REDISTRIBUTING LIGHT

Magali Contensin

Laboratoire d'Informatique de Marseille
CMI - 39, rue Joliot-Curie - F-13453 Marseille Cedex13.
e-mail : contensin@lim.univ-mrs.fr
tel: +33 (0)4 91 11 35 54 – fax: +33 (0)4 91 11 36 02

ABSTRACT

We propose strategies for reducing the number of light sources in a scene preserving the illumination obtained with the full set of light sources. This reduction comes as a post processing to a mathematical phase of an inverse lighting method we developed. The method allows to graphically define a targeted effect in a scene with fixed geometry, the computer producing causes that lead to the desired effect i.e. a lighting configuration (number of light sources, their position and self exitances). Of course our reduction strategies may also be used in the case of direct lighting.

Keywords: energy conservation, inverse lighting, radiosity, global illumination.

1. INTRODUCTION

Existing simulation tools allow users who are willing to illuminate a scene to achieve this work in a try-and-correct fashion (Fig. 1). Users fix causes later producing effects like shadows, highlights and colors in the scene. This task is iterated until the result becomes satisfactory.

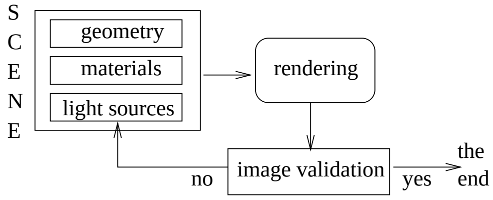


Figure 1: Direct method: try and correct

Automatic view dependent approaches for lighting design have been proposed [Marks97, Shack01]. For several years, a research axis has been to develop inverse tools that help illuminating a scene or creating the lighting design. In inverse lighting, the user no longer has to be concerned with directly configuring the light sources, but concentrates only on the desired effects. The computer produces causes that would lead to the desired lighting. Some geometrical methods have been proposed [VanWi85, Hanra90, Pouli92, Pouli97] but they do not take into account the light reflection in the scene. Constraints optimization approaches permit to deal with the global illumination: some methods [Kawai93, Schoe93] compute self exitances knowing the number of light sources and their position ; Costa et al. method [Costa99] is time expensive and its goal description is difficult to use.

We proposed an inverse lighting method which mathematically provides a lighting configuration (a number of light sources, their position and their self exitances) generating a targeted effect in the scene [Conte02]. Using the radiosity method [Cohen93], the scene is sampled in n patches and we have a set of n simultaneous equations that describe the interaction of light energy in the scene. These equations take the form of a linear system $Ab = e$ where A stands for the matrix of coefficients, b the vector of unknown radiosities and e the vector of self exitances. An interface allows the user to graphically forbid the usage of certain areas (e.g. no light source on the floor), and to paint desired radiosities on some patches. We obtain a set of constraints which are substituted in the system of equations. The unknowns are then grouped in a single vector. The obtained inverse system is solved by means of a pseudo-inverse using the singular value decomposition [Golub71]. The solution is a lighting configuration which generates an effect close to the painted one, but it has too many light sources and may contain negative values. Thus this mathematical solution is not physically valid. We use it as a starting point to propose a new lighting configuration with positive self exitances, which minimizes the number of light sources and generates an illumination close to the desired one.

In this paper we describe our method to reduce the number of light sources. This method is based on two processes: a selection of a subset of light sources within the lighting configuration and an energy conservation. Indeed the energy of the non selected light sources must be taken into account to avoid over or under-exposed scenes. We propose, analyse and compare selection methods and we give some energy conservation heuristics. We justify the choice that is made for our inverse lighting method and we give experimental results. Given a physically valid configuration which generates an effect in the scene, the try-and-correct method is time expensive to decide which light sources may be suppressed. Thus, the algorithms proposed in the case of the inverse lighting method may also be useful to reduce the number of light sources in the direct lighting case.

