OPTIMISATION AND LIGHTING DESIGN

António Cardoso Costa[I II]
acc@dei.isep.ipp.pt

António Augusto Sousa[III II]
augusto.sousa@inescn.pt

Fernando Nunes Ferreira[III II]
fnf@fe.up.pt

ABSTRACT

Nowadays, the available tools for lighting design are limited in scope and inappropriate for interactive or creative use. In this paper, we present a different approach to the lighting design problem – a new methodology that includes the geometry and materials of the scene, the lighting design goals and a computational search for lighting solutions. The objective is to improve and shorten the lighting design cycle. The most common result is a set of luminaire characteristics that maximises the attainment of the user defined lighting goals. These goals may include different kinds of constraints or objectives. We also discuss the use of optimisation in lighting design, describe an algorithm that implements the proposed methodology and present several examples.

Key words: lighting design, optimisation, global illumination, light transport.

INTRODUCTION

The development of a computer program that uses the geometry and materials of the scene plus user defined lighting goals and that determines luminaire characteristics seems to be a difficult task. A tool like this could free the lighting designer from most of the computational and mathematical details of existing computer programs and allow the exploitation of lighting solutions in virtual or real spaces. In virtual spaces, the use of such a tool could also help the architectural design phase because lighting and geometry are very closely related.

In this paper, we describe an approach that helps to solve the problem of finding the luminaire characteristics from the geometry of the scene, the properties of materials and the lighting goals. This approach has its own difficulties because there can be many incompatibilities in the input information. These have a physical meaning and can be dealt with by changing some of the input data (geometry, material properties or lighting goals). However, this feature is advantageous because it is possible to avoid searching for lighting solutions that are "impossible" (in a physical or architectural sense). The solutions found can be confirmed with a direct illumination approach using conventional computer programs.

Fig. 1 shows the standard approach in lighting design. The designer defines the scene's geometry, material properties and luminaire characteristics. Quantitative data is then computed so that the designer can make a decision by comparing this data with the lighting goals. If he is not pleased with the results, we will have to change some parameters and repeat the process several times until a satisfactory solution is found.

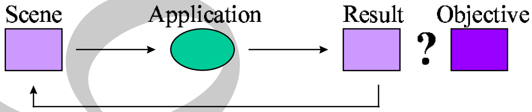


Fig. 1. Standard Approach in Lighting Design

Fig. 2 shows the a different approach, where the designer includes lighting goals in the input data. A user-driven computational search iterates until some preset termination criterion is achieved. The best solutions are shown and a decision can then be made. The three types of solutions in the right part of Fig. 2 show that it is possible to generate impossible, unwanted or adequate solutions. The designer can choose to refine the lighting goals and restart the

---

[I] Departamento de Engenharia Informática do ISEP-IPP, Rua S. Tomé, 4200 Porto, Portugal.
[II] INESC Porto, Praça da República 93 R/C, Apartado 4433, 4007 Porto CODEX, Portugal.
[III] Faculdade de Engenharia da Universidade do Porto, Rua dos Bragas, 4099 Porto CODEX, Portugal.

process or, alternatively, to apply his own judgement and choose the most suited solution.

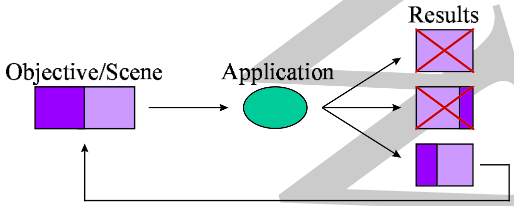


Fig. 2. Proposed Approach in Lighting Design

The methodology described in this paper fits within the frame of Fig. 2. It is based on well known algorithms and techniques from Global Illumination and Optimisation (the feedback loop depicted in this figure is implemented with optimisation techniques).
The main goal of this work is to improve and desirably shorten the lighting design cycle. Because existing tools do not correctly address this problem, the authors think this is an interesting and promising field of research.
In this paper, the "Related Work" section describes related research work, while the "Overview" section explains the new approach and our methodology. The "Implementation" section describes the algorithm implemented and the techniques used. Some examples are shown in the "Results" section. Finally, the conclusions are presented in the last section.

