PEDECIBA Informática
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Tesis de Doctorado
en Informática

Efficient global illumination calculation for inverse lighting problems.

Eduardo Fernández

2014

Efficient global illumination calculation for inverse
Lighting problems
ISSN 0797-6410
Tesis de Doctorado en Informática
Reporte Técnico RT 14-02
PEDECIBA
Instituto de Computación – Facultad de Ingeniería
Universidad de la República.
Montevideo, Uruguay, marzo, 2014

PEDECIBA INFORMÁTICA
Instituto de Computación, Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

TESIS DE DOCTORADO EN INFORMÁTICA

Efficient Global Illumination Calculation For Inverse Lighting Problems

Eduardo Fernández
eduardof@fing.edu.uy

Febrero de 2014

Dr. Gonzalo Besuievsky	Director de Tesis
Dr. Franco Robledo	Director Académico
Dr. Hector Cancela	Presidente
Dr. Gustavo Patow	Revisor
Dr. Claude Puech	Revisor
Dr. Andrés Almansa	
Dr. Benoit Beckers	

Related Publications

This thesis is a compendium of the following papers:

- E. Fernández, P. Ezzatti, S. Nesmachnow and G. Besuievsky *Low-rank Radiosity using Sparse Matrices.* Proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP 2012) and International Conference on Information Visualization Theory and Applications (IVAPP 2012), Rome, Italy, 24-26 February, 2012, Pages 260-267.
- E. Fernández and G. Besuievsky *Inverse lighting design for interior buildings integrating natural and artificial sources*, Computers & Graphics, Volume 36, Issue 8, December 2012, Pages 1096-1108.
- V. Tourre, E. Fernández and G. Besuievsky *From Lighting Intention to Light Filters.* Proceedings of the International Conference on Cleantech for Smart Cities & Buildings from Nano to Urban Scale (CISBAT 2013), EPFL, Lausanne, Switzerland, 3-6 September, 2013, Pages 1181-1186.
- E. Fernández and G. Besuievsky *Statistical Inverse Lighting.* Proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP 2013) and International Conference on Information Visualization Theory and Applications (IVAPP 2013), Barcelona, Spain, 21-24 February, 2013, Pages 185-190.
- E. Fernández and G. Besuievsky *Efficient Inverse Lighting: A Statistical Approach*, Automation in Construction, Volume 37, January 2014, Pages 48-57.
- E. Fernández and G. Besuievsky *A Sample-Based Method for Computing the Radiosity Inverse Matrix*, Accepted by Computers & Graphics.
  doi: 10.1016/j.cag.2014.02.001

Resumen

La luz es un elemento clave en la manera en que percibimos y experimentamos nuestro entorno. Como tal, es un objeto más a modelar en el proceso de diseño, de forma similar a como ocurre con las formas y los materiales. Las intenciones de iluminación (LI) son los objetivos y restricciones que el diseñador pretende alcanzar en el proceso del diseño de iluminación: ¿qué superficies se deben iluminar con luz natural y cuáles con luz artificial?, ¿qué zonas deben estar en sombra?, ¿cuáles son las intensidades máximas y mínimas permitidas? Satisfacer las LI consiste en encontrar la ubicación, forma e intensidad adecuada de las fuentes luminosas. Este tipo de problemas se define como un problema inverso de iluminación (ILP) que se resuelve con técnicas de optimización.

En el contexto anterior, el objetivo de esta tesis consiste en proponer métodos eficientes para resolver ILP. Este objetivo es motivado por la brecha percibida entre los problemas habituales de diseño de iluminación y las herramientas computacionales existentes para su resolución. Las herramientas desarrolladas por la industria se especializan en evaluar configuraciones de iluminación previamente diseñadas, y las desarrolladas por la academia resuelven problemas relativamente sencillos a costos elevados.

Las propuestas cubren distintos aspectos del proceso de optimización, que van desde la formulación del problema a su resolución. Están desarrolladas para el caso en que las superficies poseen reflexión e iluminación difusas y se basan en el cálculo de una aproximación de rango bajo de la matriz de radiosidad. Algunos resultados obtenidos son: el cálculo acelerado de la radiosidad de la escena en una unidad de procesamiento gráfico (GPU); el uso de la heurística “variable neighborhood search” (VNS) para la resolución de ILP; el planteo de una estructura multinivel para tratar ILP de forma escalonada; y el uso de estas técnicas para optimizar la configuración de filtros de luz.

Otros resultados obtenidos se basan en la formulación de las LI en función de la media y desviación estándar de las radiosidades halladas. Se propone un método para generar LI que contengan esos parámetros estadísticos, y otro método para acelerar su evaluación. Con estos resultados se logran tiempos de respuesta interactivos. Por último, las técnicas anteriores adolecen de una etapa de pre-cómputo relativamente costosa, por tanto se propone acelerar el cálculo de la inversa de la matriz de radiosidad a partir de una muestra de factores de forma.

Los métodos aquí presentados fueron publicados en seis artículos, tres de ellos en congresos internacionales y tres en revistas arbitradas.

Palabras Clave: problema inverso de iluminación, intenciones de iluminación, radiosidad, matriz inversa de radiosidad, optimización, problema multi-objetivo, descomposición aproximada de matrices, matrices de rango bajo

Abstract

Light is a key element that influences the way we perceive and experience our environment. As such, light is an object to be modeled in the design process, as happens with the forms and materials. The lighting intentions (LI) are the objectives and constraints that designers want to achieve in the process of lighting design: which surfaces should be illuminated with natural and which with artificial light?, which surfaces should be in shadow?, which are the maximum and minimum intensities allowed? The fulfillment of the LI consists in finding the location, shape and intensity appropriate for the light sources. This problem is defined as an inverse lighting problem (ILP), solved by optimization techniques.

In the above context, the aim of this thesis is the proposal of efficient methods to solve ILP. This objective is motivated by the perceived gap between the usual problems of lighting design, and the computational tools developed for its resolution. The tools developed by the industry specialize in evaluating previously designed lighting configurations, and those developed by the academia solve relatively simple problems at a high computational cost.

The proposals cover several aspects of the optimization process, ranging from the formulation of the problem to its resolution. They are developed for the case in which the surfaces have Lambertian reflection and illumination, and are based on the calculation of a low rank approximation to the radiosity matrix. Some results are: rapid calculation of radiosity of the scene in a graphics processing unit (GPU), the use of heuristics "variable neighborhood search" (VNS) for solving ILP, the proposition of a multilevel structure to solve ILP in a stepwise approach, and the use of these techniques to optimize the configuration of light filters.

Other results are based on the formulation of LI that use the mean and standard deviation of the radiosity values found. A method is proposed for generating LI containing these parameters, and another method is developed to speed up their evaluations. With these results we achieve interactive response times. Finally, the above techniques suffer from a costly pre-computing stage and therefore, a method is proposed to accelerate the calculation of the radiosity inverse matrix based on a sample of the form factors.

The methods presented here were published in six articles, three of them at international conferences and three in peer reviewed journals.