2. OVERVIEW OF THE METHOD

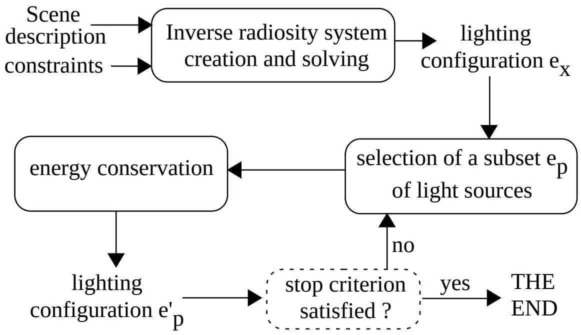


Figure 2: Inverse method general diagram

Fig. 2 presents the general working diagram of our inverse method. Given a set of constraints and a scene description the mathematical phase creates and solves an inverse radiosity system. The solution is a lighting configuration e_x which globally satisfies the targeted effect fixed by a user. Nevertheless e_x may contain negative self exitances. Moreover, the method tends to spread out the light sources over almost the whole set of possible light source patches. We propose a light source reduction method and an elimination of negative self exitances. First we select a subset e_p of light sources in e_x. This subset becomes the new lighting configuration. We give and analyse heuristics to realize this selection in section 3. Heuristics that can be used to conserve energy in the scene are described and compared in section 4. The reduction of the number of light sources is performed until a stop criterion is satisfied.

This stop criterion may be e.g., the number of desired light sources, absence of negative self exitances, an error measure, or a combination of criteria. In order to estimate the adequation between the desired effect (the m patches with radiosities b_p painted by a user) and the obtained effect (radiosities b_g generated by e'_p) we measure the global error in the scene using the L^2norm:

$$\sqrt{\sum_{i=1}^{m}(b_{p_i} - b_{g_i})^2} \qquad (1)$$

To compute Eq. 1, function $ComputeError$ used by the stop criterion test, forms and solves a system of linear radiosity equations for set e'_p of light sources. The form factors needed for the computation of b_g have already been computed and stored during the system resolution phase, they are simply reused here. This way we obtain the result faster, as the form factor computations represent about 90% of the complete radiosity solution computation time. Moreover, the closer to the solution the initial vector, the faster the Gauss-Seidel method used to solve the system converges. By using the solution vector obtained with configuration e_x, we save on an average of 60% of the iterations in the Gauss-Seidel method.

The reduction of light sources may also be used with a direct lighting method (e_x contains no negative self exitances). In that case, we consider the radiosities of all the patches of the scene as painted radiosities b_p, excepted those of e_x. Radiosities b_p of patches illuminated by e_x are compared with radiosities obtained using e'_p.

3. SELECTION OF LIGHT SOURCES

In order to eliminate negative self exitances and to reduce the number of light sources, heuristics are necessary to select a subset e_p of light sources within the full lighting configuration e_x. The goal is to obtain with e_p a result in the scene as close as possible to the one obtained with e_x. As we shall see in section 4, the self exitances of set e_q of non selected light sources must be taken into account and distributed to e_p. We suppose we have a procedure $EnergyConservation$ which does this distribution, and stores the lighting configuration after the distribution in e'_p. To decide if e'_p provides a result close to the result obtained with e_x we use the error Eq. 1.

3.1 Construction

A first solution is to iteratively add one or more light sources to an initially empty lighting configuration e_p until the error produced by e'_p becomes lesser than a threshold (or all the patches have been processed, test which is not shown hereafter). Procedure *ExitanceConstruction* adds in e_p patches that have a major influence in the illumination i.e., the highest self exitance patches. After a distribution phase, if the error generated is not acceptable, another patches with high self exitances are added, and so on. Function *DecreasingOrder* sorts the light sources taking into account their self exitance for each wavelength and their area. In Fig. 3 light sources have the same area and we consider one wavelength. First $e_p = \{S_1(30)\}$ i.e., patch S_1 is added, its self exitance is 30. After a local distribution we obtain $e'_p = \{S_1(18)\}$ the new self exitance of S_1 is 18 $(30 - 29 + 10 + 7)$. After another iteration $e'_p = \{S_1(1), S_4(17)\}$ satisfies the error threshold (S_1 take the self exitance of S_2 and S_4 the self exitance of S_3). But $e_p = \{S_4(18)\}$ could satisfy the stop criterion, because S_1 has a low self exitance after a distribution. Thus, this method tends to lighting configurations where the number of light sources is not minimal when e_x contains negative self exitances.

```
procedure ExitanceConstruction($e_x$, $threshold$)
$e_p \leftarrow \emptyset$; $e_q \leftarrow e_x$; $e'_p \leftarrow \emptyset$;
$p \leftarrow 0$;
$order \leftarrow$ DecreasingOrder($e_x$);
do
  TransferPatch($order[p]$, $e_p$, $e_q$);
  /* puts $order[p]$ in set $e_p$ and suppresses it from $e_q$ */
  EnergyConservation($e_p$, $e_q$, $e'_p$);
  $p \leftarrow p + 1$;
while (ComputeError($e'_p$) $> threshold$);
```