RELATED WORK

Schoeneman *et al.* [Schoe93] defined very well the interest of inverse lighting simulation, but solved it only partially for environments where the position and light emission distribution were known *a priori*, computing only the actual emitting luminaires in respect of their colours and intensities. They used a radiosity based algorithm [Cohen93].
Another interesting work was done by Kawai *et al.* [Kawai93], which was also based on the radiosity algorithm. Their method allowed the calculation of intensity, colour and kind of lighting distribution for each luminaire, and also computed the reflectance on any scene element or patch.
Poulin *et al.* [Pouli97] described an approach that allows the interactive positioning of light sources (simplified luminaires) from sketches of shadows, umbra, penumbra or highlights. They did not use a global illumination algorithm to guide the calculations, so results are questionable for indirectly lit spaces.
The Design Gallery approach of Marks *et al.* [Marks97] is also interesting, because it presents a new methodology for the exploration of solutions in large multidimensional problems, different from conventional methodologies like interactive evolution or inverse design. Although one of their examples is a lighting design study, they do not use a global illumination algorithm (only a simpler local ray tracer); also the dispersion phase of Design Galleries (solution generation and filtering) can become prohibitively expensive in computation time if many solutions have to be generated for the following browse and select phase.
A simple methodology for the proposed approach has been previously described by Costa *et al.* [Costa98]. In [Costa99], the authors describe in detail the different methodologies used to address the lighting design problem within the context of the proposed approach.

OVERVIEW

The "ideal" lighting design approach is simple to describe – the inputs are:

- the geometry of the scene
- the properties of the materials
- a set of statements (more or less abstract) about lighting goals.

The outputs are:

- a set of solutions (possibly empty)
- suggestions to overcome the limitations found.

The current technology does not yet allow this "ideal" approach, so several simplifications have to be made to make this problem manageable. On one hand, the level of interactivity is very low, which means that it is advisable to introduce some automated search strategy in the process in order to find the solutions as fast as possible (within predefined limits). On the other hand, the set of statements about lighting goals has to be restricted to simple quantitative lighting constraints, for now. Finally, the lighting computation must be simplified as possible, without compromising correctness, so that the iterative search does not need large amounts of computation resources.
In this work, we did not intend to reduce the computation time but mainly to find new ways to improve and shorten the lighting design cycle; research should be done to improve the algorithms and techniques used.

Background

The general approach to the global illumination problem has to determine how reflected light in a particular direction from some point affects light reflected in all directions in all other points in the scene. For any point in space, it is possible to find how radiance (light) behaves when passing through the point. In non scattering media like air, radiance does not change along a non obstructed direction, so radiance can be computed only at surface points. The directional radiance distributions shown in Fig. 3 are two examples of this behaviour. We call them SRD (spherical radiance distribution) and each angular contribution can be different from the others, as these SRDs show. To improve the clarity of images in Fig. 3, a triangulation was applied to the SRDs vector extremities and the back faced triangles were then removed.

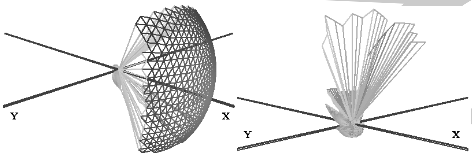


Fig. 3. Examples of Spherical Radiance Distributions

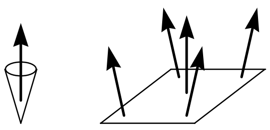
To collect lighting data for processing, we use SRDs all over the scene. A SRD is computed by sampling a set of directions inside some predefined solid angle, because normally it is not possible to compute analytically the function that describes radiance behaviour around a point.

One advantage of this approach is its independence from any particular computer program for global illumination simulation. The only requisites that must be respected by a program of that type are:

- physically valid *BSDF*[1] model
- no restrictions for the properties of materials
- light sources with arbitrary radiance distribution
- radiance calculation anywhere in space.

Luminaires

The proposed approach accepts a set of simplified quantitative statements about lighting to guide the search for solutions. These statements can be constraints: geometrical (distances, volumes, etc.), lighting related (radiance, irradiance, etc.) or others. One of the challenges this work presents is how to include complex restrictions. Many lighting related restrictions can be modelled by the use of inverse luminaires (IL), which are fictitious luminaires that emit fictitious radiance. This is used, through calculations with SRDs, to guide the search for solutions. The ILs can be of two basic geometric types, point or area (Fig. 4).



a) Point b) Area

Fig. 4. Types of Inverse Luminaires (IL)

For each type, the designer must define the fictitious radiance to emit and some solid angle of emission.

If there are already luminaires in the scene (lamps, day light, sun light, etc.), which we call previous luminaires (PL), they must be taken into account in the attainment of lighting goals. ILs definition must be monitored considering the PLs, so that physically impossible situations are avoided as soon as possible. Impossibilities or conflicts may be overcome by input data modifications or by redefinition of goals. The lighting solutions will normally be luminaire characteristics – the luminaires that result from it are called desired luminaires (DL).

