelet bases.

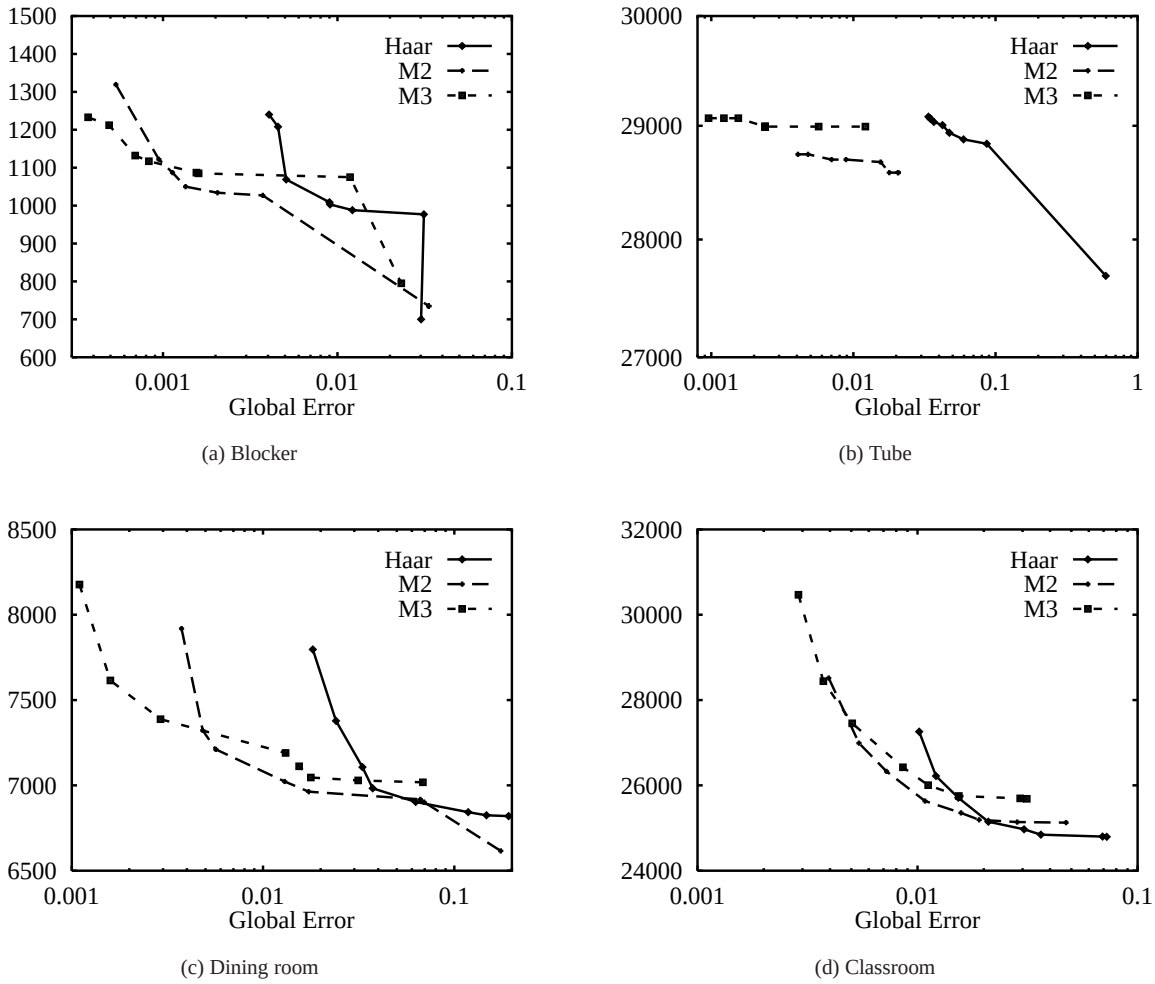
4.3. Discussion and comparison with previous studies

Despite the fact that we are reaching opposite conclusions, we would like to point out that our study is actually consistent with the previous study by Willmott[13, 14].