Keywords: inverse lighting problem, lighting intention, radiosity, radiosity matrix inverse, optimization, multi objective problem, approximated matrix decomposition, low rank matrix

Contents

List of Figures

List of Tables

Agradecimientos

En primer lugar agradezco enormemente a Gonzalo Besuievsky por haberme guiado en la tesis. Su disponibilidad permanente facilitó la comunicación y achicó las distancias. Ha estado fuertemente involucrado en todos los detalles, no sólo en la discusión de las ideas iniciales y en la etapa experimental, sino que también ha realizado aportes sustanciales en la redacción de los artículos y en la compaginación final de este texto. Sólo en el último año mantuvimos unas cien horas de reuniones a pesar de la distancia, la diferencia horaria y su complicada agenda de actividades.

En segundo lugar quiero agradecer a Franco Robledo por la confianza y apoyo otorgados, que se evidenciaron en varias instancias clave para el doctorado.

Agradezco especialmente a las personas que de forma desinteresada brindaron ideas, consejos y opiniones sobre los temas de la tesis. En particular debo agradecer a Pablo Ezzatti y a Martin Pedemonte, así como a los revisores anónimos de los artículos publicados. Sus aportes mejoraron enormemente los resultados obtenidos en profundidad, variedad y sobre todo claridad.

Mi agradecimiento al Instituto de Computación por las facilidades brindadas durante estos años. También agradezco a la Asociación Universitaria Iberoamericana de Posgrado, a la Universidad de la República, a la Universitat de Girona, a la Comisión Sectorial de Investigación Científica, a la Facultad de Ingeniería y al Programa de Ciencias Básicas, por el apoyo financiero brindado en la inscripción a congresos, pasajes y viáticos durante las estadías en Girona y Roma.

Un agradecimiento aparte merece mi familia. Mariela ha aceptado sin chistar :-) que realice tres largas estadías en Girona. Mi madre me ha soportado unas cuantas tardes en su casa. Gabi me organizó dos cumpleaños en Les Borges Blanques. Por último, Juliana y Natalia son la fuerza y razón última de mis acciones. A ellas le dedico este trabajo.

Eduardo Fernández

Chapter 1

Introduction

Lighting intentions (LI) are the goals and constraints that designers would like to achieve in an illumination design process. Given an architectural interior space to illuminate, the lighting process requires several steps that go from revealing the general use of the space, to the specific light accentuation that must be obtained [1]. Concerning to the LI, they can be defined in different ways according to the reference observation (incoming light or reflected light) or to the global feature of the intention (local subset of surfaces or the overall scene). Once the LI are specified, the designer must provide the parameter setting that fulfills the desired intentions. The main features, such as the position of the lights, their shape and spectral power, must be provided. Therefore, the inverse lighting problem (ILP) can be defined as the problem of finding a parameter setting that meets the LI. Also, ILP could be characterized as an optimization problem, where the LI are transformed into a set of objectives, equations and inequalities to satisfy, and the parameters are the set of variables that could be modified in pursuit of the satisfaction of the LI.

In the above context, the objective of this thesis is the proposal of new ways to solve ILP more efficiently. The components of the ILP solver studied in the context of this thesis, are described in Figure 1.1. In this figure, the ILP process starts with the input of the geometry and the materials of a scene. These elements remain static during the process. In a second stage, a designer defines which are the set of lighting intentions to be satisfied, and the variables of the problem. If necessary, a precomputation process might be performed. After these steps are fulfilled, an optimization process is executed and the results are visualized.

The transition from one stage to the next is not straightforward, it may result in many trials and adjustments. Special consideration has the evaluation of the quality of the solution obtained. It may not be good enough from the point of view of the designer or, eventually, there may not find a solution satisfying all the LI. Therefore, these results can induce a change in the lighting intentions and a new execution of the optimization process (red cycle in Figure 1.1).

The above process is useful in working scenarios where the architectural elements are already designed or built, and only the openings (like windows or skylights) or the luminaires can be reconfigured.

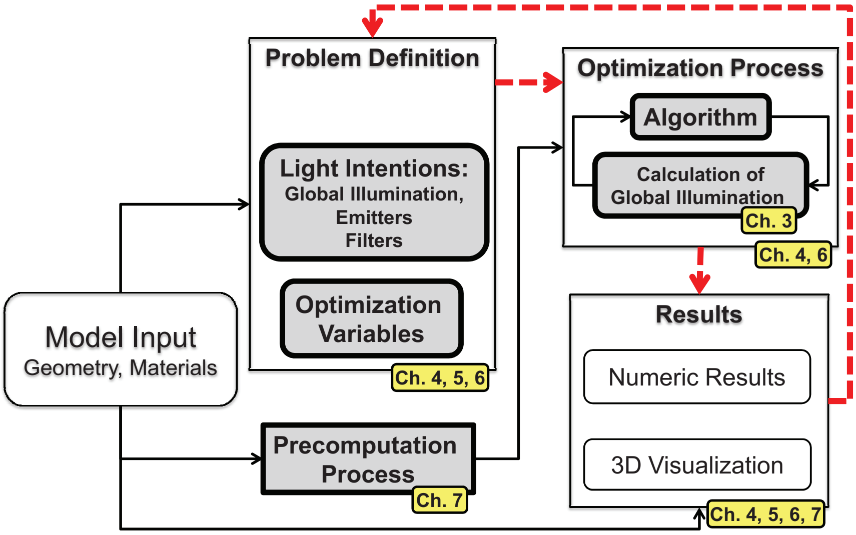


Figure 1.1: Scheme of the main components of the ILP solver used in the thesis, and the cycle of lighting design (red dashed arrows).

1.1 Motivation

One of the main motivations to center our efforts in the study of ILP, is the room for innovation perceived in the existing gap between the lighting intentions, as defined by a designer (Figures 1.2 and 1.3), and the current computational tools developed to help in the designing stage.

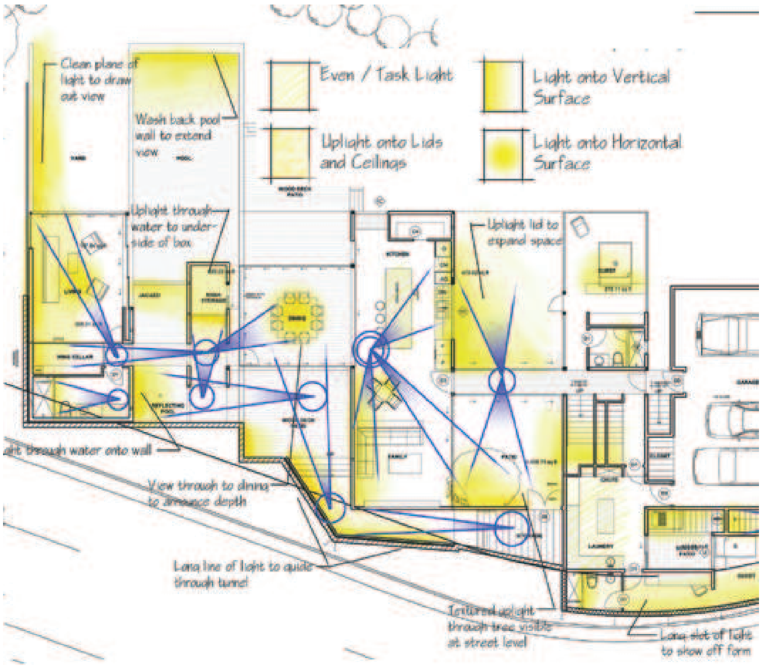

Clean plane of light to draw out view
Wash back pool wall to extend view
Even / Task Light
Light onto Vertical Surface
Uplight onto Lids and Ceilings
Light onto Horizontal Surface
Uplight through water to underside of box
Uplight lid to expand space
ght through water onto wall
View through to dining to increase depth
Long line of light to guide through tunnel
Textured uplight through tree visible at street level
Long slot of light to show off form

Figure 1.2: Example of a lighting design scheme in an architectural plan [1].

