

Inverse Direct Lighting with a Monte Carlo Method and Declarative Modelling

Vincent Jolivet¹, Dimitri Plemenos², Patrick Poulingeas³

Laboratoire MSI.
83, rue d'Isle. 87000 Limoges. France.

¹ jolivet@unilim.fr
² plemenos@unilim.fr
³ poulinge@msi.unilim.fr

Abstract. In inverse lighting problems, a lot of optimization techniques have been used. A new method in the framework of radiosity is presented here, using a simple Monte-Carlo method to find the positions of the lights in a direct lighting. Declarative modelling will also be used to allow the designer to describe in a more intuitive way his lighting wishes. Declarative modelling will propose to the user several solutions, and probably some interesting scenes not previously imagined by the designer.

1 Introduction

Usually, in a rendering process, the user gives the physical characteristics of the elements of the scene. After this, he chooses a position for the lights and describes their properties. Then, using an illumination model, an algorithm is able to render the scene.

The aim of inverse lighting is to help the designer to set the lights in a more intuitive manner. An inverse lighting software should automatically compute the characteristics of the lights from the designer's wishes. This approach would be very interesting if the designer could specify high-level properties (like the ambience in a room, or the brightness of an object). These possibilities are given by declarative modelling [11]. This could drastically shorten the classical design cycle where the designer have to adjust empirically the lighting parameters.

In section 2 of this paper, we will specify the inverse lighting problem and will present the state of the art. The section 3 will introduce declarative modelling, its motivations, concepts and methods. We will then expose in section 4 our work combining radiosity, a basic Monte-Carlo method and declarative modelling, and finally we will conclude in section 5.

2 The Inverse Lighting Problem

In computer graphics, the problems of inverse lighting consist in determining, from a given scene, the characteristics of one or several lights to obtain certain type of lighting. An algorithm finds the various characteristics of lights which, integrated into the scene, will allow to create a specific ambience wished by the designer. The characteristics are for example :

- The light position.
- The light orientation (for example if it is a spotlight).
- The physical parameters of the light, as the intensity.
- The geometrical nature of the light (its shape).
- Etc.

Mostly, for a lighting asked by the user, there is not a single possible solution. A set of scenes corresponding more or less to the designer's wishes is generally generated by an algorithm of inverse lighting.

To clarify the studied problem, one is generally brought to specify three fundamental characteristics which are going to influence the proposed algorithms :

1. The nature of the lighting (direct or global).
2. The physical model of illumination (Phong's model, radiosity, computation of the radiance at a point, etc.).
3. The a priori constraints on used lights (The position of lights can be fixed, and then only the physical parameters of these lights have to be found. One can also have a set of authorized positions, or simply no constraint on the positions of the lights).

In [13], [14] and [12] are developed taxonomies from the criteria given in the previous paragraph. By following [12], one can take as classification criterion the physical model of illumination chosen in the researches made until now.

- For Phong's model: [10] and [15].
- For radiosity: [16], [5], [10], [1] and [9].
- For radiance computation at a point: [2] and [3].

In these papers, a lot of optimization techniques are used. The underlying idea is to carry out at best the designer's wishes, supposed very precise. With declarative modelling, we shall see how this presupposition can be an obstacle to the discovery of interesting solutions of lighting.

3 Declarative Modelling

Declarative modelling [11] is an investigation method of a set of solutions corresponding to a linguistic description of a problem. The studied problem is generally in scene modelling, but the paradigm of declarative modelling could apply to other areas than computer graphics. Declarative modelling tries to facilitate the work of design by finding scenes corresponding to the general idea of the designer's project.

It's frequently admitted that declarative modeling requires 3 phases:

1. The description phase where the designer specifies his desires, not by means of precise geometrical or numeric data, but by means of linguistic terms. This allows to give a vague description of certain properties of the scene.
2. The generation phase where an algorithm generates scenes corresponding to the description done in the previous stage.
3. The exploration phase where the software displays solutions, and the user retains the best ones for him (This choice can be taken into account for the continuation of this phase: One will present to the user only scenes "looking like" the previously selected ones).

