Group Accelerated Shooting Methods for Radiosity

François Rousselle and Christophe Renaud

Laboratoire d'Informatique du Littoral
Université du Littoral Côte d'Opale
BP 719 62228 Calais Cedex France
roussell,renaud@lil.univ-littoral.fr

Abstract. The introduction of the Progressive Refinement method was the starting point of interactivity in the radiosity illumination process. Overshooting methods brought an important acceleration to the convergence particularly for scenes with a high mean reflectivity.

In this paper we present a new acceleration technique to PR and overshooting methods based on group shooting methods. The acceleration is obtained by occasionally selecting groups of interacting patches and by solving the subsystem built from this group.

This technique allows us to reduce the number of iterations that are required to solve the radiosity system and only involves a small computation overhead.

Comparing different algorithms for scenes with particular properties, we highlight interesting results of the Group Accelerated Shooting Methods especially when considering complex scenes with many occlusions.

1 Introduction

Many efforts to improve the radiosity method have been made since its introduction in 1984 by C. M. Goral & al. [1]. Progressive Radiosity [2] was the first step toward interactivity and overshooting methods [3] [4] [5] improved its convergence rate, especially for scenes with high mean reflectivity. However these methods still need more iterations than the total number of patches in the scene when a nearly exact solution of the system is needed.

In this paper we introduce a new approach based on group iterative methods [6]. Unlike Greiner & al. [7] who select sets of important patches to reduce the size of the radiosity system, we accelerate existing progressive methods by shooting from group of patches in order to improve the diffusion of light. Moreover, interactions between all the patches of the group are resolved.

In the next section we describe group iterative methods and notations used to simplify the expression of the system. Definitions of group Jacobi and group Gauss-Seidel methods are given with their cost in term of operation per iteration. Direct application of these methods to the radiosity system is discussed in the case of large scenes where the form factor matrix is unknown. Greiner's blockwise refinement method is described and discussed in the same case. Then we compare the group methods approach with bi-level [10] and hierarchical [11] radiosity methods and show that it does not relate to any of these two methods.

In the third section, after rewriting the equation of group Gauss-Seidel iteration by using the residual, we give algorithms for group Southwell, group Progressive Radiosity and group overshooting methods and briefly discuss their advantages and drawbacks.

We expose in section four the implementation of group shooting iterations into Progressive Radiosity and Xu's overshooting method followed by a discussion on the optimization of group construction and efficiency.

Finally, comparative results of some group accelerated progressive methods applied on three different scenes are presented.

2 Group iterative methods

Group iterative methods are an extension of point iterative methods using several equations of the system at a time. The resolution of a group of n equations instead of the successive resolution of these n equations leads to a better convergence of the unknowns to the solution but implies a much higher cost.

Let us consider a system of n linear equations :

$$\Phi B = E \Leftrightarrow \sum_{j=1}^{n} \varphi_{i,j} b_j = e_i \qquad i = 1, \ldots, n \tag{1}$$

Point Jacobi and point Gauss-Seidel iterations are resolution schemes considering one equation of the system at a time. For example the point Jacobi's $(k+1)^{th}$ iteration is the resolution of the i^{th} equation below :

$$\varphi_{i,i} b_i^{(k+1)} = e_i \Leftrightarrow \sum_{j=1, j\neq i}^{n} \varphi_{i,j} b_j^{(k)}$$

$b_i^{(k+1)}$ is the only unknown while the b_j's are taken from the previous iteration.

Group Jacobi and group Gauss-Seidel iterations are successive resolution of groups of the system equations. For example group Jacobi's $(k+1)^{th}$ iteration restricted to a group built from equations i and j leads to the resolution of the following subsystem :

$$\left\{ \begin{array}{ccccccc} \varphi_{i,i} b_i^{(k+1)} & + & \varphi_{i,j} b_j^{(k+1)} & = & e_i & \Leftrightarrow & \sum_{t=1, t\neq i, t\neq j}^{n} \varphi_{i,t} b_t^{(k)} \\ \varphi_{j,i} b_i^{(k+1)} & + & \varphi_{j,j} b_j^{(k+1)} & = & e_j & \Leftrightarrow & \sum_{t=1, t\neq i, t\neq j}^{n} \varphi_{j,t} b_t^{(k)} \end{array} \right.$$

$b_i^{(k+1)}$ and $b_j^{(k+1)}$ are the unknowns while the b_t's are taken from the previous iteration. Thus, a group iteration using n equations leads to the resolution of a subsystem of system (1) with n unknowns.