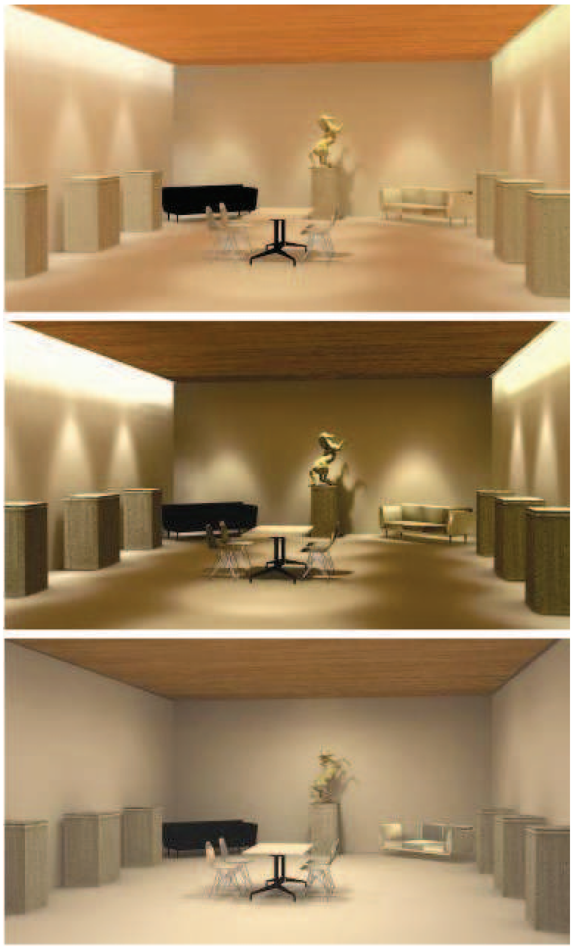


Figure 1.3: Examples of possible lighting configurations [1].

The designers usually follow a working-forward approach, going from the initial ideas to the final lighting configuration. They start studying the controllable aspects of light and defining the lighting goals and constraints (i.e., the LI). In Figure 1.2, there is a lighting scheme, where different "effects" of the light over the surfaces are described, and where the blue cones indicate where the visitor's attention should be directed. After this stage is completed, the designers develop a layout with the main ideas about the location and the properties of the luminaires, and finally they produce a set of final construction documents [1]. Hundreds of combinations and possibilities arise at each step, and the designer must decide which ones are better, taking into consideration the quality of the result, the difficulties in the construction, the initial costs, and the operating and maintenance costs.

On the other hand, the existing industrial packages (like for example Dialux [2], Relux [3] or AGi32 [4]) do not provide tools of inverse design for the generation of feasible solutions and the selection of the better one. Instead, the existing CAD systems bring helpful tools for the simulation of the scene (providing a database of possible furniture, luminaires, and surfaces). The calculation of the incoming and reflected radiance on all the surfaces is then computed through a physically-based renderer. This can be seen as a working-backward approach, where the designer must propose a solution and controls whether the desired initial intentions are fulfilled. The connection between the intentions and the lighting configuration is a hard work entirely done by designers, based on their experience or intuition, and usually it is found after many trials.

An additional example of the existing gap between the designers' needs and the existing tools is found in the literature devoted to ILP. The models found there are very simple. For example, a box-shaped room without furniture, with a small set of possible light positions. Also, the algorithms developed there are usually computationally demanding. These mentioned drawbacks conspire against the development of an interactive cycle of lighting design.

Another motivation is based on the results achieved in my master's thesis. In that previous work, it was developed a methodology to find a low-rank approximation of the radiosity inverse matrix (LRR), where the radiosity equation (see Equation 2.4) is transformed into a matrix-vector product (Equation 1.1), where $\mathbf{Y}$ and $\mathbf{V}$ are $n\times k$, with $n\gg k$.

$$B = \mathbf{M}E = (\mathbf{I} - \mathbf{RF})^{-1}E \qquad \approx \qquad (\mathbf{I} - \mathbf{YV}^T)E = \widetilde{\mathbf{M}}E = \widetilde{B} \tag{1.1}$$

Following this methodology it is possible to solve the radiosity equation in a fraction of a second, in scenes composed by more than $200,000$ patches and infinite bounces of light. The LRR method includes a rather slow precomputational step, but after that it performs a fast radiosity calculation. This result encourage us to think that the LRR method could be used as the global illumination engine in an ILP solver.

1.2 Context

Several assumptions were taken into account during the development of the thesis. These assumptions delimit and give context to the area of work covered by the thesis.

A first assumption is that in the lighting design process the light comes after the geometry is configured. When the illumination process is performed on a building with historic or patrimonial value, seems natural that few modifications to the structure are allowed, and even their colors and textures suffer of severe constraints. Further examples occur when the lighting process starts in an advanced stage of the architectural project. This assumption has a deep and beneficial consequence in the pipeline of the ILP solver. A static geometry enables the development of a powerful precomputational process, where many of the lighting interactions are calculated only once, then, the optimization process could be accelerated greatly. This also accelerates the iterative cycle of design (red dashed cycle in Figure 1.1), facilitating the interactivity of the design application, for the designers' benefit.

Another important contextual assumption is the nonexistence of a formal language to describe LI. Designers express their intentions in a descriptive manner (Figure 1.2), and the transformation of them into variables, goals and mathematical constraints is not a straightforward task. Also, the mathematical description of the LI must take advantage of the ILP solver and the global illumination engine used in the optimization process. Then, it seems that a fruitful approach consists in the use of an "inverse" strategy for the development of a ILP process. The strategy starts with the study of the non computationally demanding results, that take into account global illumination issues, and ends with the study of the possible connections between these results and the LI that a designer may want to satisfy. As a consequence of this strategy, some of the results presented here are centered in the development of efficient lighting calculations that speeds-up the evaluation of the LI, and the definition of simple mathematical tools that facilitate the translation of intuitive LI into more efficient ones.

The last assumption takes into account the importance that surfaces with diffuse reflection still have in different areas of engineering and design. In the areas of radiative heat transfer [5], and architecture [6], there are hundreds of practical examples and an extense area of research pointing in that direction. In that regard, the global illumination methodology used in this thesis is based in the radiosity equation.

1.3 Contributions and Thesis Structure

The contributions mentioned here are described extensively in Chapters 3 to 7. This thesis is a compendium of the papers described below. For this reason, some contents are repeated throughout the thesis, as it happens with the bibliography and the previous work.