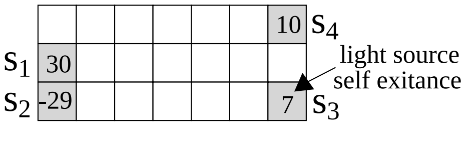


Figure 3: Light sources on a wall

Alternatively, procedure *ErrorConstruction* computes for each light source l the error obtained when l is added to the lighting configuration. It chooses patch l for which the computed error is minimal and applies this selection until the stop criterion is satisfied. In Fig. 4 the high self exitances on the left wall provide high radiosities on patch z. Negative self exitances on the right wall subtract the energy reflected by the ceiling to obtain low radiosities on y. First S_1 is added because it is the most important patch in the scene considering the error, after a distribution $e'_p = \{S_1(45)\}$. Patch S_2 is then added because it satisfies the low radiosities on y, and $e'_p = \{S_1(56), S_2(-11)\}$. Thus when e_x contains negative self exitances, we have no warranty that the final lighting configuration has only positive self exitances.

```
procedure ErrorConstruction($e_x$, $threshold$)
$e_p \leftarrow \emptyset$; $e'_p \leftarrow \emptyset$; $e_q \leftarrow e_x$;
do
  $error\_min \leftarrow +\infty$;
  for each light source $l$ in $e_q$
    TransferPatch($l$, $e_p$, $e_q$);
    EnergyConservation($e_p$, $e_q$, $e'_p$);
    $error \leftarrow$ ComputeError($e'_p$);
    if ($error < error\_min$) then
      $error\_min \leftarrow error$; select $\leftarrow l$;
    TransferPatch($l$, $e_q$, $e_p$);
  TransferPatch($select$, $e_p$, $e_q$);
while ($error\_min > threshold$);
```

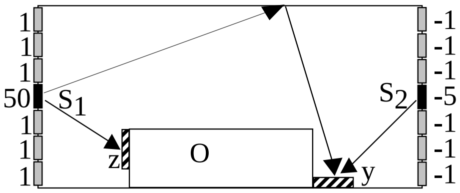


Figure 4: 2D view of a scene

3.2 Elimination

Rather than adding light sources to an initially empty lighting configuration we could suppress iteratively a set of light source patches. Procedure *ErrorElimination* computes the error obtained when light source l is suppressed from the lighting configuration. This is done for each light source. Patch l for which the computed error is minimal is suppressed. This selection is applied until the stop criterion is satisfied. In Fig. 4 suppressing patch S_1 or S_2 leading to a high error, all other light source patches are suppressed and we obtain $e'_p = \{S_1(56), S_2(-11)\}$. Thus, when e_x contains negative self exitances, we have no warranty that the final lighting configuration has only positive self exitances. Indeed, the method suppresses the low self exitances in absolute value and preserves the high self exitances in absolute value.

```
**procedure** ErrorElimination($e_x$, $threshold$)
$e_p \leftarrow e_x$; $e_q \leftarrow \emptyset$; $e'_p \leftarrow e_x$;
**do**
  $error\_min \leftarrow +\infty$;
  **for each** light source $l$ in $e_p$
    TransferPatch($l$, $e_q$, $e_p$);
    EnergyConservation($e_p$, $e_q$, $e'_p$);
    $error \leftarrow$ ComputeError($e'_p$);
    **if** ($error < error\_min$) **then**
      $error\_min \leftarrow error$; $select \leftarrow l$;
    TransferPatch($l$, $e_p$, $e_q$);
  TransferPatch($select$, $e_q$, $e_p$);
  EnergyConservation($e_p$, $e_q$, $e'_p$);
  $e_p \leftarrow e'_p$;
**while** ($error\_min < threshold$);
```

Alternatively, procedure *ExitanceElimination* sorts patches considering their self exitances and their area. It iteratively suppresses the patches from high negative to positive ones. This method eliminates negative self exitance patches, and removes patches with low self exitance values which have few influence in the illumination. Thus the stop criterion may not be based only on the error because the high negative self exitances removal implies a high error which will decrease when positive light sources are removed. First, all the negative self exitance patches are suppressed. In a second step the elimination of positive self exitance patches is performed until the error becomes not acceptable. Notice that the imposed radiosities in the scene are positive so globally the energy in the scene is positive, and we are sure to end at a positive lighting configuration.

```
**procedure** ExitanceElimination($e_x$, $th$)
$e_p \leftarrow e_x$; $e_q \leftarrow \emptyset$; $e'_p \leftarrow e_x$;
$p \leftarrow x$; /* $x$ is the number of light sources in $e_x$ */
**do**
  $order \leftarrow$ DecreasingOrder($e_p$);
  $p \leftarrow p - 1$;
  TransferPatch($order[p]$, $e_q$, $e_p$);
  EnergyConservation($e_p$, $e_q$, $e'_p$);
  $e_p \leftarrow e'_p$;
**while** (Neg($e'_p$) **or** ComputeError($e'_p$) $< th$);
```