2.1 Notations and definitions

Using an ordered grouping $\pi : G_k, k = 1, \ldots, q$ of $I = \{1, \ldots, n\}$ we define $\Phi_{r,s}$ as the sub matrix of Φ consisting of rows $R_i, i \in G_r$ and columns $C_j, j \in G_s$. B_r is the sub vector of B consisting of elements $b_i, i \in G_r$, and E_r is the sub vector of E consisting of elements $e_i, i \in G_r$. Using these notations equation (1) is rewritten as :

$$\sum_{s=1}^{q} \Phi_{r,s} B_s = E_r, \qquad r = 1, \ldots, q \tag{2}$$

Thus, an iteration of the group Jacobi solver is written as :

$$\Phi_{r,r} B_r^{(k+1)} = E_r \Leftrightarrow \sum_{s=1, s\neq r}^{q} \Phi_{r,s} B_s^{(k)}, \qquad r = 1, \ldots, q \tag{3}$$

In the same way, a group Gauss-Seidel iteration is written as :

$$\Phi_{r,r} B_r^{(k+1)} = E_r \Leftrightarrow \sum_{s=1}^{r-1} \Phi_{r,s} B_s^{(k+1)} \Leftrightarrow \sum_{s=r+1}^{q} \Phi_{r,s} B_s^{(k)}, \qquad r = 1, \ldots, q \tag{4}$$

2.2 Point versus group methods for radiosity

In the case of radiosity, point iterative methods previously described are known as gathering methods. Each iteration corresponds to the energy gathering from the scene to the patch associated with the unknown of the equation being solved. Group iterative methods correspond to group gathering. The resolution of the subsystem of Φ associated to the group G_r composed of m equations corresponds to the 'simultaneous' gathering from the scene to the m patches associated with the unknowns of the equations. By 'simultaneous' we mean that this group gathering also resolves the interactions between the m patches involved in this group, this leads to a better estimation of their radiosities than m successive gathering steps.

But group iterative methods have a higher cost in term of number of operations per iteration. Considering the group G_r associated to m equations of the system, we need approximatively m^3 operations to compute $\Phi_{r,r}^{-1}$ using a Gauss elimination method and approximatively $2nm^2$ operations to compute the unknowns of group Jacobi or group Gauss-Seidel iteration. Compared to the $3n$ operations that are required for a point Jacobi or point Gauss-Seidel iteration (even $2n$ in the radiosity case where $\varphi_{i,i} = 1$), the benefit of group methods over point methods vanishes in the general case. In fact, group iterative methods are used when Φ can be partitionned in groups with special properties permiting the resolution of each iteration in $O(n)$ operations. For example [8] gives a procedure to solve directly the subsystem when $\Phi_{r,r}$ is a symmetric and positive definite tridiagonal matrix.

Another problem of gathering methods is the lack of interactivity during the process. Most nowadays radiosity scenes consist of tens or even hundreds of thousands of patches and need a lot of time for form factors computation. Each iteration step of a gathering method requires all the form factors matrix to be available, each row of the matrix allowing the radiosity of a unique patch to be updated. Thus, one have to wait almost an entire iteration step to have a first idea of what the result of illumination will be.

In 1988 Cohen & al. [2] introduced the progressive refinement method also called progressive radiosity (PR) which shoots energy from a patch, and updates the radiosities of all the visible patches at each iteration by using a column of the form factors matrix. This method was proved to be an equivalent of the Southwell iterative method in [9]. Several overshooting methods [3] [4] [5] increased the convergence speed of PR by anticipating the amount of energy which will come back to the shooting patch in future iteration steps, and shooting it along with the current unshot radiosity. Xu's method gives the best acceleration by considering direct illumination plus an estimation of the energy due to multiple reflections in the scene before shooting.