The Speed-up of the LRR solver through a GPGPU strategy (Ch. 3): In this work a graphics processor hardware is used to accelerate the matrix·vector product operations, needed in the low-rank radiosity methodology (Equation 1.1). One of the low-rank matrices is stored as a sparse matrix, which contributes greatly to the speed-up of the process.

Chapter 3 is based on: *E. Fernández, P. Ezzatti, S. Nesmachnow and G. Besuievsky, Low-rank Radiosity using Sparse Matrices. Proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP 2012) and International Conference on Information Visualization Theory and Applications (IVAPP 2012), Rome, Italy, 24-26 February, 2012, Pages 260-267.*

The use of VNS and LRR to solve ILP (Chs. 4 and 5): A Variable Neighborhood Search (VNS) algorithm is implemented as a global optimization method to solve ILP. Several shortcuts are implemented in both the VNS algorithm and LRR method, to speed-up the optimization process, and to solve multi-objective optimization problems.

As an extension of this work, the methodology already developed is used to find optimal filters (Ch. 5). A filter acts on an area dividing it into elements, whose optics properties are calculated in order to produce a given lighting intention.

Chapter 4 is based on: *E. Fernández and G. Besuievsky, Inverse lighting design for interior buildings integrating natural and artificial sources, Computers & Graphics, Volume 36, Issue 8, December 2012, Pages 1096-1108.*

Chapter 5 is based on: *V. Tourre, E. Fernández and G. Besuievsky, From Lighting Intention to Light Filters. Proceedings of the International Conference on Cleantech for Smart Cities & Buildings from Nano to Urban Scale (CISBAT 2013), EPFL, Lausanne, Switzerland, 3-6 September, 2013, Pages 1181-1186.*

The use of statistics in the definition of LI (Ch. 6): The availability of $\widetilde{\mathbf{M}}$ triggers a new line of thought on the definition of mathematical goals and constraints of easy evaluation, based on the calculation of the mean and standard deviation of the radiosity values obtained from the scene. As a result, hundreds of constraints are eliminated, and the amount of iterations per second of the ILP solver rises to the thousands, for the tested scenes.

Chapter 6 is based on: *E. Fernández and G. Besuievsky, Efficient Inverse Lighting: A Statistical Approach, Automation in Construction, Volume 37, January 2014, Pages 48-57.*

Improvements in the precomputation stage (Ch. 7): Here it is explained a new method to find $\widetilde{\mathbf{M}}$, based on a randomized algorithm. With this methodology, the form-factor calculation is reduced to a subset of the **F** rows, accelerating significantly the precomputation stage, and facilitating the use of the previous results with greater 3D models.

Chapter 7 is based on: *E. Fernández and G. Besuievsky, A Sample-Based Method for Computing the Radiosity Inverse Matrix, Accepted by Computers & Graphics.*

In Chapter 8 the main conclusions and future work are discussed.

Bibliography

[1] S. Russell, The Architecture of Light: Architectural Lighting Design, Concepts and Techniques: a Textbook of Procedures and Practices for the Architect, Interior Designer and Lighting Designer, Conceptnine, 2008.

[2] DIALux, Dial gmbh, www.dial.de (Nov 2013).

[3] Relux, Relux informatik AG, www.relux.biz (Nov 2013).

[4] AGi32, Lighting analysts inc., www.agi32.com (Nov 2013).

[5] M. F. Modest, Radiative Heat Transfer, Second Edition, Chemical, Petrochemical & Process, Academic Press, 2003.

[6] H. Plummer, The Architecture of Natural Light, Monacelli Press, 2009.

Chapter 2

State of the Art

This chapter offers a brief overview of the most important aspects related to this thesis: the physical based global illumination, the low-rank radiosiy method, the inverse rendering problem, the concept of optimization and the radiosity inverse matrix.

2.1 Global Illumination

Light in the real world interacts with surrounding media and surfaces creating visual phenomena such as color bleeding, reflections, crepuscular rays, and caustics [1]. Although the underlying principles of light-matter interaction are well understood, their efficient computation is still a challenging problem. Many processes require simplified models, customized algorithms and data structures.

In general, light arriving at a point A from some other point B is considered as "direct" if it is generated in B. If the arriving light was reflected or transmitted before arriving at A, then it is considered as "indirect" illumination.

Those algorithms that explicitly gather illumination information only from the direct light source are "local illumination" models. On the other hand, algorithms that estimate indirect lighting information are called "global illumination" models.

According to Glassner [2], the most general equation that relates the global interaction between light and matter is the "full radiance equation", a half-page equation that includes phenomena such as photon transport in scattering media, polarization, phosphorescence and fluorescence. This equation could be briefly expressed in operator notation (Equation 2.1),

$$I = (\mathcal{M} + \mathcal{V})\left[\varepsilon + \mathcal{P}\mathcal{A}I + \mathcal{K}\mathcal{F}I\right] \tag{2.1}$$

where I stands for the intensity of light at wavelength λ, arriving at a point r from direction $\vec{w}$ at time t (see Figure 2.1), ε is the emitted light at wavelength λ at a point s in direction $\vec{w}$ at time t, $\mathcal{M}$ represents the attenuation of radiance from point s, $\mathcal{V}$ is the attenuation of radiance from point a between r and s, $\mathcal{P}$ is the phosphorescence operator, $\mathcal{F}$ is the fluorescence operator, $\mathcal{A}$ is the absortion operator, and $\mathcal{K}$ is the bidirectional surface-scattering distribution function (BSSDF).

More simplified and widely used versions of the above equation are the "rendering equation" [3] (where the light is transported with no surrounding media), and the "volumetric rendering equation" [2] (which includes the presence of participating media).

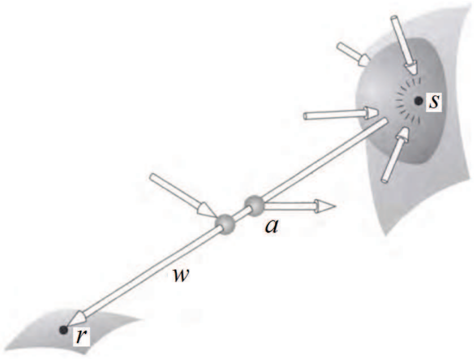


Figure 2.1: Light arriving at r, from s, from direction $\vec{w}$, through the point a [2].

According to Kajiya [3], the rendering equation is defined in Equation 2.2,

$$I(x,x') = g(x,x')\left[\varepsilon(x,x') + \int_S \rho(x,x',x'')I(x',x'')dx''\right] \tag{2.2}$$

where $I(x,x')$ is related to the intensity of light (W/m^2) passing from x' to x (Figure 2.2), $g(x,x')$ is 0 when x and x' are occluded from each other and $1/r^2$ in the other case (r is the distance between them), $\varepsilon(x,x')$ is related to the intensity of light that is emitted from x' to x, and $\rho(x,x',x'')$ is related to the intensity of light scattered from x'' to x by a patch of surface at x' (bidirectional reflectance distribution function, or BRDF). The rendering equation is time invariant (no diffraction is calculated), the media is of homogeneous refractive index and does not itself participate in the scattering light. Finally, the equation is not influenced by light wavelength or polarization properties.