In the usual inverse lighting methods, the user specifies in a very precise way the objectives (for example by giving to certain elements of the scene the color which they should have after lights will have been found by the algorithm), what justifies the use of optimization techniques to satisfy as well as possible the user's requirements.

With declarative modelling, the designer does not need to strongly detail his desires in lighting. He can give a rough description of the ambience for the scene, or for some parts of it. The resolution engine (occurring in the generation phase) of the declarative modeller is then going to propose him several solutions, among which some that he can consider interesting while he wouldn't have think about them a priori (It is moreover likely that it would not have obtained them if he had had to supply an extremely precise description of his wishes by means of numerical values, as it is the case with usual methods).

Thus, declarative modelling seems an excellent designing process because it does not enclose the user into the a priori ideas (more or less conscious) that he can have of the scene to be produced.

4. Inverse Lighting and Declarative Modelling

The inverse lighting algorithm that we propose takes place within the framework of radiosity and limits temporarily itself to direct lighting. Using a Monte-Carlo ray-casting technique [17], it extends the work of [9] made in the field of radiative exchanges and tested only for environments in two dimensions.

Moreover, we would like to use declarative description of the objectives, that is, description where only expected properties have to be given to the system. In the case of inverse lighting, the property "This part of object O has to be lighted" is an imprecise declarative description and can produce several solutions. It's possible to reduce the imprecision of a property by using fuzzy sets based modifiers like "very", "enough", etc. [4].

4.1 A New Method of Inverse Lighting

From a set O of patches to be lit, one tries to position a light, which will be a new patch added to the scene, whose shape is a quadrilateral and which should satisfy

some conditions of lighting expressed in the form of a numerical interval for the luminous flow of each patch of O.

Unlike other methods of inverse lighting radiosity, the light does not belong to the patches of the initial meshing of the scene.

To search the patches which could be lights for a scene, a simple Monte-Carlo method is used, as described in [17]. Stochastic rays are shot from every patch to be lighted, and the patches of the scene reached at first by the rays are marked.

Patches able to light directly the patches to be illuminated (belonging to the list O) are collected in a list L. When no more new patches are added to the list L after maxRays sent, no more rays are shot. It's then admitted that L contains all the directly visible patches from the patch to be lit.

The form factors between the patch to be illuminated and the patches in the list L can then be estimated. This will help us for the calculation of the emittance of the patch which will act as a light.

The proposed algorithm is the following one :

```

Procedure findLightSourcesForPatch
  // Pi is the patch to be lighted
  L ← ∅ // L is the list of the directly visible
        // patches from Pi
  numRays ← 0
  repeat
    newPatch ← false
    Shoot a ray stochastically from patch Pi
    Let Pj be the first patch hit by the ray
    if Pj ∉ L then
      L ← L ∪ Pj
      newPatch ← true
    endif
    numRaysGot(Pj) ← numRaysGot(Pj)+1
    numRays ← numRays+1
    if newPatch = false then
      alreadyHit ← alreadyHit+1
    else
      alreadyHit ← 0
    endif
  until alreadyHit = maxRays
  for all patches Pj ∈ L do
    formFactor(i,j) ← numRaysGot(Pj) / numRays
  endforall
endProcedure

```

Remark on the results:

From a certain threshold value for maxRays, the list L increases little. The patches that are added then in L have not enough influence on the patch to be lit. They present not enough importance:

- because they can lead to a contradiction by lighting a zone which the user would like to leave in shadow.

- because if they have a real importance for the objectives to achieve, they will be taken into account by the patches to be lit near the one that has just been studied (It is indeed a rather common situation: the designer is generally interested in the lighting of a set of patches forming an object or a piece of an object of the scene, and not in the lighting of a single patch).

To limit the calculation time, it is necessary to find a good value for maxRays. However determining a good quality heuristic for maxRays is difficult, because this value depends strongly on the geometry of the scene.

That is why we developed a new version of the previous algorithm in which maxRays does not appear any more.

