TWO-LEVEL ITERATIVE SHOOTING METHODS WITH GROUPS

F. Rousselle[1] **M. Leblond**[2] **C. Renaud**[1]

[1] Laboratoire d'Informatique du Littoral
Université du Littoral Côte d'Opale
BP 719 - 62228 Calais Cedex
France
[2] Laboratoire de Mathématiques Pures et Appliquées
Université du Littoral Côte d'Opale
BP 699 - 62228 Calais Cedex
France
(roussell,leblond,renaud)@lil.univ-littoral.fr

ABSTRACT

In this paper, we present an acceleration technique for Progressive Radiosity based on group iterative methods. This technique uses groups of shooting patches to accelerate the diffusion of light. The quality of the results depends on the amount of energy that is exchanged between the patches of each group. We propose two group building techniques which guarantee a high level of interaction between the shooting patches. The resolution of the sub systems generated by the groups is done rapidly thanks to a new technique of hybridization applied to the Gauss Seidel method. This new PR method using groups is especially efficient in the case of scenes having many occlusions.

Keywords: progressive radiosity, group resolution, iterative solver, hybridization.

1 INTRODUCTION

Since progressive radiosity (PR) has been introduced in computer graphics [Cohen88b], a lot of researches have focused on accelerating its convergence rate. Overshooting methods [Feda92a] [Shao93a] [Xu94a] have been proposed for this purpose. They attempt to accelerate PR by reducing the number of times a patch is selected for shooting. Minimizing the number of selection per patch is expected by estimating the energy that would come back to the emitter from direct and indirect reflections. Results of such approaches have highlighted their interest, but they are self-limited by using only one form factors row at a time. Thus when occlusions are numerous, overshooting only improves the convergence rate of a few number of patches.

High level of occlusions can arise both locally and globally in image synthesis scenes. Furthermore some other research areas are concerned with radiosity computation in high level of occlusion environments. Radiative transfers in plant canopies is one of them [Chell98a]. We are currently studying this kind of problem with agronomists and it appears that more powerful radiosity algorithms like hierarchical radiosity [Hanra91a] are useless in such problems.

In this paper we are interested in studying some improvements to a Group Accelerated Progressive Method (GASM) [Rouss99a]. This method allows existing PR based methods to be successfully accelerated by periodically shooting energy from a group of patches. Each group shooting step takes into account both all the internal group exchanges and the exchanges between the group and the entire scene. GASM provides efficient results for high occlusion levels but previous approach required several additional studies.

Two main improvements are investigated in this paper. At one hand, efficient group sub-systems solvers have to be used in order to reduce the overhead of the groups usage. At the other hand, the efficiency of a group depends on how it has been built : higher is the amount of energy that is exchanged inside the group, more efficient is this group. In the next part of this paper we summarize the GASM approach. Then in part 3 we present a new efficient technique for radiosity linear systems solvers. Two group building techniques are investigated in part 4. Some results are provided for scenes with high level of occlusion and finally we give some perspective to our work.

2 GROUP ACCELERATED SHOOTING METHOD

GASM is an acceleration technique on top of a standard shooting method such as Progressive Radiosity (PR). Periodically, during the PR iteration steps, a group of patches is built, the corresponding sub system is solved and the overall radiosity and unshot radiosity vectors are updated. This means that all the interactions between the patches of the group are solved and that the energy of the entire group is shoot towards the rest of the patches in the scene.

The shooting patches and their corresponding form factor rows computed during the successive shooting steps of PR are stored into memory. The number of patches stored depends on the amount of available memory ; it is generally several hundreds. Each group is built from the patches in memory and the size of the group is fixed by the user. The periodicity of group resolution is computed automatically in order to obtain a small overhead. The periodicity is deduced from the average time spent computing form factors and doing the single shooting steps and the average time spent solving the sub systems.

The following notations will be used throughout this article :

- $\Phi B = E$ is the radiosity system, with $\Phi = (\varphi_{i,j})_{n \times n}$ the radiosity matrix ;
- $E = (e_1, \ldots, e_n)^T$ is the vector of initial energy values ;
- $B = (b_1, \ldots, b_n)^T$ is the vector of radiosity values (iterate vector) ;
- $\Delta B = (\Delta b_1, \ldots, \Delta b_n)^T$is the vector of unshot radiosity values (residual vector) ;
- B_r is the sub vector of B composed of elements b_i, $i \in$ group r ;
- ΔB_r is the sub vector of ΔB composed of elements Δb_i, $i \in$ group r ;
- $\Phi_{r,r}$ is the sub matrix of Φ composed of the intersection of rows $\Phi_{i,*}$, $i \in$ group r and columns $\Phi_{*,j}$, $j \in$ group r.