In short, the luminaire classification is:

- PL – previous luminaire; a luminaire in the scene
- IL – inverse luminaire; a fictitious luminaire used in the lighting design process representing an objective to achieve:
  1) positive; satisfy some minimum threshold
  2) negative; satisfy some maximum threshold
- DL – desired luminaire; a lighting design result.

Methodology

Previous luminaires (PLs) produce an initial radiance in the scene that must be taken into account in the definition and setup of lighting goals, most of them modelled by ILs, so the initial processing phase (direct lighting simulation) calculates the lighting generated by the PLs in the form of SRDs. Then, for each point type IL, we subtract the weighted SRDs previously calculated from its own SRD and the result will be the IL fictitious radiance to emit into the scene. If the difference is negative in at least one direction, we conclude that a conflict exists between the PLs present in the scene and the lighting goals represented by that IL. This is clearly a design conflict and can only be overcome by user intervention. For an area type IL, several SRDs must be sampled; if there are no significant differences, a representative SRD can be chosen; otherwise, it is advisable to subdivide it into several smaller ILs, repeating this process until some termination criterion is achieved.

Other lighting goals (geometrical restrictions, etc.) are handled differently and will be explained later.

This methodology allows the designer to define the scene's volumes where the desired luminaires (DLs) will be located and DLs parameters or relationships between them. Each DL normally has 6 degrees of freedom (6df), 3 for position, 2 for direction and 1 for emission angle. The obtained solutions are sets of DLs characteristics (position, direction, etc.) that maximise some function of the fictitious radiance each DL receives. This function can be as simple as the sum of those incoming radiances for all DLs, or weighted by design goals, usually of geometrical type[2]. As stated before, there are positive and negative ILs: positive ILs emit fictitious radiance that increases the global incoming radiance; negative ILs emit fictitious radiance that decreases it.

Using an optimisation algorithm it is possible to calculate automatically solutions that maximise some

---

[1] Bi-directional Scattering Distribution Function.

[2] These are the types of lighting goals which can not be modelled using ILs.

function of incoming DL radiance[3]. This will be the cost function $F(x)$ of the optimisation phase; the main goal is to find its global minimum. The flexibility in the definition of a cost function requires a sophisticated optimisation algorithm, capable of solving problems with many degrees of freedom and able to cope with complicated cost functions, eventually non-linear or mathematically complex.

A straightforward direct simulation can later be made so that the user can check the match between desired objectives and obtained results.

Luminaire Calculation and Optimisation

Inverse lighting simulation produces raw data that can be used to derive valuable information. For example, in a certain point in the scene, several directions can be sampled and radiances computed. By itself, this does not give any clue about the best place to position an hypothetical luminaire with a particular emission solid angle and distribution efficiency. To address this issue it is necessary to have some iterative process to test points in the scene inside the luminaires predefined volumes. For each iteration it is necessary to compute some measure of quality in the search for the "best" solution.

Even though position and direction are continuous quantities it is reasonable to consider them discrete; in this instance, the configuration space becomes a large set of points and directions. It is not reasonable to do an exhaustive search because the number of candidate solutions may be very large. The time needed to compute radiance along a certain direction from any point is significant and invalidates this brute force methodology. Even if some filtering is done on the parameter space, for example, considering only downward directions, the number of candidate solutions is still too large.

The most interesting search strategies [Pirlo96] for use in lighting design are local search strategies, which basically move from one solution to another one in its neighbourhood according to some well-defined rules. In lighting design, because the cost function $F(x)$ is usually very complex and highly non-linear, straightforward strategies are not adequate for solving the problem, because the global minimum may be very difficult to find. Among local search strategies explicitly designed to cope with these difficulties (like temporary deterioration of the cost function $F(x)$), a very interesting one is Simulated Annealing (SA). Simulated Annealing [Kirkp84] has attracted attention as a suitable strategy for large scale optimisation problems, especially those where the global minimum is hidden among many local minima. A typical SA algorithm requires:

- a description of system configurations
- a configuration change random generator
- a cost function, whose minimisation is the goal of the algorithm
- a control parameter and an annealing schedule.

According to [Ingbe93], SA is an optimisation strategy that can:

- process cost functions with quite arbitrary degrees of non-linearities, discontinuities and stochasticity
- process arbitrary boundary conditions and constraints imposed on the cost function
- be statistically guaranteed to find an optimal solution.