In every stage, we shoot only a limited number of rays (of the order of 100). If at the end of a stage, the number of new patches hit by the rays is greater than 20 % of the total number of rays sent, the process is repeated. In practice, we notice that we get, in the same calculation time, almost all of the patches obtained by adjusting empirically maxRays' value for the previous algorithm.

Generally, the surface to be lighted is made of several patches. So, the problem to resolve is the lighting of a set of surface patches. In this case, the obtained lights must lie in the intersection of all the lists obtained for all the patches to be lighted. Let us call this list Λ .

Remark:

If we want to light a surface made of several patches, it is not interesting to look for the possible lights for all these patches. With a small number of patches selected on the surface, almost the whole set of the patches able to be lights for direct lighting can be found. With this process, a lot of calculation time can be saved.

To achieve this, it is necessary to choose some "good" (i.e. representative) patches of a surface (or of an object) to be lighted.

The following ideas could be applied, although they have not yet been tested in our prototype:

For every vertex of the surface, the patch (or one of the patches) including this vertex is selected. The patch containing the centre of the surface is added in the list of "good" patches.

If a whole object (or a part of an object) has to be lighted, the previous process has to be applied to all the surfaces of the object. The value of the angle formed by two surfaces will allow new optimizations. If this angle is small, it will be possible to take into account only one time some neighbor patches in the list of the patches of the scene to be lit.

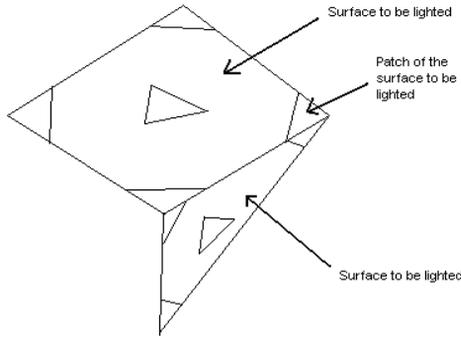


Fig. 1. Choice of the representative patches of the surface of the object to be lit.

When one deals with scenes containing few objects, it is not unusual that for 5 patches to be lit, a list Λ of much more than 300 patches able to be lights is obtained. A lot of patches belonging to the surfaces of the bounding box of the scene are indeed added to Λ if there is no object put between these surfaces and the patch to be lit, even by fixing a priori constraints to the position of the potential lights.

We thus decide to delete a patch j from the list Λ if the form factor F_{ij} is less than a threshold value ($\text{minThreshold}(i)$) for at least one of the patches i to light. The proposed heuristic for $\text{minThreshold}(i)$ is :

$$\text{minThreshold}(i) = \frac{1}{\text{Number of visible patches from } i} \quad (1)$$

With this heuristic, the number of elements of Λ decreases drastically.

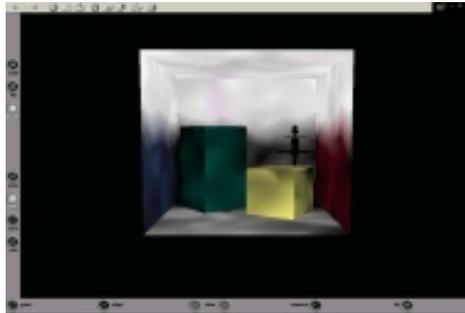


Fig. 2. All the potential lights are painted in white. The surface to be lighted lies on the top of the left side box.

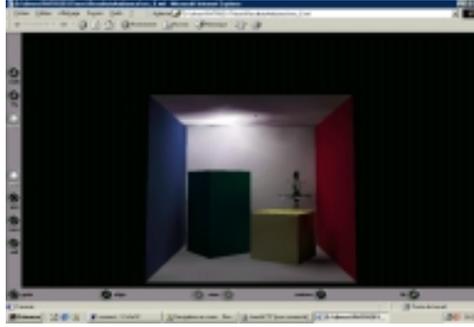


Fig. 3. Remaining lights when removing patches with low value form factors and limiting lights to be to the ceiling.

The obtained area for lights in Fig. 3 is now limited enough to start the research of a position of a light.