Using these notations, GASM applied to PR is summarized in the pseudo algorithm 1.

Algorithm 1 Group Accelerated Progressive Radiosity

```
 1: choose $m$ the size of the group
 2: initialize $B$ and $\Delta B$ with $E$
 3: do $m$ shooting steps and store
    each shooting patch along with its ff's
 4: solve the $m \times m$ sub system
 5: compute the frequency of group
     resolution
 6: while not converged do
 7:   choose patch $i$ with max $\mid \Delta b_i.a_i \mid$
 8:   if patch $i$ not in memory then
 9:     compute the form factors $F_{*,i}$
10:     if no place left in memory then
11:       replace oldest ff's with new one's
12:     endif
13:   endif
14:   perform a shooting step from patch $i$
15:   if group frequency reached then
16:     choose the $m$ patches among those
        in memory having max $\mid \Delta b_i.a_i \mid$
17:     compute $\Phi_{r,r}^{-1}$
18:     $\Delta rad_r = \Phi_{r,r}^{-1} \Delta B_r$
19:     $B_r = B_r + \Delta rad_r - \Delta B_r$
20:     for all patches $j \notin$ group $r$ do
21:       $\Delta rad = -\Phi_{j,r} \Delta rad_r$
22:       $\Delta b_j = \Delta b_j + \Delta rad$
23:       $b_j = b_j + \Delta rad$
24:     endfor
25:     $\Delta B_r = 0$
26:   endif
27: endwhile
```

3 ITERATIVE RESOLUTION OF THE SUB SYSTEM

To shoot from group r, it is necessary at first to compute $\Phi_{r,r}^{-1}$ (line 17 of algorithm 1). This matrix is subsequently used to update the vector of radiosity values (B) and the vector of unshot radiosity values (ΔB) (lines 18 to 24). Assuming that n is the number of patches in the scene and that m is the size of a group, the computational cost of each group shooting step is :

- $O(m^3)$ to compute $\Phi_{r,r}^{-1}$ with a standard inversion method such as the Jordan method ;

- $O(m^2)$ to update B_r ;
- $O(mn)$ to update patches which do not belong to group r.

The total cost is thus $O(m^3)$ if $m^2 > n$ and $O(mn)$ otherwise.

When groups of a relatively large size (more than a few hundreds patches) are used, the computational time of $\Phi_{r,r}^{-1}$ becomes prohibitive. Besides, this cost cannot be compensated with other uses of this matrix since it corresponds to a particular group and that a new group is built at each new group shooting step.

We propose therefore to use an iterative method to resolve the system $\Phi_{r,r}\Delta rad_r = \Delta B_r$, these methods having a much lower cost. Thus, we introduce a second iterative level in the global radiosity system resolution.

Any algorithm usually proposed in radiosity could suit the resolution of this sub system. Nevertheless, since the matrix $\Phi_{r,r}$ is known (it is build from the form factors rows stored into memory), general matrix techniques are faster in this case [Baran95a].

According to the definition of the matrix $\Phi_{r,r}$ in section 2, the latter inherits the strictly diagonally dominance property from the matrix Φ since all its diagonal elements are 1 and all other elements of a row of $\Phi_{r,r}$ are part of a row of Φ. Therefore the Gauss Seidel resolution technique can be applied. Other methods such as Chebichev or Conjugate Gradient can be used also, however we point out in [Leblo99a] that the application of a hybridization technique allow us to outperform general matrix techniques both in term of rapidity of convergence and computational time. We described below the hybridization principle [Brezi97a].

3.1 Hybridization of two iterative methods

Let us consider the linear system $Ax = b$, where A is a $n \times n$ matrix and x and b are two vectors of dimension n. We assume that x^* is the solution of this system and that $\{x_k^{'}\}$ and $\{x_k^{''}\}$ are two vectors sequences which converge to x^*. Residuals of these two sequences are given by :

$$\left\{ \begin{array}{lcl} r_k^{'} & = & b - Ax_k^{'} \\ r_k^{''} & = & b - Ax_k^{''} \end{array} \right.$$