Luminaire optimisation is the problem of finding the best combination of luminaire parameters that satisfies some user defined criteria, including restrictions. The parameters are related to geometrical and lighting issues and the restrictions are related to possible luminaire locations, orientations, etc. It is also possible to impose relationships between luminaires and the scene's parameters. These can be of many types such as positional, directional, logical and others. This shows the great flexibility at the disposal of the user to define the optimisation context, supposing it is reasonably and accurately formulated. A badly specified optimisation context will probably produce "erroneous" answers.

IMPLEMENTATION

Fig. 5 shows an outline of the algorithm that implements the new methodology. The implementation of this methodology makes use of several algorithms and techniques:

- a light transport algorithm
- an IL refinement algorithm (account for PLs)
- a SRD calculation algorithm
- an optimisation algorithm
- a script language for communication between the optimisation and the light transport tools (the double arrow in Fig. 5)

The input data is the geometry of the scene, properties of materials and lighting goals. The geometry and properties are usually described in a quantitative way using a conventional data format. Any existing scene's luminaires and lighting goals are described by the use of ILs and PLs.

---

[3] Or equivalently, that minimises its negative.

WSCG

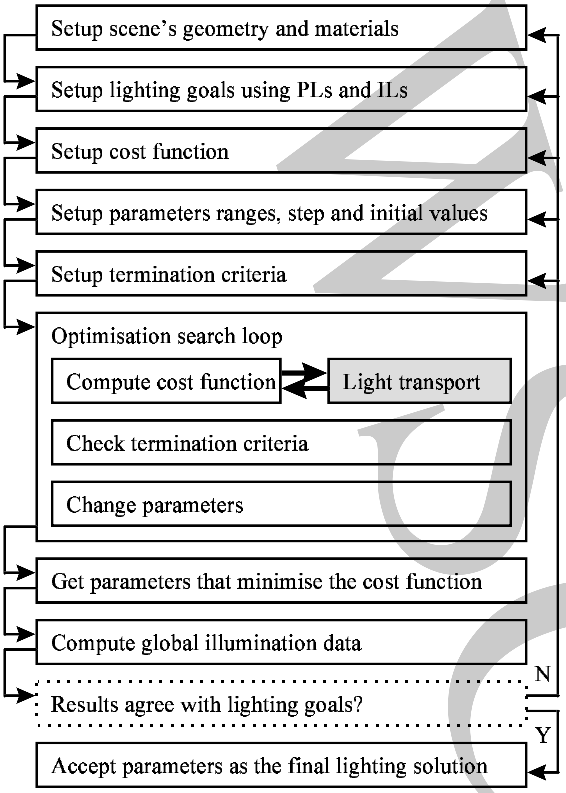


Fig. 5. Algorithm for the new methodology

Light Transport Calculation

The light transport algorithm we use is the *RADIANCE* lighting software [Ward94]. In *RADIANCE*, ILs are treated like conventional light sources (point or area). A considerable flexibility is allowed in light source specification, including spotlight effects (user defined emissive solid angles) and arbitrary radiance emissive distribution curves for light sources.

The light transport calculations done inside the optimisation loop of Fig. 5 are the main responsible for the large computation times in this implementation. Each calculation is similar to a standard global illumination light transport calculation, although we are not computing through a rectangular array of pixels in 3D space like in a conventional image but in a set of directions around a 3D point. As the geometry does not change during the optimisation loop, any cache/coherence technique that is able to speed up calculations without introducing significant errors is useful to reduce computation times.

SRD Sampling

Spherical radiance distributions (SRD) are calculated by collecting radiance samples around a point. Due to the complexity of lighting, many samples are necessary to capture the most relevant phenomena (shadow boundaries, specular highlights, etc.). In our test cases, we have found that 4096 distributed samples on the direction sphere are a good compromise between accuracy and performance, although 2048 samples usually produce similar results more quickly. To calculate these directions (i.e., points in the unit sphere surface), we have implemented a Metropolis algorithm to compute a quasi-optimal set of "evenly distributed" points that minimise a metric based on geodesic distance between points.

Optimisation

The optimisation algorithm used is the *ASA* optimisation software [Ingbe1998] (a SA algorithm). In all tested cases, it has shown good results in less than 5000 iterations (usually producing less than 200 intermediate solutions). When constraints are very complex the number of iterations must be increased because invalid parameter configurations are only rejected after being generated (without performing global illumination calculations, which take the largest part of computation time).