In Willmott's study, higher order wavelets were carrying a strong memory cost, due to link storage. As a consequence, radiosity computations with higher order wavelets were restricted to low precision computations. According to our experiments, for low precision computations, lower order wavelets are indeed providing a faster approximation, with a lower memory use.

Our study can therefore be seen as an extension of Willmott's study to high precision computations. Such high precision computations were made possible only by getting rid of links[12]. Once you have eliminated link storage, the memory cost of the radiosity algorithm is almost reduced to the cost of mesh storage. The refinement oracle (see section 3.1) ensures that the mesh produced is close to optimal with respect to the radiosity on the surfaces.

Also, by concentrating the oracle on the mesh instead of the interactions, we are able to exploit the power of wavelet bases functions to efficiently approximate functions. This results in a coarser mesh, both at places where the radiosity



(a) Blocker

(b) Tube

(c) Dining room

(d) Classroom

Figure 7: *Memory requirements (in kB) with respect to global error*

function has slow variations, such as an evenly lit wall, and at place with rapid variations, such as shadow boundaries.

5. Conclusion and future work

We have presented an implementation of wavelets bases in the radiosity algorithm. With this implementation, we have conducted experimentations on several wavelet bases. Our experiments show that for high precision computations, higher order wavelets are providing a better approximation of the radiosity function, faster, and with a lower cost in memory. Please note that our implementation is not putting any disadvantage on lower order wavelets; for Haar wavelets, our refinement oracle only uses a few tests and the visibility estimation only requires one visibility test. Similarly, the benefit of not storing links is independant of the wavelet base.

Although in this paper we have only conducted tests on relatively small test scenes (up to 3000 input polygons), our implementation (Candela[15]) enables us to use higher order wavelets on arbitrarily large scenes. Fig. 8* shows a radiosity computation with $\mathcal{M}_2$ wavelets made with our implementation on a scene with 144255 input polygons. The computations took 3 hours, and required approximately 2 Gb of memory on 32 processors of a SGI Origin 2000. The complete solution had approximately 1.5 million patches.

The optimal choice for radiosity computations depends on the degree of precision required. Lower order wavelets are better for low precision computations, and higher order wavelets are better for high precision computations. Each wavelet base corresponds to a certain degree of precision, where it outperforms all the other wavelet bases, both for the computation time and the memory footprint. Although

our computations have been limited to Haar, $\mathcal{M}_2$ and $\mathcal{M}_3$ wavelets, we think that this effect applies to all the other wavelet bases, such as $\mathcal{M}_4$, $\mathcal{M}_5$... and that for even more precise computations, $\mathcal{M}_4$ would outperform $\mathcal{M}_3$, and so on.

However, for moderately precise computations, $\mathcal{M}_2$ wavelets are quite sufficient. The precision level that corresponds, in our experience, to visually acceptable results is at the boundary between the areas of competence of $\mathcal{M}_2$ and $\mathcal{M}_3$, so $\mathcal{M}_2$ wavelets can be used. $\mathcal{M}_2$ wavelets also have a distinct advantage over all the other wavelet bases: they result in linearly varying functions that can be displayed directly on current graphics hardware (using Gouraud shading), as opposed to constant, quadric or cubic functions.

In our future work, we want to explore the possibility to use several different wavelet bases in the resolution process. In this approach, it would be possible to use Haar wavelets for interactions that do not require a lot of precision, such as interactions that do not carry a lot of energy, and $\mathcal{M}_2$, and perhaps $\mathcal{M}_3$, $\mathcal{M}_4$..., wavelets for interactions that require a high precision representation. We think that this approach could be especially interesting with shooting since the first interactions will carry a lot of energy, while later interactions will only carry a small quantity of energy.