These methods introduce interactivity during the process as a few shooting steps give a first idea of the illumination of the scene. But they also need more than n iteration steps to give a result close to the exact solution. They involve multiple selections of the same shooting patches requiring several computation of the same form factors since the whole matrix cannot be stored in memory. Moreover these methods do not converge in the same manner, it is difficult to compare for example scenes from PR and Xu's methods before the convergence is achieved. When stopping PR and Xu's methods at the same RMS error, the scene computed by Xu's method has a much higher contrast than PR solution. The patches well exposed to light sources have radiosities much closer to the exact solution than the less exposed ones. On the contrary, the more uniform illumination of the PR algorithm leads to a better solution in term of visualisation since the error with the exact solution is well distributed among all patches of the scene. Xu's method has a good convergence rate in term of mean error measure but not in term of maximum error measure. The overshooting term added to the unshot radiosity permit the acceleration of convergence only for patches that are visible from the shooting patch.

2.3 Blockwise refinement

Greiner & al. proposed in [7] a method also based on the selection of groups of important patches. The Blockwise Refinement approach solves the radiosity system by selecting in turn a small set of patches. An approximate solution of the whole system is computed for this set of patches taking into account the exact interactions between the selected patches and all other patches, and an approximation for the interactions between the others patches. This solution is refined by selecting other sets of patches that successively correct the error introduced by the approximation.

This method implies a matrix of unshot radiosities to be stored in order to keep the energy exchanges between the different patches. Thus, this method does not fit the resolution of large systems where a $n \times n$ matrix for each wavelength cannot be stored.

The comparison of Blockwise Refinement with other methods is done considering the number of floating points operations needed by the resolution of the system. This implies that all form factors are already known and that their computation is not taken into account. This does also not apply to the resolution of large systems.

Taking the form factors computation into account, our goal is to obtain a method that takes advantage both of the interactivity of shooting methods and of the accelerated convergence provided by group resolution without any $n \times n$ matrix storage. Group shooting methods described in the next section are a step towards this goal.

2.4 Comparison with substructuring and hierarchical methods

The notion of group exposed in this paper is very different from the grouping of surface elements into patches proposed in the bi-level radiosity method or even from the hierarchical organization of surfaces and energy exchanges of the hierarchical radiosity method.

In the bi-level radiosity method, a coarse meshing of the scene in N patches is subdivided in M surface elements $(M \gg N)$ for patches presenting a high radiosity gradient. A patch can be considered an element itself or consisting of a group of elements. The elements are the energy receivers thus capturing details of the illumination, while the patches are the emitters. The elements that are derived from a patch do not have any interaction between themselves unlike our groups. In our approach, groups are built from elements which exchange energy in order to accelerate the diffusion of this energy. In bi-level radiosity, groups are built in order to reduce the size of the system.

The solution obtained by the bi-level radiosity method is comparable to the solution obtained by a complete resolution of a system implying a $M \times M$ matrix constructed with all the elements, but requires only the use of a $M \times N$ matrix. This method can be viewed as a simplification of the system by compressing the matrix Φ by merging columns.

The hierarchical organization of surfaces and energy exchanges of the hierarchical radiosity method results in reducing the number of form factors needed to $O(n)$ [12]. Thus, this hierarchical organization can be viewed as a compression technique of the form factors matrix reducing the complexity of the system.

The aim of group methods is to accelerate the convergence of the iterate not by reducing the size or complexity of the system but by relaxing several unknowns at a time.

3 Group shooting methods

3.1 Group Southwell

Southwell and Gauss-Seidel relaxation methods are very similar. The only difference is that in Gauss-Seidel, the equations of the system are considered successively while in Southwell, the order in which the equations are selected is performed according to a residual criterion. Unlike group Gauss-Seidel where an iteration step consists in the resolution of all the groups of equations using equation (4), a group Southwell iteration step solves only one group of equations. The $(k+1)^{th}$ Southwell iteration step using a group r computes $B^{(k+1)}$ where $B_r^{(k+1)}$ is updated using the equation (6) described below and $B_s^{(k+1)} = B_s^{(k)}$ for all $s \neq r$.

The residual vector $R^{(k)}$ defined by

$$R^{(k)} = E \Leftrightarrow \Phi B^{(k)} \tag{5}$$

is computed at each iteration and the element of $R^{(k)}$ with the greatest absolute value defines the next group of equations to consider.