Using a linear combination of these two residuals we build the residual r_k :

$$r_k = \alpha_k r_k^{'} + \beta_k r_k^{''}$$

It is assumed to be also the residual of a new solver providing a sequence $\{x_k\}$ which converges to x^* :

$$r_k = b - Ax_k$$

It is then easy to derive an expression for x_k using the linear combination of $r_k^{'}$ and $r_k^{''}$:

$$\begin{array}{rcl} r_k & = & \alpha_k r_k^{'} + \beta_k r_k^{''} \\ b - Ax_k & = & \alpha_k(b - Ax_k^{'}) + \beta_k(b - Ax_k^{''}) \\ Ax_k & = & (1 - \alpha_k - \beta_k)b + A(\alpha_k x_k^{'} + \beta_k x_k^{''}) \end{array} \quad (1)$$

But using the last equation of (1) requires to reverse A in order to compute x_k . Therefore, A is suppressed from this equation by assuming that $\alpha_k + \beta_k = 1$. This equation is then rewritten as :

$$x_k = \alpha_k x_k^{'} + \beta_k x_k^{''}$$

and replacing β_k by $(1 - \alpha_k)$ we get the following expressions for the terms x_k and r_k :

$$\left\{ \begin{array}{lcl} x_k & = & x_k^{''} + \alpha_k(x_k^{'} - x_k^{''}) \\ r_k & = & r_k^{''} + \alpha_k(r_k^{'} - r_k^{''}) \end{array} \right.$$

Computing these two terms requires however to know a value for α_k. Because we want x_k to converge more quickly to x^* than both $x^{'}$ and $x^{''}$, r_k has also to converge more quickly to 0 than both $r_k^{'}$ and $r_k^{''}$. In order to ensure the best convergence for r_k we choose α_k in such a way that it minimizes the euclidean norm of r_k which is equivalent to minimize $f(\alpha_k)$ defined by $f(\alpha_k) = (r_k, r_k)$. Thus :

$$f(\alpha_k) = \\ \alpha_k^2(r_k^{'} - r_k^{''}, r_k^{'} - r_k^{''}) + 2\alpha_k(r_k^{''}, r_k^{'} - r_k^{''}) + (r_k^{''}, r_k^{''})$$

Two cases can appear :

- $r_k^{'} = r_k^{''}$: in this case we take $r_k = r_k^{'} = r_k^{''}$
- $r_k^{'} \neq r_k^{''}$: in this case it is easy to show that f, a polynomial of degree 2 in α_k , has a minimal value in α_k such that $f^{'}(\alpha_k) = 0$, and that this α_k is given by :

$$\alpha_k = -\frac{(r''_k, r'_k - r''_k)}{(r'_k - r''_k, r'_k - r''_k)}$$

The hybrid residual vector is proved to have a norm below or equal to the lowest norm of the two residual vectors used during the hybridization. This property guarantees a convergence at least as rapid as the best of two iterative methods used.

3.2 Hybridization of Gauss Seidel

The hybridization technique described above can be applied to two vector sequences converging to the solution, independently of the way these vectors are computed. In particular, they can be computed by the same iterative method.

In our approach (case 3 of [Leblo99a] : see Fig. 1), we use a single iterative method defined by $x'_k = u(x'_{k-1})$ and x_0 fixed. The hybrid sequence x_k is initialize using $x_0 = x'_0$. At each iteration step, we compute :

- the $k+1^{th}$ iterate of the iterative method using $x'_{k+1} = u(x_k)$
- we hybridize x_k and x'_{k+1} : $x_{k+1} = \alpha x'_{k+1} + (1-\alpha)x_k$

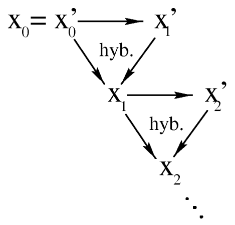

Figure 1: Hybridization of an iterative method with restart

This technique we call hybridization method with restart is described in algorithm 2.

At this time, we obtain the best results by hybridizing the Gauss Seidel method with restart. The computational cost of Δrad_r reduces to $O(m^2)$ and the final cost of group shooting becomes $O(mn)$.

3.3 Comparison of algorithms

In Table 1 we present the average shooting times for groups of different sizes in the case of the Multi-cubes scene (see Fig. 3). These times include the resolution of the sub system (computation of $\Phi^{-1}_{r,r}$ using the Jordan method in the case of GASM or the application of hybrid GS with restart in our two-level iterative method, TLISM) and the update of the overall iterate and residual vectors.