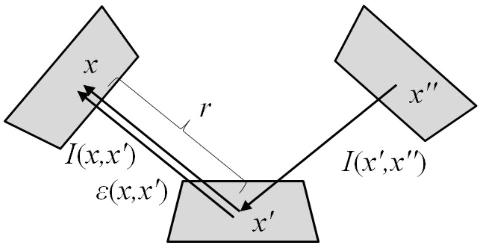


Figure 2.2: Some elements of the rendering equation.

These equations can be solved through different approaches. Ritschel et al. [1] established that the classical approaches to compute (interactive) global illumination are: Finite elements (e.g., radiosity) [4], Monte Carlo ray tracing [3], photon mapping [5], instant radiosity [6], many-light-based global illumination [7, 8], point-based global illumination [9, 10], discrete ordinate methods [11], and precomputed radiance transfer [12]. Through the previous techniques it is intended to implement a wide range of phenomena: direct (local) and indirect (global) illumination, ambient occlusion, natural illumination, single and multiple bounces, caustics, diffuse and glossy bounces, and scattering.

Some of the implementations of the above approaches claim to compute the global illumination of a scene at real time rates, but all of them have difficulties with many of the following open problems: the scalability, the amount of bounces, multiple glossy bounces, multiple scattering, and complex light sources.

2.1.1 The Radiosity Problem

The radiosity method [4] is a pioneering technique for global illumination on scenes with Lambertian surfaces, which has been applied in many areas of design and computer animation [13]. The term "radiosity" stands for the sum of the emitted and reflected light power leaving a surface (W/m^2). Lambertian surfaces are ideally diffusely reflecting surfaces, which holds that the apparent brightness of them to an observer are the same regardless of the observer's angle of view (Figure 2.3).

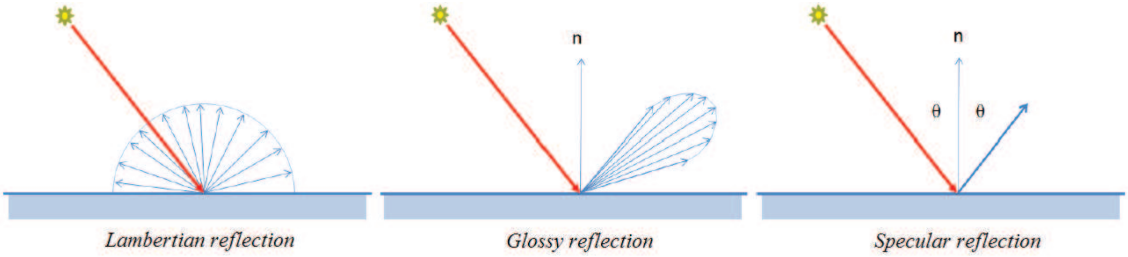


Figure 2.3: Typical kinds of reflections.

Lambertian surfaces simplify the BRDF term in the rendering equation (Equation 2.2), which is transformed into the "radiosity equation" (Equation 2.3). In this equation, $B(x)$ is the radiosity at point x, $E(x)$ is the light emission in x, $\rho(x)$ is the Lambertian diffuse reflectivity of x, and $G(x, x')$ is a geometric factor that determines how much the radiosity in x' contributes to the radiosity of x [14].

$$B(x) = E(x) + \rho(x) \int_S B(x')G(x, x')dA' \tag{2.3}$$

The radiosity equation could be discretized through the use of a finite element methodology (Figure 2.4). The scene is discretized into a set of patches, and all the terms in Equation 2.3 are transformed into a finite set of discrete elements. Therefore, $B(x)$ is transformed into B_i, $E(x)$ into E_i, $\rho(x)$ into R_i, and $G(x, y)$ into $\mathbf{F}(i, j)$. Then, Equation 2.3 is transformed into a set of linear equations:

$$B_i = E_i + R_i \sum_{j=1\ldots n} B_j \mathbf{F}(i, j) \quad , \quad \forall i \in \{1 \ldots n\}$$

This set of linear equations is expressed in a succinct manner in Equation 2.4.

$$(\mathbf{I} - \mathbf{RF})B = E \tag{2.4}$$

In this equation, $\mathbf{I}$ is the identity matrix, $\mathbf{R}$ is a diagonal matrix containing the R_i values, $\mathbf{F}$ is the form factor matrix, B is a vector containing all the radiosities to be found, and E is another vector containing the emission of the scene. The form factor $\mathbf{F}(i, j)$ is defined as a number between 0 and 1, that indicates the fraction of the light

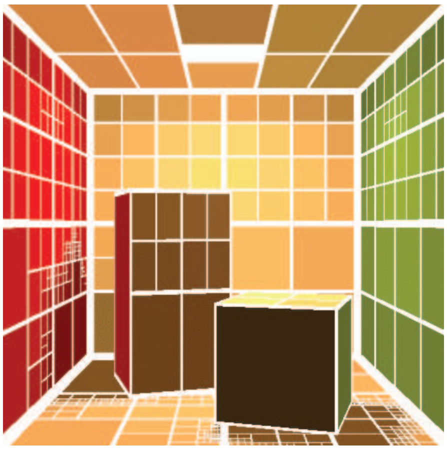


Figure 2.4: Division of a radiosity scene into a finite set of elements [15].

power emitted by patch i that arrives to patch j. It can be computed as:

$$\mathbf{F}(i,j) = \frac{1}{A_i}\int_{A_i}\int_{A_j} G(x_{A_i}, x_{A_j})dx_{A_j}dx_{A_i}$$

When the scene is simple (no labyrinthine design), the form factor matrix is a non-sparse matrix with dimension $n{\times}n$, where n is the amount of patches in the scene.

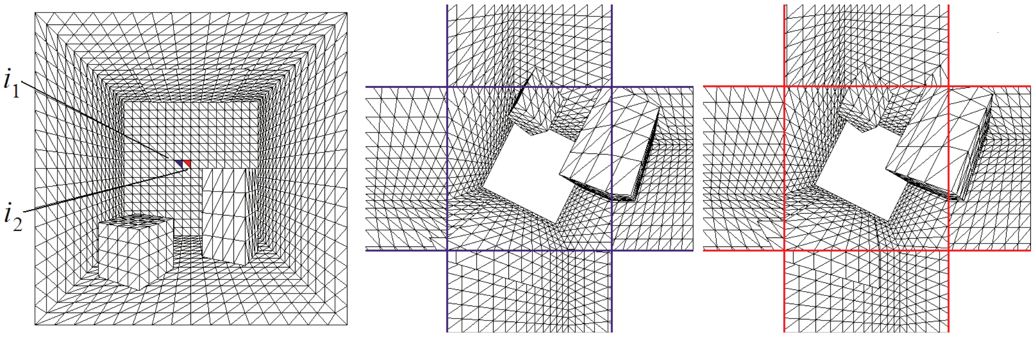
In the classic approach, the explicit computation of the radiosity matrix $\mathbf{I} - \mathbf{RF}$ is avoided due to the huge computational cost. One of the classic formulations proposed is the progressive refinement solution [16], where a selection of form factors is computed to shoot energy from significant surfaces. Another important contribution is the hierarchical radiosity method [17], which reduces the $O(n^2)$ form factors calculation to $O(n)$, for a scene composed of n patches. Hierarchical radiosity with clustering [18, 19] overcomes the intial $O(n^2)$ cost of linking n patches. For computing global illumination these methodologies were succesful and resulted into interactive techniques that are currently used in architectural lighting design applications and in visualization.