Cost Functions and Light Transport

In any optimisation problem, a cost function must be able to measure the quality of the obtained solution during the iterative search. Choosing the most adequate function that leads to the best solution is not trivial. There are infinite variations of cost functions, but only a small fraction will produce useful results.

The simplest situation is when there exist only ILs. The cost function $F(x)$, which depends on the parameters of configuration space X (positions, directions, etc.), should be proportional to some integral value of the fictitious radiance in the scene, scaled by auxiliary functions which are used to introduce design constraints or relationships in the design. The configuration space X should be appropriately chosen so that it agrees with lighting design goals (for example, it does not make any sense to search for solutions near the floor if the user desires to have solutions near the ceiling).

The cost function can also be considered the interface between the optimisation and the light transport tools. We have created a script language to represent this interface[1]: the optimisation parameters are predefined variables (x_1, x_2, etc.) and there are several auxiliary functions to help the calculation of the cost function value. There is also the special function **Importance**() that actually runs the light transport simulation to compute SRDs for points in the scene. Consider the following example:

```
# ($x_1$,$x_2$,$x_3$):position
if Distance(($x_1$,$x_2$,any),Point$_A$)<1.0 return FAILURE
$W_{IL1}$=Importance(Scene$_{IL1}$,$x_1$,$x_2$,$x_3$,...)
return -($W_{IL1}$*K*Distance(($x_1$,$x_2$,$x_3$),Point$_B$))
```

The configuration parameters x_1, x_2 and x_3 represent

---

[1] A straightforward free form language not described in detail due to lack of space.

spatial ordinates. This cost function ignores configurations where points are inside the cylinder of radius 1 centred on some point A (line 2), scaling all other cases using the distance between configuration spatial ordinates and some pre-defined point B (lines 3 and 4).

If there is more than one DL to optimise, then the cost function must combine the influence of the several DLs.

```
1. # $(x_1,x_2,x_3),(x_6,x_7,x_8)$:positions $(x_4,x_5),(x_9,x_{10})$:directions
2. if Distance($(x_1,x_2,$any$)$,$Point_A$)>4.0 return FAILURE
3. if Angle(Dir($x_4,x_5$),(0,0,1))>45° return FAILURE
4. if Distance($(x_1,x_2,x_3),(x_6,x_7,x_8)$)<$D_{threshold}$ return FAILURE
5. $W_{IL1/DL1}$=Importance($Scene_{IL1},x_1,x_2,x_3,x_4,x_5$)
6. $W_{IL1/DL2}$=Importance($Scene_{IL1},x_6,x_7,x_8,x_9,x_{10}$)
7. if $W_{IL1/DL2}>W_{threshold}$ return FAILURE
8. return $-(K_1*W_{IL1/DL1}+K_2*W_{IL1/DL2})$
```

As the example shows, it is possible to include many different boundary search conditions and calculation criteria. It could be translated into plain English as *"I want one luminaire directed to the ceiling* (line 3) *and inside a cylindrical volume around this vertical line* (line 2)*; the other luminaire must not be very near* (line 4) *and is not as important as the first one* (line 7)*"*.

To show a more complex cost function example in a lighting design problem with PLs, suppose the user states *"I would like to enhance the lighting of this room by adding another projector type luminaire (an halogen projector luminaire) hanging from the ceiling so that illumination on the top of this table improves, but without interfering with the computer screen by creating any kind of glares or other undesirable lighting effects"*. This could be achieved by the definition of 2 ILs, one for the desktop (positive) and another for the user face in front of the computer screen (negative). As stated before, PLs affect ILs. An initial global illumination light transport simulation must be performed and the resulting ILs calculated so that they can be used in the optimisation (IL_1 and IL_2). An adequate cost function is:

```
1. # $(x_1,x_2,x_3)$:position $(x_4,x_5)$:direction
2. V=Vector($Face_{Center},(x_1,x_2,x_3)$)
3. if Angle($Face_{Perp}$,V)<$V_{threshold}$ and
   Angle($Face_{Perp}$,Dir($-x_4,-x_5$))<$A_{threshold}$ return FAILURE
4. $W_{IL1/DL1}$=Importance($Scene_{IL1},x_1,x_2,x_3,x_4,x_5$)
5. $W_{IL2/DL1}$=Importance($Scene_{IL2},x_1,x_2,x_3,x_4,x_5$)
6. return $-(K_1*W_{IL1*/DL1}-K_2*W_{IL2*/DL1})$
```

This example shows how to include directional constraints in the cost function (lines 2 and 3). The purpose of these lines is to avoid the specular problems stated above. Constant K_2 (the weight associated to the negative IL_2) should also be considerably greater than K_1 to emphasise the avoidance of glare effects.