For a direct lighting method (e_x has only positive self exitances) all the heuristics are usable. Globally direct methods should avoid selection based on minimal error which is computationally expensive. As the number of light sources desired is minimal, the most efficient method is the construction considering maximal self exitances. When the light source reduction is performed in the case of an inverse lighting method the minimal exitance patch suppression is the only heuristic which warrants a minimal lighting configuration without negative self exitances.

4. ENERGY CONSERVATION

Let e_x be a lighting configuration with x light sources, and b_x the corresponding radiosities produced in the scene. We want to suppress a set q of light sources and preserve the global illumination in that scene. The p remaining light sources produce radiosities $b_p = b_x - b_q$. We shall obtain a globally over or under-exposed scene depending on whether negative or positive self exitances have been removed. In order to produce radiosities close to b_x we have to create a new lighting configuration e'_p by modifying the p self exitances of the patches of configuration e_p.

This task could be treated as an inverse problem in which the possibly emissive patches are the set of p light sources and the imposed radiosities are either the radiosities fixed by a user (in the case of our inverse lighting method) or the radiosities of the whole scene except the radiosities of patches p and q[1] (for a light source reduction following a direct lighting method). This method will mathematically provide a set e'_p which may contain negative self exitances. Furthermore the computation of e'_p is time expensive and so an iterative selection of patches is not possible.

Alternatively, we propose to use the self exitances of the suppressed light sources to determine e'_p. Considering that exitance $e_q[\lambda][s]$ of a suppressed patch s for wavelength λ is area dependent, we use Φ_s the power leaving s i.e., its self exitance multiplied by its area A_s. Procedure *EnergyConservation* computes for each suppressed patch its power and distributes it. We propose and compare some heuristics of distribution and we present different algorithms for *XDistribute*.

```
**procedure** EnergyConservation($e_p$, $e_q$, $e'_p$)
**for each** suppressed light source $s$ in $e_q$
  $A_s \leftarrow$ ComputeArea($s$);
  **for each** wavelength $\lambda$
    $\Phi_s[\lambda] \leftarrow e_q[\lambda][s] \times A_s$;
  XDistribute($s$, $\Phi_s$, $e_p$, $e'_p$);
```

4.1 Uniform distribution

A first solution is to uniformly distribute the self exitances of the non selected patches to the p light sources taking the area of the patches into ac-

---

[1] Indeed the radiosity of a light source patch equals its self exitance plus a fraction of the energy gathered from all other patches in the scene[Cohen93].

count (procedure *UniformDistribute* which is a first version of *XDistribute*).

```
procedure UniformDistribute($s$, $\Phi_s$, $e_p$, $e'_p$)
for each light source $l$ in $e_p$
    $A_l \leftarrow$ ComputeArea($l$);
    for each wavelength $\lambda$
        $e'_p[\lambda][l] \leftarrow e_p[\lambda][l] + (\Phi_s[\lambda]/(A_l \times p))$;
```

Let us consider the three following cases of lighting configurations: the x initial light sources are either located on different walls that may be opposed (Fig. 5) or adjacent (Fig. 6), or they are coplanar (Fig. 7). In these figures representing 2D views, the p light source patches are shown in black and the q suppressed light sources in grey.