The arrival of the GPU architecture sped-up previous approaches and led to the birth of new techniques. Dong et al. [20] precomputed links for a pre-defined geometry, similar to Immel et al. [21], but now they can be deformed at runtime, when the link structure remains valid. Dachsbacher et al. [22] also used links to exchange antiradiance, avoiding the visibility computation. Meyer et al. [23] introduced the updating of the link structure at runtime. All these techniques are only well suited to low-complexity scenes with moderate transformations.

Despite the above problems and the solutions reached, the inverse of the radiosity matrix, $\mathbf{M} = (\mathbf{I} - \mathbf{RF})^{-1}$, provides important information about the global radiosity of the scene. Each element $\mathbf{M}(i,j)$ gives the contribution of the emission in j to the final radiosity value in i. Then, a single scalar product between a row $\mathbf{M}(i,:)$ and the emission vector E provides the final radiosity value in i. For scenes where the geometry is fixed and only the light output varies, $\mathbf{M}$ is a constant matrix. Therefore, regardless of the initial computational cost, an estimate of the inverse matrix brings the possibility of hundreds of radiosity calculations per second, each even with infinite light bounces.

2.2 Low Rank Radiosity

It is very likely that the **RF** matrix included in Equation 2.4 has a low numeric rank. This is because **R** is a diagonal matrix and each row $\mathbf{F}(i,:)$ is computed based on the scene view from the element i, where usually most pairs of close patches have a very similar view of the scene (see Figure 2.5). Then, **RF** has many similar rows, which results in a reduction of the numerical rank.



(a) Two close patches i_1 and i_2. (b) Hemicube view from i_1. (c) Hemicube view from i_2.

Figure 2.5: Close patches have similar views of the scene.

The rank reduction of **RF** matrices allows approximating them by the product of two matrices $\mathbf{U}_k\mathbf{V}_k^T$, both with dimension $n{\times}k$ $(n{\gg}k)$, without losing relevant information about the scene. References about low-rank properties of radiosity and radiance matrices can be found in Baranoski et al. [24], Ashdown [25], Hašan et al. [8], and Fernández [26]. Figure 2.6 shows the singular values and the eigenvalues of the form factor matrix **F**, for a Cornell box with $10,240$ patches. We can observe that only 1,500 singular values are greater than 10^{-2}. Eckart and Young demonstrated that the minimum distance to **F** by a rank k matrix is $\|\mathbf{F}-\mathbf{F}_k\|_2 = \sigma_{k+1}$ [27], where $\mathbf{F}_k$ is a matrix containing the k greatest singular values and vectors of **F**, and σ_{k+1} is the $(k+1)^{th}$ largest singular value of **F**.

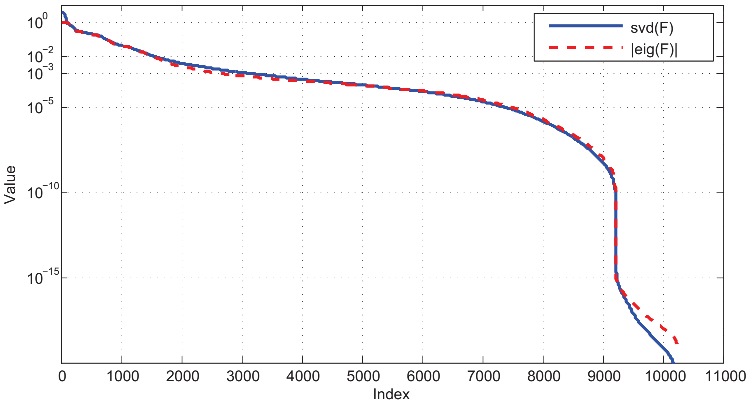


Figure 2.6: Singular values and eigenvalues of **F**.

$\mathbf{U}_k$ and $\mathbf{V}_k$ can be computed using factorization techniques, such as singular value decomposition (SVD or PCA) [28], CUR factorization [29], and discrete transformations based on Fourier and wavelets [30]. In the truncated SVD (TSVD) methodology, only the k largest singular values are calculated and the rest are discarded. This methodology produces optimal results, but has $O(n^2k)$ complexity. The other methods mentioned not based on SVD, have lower complexity orders, but also produce larger errors in the **RF** approximation. To overcome the weakness of the traditional factorization techniques, the low-rank radiosity (LRR) method [26] uses the concept of spatial coherence [31]. This method generalizes the two-level hierarchy methodology [32], where two meshes with different granularity levels are generated for the scene (a coarse mesh with k patches and a fine mesh with n elements). The LRR method allows to build the matrices $\mathbf{U}_k$ and $\mathbf{V}_k$ after $O(n^2)$ floating point operations. All the methods mentioned above requires the calculation of the **RF** matrix.

The product $\mathbf{U}_k\mathbf{V}_k^T$ generates a matrix with dimension $n{\times}n$ and rank k. The memory space needed to store $\mathbf{U}_k$ and $\mathbf{V}_k^T$ is $O(nk)$, significantly less than the $O(n^2)$ required to store the matrix **RF**, especially when $n{\gg}k$. This memory saving allows reducing the space needed to store the information for large scenes.

In the LRR equation (Equation 1.1), **RF** is replaced by the low-rank approximation $\mathbf{U}_k\mathbf{V}_k^T$, and B is only found approximately ($\widetilde{B}$). Then, the matrix $\left(\mathbf{I} - \mathbf{U}_k\mathbf{V}_k^T\right)$ is invertible using the Sherman-Morrison-Woodbury formula [28]:

$$\widetilde{B} = E - \underbrace{\left(-\mathbf{U}_k\left(\mathbf{I}_k - \mathbf{V}_k^T\mathbf{U}_k\right)^{-1}\right)}_{\mathbf{Y}_k}\mathbf{V}_k^T E = E - \mathbf{Y}_k\mathbf{V}_k^T E \tag{2.5}$$

When finding $\widetilde{B}$ through Equation 2.5, $O(nk^2)$ operations are required and $O(nk)$ memory is needed.

In scenes with static geometry and dynamic lighting (i.e., only the independent term E varies in Equation 2.5), part of the computation can be executed before the real-time stage, because the $n{\times}k$ matrices $\mathbf{Y}_k$, $\mathbf{U}_k$, and $\mathbf{V}_k$ can be calculated only once. Thus, the real-time stage has now complexity $O(nk)$. This is an auspicious result in order to develop a methodology for real-time radiosity and inverse lighting problems.

Equation 2.5 can also be formulated as $\widetilde{B} = \widetilde{\mathbf{M}}E$, so $\widetilde{\mathbf{M}} = \mathbf{I} - \mathbf{Y}_k\mathbf{V}_k^T$ is considered as the global operator that manages the infinite bounces of light in a single operation. This global operator is computed without the use of Neumann series, as it happens in other works [33, 34], taking advantage of the fact that **RF** is a low-rank matrix.