RESULTS

The performance of the described algorithm is already comparable to a standard global illumination light transport simulation producing a high quality image. For example, a design study using our algorithm usually produces good lighting solutions in 3000 iterations. Each iteration requires usually no more than 1024 radiance samples per IL used (one direction hemisphere). If there are 3 ILs in the study, that will mean 9216000 radiance calculations (3000x1024x3).

In a conventional design study, an image of size 1024x1024 with 16 radiance samples per pixel requires 16777216 radiance calculations. If more images are needed the computation time will increase linearly. One method designers use to overcome this is to compute smaller images (at 512x512 with 16 radiance samples per pixel, 3 images will require more radiance calculations than our algorithm).

Examples

The first example shows the flexibility of this methodology and explains the use of cost functions. The second example is more complex and is similar to a real world lighting design study (hall room).

Simple Example

This example uses a very simple geometry and describes the flexibility of cost functions for luminaire parameter optimisation. Two desktops are represented by two polygons, at height 0.8m (Fig. 6). The initial lighting objective is to have a homogeneous lighting distribution over the desktops, which will act as two independent positive ILs.

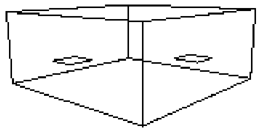
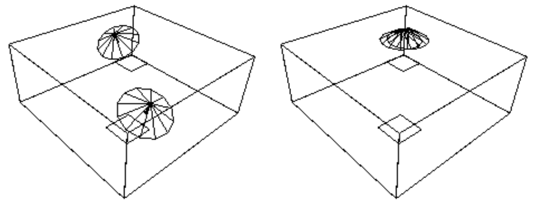


Fig. 6. Simple Room geometry

Fig. 7a shows one solution for the above problem, assuming the use of two DLs. If this lighting problem is not clearly specified, Fig. 7b shows another valid solution, although perhaps not the desired one.



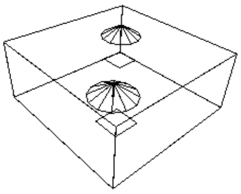
a) Solution b) Solution

Fig. 7. Room with 2 ILs and 2 DLs

The cones depicted in Fig. 7 represent the solution luminaires (DLs). These images show the importance of specifying "adequate" cost functions for luminaire optimisation.
To ensure that DLs are not concentrated over one IL, a weighting factor is introduced that depends on the difference of each IL contribution to both DLs. This weight will penalise the search for "undesirable" solutions (both DLs over one desktop). One appropriate cost function is:

```
# $(x_1,x_2,x_3),(x_6,x_7,x_8)$:positions $(x_4,x_5),(x_9,x_{10})$:directions
$W_{IL1/DL1}$=Importance($Scene_{IL1},x_1,x_2,x_3,x_4,x_5$)
$W_{IL1/DL2}$=Importance($Scene_{IL1},x_6,x_7,x_8,x_9,x_{10}$)
$W_{IL2/DL1}$=Importance($Scene_{IL2},x_1,x_2,x_3,x_4,x_5$)
$W_{IL2/DL2}$=Importance($Scene_{IL2},x_6,x_7,x_8,x_9,x_{10}$)
$K_1$=Abs($W_{IL1/DL1}-W_{IL1/DL2}$)
$K_2$=Abs($W_{IL2/DL1}-W_{IL2/DL2}$)
return -($K_1*(W_{IL1/DL1}+W_{IL1/DL2})+K_2*(W_{IL2/DL1}+W_{IL2/DL2})$)
```

Fig. 8 shows the solution found if this cost function is used.