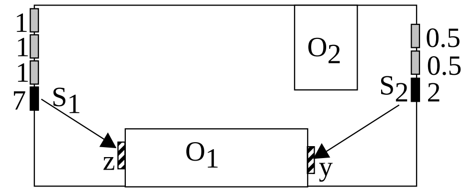


Figure 5: Initial configuration on opposed walls

In Fig. 5 all the light source patches have the same area, patches S_1 and S_2 are respectively assigned a self exitance of 9 and 4 after a uniform distribution. Object O_1 will receive with this new lighting configuration much more energy on its right part that it should. This example leads to conclude that the self exitance of a suppressed patch has to be distributed to the coplanar light source patches (S_1 and S_2 take the values 10 and 3). We obtain the procedure *CoplanarDistribute* in which the self exitance of a patch s is given to a patch l only if l and s are coplanar and the portion of self exitance distributed is function of the number n of coplanar light source patches.

```
procedure CoplanarDistribute($s$, $\Phi_s$, $e_p$, $e'_p$)
$n \leftarrow$ ComputeCoplanarNumber($s$, $e_p$);
if ($n = 0$) then UniformDistribute($s$, $\Phi_s$, $e_p$, $e'_p$);
else
    for each light source $l$ in $e_p$
        if Coplanar($s$, $l$) then
            $A_l \leftarrow$ ComputeArea($l$);
            for each wavelength $\lambda$
                $e'_p[\lambda][l] \leftarrow e_p[\lambda][l] + (\Phi_s[\lambda]/(A_l \times n))$;
```

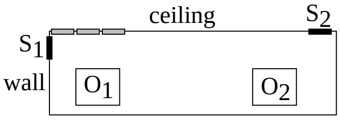


Figure 6: Initial configuration on adjacent walls

Though a uniform coplanar distribution solves the problem of Fig. 5, this solution may lead to undesired energy migration in the scene, as depicted in Fig. 6 and 7. In Fig. 6, with the uniform coplanar method, all the energy is distributed to S_2 while it should be distributed to S_1 to avoid an energy migration to the right corner of O_2. Again in Fig. 7, where all suppressed patches are close to S_2, the uniform coplanar method will equally distribute the energy of suppressed patches to S_1 and S_2 producing a result far from b_x.

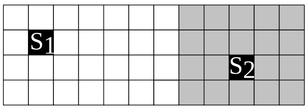


Figure 7: Initial configuration coplanar

4.2 Local distribution

We can distribute the self exitance of a suppressed patch to a subset of light source patches. Let t and y be two receiver patches in the scene (Fig. 8). The new lighting configuration must preserve their radiosities. If we consider only the direct illumination from the light source A, its self exitance cannot be attributed to E. Indeed E illuminates patch t and provides no light to patch y. But we can distribute the self exitance to B, C or D illuminating y. In this case of direct illumination a patch can take the self exitance of a suppressed patch only if it sees the same receiver patches. When the indirect lighting is taken into account, a receiver patch z may transmit energy to an occluded patch t (A lights t via z). In that case we cannot distribute self exitance to B because B does not light z. The subset of light source patches receiving the self exitance of A is then C, D. As a matter of fact, the distribution should be done taking the visibility into account. A suppressed light source can only transmit its self exitance to light source patches seeing the scene under a similar angle.

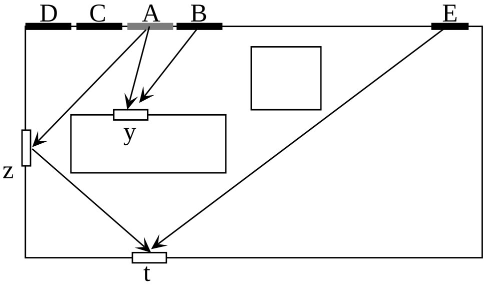


Figure 8: local distribution

Let n be the number of patches in the scene, and let F_{si} be the form factor between patches s and i. We can define a distance form factor criterion d_l between a candidate light source patch l and a suppressed patch s:

$$d_l = \sqrt{\sum_{i=1}^{n} (F_{s_i} - F_{l_i})^2}$$

The smaller d_l, the more the patches s and l see the scene under a similar angle. We can use this criterion and decide to distribute self exitance of s to the light source patch with the smallest distance d_l (procedure *LocalDistribute*). If there is more than one patch which has a minimal value, the self exitance is shared between candidates.

```
procedure LocalDistribute($s$, $\Phi_s$, $e_p$, $e'_p$)
$min \leftarrow +\infty$;
for each light source $l$ in $e_p$
   $d[l] \leftarrow$ ComputeDistanceFormula$d_l(l, s)$;
   if $(d[l] < min)$ then $min \leftarrow d[l]$;
$m \leftarrow$ ComputeNumberOfMin$(d, min)$;
for each light source $l$ in $e_p$
   if $(d[l] = min)$ then
      $A_l \leftarrow$ ComputeArea$(l)$;
      for each wavelength $\lambda$
         $e'_p[\lambda][l] \leftarrow e_p[\lambda][l] + (\Phi_s[\lambda]/(A_l \times m))$;
```