From equation (4) and using the iteration numbering described above, group Southwell iteration can be written using the residual R :

$$\begin{aligned} B_r^{(k+1)} &= \Phi_{r,r}^{-1}(E_r \Leftrightarrow \sum_{s=1, s\neq r}^{q} \Phi_{r,s} B_s^{(k)}) \\ B_r^{(k+1)} &= \Phi_{r,r}^{-1}(E_r \Leftrightarrow \sum_{s=1}^{q} \Phi_{r,s} B_s^{(k)} + \Phi_{r,r} B_r^{(k)}) \\ B_r^{(k+1)} &= \Phi_{r,r}^{-1} R_r^{(k)} + B_r^{(k)} \qquad (6) \end{aligned}$$

Thus, each unknown is updated using its value from the previous iteration. The residual can also be updated instead of directly computed from its definition :

$$R_s^{(k+1)} = E_s \Leftrightarrow \sum_{l=1}^{q} \Phi_{s,l} B_l^{(k+1)}$$

$$\begin{aligned}
R_s^{(k+1)} &= E_s \Leftrightarrow \sum_{l=1,l\neq r}^{q} \Phi_{s,l} B_l^{(k)} \Leftrightarrow \Phi_{s,r} B_r^{(k+1)} \\
R_s^{(k+1)} &= E_s \Leftrightarrow \sum_{l=1,l\neq r}^{q} \Phi_{s,l} B_l^{(k)} \Leftrightarrow \Phi_{s,r} (B_r^{(k)} + \Phi_{r,r}^{-1} R_r^{(k)}) \\
R_s^{(k+1)} &= E_s \Leftrightarrow \sum_{k=1}^{q} \Phi_{s,k} B_k^{(k)} \Leftrightarrow \Phi_{s,r} \Phi_{r,r}^{-1} R_r^{(k)} \\
R_s^{(k+1)} &= R_s^{(k)} \Leftrightarrow \Phi_{s,r} \Phi_{r,r}^{-1} R_r^{(k)} \qquad (7)
\end{aligned}$$

From equations (6) and (7), group Southwell algorithm is presented below :

```
for all $i$
  $b_i = 0$
  $r_i = e_i$
while not converged
  choose group $r$ with largest $\| R_r \|$
  $B_r = B_r + \Phi_{r,r}^{-1} R_r$
  for all groups $s \neq r$
    $R_s = R_s \Leftrightarrow \Phi_{s,r} \Phi_{r,r}^{-1} R_r$
$R_r = 0$
display image using $b_i$ as intensity of patch $i$
```

3.2 Group PR and group overshooting

We can express group PR in the group Southwell form using ∇B_r and ΔB_r as the shot and unshot radiosities of group r and A_r as the surfaces vector of patches belonging to group r :

```
for all $i$
  $\nabla b_i = 0$
  $\Delta b_i = e_i$
while not converged
  choose group $r$ with largest $\| \Delta B_r . A_r^T \|$
  compute form factors columns corresponding to group $r$
  $\nabla B_r = \nabla B_r + \Phi_{r,r}^{-1} \Delta B_r$
  for all groups $s \neq r$
    $\Delta B_s = \Delta B_s \Leftrightarrow \Phi_{s,r} \Phi_{r,r}^{-1} \Delta B_r$
$\Delta B_r = 0$
display image using $\nabla b_i + \Delta b_i$ as intensity of patch $i$
```

Finally, replacing the shot radiosity vector ∇B by $B = \nabla B + \Delta B$ we get the following group PR algorithm :

```
for all $i$
  $b_i = e_i$
  $\Delta b_i = e_i$
while not converged
```

```
  choose group $r$ with largest || $(\Delta B_r[+\Delta B_r^+].A_r^T$ ||
  compute form factors columns corresponding to group $r$
  compute $\Phi_{r,r}^{-1}$
  for all groups $s$
    $\Delta rad = \Leftrightarrow\Phi_{s,r}\Phi_{r,r}^{-1}(\Delta B_r[+\Delta B_r^+])$
    $\Delta B_s = \Delta B_s + \Delta rad$
    $B_s = B_s + \Delta rad$
$\Delta B_r = 0[\Leftrightarrow\Delta B_r^+]$
display image using $b_i$ as intensity of patch $i$
```

Group overshooting is achieved by considering ΔB_r^+ (term inside the [ ] in the previous algorithm) computed by the different overshooting methods.