Algorithm 2 Hybrid iterative method with restart

```
initialize $x_k$ and $r_k$
while not converged do
  do an iterative step : $x_k \rightarrow x'_{k+1}$   $r_k \rightarrow r'_{k+1}$
  $\alpha = -\frac{(r_k, r'_{k+1} - r_k)}{(r'_{k+1} - r_k, r'_{k+1} - r_k)}$
  $x_k = x_k + \alpha(x'_{k+1} - x_k)$
  $r_k = r_k + \alpha(r'_{k+1} - r_k)$
endif
```

These times are in seconds and have been obtained on a PC running a Pentium II at 300 Mhz.

	100	200	300	400	500
GASM	1,18	6,15	20,19	47,47	90.96
TLISM	0.69	1.50	2.49	3.70	5.04

Table 1: Average group shooting times

The GASM method using the Jordan technique to compute the inverse of $\Phi_{r,r}$ allows us to use groups of a maximum size of 200. Beyond, the cost of group shooting becomes prohibitive and involves a low frequency of application which lead to a worse convergence rate. On the contrary, the hybrid GS method allows us to use frequently groups of size 500 and leads to a better convergence rate.

4 GROUP BUILDING

The choice of the patches to include in the group is essential to obtain a significant convergence speedup. The advantage of group shooting is due to the fact that each patch of the group benefits from the energy of all other patches in the group by multiple reflections. Consequently, it is important to take into account the level of interaction between the patches during group building.

In order to improve the quality of groups, we took inspiration from the techniques of Funkhouser [Funkh96a] in which surfaces of the scene are subdivided in groups according to a form factors graph.

In our method, groups are build dynamically during the resolution. One group is considered at a time so we have only one group to build from the graph. The form factors are already known, nevertheless their values alone are not sufficient to build the most efficient group. We have to consider also the unshot radiosity values of each patches.

Thus, we define a graph in which each node represents a patch in memory and each edge represents the level of interaction between two patches. The edges values are stored in the interaction matrix N. Each element N_{ij} represents the quantity of energy exchanges between patches i and j and is defined by :

$$N_{ij} = \rho_i F_{ij} \Delta B_j + \rho_j F_{ji} \Delta B_i$$

N satisfies $N_{ij} = N_{ji}$ and $N_{ii} = 0 \ \forall i, j$. Moreover, the level of interaction between a patch i and a group r is defined by $N_{ir} = N_{ri} = \sum_{j \in group \ r} N_{ij}$.

The size of N depends on the number of patches stored in memory. It has to be computed each time a group shooting step is done since ΔB varies all along the radiosity resolution.

We use N to define new efficient group building techniques that guarantee a high interaction level in each group. Indeed, when the energy exchanges are high in a group, the impact of its resolution on the rest of the scene is important.

We present below two new group building techniques inspired from the *"merge"* and *"split"* techniques proposed by Funkhouser.

4.1 Additive technique

The additive technique consists firstly in choosing the two patches having the highest interaction level, *i.e.* patches i and j such that $N_{ij} = max_{kl}(N_{kl})$. Subsequently, patches in memory having the highest interaction level with the group are iteratively added. In fact, the group r is build by iteratively adding the patches i for which $N_{ir} = max_j(N_{jr})$, $\forall i \notin$ group r, until the desired group size is reached.

This technique builds homogeneous groups for which the corresponding $\Phi_{r,r}$ matrix is irreducible : all the patches of each group exchange some energy with each other through direct or indirect illumination. This property is always verified as long as all the patches that have been added to group r had non zero N_{ir}.

4.2 Subtractive technique

The subtractive technique consists in initializing group r with all the patches available in memory and iteratively removing the patches having the lowest interaction level with the group. The patches i such that $N_{ir} = min_j(N_{jr})$ are removed until the desired group size is reached.

Groups built using this technique are not necessarily homogeneous as those built using the additive technique. Indeed, the choice of the patches to remove does not guarantee that the patches remaining do all interact with each other. But generally, the amount of energy of groups built using the subtractive technique is higher since the quality of groups built with the additive technique is strongly dependent on the choice of the two initial patches.

4.3 Comparison of building techniques

In GASM, groups are built by choosing among the patches in memory those having the maximum unshot radiosity values [Rouss99a]. The additive and subtractive techniques allow a higher interaction inside groups that can be easily measured. Using the definition of N in section 4, we use $N_r = \Sigma_{i,j \in group \ r} Ni, j$ to measure the interaction level of group r.

In Table 2 the average interaction levels of groups built during the Labyrinth scene illumination are presented for the original group building technique of GASM (max), the additive technique (add.) and the subtractive technique (sub.).