Establishing the similarity between the form factors of both suppressed and candidate patches is computationally expensive. Instead we rely on the distance between suppressed and candidate patches. As a matter of fact the form factors of two neighbours patches are relatively close. We compute the distance from the center of a non selected patch to the center of each candidate patch. The self exitance of the suppressed patch is distributed to the local candidate patches accordingly to their distance (procedure *DistanceDistribute*). Locality is defined by a distance threshold, e.g. the average distance.

```
procedure DistanceDistribute($s$, $\Phi_s$, $e_p$, $e'_p$)
$sumpond \leftarrow 0$;
$c_s \leftarrow$ ComputePatchCenter$(s)$;
for each light source $l$ in $e_p$
   $c_l \leftarrow$ ComputePatchCenter$(l)$;
   $d[l] \leftarrow$ ComputeEuclidianDistance$(c_s, c_l)$;
$th \leftarrow$ ComputeThreshold$(d)$;
$total\_dist \leftarrow$ SumDistancesInfToThreshold$(d, th)$;
for each light source $l$ in $e_p$
   if $(d[l] < th)$ then
      $pond[l] \leftarrow total\_dist/d[l]$;
      $sumpond \leftarrow sumpond + pond[l]$;
for each light source $l$ in $e_p$
   if $(d[l] < th)$ then
      $A_l \leftarrow$ ComputeArea$(l)$;
      $pond[l] \leftarrow pond[l]/sumpond$;
      for each wavelength $\lambda$
         $e'_p[\lambda][l] \leftarrow e_p[\lambda][l] + (\Phi_s[\lambda] \times pond[l]/A_l))$;
```

In Fig. 9 and 10 we compare the obtained error when the 25 light sources are iteratively suppressed and their self exitances are distributed either using a uniform or a local (neighbouring) distribution. When a light source is suppressed, the new lighting configuration e'_p must provide a global illumination b'_p as close as possible to the illumination b_x obtained with the initial lighting configuration e_x. We use Eq. 1 to measure the *error* i.e., the difference between the radiosities b'_p computed with the current lighting configuration and the radiosities due to e_x. Fig. 9 shows the results for a coplanar light source configuration whereas Fig. 10 concerns the same scene with a configuration on opposed walls. We can see that the local distribution is better than the uniform distribution.