Each resolution of the subsystem consisting of a group of size m allows a faster convergence than the m shooting steps of the patches composing the group. As explained in section 2.2, interations between all the patches composing a group are resolved. Thus, unshot radiosities of patches within the group are all set to 0 after an iteration of group PR, which is not the case after the shooting steps of point PR. In the same manner, unshot radiosities of patches within the group are all set to $\Leftrightarrow\Delta B_r^+$ after an iteration of a group overshooting method.

Group shooting methods are suitable for large scenes illumination because they only require the storage of m columns of Φ, m corresponding to the size of the largest group.

3.3 Group building

All group methods described in sections 2.2 and 3 use an ordered grouping as described in section 2.1. The system is therefore partitioned before starting the resolution. But in order to define well suited groups for the radiosity system we have to consider the unshot radiosity values which vary all along the resolution. It is thus preferable to construct groups dynamically during the resolution of the system. Instead of having several groups chosen at the beginning of the algorithm we construct one group at the beginning of each iteration. This leads to the algorithm below :

```
choose $m$ the size of the group
for all $i$
  $b_i = e_i$
  $\Delta b_i = e_i$
while not converged
  construct group $r$ with the $m$ patches having largest || $(\Delta b_i[+\Delta b_i^+]).a_i$ ||
  compute form factors columns corresponding to the patches of the group
  compute $\Phi_{r,r}^{-1}$
  for all patches $j$
    $\Delta rad = \Leftrightarrow\Phi_{j,r}\Phi_{r,r}^{-1}(\Delta B_r[+\Delta B_r^+])$
    $\Delta b_j = \Delta b_j + \Delta rad$
    $b_j = b_j + \Delta rad$
$\Delta B_r = 0[\Leftrightarrow\Delta B_r^+]$
display image using $b_i$ as intensity of patch $i$
```

The cost of solving of a group is not a problem when restricting to small size as compared to the number of patches in the scene. In fact, the time spent in computing form factors is much more important than the time spent in solving the subsystems (see section 5).

But those group shooting methods result in a lack of interactivity since the time elapsed between each update of the radiosities is much more important than in the case of point shooting methods. Using a group of size m, we have to wait for the computation of m columns of the form factors between each update.

4 Group accelerated shooting methods

Our goal is to find an interactive method suited to large scenes illumination and that converges to an accurate solution in a time smaller than the time required to compute the form factors matrix. The interactivity is conserved from the existing shooting methods and the acceleration of convergence is obtained from the periodic use of group shooting steps.

4.1 First approach

Giving a scene, we choose the size of the group, typically from 20 to 50 in our experimentations. It determines the number of form factors columns stored into memory. The chosen shooting method is then started and each column of the form factors matrix computed to shoot energy from a patch is stored. We keep the same order of selection of shooting patches as the shooting method.

When the number of stored patches and form factors columns reaches the size of the group, the subsystem generated by these patches is solved. The time spent solving this subsystem is measured and compared to the computation time of form factors in order to compute a frequency of group resolution applications which ensure a small overhead for the shooting method. The determination of the frequency is based on computation time because we cannot estimate precisely the cost of form factors computation in term of operations. It is not only dependent on the method used nor the number of patches in the scene but also of its geometry.