2.3 Inverse Rendering Problems

According to Tarantola [35], physical theories allow us to make predictions: given a complete description of a physical system, we can predict the outcome of some measurements. The problem of predicting the result of measurements is called the modelization problem, the simulation problem, or the forward problem. In contrast, inverse problems generally infer the properties (or model parameters) of a physical system from observed or desired data. Inverse problems are usually numerically complex, have many or even infinite solutions, and are of interest in a wide range of fields, including lighting engineering and lighting design.

Marschner [36], studied the rendering equation (Equation 2.2), and considered that there are three main kinds of inverse rendering problems: the inverse lighting problems (ILP), the inverse reflectometry problems, and the inverse geometry problems. In the inverse lighting problems (related to $\varepsilon(x, x')$ in Equation 2.2), the shape and reflection properties of a scene are known and the reflected light is at least partially known. The problem consists in finding the emission (or the incident light due to direct illumination) in the scene. In the inverse reflectometry problems (related to $\rho(x, x', x'')$) the variables of the problem are related to the reflection properties of the surfaces. Finally, in inverse geometry problems (related to $g(x, x')$) the unknowns are linked to the geometry of the object. Two interesting surveys on the state of the art of inverse rendering problems can be found in the work of Patow and Pueyo [37, 38].

2.4 Inverse Lighting Problems

According to Patow and Pueyo [37], the inverse lighting problems can be further classified into problems of inverse emittance and inverse light positioning. In the former problems the unknowns are the emittances of a given subset of surfaces of the scene. In the latter problems the unknowns are the locations of the light sources in order to achieve a desired illumination. Another classification can be fulfilled related to the treatment done to the rendering equation. In this alternate approach, the main classification is between the classic finite element radiosity setting, the inverse Monte Carlo framework, or even by posing them as direct illumination problems. Given the above classification, it is possible to say that this thesis is centered in the resolution of inverse light positioning and emittance problems, through the setting of a finite element radiosity configuration.

Figure 2.7 shows an example of inverse lighting problem (ILP) as considered in this thesis, where the emitter position and shape (parameters) are inferred, given expressed lighting intentions (desired data).

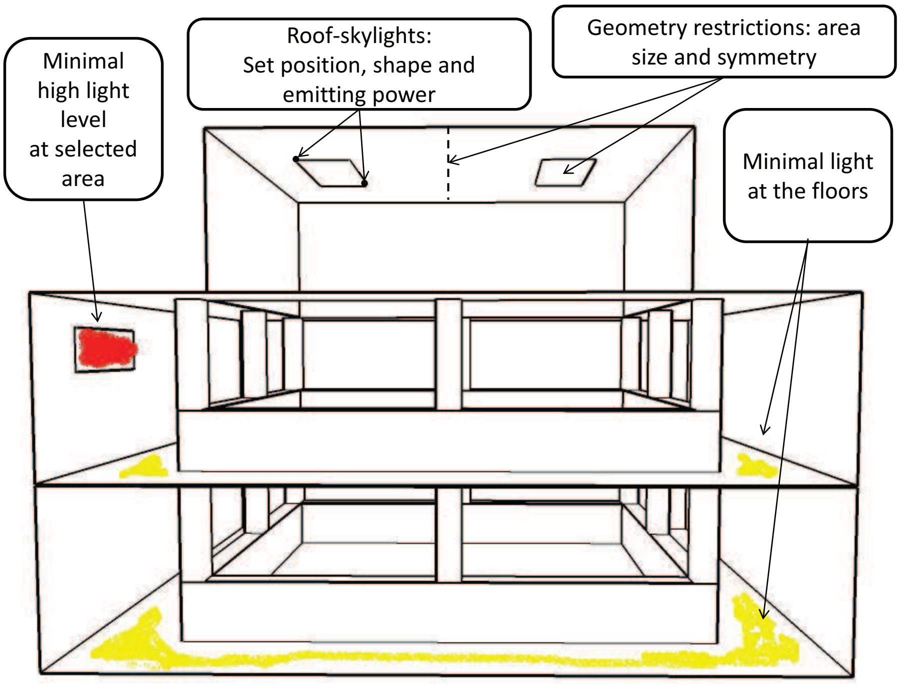

Minimal high light level at selected area
Roof-skylights: Set position, shape and emitting power
Geometry restrictions: area size and symmetry
Minimal light at the floors

Figure 2.7: An example sketch of a design configuration.

For inverse lighting problems with Lambertian surfaces, the problem of solving the emission of a surface given the reflection on the surfaces can be stated from the Equation 2.6, which can be derived directly from Equation 2.4.

$$\mathbf{RF}E = (\mathbf{I} - \mathbf{RF})C \tag{2.6}$$

In this equation, the emission E is unknown, C is the known reflected radiosity of all the surfaces (i.e., $C = B - E$). Therefore, $(\mathbf{I} - \mathbf{RF})C$ is a defined vector.

The above problem is ill-posed, due to the low numerical rank of matrix $\mathbf{RF}$. This means that the solucion of E is extremely sensitive to the input data C. Ramamoorthi and Hanrahan [39] found similar results when they tried to estimate the light emission from observations of the reflection on a Lambertian surface. Also, Harutunian et al. [40] observed similar results in the context of radiative heat transfer. Another problem is that the solution found could have negative values (Contensin [41]). Therefore, the resolution of the inverse problem through a single inversion of the radiosity matrix (using truncated singular value decomposition (TSVD) or modified TSVD (MTSVD) [42]), by itself offers limited results. Moreover, when the surfaces are glossy or specular, it is more likely to find a precise configuration of emitters, because the path followed by the light is better defined.

Another element to take into consideration in ILP is that usually the exact values of the reflected radiance C are unknown. In lighting design, the designer establishes coarsely some constraints of the incoming or reflected light, and hope to find an emitter configuration that fulfills such constraints. Therefore, Equation 2.6 is not well defined, hindering its resolution. An alternative to that approach is to solve the radiosity equation many times, trying different configuration of the lights E, looking for the configuration that better fulfills the constraints. As a consequence of this approach, many of the papers published of inverse lighting are focused on the resolution of ILP as an optimization problem (see Section 2.5).

2.4.1 Previous Work in Inverse Lighting Problems

One of the first attempts into the ILP domain was presented by Schoeneman et al. [43]. These authors introduced the idea of providing an iterative numerical solution to achieve results from a “spray-painting” user-interface description. The interactivity was achieved only for direct illumination. Several works searching for similar goals but driven by different motivations and assumptions were later proposed [37]. Contensin [41] formulated an inverse radiosity method based on a pseudo-inverse analysis of the radiosity matrix. Costa et al. [44] proposed an optimization engine to deal with complex light specifications. Kawai et al. [45] performed the optimization over the intensities and directions of a set of lights as well as surface reflectiveness to best convey the subjective impression of certain scene qualities (e.g., pleasantness or privacy), as expressed by users. Their so-called radioptimization system is a framework that determines optimal setting parameters based on unconstrained optimization techniques used in conjunction with penalty and barrier methods to manage constraints [46], and a hierarchical radiosity solver [17]. Castro et al. [47] used heuristic search algorithms combined with linear programming to optimize light positioning with an energy-saving goal. Similar problems were solved with closely related techniques. Pellacini et al. [48] presented an interactive system for computer cinematography that allows users to paint the desired lighting effects in the scene, and a solver provides the corresponding parameters to achieve these effects using a non-linear optimization method. Gibson et al. [49] solved the inverse lighting problem through iteration of virtual light sources in the context of photometric reconstruction data. Shesh and Chen [50] developed a sketch guided illumination of models, to specify target lighting through the sketch of bright and dark regions on the model as if coloring it with crayons.