We also want to explore the possibility to use higher order wavelets on non-planar objects. Since they have a better ability to model rapidly varying radiosity functions, they seem to be the ideal choice for curved surfaces, such as spheres or cylinders.

6. Acknowledgements

The authors would like to give a very special thank to Jean-Claude Paul. It was his insight that started this work on higher order wavelets, and it was his advice and support that ensured its success.

References

1. B. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin. Wavelet-like bases for the fast solution of second-kind integral equations. *SIAM Journal on Scientific Computing*, 14(1):159–184, January 1993. 5

2. Philippe Bekaert and Yves Willems. Error Control for Radiosity. In *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pages 153–164, New York, NY, 1996. Springer-Verlag/Wien. 4

3. Philippe Bekaert and Yves D. Willems. Hirad: A Hierarchical Higher Order Radiosity Implementation. In *Proceedings of the Twelfth Spring Conference on Computer Graphics (SCCG '96)*, Bratislava, Slovakia, June 1996. Comenius University Press. 4

4. Reid Gershbein. Integration Methods for Galerkin Radiosity Couplings. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering) in Dublin, Ireland, June 12-14, 1995)*, pages 264–273, New York, NY, 1995. Springer-Verlag. 3

5. Steven J. Gortler, Peter Schroder, Michael F. Cohen, and Pat Hanrahan. Wavelet Radiosity. In *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, pages 221–230, 1993. 1, 2, 3, 5

6. Pat Hanrahan, David Salzman, and Larry Aupperle. A Rapid Hierarchical Radiosity Algorithm. In *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, volume 25, pages 197–206, July 1991. 1, 2, 3

7. Paul Heckbert. Discontinuity Meshing for Radiosity. In *Third Eurographics Workshop on Rendering*, pages 203–226, Bristol, UK, May 1992. 3

8. Nicholas Holzschuch and Francois. X. Sillion. An exhaustive error-bounding algorithm for hierarchical radiosity. *Computer Graphics Forum*, 17(4):197–218, December 1998. 3

9. Nicolas Holzschuch, Francois Sillion, and George Drettakis. An Efficient Progressive Refinement Strategy for Hierarchical Radiosity. In *Fifth Eurographics Workshop on Rendering*, pages 343–357, Darmstadt, Germany, June 1994. 3

10. Dani Lischinski, Brian Smits, and Donald P. Greenberg. Bounds and Error Estimates for Radiosity. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 67–74, 1994. 3

11. Philipp Slusallek, Michael Schroder, Marc Stamminger, and Hans-Peter Seidel. Smart Links and Efficient Reconstruction for Wavelet Radiosity. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 240–251, New York, NY, 1995. Springer-Verlag. 3

12. M. Stamminger, H. Schirmacher, P. Slusallek, and H.-P. Seidel. Getting rid of links in hierarchical radiosity. *Computer Graphics Journal (Proc. Eurographics '98)*, 17(3):C165–C174, September 1998. 4, 5, 8

13. Andrew Willmott and Paul Heckbert. An empirical comparison of progressive and wavelet radiosity. In Julie Dorsey and Phillip Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 175–186, New York, NY, 1997. Springer Wien. ISBN 3-211-83001-4. 1, 2, 3, 4, 5, 8

14. Andrew J. Willmott and Paul S. Heckbert. An empirical comparison of radiosity algorithms. Technical Report CMU-CS-97-115, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, April 1997. Available from http://www.cs.cmu.edu/ radiosity/emprad-tr.html. 3, 8

15. Christophe Winkler. *Expérimentation d'algorithmes de calcul de radiosité à base d'ondelettes*. Thèse d'université, Institut National Polytechnique de Lorraine, 1998. 4, 9

16. Harold R. Zatz. Galerkin Radiosity: A Higher Order Solution Method for Global Illumination. In *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, pages 213–220, 1993. 3

Figure 8: *Radiosity computation on a large scene (with $\mathcal{M}_2$ wavelets)*