Then, at the beginning of each new iteration, the older patch stored in the group is deleted along with its corresponding form factors and replaced by the new shooting patch and its form factors. The resulting algorithm is outlined below. The m first shooting steps, the resolution of the resulting subsystem and the computation of the frequency of group resolution are not detailed in order to clarify the algorithm reading.

```
choose $m$ the size of the group
for all $i$
  $b_i = e_i$
  $\Delta b_i = e_i$
do $m$ shooting steps and store each shooting patch along with its form factors
solve the subsystem generated by these $m$ patches
compute the frequency of group resolution
while not converged
  choose patch $i$ with max $\mid \Delta b_i . a_i [+\Delta b_i^+] \mid$
  if patch $i$ is not already stored in memory
    compute the form factors column $F_{*,i}$
    replace oldest patch and ff's in the group with patch $i$ and $F_{*,i}$
  if the number of shooting steps matches the frequency of group resolution
    compute $\Phi_{r,r}^{-1}$
    for all patches $j$
```

```
      $\Delta rad = \Leftrightarrow\Phi_{j,r}\Phi_{r,r}^{-1}(\Delta B_r[+\Delta B_r^+])$
      $\Delta b_j = \Delta b_j + \Delta rad$
      $b_j = b_j + \Delta rad$
    $\Delta B_r = 0[\Leftrightarrow\Delta B_r^+]$
  else
    for all patches $j$
      $\Delta rad = \Leftrightarrow\varphi_{j,i}(\Delta b_i[+\Delta b_i^+])$
      $\Delta b_j = \Delta b_j + \Delta rad$
      $b_j = b_j + \Delta rad$
    $\Delta b_i = 0[\Leftrightarrow\Delta b_i^+]$
display image using $b_i$ as intensity of patch $i$
```

Generally, source patches have low reflectivities in radiosity scenes. Considering these patches in a group is useless and has no gain as compared to successive shooting steps because there is almost no energy transfer between them. Thus, it is preferable to shoot from all source patches before starting the construction of the group.

4.2 Optimisation of groups

By storing columns of the matrix Φ instead of columns of the form factors matrix we would save some operations needed to compute $\Phi_{j,r}$ during the group r resolution. But storing columns of Φ requires much more memory since it is proportionnal to the number of wavelenghts used to illuminate the scene. We did not choose this solution because the quality of groups constructed during the resolution can be improved using the memory available in the host machine.

Indeed, we store as much patches and form factors as possible that will be used to build the groups. The size of the group is not changed but the choice of patches included in the group is no longer set by the shooting method. For the optimized algorithm presented in section 5, the group were constructed by choosing among all the patches available in memory those having the maximum $\Delta b_i.a_i$ ($\mid(\Delta b_i + \Delta b_i^+).a_i \mid$ for overshooting methods).

But the amount of unshot energy is not always a sufficient criterium to ensure a significant speedup. The advantage of using groups is to simultaneously illuminate a great number of patches and to resolve all the interactions due to multiple reflections between the patches within the groups. It is thus essential for our acceleration method to construct the groups by maximizing the number of visible patches from the group and the amount of energy exchange between patches of the group. We are currently working on new techniques for group building based on the knowledge of the form factors stored in memory in order to suit best to these criteria.

5 Comparison of the algorithms

We use the normalized Root Mean Square error metric to compare the convergence of each method. The computational cost of each step of the different methods is dominated by the form factor computation. For the different scenes we used, the mean cpu time spent for shooting the energy is 1% in the case of PR and Xu and 4% to solve the subsystems in the case of group accelerated shooting methods, the rest of the time being spent for form factors computation. Thus, we simply use iteration steps to present each method.

We have selected three different scenes for comparing the performances of the algorithms. The first one called multi-cubes contains 480 patches and has a mean reflectivity of 0.44 (see figure 4). It is built from a large external cube including 64 smaller cubes that generate a lot of occlusions. The comparison of PR, Xu, group accelerated PR and group accelerated Xu with or without the storage of more patches than required by the group is shown in figure 1. All group accelerated methods perform very well all along the iterations giving an important acceleration at the beginning of the convergence and leading to a very good solution in less than n iteration steps. This kind of scene where light propagation is difficult seems to be well handled by our method. In this case group shooting is interesting since it allows an update of a larger number of patches than the sole shooting patch of PR or Xu's method.