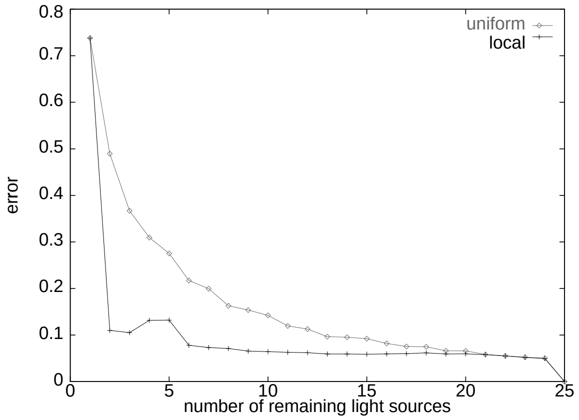


Figure 9: coplanar light sources

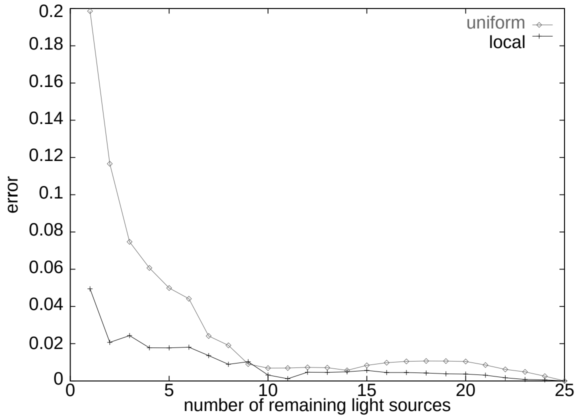


Figure 10: opposed light sources

5. EXPERIMENTAL RESULTS

We have chosen for our inverse lighting method the procedures *ExitanceElimination* and *DistanceDistribute*. The test scene is a staircase joining 3 floors, the possible light sources are the patches on the walls and the ceiling of each floor. The user imposed a bright lighting on the floor, and on the middle of the first steps (3 % of the 1500 patches have imposed radiosities). Fig. 12(a) shows the resulting scene illuminated by the lighting configuration e_x issued from the system resolution phase. The solution contains negative self exitances, and is spread out over all 685 possible light source patches. In Fig. 11 abscissa represents the number of light sources and y-axis the error of Eq. 1. We see for e_x that the difference between desired and computed radiosities is about 10^{-2}. A click on a point of the curve (Fig. 11) displays the scene illuminated by the corresponding lighting configuration in a window, examples are shown in Fig. 12(a)–(d). In Fig. 12(b) the remaining 101 light sources have been gathered towards the stage, where the most important radiosities have been fixed. The error is about 10^{-1}, which is not human visible. Indeed each point of the error curve represents a sum of differences between a target and a result for a lighting configuration. In Fig. 12(c) only 3 light sources from e_x are remaining. One is located on the ceiling, and two others on the walls on each side. The one visible on the left helps respecting the constraints fixed by the user on the steps. In this case, half of the patches has a difference about 10^{-2} and the difference for the others is 10^{-1}, which leads to a global error of 10^{-1}. Finally Fig. 12(d) proposes a smoothed view of the same scene. Total computation time is 2 hours 30 minutes for the mathematical phase (Pentium III 450 MHz / Linux architecture). The light source reduction is quasi immediate as the form factors are reused and the solution radiosity vector of the old lighting configuration is used.

6. CONCLUSION

We have presented a light source reduction method based on two processes: a selection of a subset of light sources within a lighting configuration, and an energy conservation. For each of them we have provided, analysed and compared heuristics. This light source reduction is used by our inverse lighting method to reduce the lighting configuration proposed by the mathematical phase and suppress all the negative self exitances. But it could be used in the case of a direct lighting method to reduce the number of light sources without try and correct iterations. One of the perspective of our work is to take into account the directionality in the reducing light source phase. Treating surfaces with any bi-directional reflectance distribution function can be explored using the three point method [Auppe93].

Acknowledgements Many thanks to L. Pierre.

REFERENCES

[Auppe93] Aupperle,L, Hanrahan,P: Importance and discrete three point transport. In *Eurographics Workshop on Rendering*, 1993.

[Cohen93] Cohen,M.F, Wallace,J.R: *Radiosity and realistic image synthesis.* Academic Press Professional, 1993.

[Conte02] Contensin,M: Inverse lighting problem in radiosity. *Inverse Problems in Engineering*, to be published in 2002.

[Costa99] Costa,A.C, *et al.*: Lighting design: A goal based approach using optimisation. In *Eurographics Workshop on Rendering*, 1999.

[Golub71] Golub,G.H, Reinsch,C: Singular values decomposition and least squares solutions. In *Handbook for automatic Computation, vol. II Linear Algebra.* Springer Verlag, 1971.

[Hanra90] Hanrahan,P, Haeberli,P.E: Direct WYSIWYG painting and texturing on 3D shapes. In *SIGGRAPH'90*, 1990.

[Kawai93] Kawai,J.K, *et al.*: Radioptimization - goal based rendering. In *SIGGRAPH'93*, 1993.

[Marks97] Marks,J, *et al.*: Design galleries: A general approach to setting parameters for computer graphics and animation. In *SIGGRAPH'97*, 1997.

[Pouli92] Poulin,P, Fournier,A: Lights from highlights and shadows. In *the 1992 Western Computer Graphics Symposium*, 1992.

[Pouli97] Poulin,P, *et al.*: Sketching shadows and highlights to position lights. In *Computer Graphics International*, June 1997.

[Schoe93] Schoeneman,C, *et al.*: Painting with light. In *SIGGRAPH'93*, 1993.

[Shack01] Shacked,R, Lischinski,D: Automatic lighting design using a perceptual quality metric. In *EUROGRAPHICS'2001*, 2001.

[VanWi85] Van Wijk,J.J, *et al.* Some issues in designing user interfaces to 3D raster graphics. *Computer Graphics Forum*, 4(1):5–10, January 1985.