After having determined the list Λ , the problem is to fully determine a light source on the chosen area.

Currently, we limited ourselves to a single emitting patch (that we will note P_1), whose shape is a quadrilateral. To put this patch, a convex polygon is built from the patches of the list Λ . For this purpose, the Graham's algorithm [6] is used and applied to the vertices of the patches of the list Λ . We sweep then the convex polygon to find the suitable positions for the emitting patch.

When a suitable position has been obtained, an emittance L^e is given to the lighting patch P_1 , taking into account values allowed for the luminous flow Φ_i of patches P_i to be lit.

For every patch P_i , we have the inequalities:

$$\min_i \leq \Phi_i \leq \max_i . \quad (2)$$

Which become:

$$\frac{\min_i}{\pi \rho_i F_{li} A_l} \leq L^e \leq \frac{\max_i}{\pi \rho_i F_{li} A_l} . \quad (3)$$

Where A_l indicates the area of the patch P_1 .

It is obvious that the form factors F_{li} between the patch P_1 and the patches to be lit have to be calculated. For that, we use again the previous Monte-Carlo method.

By taking into account the set O of patches to be illuminated, the acceptable values for L^e can be found, or conclude that we have no possible solution for this position of the illuminating patch. With a discretization of the interval of the suitable values for L^e , we will be able to give the user several solutions, for the same position of P_1 .

Figures 4 and 5 show examples of lighting solutions found by our algorithm.



Fig. 4. First solution found. The patches to be lighted are on the top of the left side box.

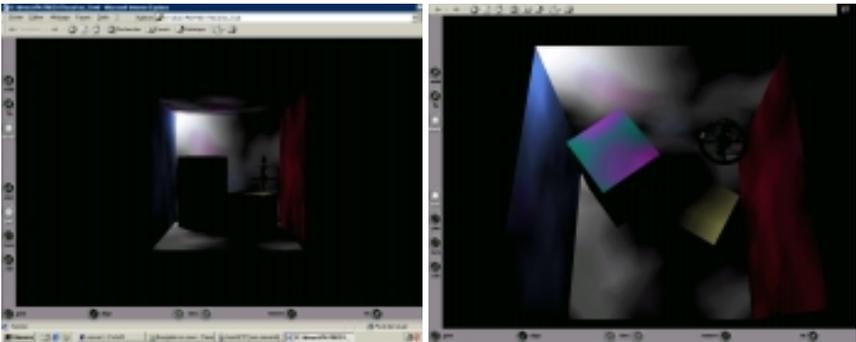


Fig. 5. Last position found for the lighting patch. The patches to be lighted are still on the top of the left side box.

Remark on the computational time of these scene: The previous scenes are composed of 5938 patches. The determination of the light source has taken about 10 seconds.

4.2 Within the Framework of Declarative Modelling

4.2.1 The Description and Generation Phases

With declarative modelling, the user is not obliged to specify details such as "patches 19 and 70 should have an energy of 1889 watts". The model of description supplied by the declarative modeller allows the designer to manipulate high-level concepts (the "table", a "weak lighting", etc.). He can so express his desires with a linguistic form more intuitive than raw numeric data. The properties which the user is going to require for the scene correspond to his still vague idea at the beginning of the modelling work.

The resolution engine will generate, from these properties, several solutions. This way of proceeding can supply to the designer a lighting that he could find interesting although he did not imagine it at the beginning of the description. The property asked by the user can be realized in several ways, especially for the number and the size of the lights:

- by a single big light put in the scene,
- by a single light, but of smaller size and put in another position,
- with several lights,
- etc.

4.2.2 The Representation Model of the Declarative Modeller

To handle the lighting properties which will be expressed with qualitative form (for example: "the top of the table is very lit"), we intend to use the works of E. Desmontils[4] where the fuzzy subsets theory is used to manage the properties of a domain (A domain is defined as the set of values that a scene, or a part of a scene, can take for a certain characteristic).