	100	200	300	400	500
max	0.11	0.20	0.29	0.35	0.43
add.	0.25	0.32	0.40	0.45	0.50
sub.	0.22	0.31	0.41	0.46	0.51

Table 2: Average interaction levels of groups

The additive and subtractive techniques give better results than the simple choice of patches having the maximum unshot radiosity values. However, the advantage of using these two new techniques seems to reduce when using larger groups. In fact, the number of patches and form factors rows stored into memory is 1000 in this example for all the different group sizes. Thus, the additive and subtractive techniques have more and more difficulties to build groups of better quality when the ratio of the number of patches in memory to the group size reduces. Therefore, by increasing the number of stored patches to 2000 for the Labyrinth scene, the average interaction levels of groups of size 300 become respectively 0.38, 0.67 and 0.69 for the three techniques.

5 RESULTS

We present results for two scenes both with a high level of occlusions. The first one is a labyrinth (see Fig. 2) with 3.362 patches. Its average reflectance is 0, 59.

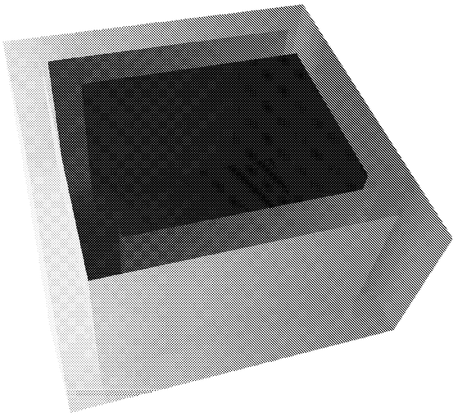

Figure 2: Labyrinth

The second scene (Fig. 3) is a cubic room including 64 smaller cubes. This geometry generates a lot of occlusions and thus involves complex energy diffusion paths. The average reflectance of the scene is 0, 41 and it includes 4.590 patches.

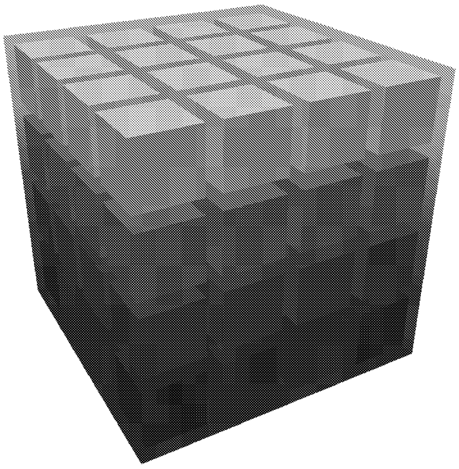

Figure 3: Multi-cubes

In Fig. 4 and 5, we compare the convergence rate for Progressive radiosity, Xu's overshooting method, GASM with and without the improvements we described in this paper (TLISM). We use the Root Mean Square error metric to measure the convergence rate of each algorithm. The results are express on a time basis in seconds.

For both scenes, PR has a very slow convergence rate due to the occlusions. More important is the low speedup provided by Xu's overshooting. As previously described, overshooting runs with an unique form factors row. Only the patches that are visible from the emitter take the advantage of the overshooting term. In the case of environments with a low level of occlusions, Xu's method outperforms the progressive algorithms. But it is clearly not designed for environments with a lot of occlusions.

The GASM approach takes the advantage of using several form factors rows. The overall visibility of the group is thus larger than the one of classical overshooting methods. This allows the GASM to provide very fast convergence rates.The original GASM is restricted to small sized groups. The group size for optimal acceleration is 200 patches. During the first steps of the algorithm (during about 2.000 seconds) convergence rate is very high. Then it becomes much lower due both to the small size of the group and the low quantity of energy that is exchanged between the patches at this point of the iterative process. By using larger groups, which is authorized by the hybrid Gauss Seidel solver, and using the additive and subtractive techniques to build them, the speedup is much more important.

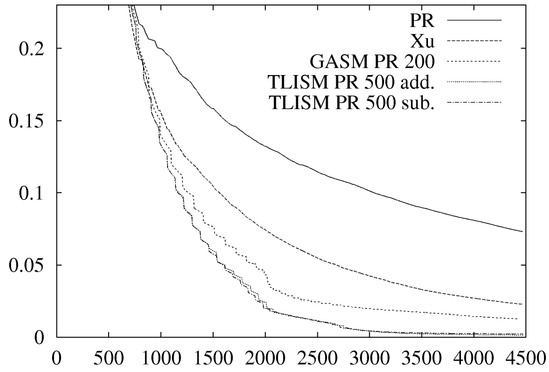

Figure 4: Convergence rate for Labyrinth

The difference is lower for the Multi-cubes scene (figure 5). However after 8.000 seconds, the RMS error is two times smaller for TLISM using 500 patches per group than for GASM using groups of size 200.