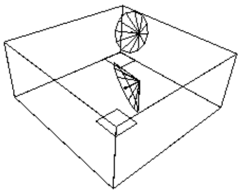
Solution using improved cost function

Fig. 8. Room with 2 ILs and 2 DLs

In Fig. 9, we present the solution if a directional constraint is added the initial problem: <u>the angle between the luminaire direction and the vector from the luminaire position to the centre of a desktop must be greater than 45°</u>.
This geometrical constraint can be used to avoid excessive direct lighting on the desktops. An appropriate extended cost function is:

```
1.  # $(x_1,x_2,x_3),(x_6,x_7,x_8)$:positions $(x_4,x_5),(x_9,x_{10})$:directions
2.  V=Vector(($x_1,x_2,x_3$),($IL_{1x},IL_{1y},IL_{1z}$))
3.  if Angle(V,Dir($x_4,x_5$))<45° return FAILURE
4.  V=Vector(($x_6,x_7,x_8$),($IL_{1x},IL_{1y},IL_{1z}$))
5.  if Angle(V,Dir($x_9,x_{10}$))<45° return FAILURE
6.  V=Vector(($x_1,x_2,x_3$),($IL_{2x},IL_{2y},IL_{2z}$))
7.  if Angle(V,Dir($x_4,x_5$))<45° return FAILURE
8.  V=Vector(($x_6,x_7,x_8$),($IL_{2x},IL_{2y},IL_{2z}$))
9.  if Angle(V,Dir($x_9,x_{10}$))<45° return FAILURE
10. $W_{IL1/DL1}$=Importance($Scene_{IL1},x_1,x_2,x_3,x_4,x_5$)
11. $W_{IL1/DL2}$=Importance($Scene_{IL1},x_6,x_7,x_8,x_9,x_{10}$)
12. $W_{IL2/DL1}$=Importance($Scene_{IL2},x_1,x_2,x_3,x_4,x_5$)
13. $W_{IL2/DL2}$=Importance($Scene_{IL2},x_6,x_7,x_8,x_9,x_{10}$)
14. $K_1$=Abs($W_{IL1/DL1}-W_{IL1/DL2}$)
15. $K_2$=Abs($W_{IL2/DL1}-W_{IL2/DL2}$)
16. return -($K_1*(W_{IL1/DL1}+W_{IL1/DL2})+K_2*(W_{IL2/DL1}+W_{IL2/DL2})$)
```

The geometrical constraint mentioned above (lines 2-9) can not be included in the definition of the luminaire parameters, so it must be checked in run time during the generation of new parameters for the optimisation.



Solution using extended cost function

Fig. 9. Room with 2 ILs and 2 DLs

Complex Example

Let us consider the hall in Fig. 10 (the image shows the geometry of this room).

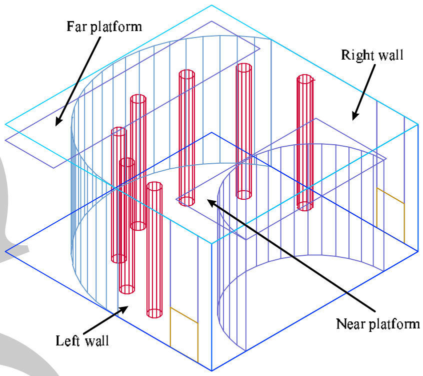


Fig. 10. Hall geometry

A designer wants to indirectly illuminate this hall with 4 projector type luminaires (4 DLs), <u>directed above</u> and <u>located in the vertical side walls and over the horizontal platforms</u>, producing an <u>homogeneous lighting on an imaginary plane at height 1.5m</u> and <u>on the bottom part of the curved walls</u>. The underlined expressions represent the lighting goals: the two initial goals are geometrical constraints and last two goals can be modelled by ILs. Because there are no relationships between the DLs parameters, each DL optimisation is performed independently (5df only, because the emission angle is predefined). A search for solutions can be performed using 3 ILs (the imaginary plane and the bottom of curved walls) and with the geometrical constraints included in the cost function:

```
# $(x_1,x_2,x_3)$:positions $(x_4,x_5)$:directions
if Angle(Dir($x_4,x_5$),(0,0,1))>45° return FAILURE
$W_{IL1}$=Importance($Scene_{IL1},x_1,x_2,x_3,x_4,x_5$)
$W_{IL2}$=Importance($Scene_{IL2},x_1,x_2,x_3,x_4,x_5$)
$W_{IL3}$=Importance($Scene_{IL3},x_1,x_2,x_3,x_4,x_5$)
return -($1.0*W_{IL1}+2.0*W_{IL2}+2.0*W_{IL3}$)
```

On a PC/*Pentium* 200 (64MB RAM, *Linux* 2.0.35), the design study for one DL ran for 2 hours, did 1000 iterations and generated 116 intermediate solutions; 1536000 radiance calculations were performed.