In the case of the illumination of an object, we have a "simple" property that can be modified (The object can be "very", "little", etc. lit). This property is represented by a fuzzy interval. The possible modifier can be itself a fuzzy operator (for example: "the table is more or less lit"). It is also possible to propose a "comparison" property between two parts of a scene. A modifier can then be applied to such a property (for example: "the chair is slightly less lit than the table").

To obtain a membership function (which is here a measure of \mathfrak{R} in $[0,1]$) for the illumination property, it will probably be necessary to use psycho-physical tests. The interval of values for L^e can thus be an α -cutting, where α will correspond to an acceptance threshold value ($\alpha \in [0,1]$).

5 Conclusion

We have proposed in this paper an efficient algorithm for inverse direct lighting within the framework of radiosity. The computational times are good enough for reasonable sizes of scenes and the method can also propose several solutions to the problem.

With declarative modelling, we will be able to qualitatively improve the design cycle in the lighting process. Declarative modelling, with its linguistic description of the goals to achieve, allows us to deal with many types of design constraints (as aligned luminaires) that conventional tools ignore.

In the future, we will also try to take into account global illumination supplied by radiosity.

References :

- [1] M. Contensin M., J.-L. Maltret. Computer Aided Lighting for Architects and Designers. IKM'97, Weimar, Février 1997.
- [2] A.C. Costa, A.A. Sousa, F.N. Ferreira. Optimization and Lighting Design. WSCG'99.
- [3] A.C. Costa, A.A. Sousa, F.N. Ferreira. Lighting Design: a Goal based Approach using Optimization. 10th EG Workshop on Rendering, 1999.
- [4] E. Desmontils. Formulation des Propriétés en Modélisation Déclarative à l'Aide des Ensembles Flous. Rapport de Recherche IRIN – 106. Décembre 1995.
- [5] J. Elorza, I. Rudomin. An Interactive System for Solving Inverse Illumination Problems using Genetic Algorithms. Computacion Visual 1997.
- [6] R.L. Graham. An Efficient Algorithm for Determining the Convex Hull of a Finite Set of Points in the Plane. Information Processing Letters, 1. 1972.
- [7] V. Harutunian, J.C. Morales, J.R. Howell. Radiation Exchange within an Enclosure of Diffuse-Gray Surfaces: The Inverse Problem. ASME/AIAA International Heat Transfer Conference, 1995.
- [8] J.K. Kawai, J.S. Painter, M.F. Cohen. Radioptimization – Goal Based Rendering. SIGGRAPH'93.
- [9] M. Oguma, J.R. Howell. Solution of the Two-Dimensional Blackbody Inverse Radiation Problem by Inverse Monte-Carlo Method. ASME/JSME Thermal Engineering Conference, 1995.
- [10] P. Poulin, A. Fournier. Lights from Highlights and Shadows. March 1992, Symposium on Interactive 3D Graphics.
- [11] D. Plemenos. Declarative Modelling by Hierarchical Decomposition. The Actual State of the MultiFormes Project. GraphiCon'95, St Petersburg, 1-5 juillet 1995.
- [12] P. Poulingeas. L'Eclairage Inverse – Etat de l'Art. Rapport de Recherche MSI 01-01. Octobre 2001.
- [13] G. Patow, X. Pueyo. A Survey on Inverse Reflector Design and Light Source Distribution Problems. Institut d'Informàtica i Aplicacions, Universitat de Girona. Private Communication.
- [14] G. Patow, X. Pueyo. A Survey on Inverse Emittance and Inverse Reflectometry Computation Problems. Institut d'Informàtica i Aplicacions, Universitat de Girona. Private Communication.
- [15] P. Poulin, K. Ratib, M. Jacques. Sketching Shadows and Highlights to position Lights. Proceedings of Computer Graphics International 1997.
- [16] C. Schoeneman, J. Dorsey, B. Smits, J. Arvo, D. Greenberg. Painting with Light. SIGGRAPH'93.
- [17] P. Shirley. Radiosity via Ray Tracing. Graphics Gems II, p. 306-310, James Arvo Ed., Academic Press, San Diego. 1991