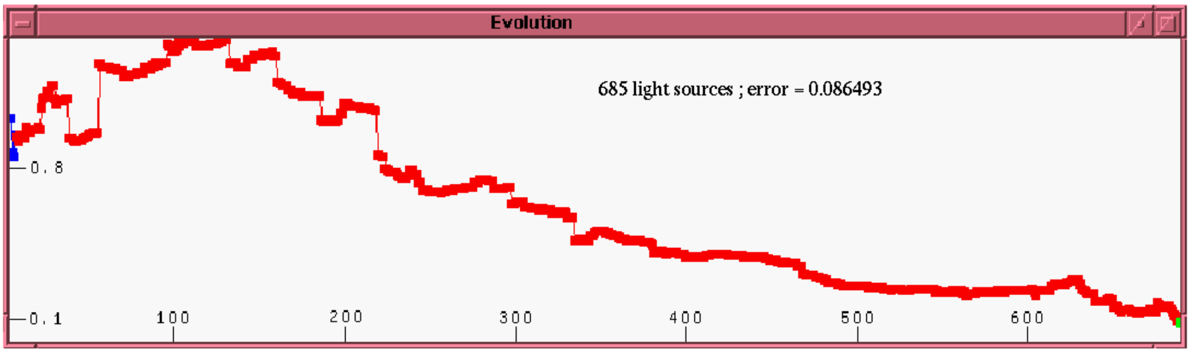
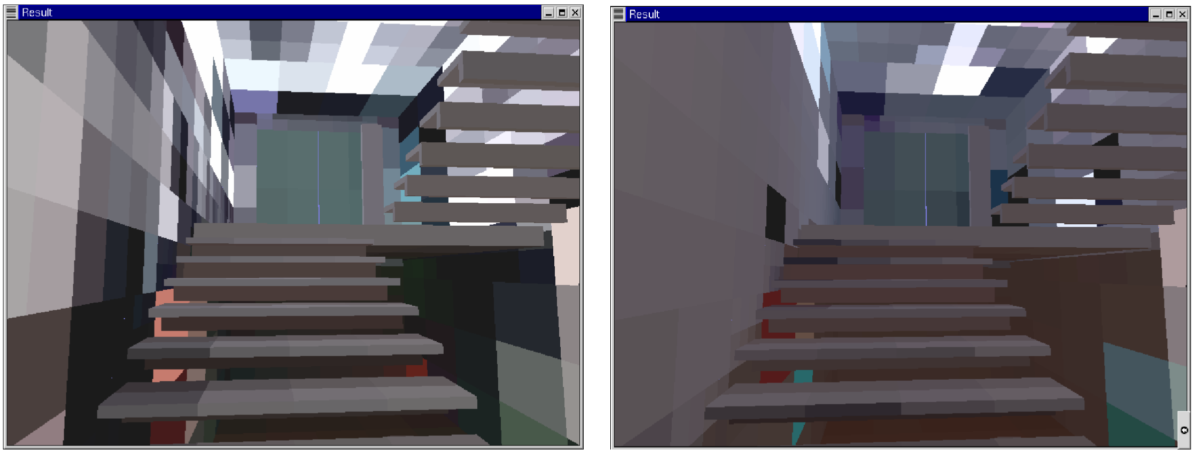
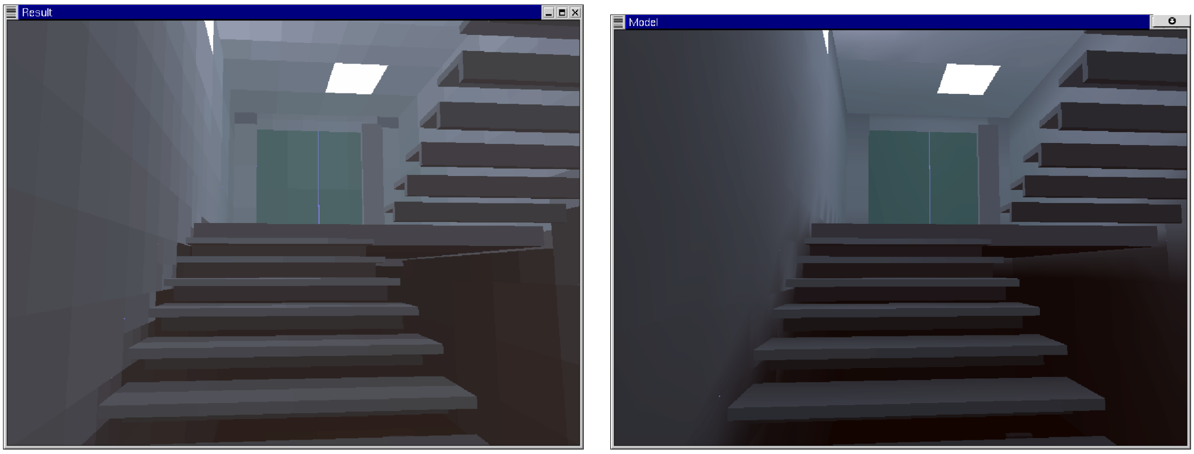


Figure 11: Error computed for each lighting configuration



(a) All light sources

(b) 101 light sources



(c) 3 light sources

(d) 3 light sources, smoothed

Figure 12: Computed radiosities for all patches