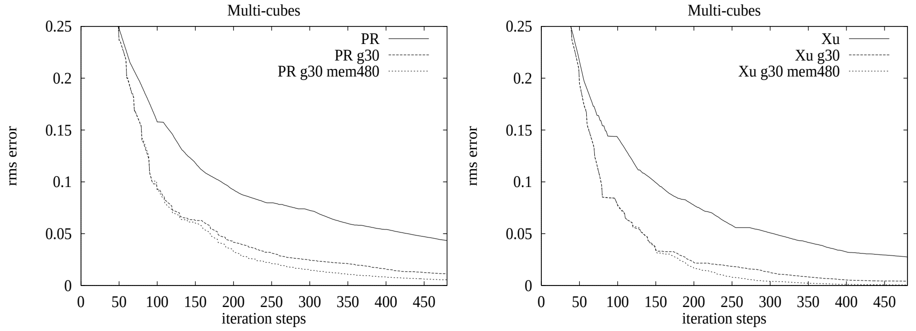

Fig. 1. Convergence for multi-cubes scene. On the left side : PR, PR g30 : group accelerated PR using groups of size 30 and PR g30 m480 : use the main memory to store all the patches and form factors. On the right side : Xu, Xu g30 : group accelerated Xu using groups of size 30 and Xu g30 m480 : store 480 patches and form factors columns

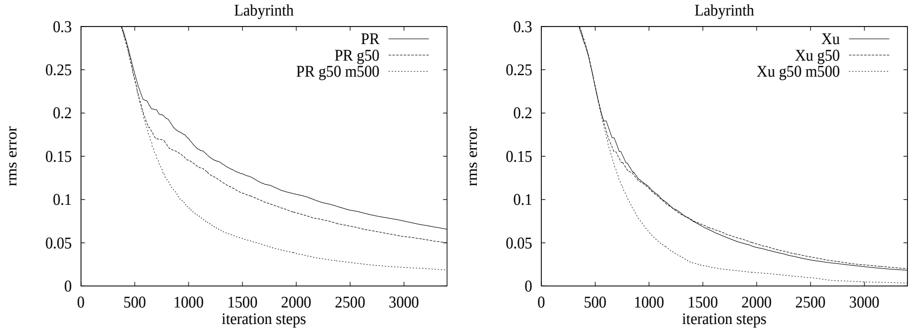

Fig. 2. Convergence for labyrinth. PR, PR g50 and PR g50 m500 on the left side. Xu, Xu g50 and Xu g50 m500 on the right side

The second scene is called labyrinth (see figure 5). It contains 3362 patches and has a mean reflectivity of 0.59. This scene is a good example of the benefit of storing several patches in order to improve groups efficiency without having to change their size when the scene has more patches (see figure 2). Again, group accelerated Xu with storage of 500 patches leads to a nearly converged solution in n iteration steps.

The last scene, shown in figure 6 is an office. It contains 10991 patches and has a mean reflectivity of 0.43. Group accelerated PR converges faster than PR, but the difference appears to be very small (see figure 3). Group accelerated Xu does not provide any acceleration even by storing some additionnal rows into memory. These poor results can be explained by several reasons. The main one is that the patches that appear in a group have a very small size as compared to the scene dimensions. Consequently when the group's patches are homogeneously distributed through the scene (which does not include a large number of occlusions) the resolution of this group does not provide any large updates both for group and group-outside patches. Furthermore we used a low number of patches per group (50) as compared to the total number of patches for this scene. Using a much more larger number of pacthes per group should provide better results. But it implies a larger computationnal cost for solving the group subsystem. We studed some new resolution techniques in [13] which could allow us to use larger group for our approach.

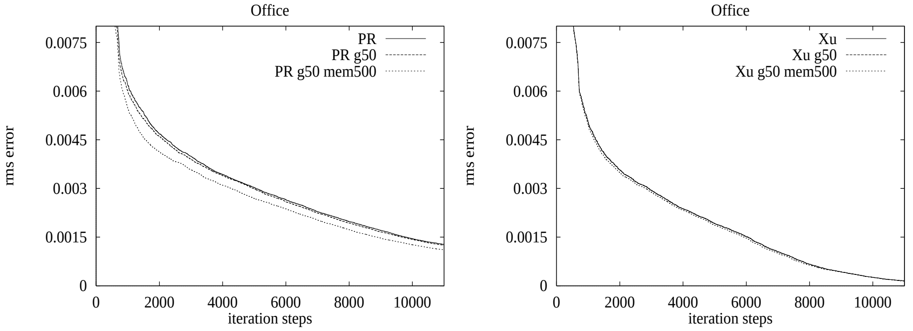

Fig. 3. Convergence for office. PR, PR g50 and PR g50 m500 on the left side. Xu, Xu g50 and Xu g50 m500 on the right side

6 Conclusion

In this paper we presented a new approach for accelerating existing shooting methods. Periodically, during the iteration of the shooting algorithm, we choose a group of important patches that have been stored into memory and solve the subsystem composed of the equations corresponding to these patches.