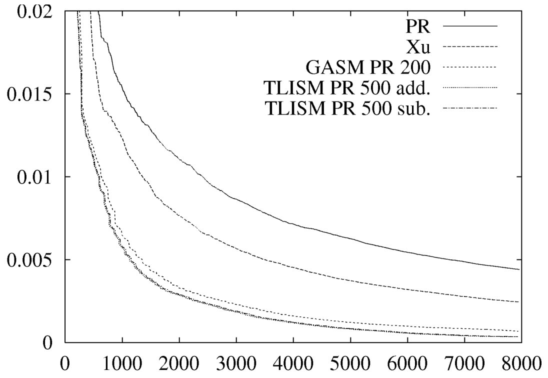

Figure 5: Convergence rate for Multi-cubes

6 PERSPECTIVES

In this paper we have presented some new enhancements to the Group Accelerated Shooting Method. This method is well suited to scenes or part of scenes with high levels of occlusion.

Using large groups is an important parameter for the speedup we can expect from GASM. However the computation time required for solving these groups was a drawback for the original method. We have presented an efficient technique that outperforms the iterative Gauss Seidel solver capabilities. The resolution of the group sub-systems is fast, even for large groups.

The second important parameter is concerned with the groups building techniques. Improving the performances of the iterative method requires to build groups with high internal energy exchanges. We have studied two groups building techniques which highlight high internal coherency and contribute to the improvement of GASM.

We just have presented TLISM for Progressive radiosity. But it can be applied to overshooting methods too. This work is today under investigation and the first results we obtained indicate the interests of such an approach. Finally, groups techniques appear as a very investigation way for both bi-level [Cohen86a] and hierarchical radiosity [Hanra91a].

REFERENCES

[Baran95a] Baranoski, G.V.G.: Bramley R., Shirley P. : Fast Radiosity Solutions For Environments With High Average Reflectance, Proceedings of Fifth Eurographics Workshop on Rendering, pp. 345-356, 1995.

[Brezi97a] Brezinski, C. : Projection Methods for Systems of Equations, North-Holland, Amsterdam, 1997.

[Chell98a] Chelle, M., Andrieu, B., Bouatouch, K.: Nested radiosity for plant canopies, The Visual Computer,Vol. 14, pp. 109-125, 1998.

[Cohen86a] Cohen, M.F., Greenberg, D.P., Immel, D.S., Brock, P.J. : An Efficient Radiosity Approach for Realistic Image Synthesis. IEEE Computer Graphics and Applications, Vol. 6, No. 3, pp. 25-35, 1986.

[Cohen88b] Cohen, M.F., Chen, S.E., Wallace, J.R., Greenberg, D.P. : A progressive refinement approach to fast radiosity image generation, ACM Computer Graphics, Vol. 22, No. 4, pp. 75-84, 1988.

[Feda92a] Feda, M., Purgathofer, W. : Accelerating radiosity by overshooting, Proceedings of Third Eurographics Workshop on Rendering, pp. 21-32, 1992.

[Funkh96a] Funkhouser, T.A. : Coarse-Grained Parallelism for Hierarchical Radiosity Using Group Iterative Methods, Computer Graphics (SIGGRAPH'96 Proceedings), pp.343-352, 1996.

[Hanra91a] Hanrahan, P., Saltzman, D., Aupperle, L. : A Rapid Hierarchical Radiosity Algorithm, ACM Computer Graphics, Vol. 25, No. 4, pp. 197-206, 1991.

[Leblo99a] Leblond, M., Rousselle, F., Renaud, C. : Accelerating Convergence of Radiosity Solvers through Hybridation, technical report LIL 99-3, 1999.

[Rouss99a] Rousselle, F., Renaud, C. : Group Accelerated Shooting Methods, Proceedings of Tenth Eurographics Workshop on Rendering, pp. 77-88, 1999.

[Shao93a] Shao, M.Z., Badler, N.I. : Analysis and Acceleration of Progressive Refinement Radiosity Method, Proceedings of Fourth Eurographics Workshop on Rendering, pp. 247-258, 1993.

[Xu94a] Xu, W., Fussell, D. : A Fast Solver of Radiosity Equation Systems, Proceedings of Pacific Graphics, 1994.