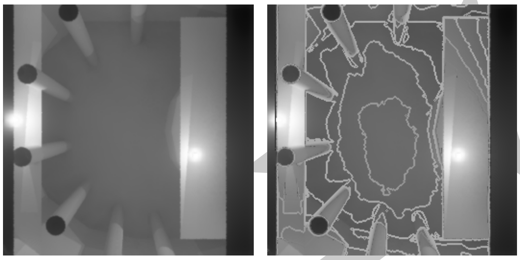
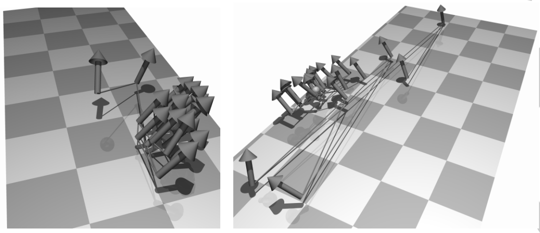


Fig. 11. Results for Hall solution (top view)

Fig. 11 shows a rendering (left) and a luminance plot (right) of the Hall, with 4 luminaires defined using the lighting solutions found. This figure shows that the luminance levels are in accordance with lighting goals and that the room has a reasonable homogeneous lighting distribution. To increase illumination, lighting emission in all the luminaires could be equally augmented without invalidating design results. Fig. 12 shows the convergence of intermediate solutions for 2 of the 4 DLs. The arrows represent intermediate solutions and the spheres mark the best solutions. The platforms DL solutions point to the centre of the ceiling and the walls DLs solutions point almost vertically – this agrees with our expected empirical solutions, i.e., use the ceiling to reflect light to the floor.



a) far platform b) near platform

Fig. 12. Convergence of solutions for Hall

CONCLUSIONS

The objective of our work is to study new approaches to the lighting design problem, trying to improve and shorten the design cycle (Fig. 2). This paper presents a new approach to the determination of luminaire characteristics in spaces with a known scene geometry and materials through the inclusion of lighting goals in the input data. We also introduced an optimisation loop to automatically search for lighting solutions. Our proposed methodology has lead to the development of a new algorithm, which is based on well known algorithms and techniques from the Global Illumination and the Optimisation fields, but linking them together in an innovative way.

The computation times are still very large for this methodology to be called interactive. The algorithms and techniques used must be tuned for this context and this will be our main concern in the near future. Radiance calculations are the main responsible for the large computation times, so any improvement in this field will bring benefits to our algorithm. Also better search strategies must be found other than Simulated Annealing. Our goal in this field is to reduce the number of iterations and obtain the final solution more quickly, which will also significantly reduce the number of radiance calculations.

REFERENCES

[Cohen93] Cohen, M.F.; Wallace, J.R.; *Radiosity and Realistic Image Synthesis*; Academic Press 1993.

[Costa98] Costa, A.; Sousa, A.; Ferreira, F.; *Design de Iluminação*; 8th Portuguese Workshop on Computer Graphics, Coimbra, Portugal 1998 (*in portuguese*).

[Costa99] Costa, A.; Sousa, A.; Ferreira, F.; *A New Approach to Lighting Design*; to be presented in the ICVC99 conference, Goa, India, February 1999.

[Inbge93] Ingber, L.; *Simulated Annealing: Practice versus Theory - Journal of Mathematical Computer Modelling*, V. 18, N. 11, pages 29-57; 1993.

[Ingbe98] Ingber, L.; *Adaptive Simulated Annealing*; *ftp://ftp.ingber.com/pub*; Lester Ingber Research; 1993-1998.

[Kawai93] Kawai, J.K.; Painter, J.S.; Cohen, M.F.; *Radioptimization - Goal Based Rendering - COMPUTER GRAPHICS Proceedings*, pages 147-154; SIGGRAPH 1993.

[Kirkp84] Kirkpatrick, S; *Journal of Statistical Physics*, V. 34, pages 975-986; 1984.

[Marks97] Marks, J; Andalman, B.; Beardlsey, P.A.; Freeman, W.; Gibson, S.; Hodgins, J.; Kang, T.; *Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation - COMPUTER GRAPHICS Proceedings*, pages 389-400; SIGGRAPH 1997.

[Pirlo96] Pirlot, M.; *General Local Search Methods - European Journal of Operational Research*, N. 92, pages 493-511; Elsevier Science B.V. 1996.

[Pouli97] Poulin, P.; Ratib, K.; Jacques, M.; *Sketching Shadows and Highlights to Position Lights - Proceedings of Computer Graphics International 97*, pages 56-63, 1997.

[Schoe93] Schoeneman, C.; Dorsey, J.; Smits, B.; Arvo, J.; Greenberg, D.; *Painting with Light - COMPUTER GRAPHICS Proceedings*, pages 143-146; SIGGRAPH 1993.

[Ward94] Ward, G.J.; *The RADIANCE Lighting Simulation and Rendering System - COMPUTER GRAPHICS Proceedings*, pages 459-472; SIGGRAPH 1994.