Shooting from groups of patches increases the diffusion of light through the scene by updating more radiosities and resolving all interactions inside the group. This method shows very good results especially in scenes with high occlusions where classical progressive algorithms have a very low convergence rate. Our results show that near-convergence can be obtained in less shooting iterations than the total number of patches in the scenes.

Studying accurately the group building choices is one of the next stage of our work. We used simple criteria for choosing both a shooting patch and patches of a group. Those criteria are too closed from the shooting algorithms ones and seems to be inefficient in some cases like the office scene. Some group properties are today under investigation in order to built the “best” group, providing the best speedup with a low computation cost.

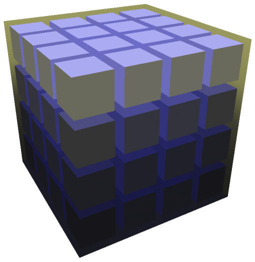

Fig. 4. Multi-cubes scene, 480 patches, mean reflectivity 0.44

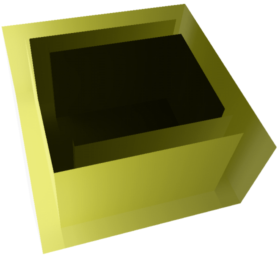

Fig. 5. Labyrinth scene, 3362 patches, mean reflectivity 0.59

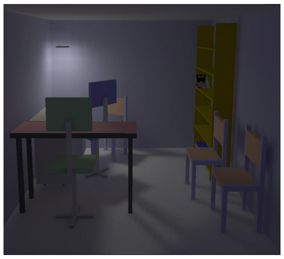

Fig. 6. Office scene, 10991 patches, mean reflectivity 0.43

Acknowledgements

We would like to thank the referees for their helpful comments which allowed us to improve the quality of our paper.

References

1. Goral C.M., Torrance K.E., Greenberg D.P., Battaile B. : Modeling the interaction of light between diffuse surfaces. ACM Computer Graphics, 18(3):pp. 213-222, July 1984.
2. Cohen M.F., Chen S.E., Wallace J.R., Greenberg D.P. : A progressive refinement approach to fast radiosity image generation. ACM Computer Graphics, 22(4):75-84, August 1988.
3. Feda M., Purgathofer W. : Accelerating radiosity by overshooting. Proceedings of Third Eurographics Workshop on Rendering, 21-32, May 1992.
4. Shao M.Z., Badler N.I. : Analysis and Acceleration of Progressive Refinement Radiosity Method. Proceedings of Fourth Eurographics Workshop on Rendering, 247-258, June 1993.
5. Xu W., Fussell D. : A Fast Solver of Radiosity Equation Systems. Proceedings of Pacific Graphics, 1994.
6. Young D.M. : Iterative Solution of Large Linear Systems. Academic Press, New York, 1971.
7. Greiner G., Heidrich W., Slusallek P. : Blockwise Refinement - A New Method for Solving the Radiosity Problem. Proceedings of Fourth Eurographics Workshop on Rendering, 233-245, June 1993.
8. Hageman L.A., Young D.M. : Applied Iterative Methods. Academic Press, 1981.
9. Gortler S.J., Cohen M.F., Slusallek P. : Radiosity and Relaxation Methods: Progressive Refinement is Southwell Relaxation. Technical Report CS-TR-408-93, Department of Computer Science, Princeton University, Princeton, NJ, February 1993.
10. 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, 6(3):pp. 25-35, March 1986.
11. Hanrahan P., Saltzman D., Aupperle L. : A Rapid Hierarchical Radiosity Algorithm. ACM Computer Graphics, 25(4):pp. 197-206, August 1991.
12. Sillion F.X., Puech C. : Radiosity and Global Illumination. Morgan Kaufmann Publishers, 1994.
13. Leblond M., Rousselle F., Renaud C. : Accelerating Convergence of Radiosity Solvers through Hybridation. Submitted to Pacific Graphics'99.