Considerably less work has been devoted to optimizing daylight sources, such as openings and skylights. The problem of finding the modeling shapes for lighting goals is more complex for daylight sources than for artificial lights, because in the former case, we are dealing with a dynamic light source. An inverse method for designing opening buildings is presented by Tourre et al. [51] based on the evaluation of potential elements with relation to an interior lighting intention. In Besuievsky and Tourre [52], the method is extended for accurate sky computation with exterior occlusion. These approaches consider an anisotropic distribution of the light over the potential opening. More recently, Liu and Granier [53] developed a system of tracking of outdoor lighting variations for augmented reality with moving cameras. These variations are reflected in the lighting and shadows of the virtual objects to achieve more visual realism.

2.4.2 Coherence in Inverse Lighting Problems

Regardless of the strategy being used to tackle inverse lighting problems, the global illumination function must be evaluated many times prior to finding a converged solution. For this purpose an efficient method should be used to compute the solution for each setting of parameter values. However, research on this point has not been emphasized. The use of coherence is crucial to improve the global illumination computation in inverse lighting. Castro et al. [47] improved this known expensive computation by re-using random walk paths. They build and store, in a preprocessing step, an irradiance matrix that allows computing, for each patch of the scene the power contribution of several fixed light points. The main restrictions of this approach are that the authorized light positions must be predefined and that all of the light sources are point lights. In chapters 3 to 6 of our work, we use a low-rank radiosity method (LRR) [26] that allows computers to solve the radiosity equation in real-time with infinite bounces for scenes with dynamic lighting and a relatively small computer storage. With this method no restrictions are required for the light sources: the sources could be anywhere in the scene and area light sources can be allowed, as well. Other approaches, such as the one presented by Kontkanen et al. [33], which uses wavelets to store a pre-computed matrix, could also be used for this purpose. This technique supports all types of light sources, and also glossy BRDFs. However, dynamic objects cannot be easily supported. Furthermore, precomputation still takes tens of minutes for very simple scenes. Arquès and Michelin [54] found that all patches in the neighborhood of a given reference patch, have form factors that are weighted sums of the reference patch form factor and a series of associated integrals of generalized orthogonal polynomials, which reduces the calculation time and allows to control the calculation errors. Vilaplana and Pueyo [55] present a brief characterization of the uses of coherence for form factor computation, and propose new uses of coherence for visibility computation in form factor determination.

2.5 Optimization

Optimization processes play an important role in the resolution of ILP, therefore, a brief introduction is given here. A deeper introduction to this area of knowledge could be found in [46], [56], [57], and [58].

According to Luenberger and Ye [46], the concept of optimization is now well rooted as a principle underlying the analysis of many complex decision or allocation problems. One approaches a complex decision problem, involving the selection of values for a number of interrelated variables, by focussing the attention on a single objective designed to quantify performance and measure the quality of the decision. This objective is optimized subject to the constraints that may limit the selection of decision variable values. Other approaches involve multi-objective optimization problems, where more than an objective function is optimized simultaneously, and where the solution found is not unique (probably an infinite number of Pareto optimal solutions [56, 57]).

The decision problems consist of the following steps [58]: formulate the problem, model the problem, optimize the problem, and implement the solution (see Figure 2.8). In inverse lighting problems, the formulation of the problem is exemplified in Figure 1.2, where a designer outlines the internal and external factors, and the objective(s) of the problem. In the modeling of the problem, an abstract mathematical model is built for the problem. Usually this stage involves approximations and sometimes some elements are eliminated due to its mathematical complexity. Once the problem is modeled, the optimization step generates a "good" solution for the mathematical problem (but perhaps not for the original problem). The solution may be optimal or suboptimal. In the final stage, the solution is tested in practice by the decision maker and is implemented if it is "acceptable". Like life cycles in software engineering, the life cycle of optimization models and algorithms may be linear, spiral, or cascade.

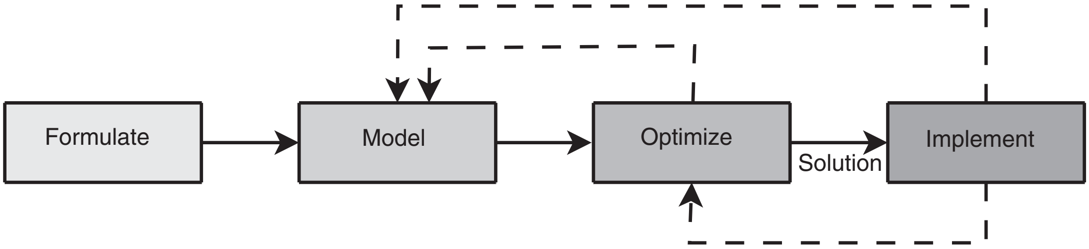


Figure 2.8: The classical process in decision making [58].

This thesis is centered in the model and optimization stages. In Figure 1.1, the model input and the design configuration step correspond to "model the problem" stage, and the precomputation process, the optimization process and the results steps correspond to "optimize the problem" stage, in the decision making process.

Different families of optimization models are used in practice to formulate and solve decision-making problems. The classical optimization models can be divided into the following classes: mathematical programming models [59] (which are divided into continuous, integer, mixed, and also can be divided into linear and nonlinear), combinatorial optimization models [60], constraint satisfaction models [61], and nonanalytic models [62]. The most successful models are based on mathematical and constraint programming.

In mathematical programming, the problem consists in maximizing or minimizing a real function, within the available input values. The combinatorial optimization problems are characterized by discrete decision variables and a finite search space. In constraint programming, the constraints define the properties of a solution, which is found through the interleaving of propagation and search processes. Finally, the nonanalytic models afford problems that cannot be solved in an exhaustive manner, because the objective function is unknown.

Considering the complexity of the problem, there are exact methods and approximate methods. Exact methods obtain optimal solutions and guarantee optimality. But for NP-complete problems the approximate methods are used, because they generate high quality solutions in a reasonable time for practical use. However, there is no guarantee of a global optimal solution. The exact methods include branch and bound, constraint programming and dynamic programming methods, among others. The approximate methods can be divided into heuristic algorithms and approximation algorithms. Heuristic algorithms can be further divided into metaheuristics (single and population based), and problem specific methods.

Unlike exact methods, metaheuristics allow to tackle large-size problem instances by delivering satisfactory solutions in a reasonable time. There is no guarantee of global optimal solutions or even bounded solutions. Application of metaheuristics falls into a large number of areas, such as: engineering design, topology optimization, machine learning, system modeling, planning in routing problems, and logistics, among others.

2.5.1 Optimization in Inverse Lighting Problems

The choice of the optimization methodology to solve ILP seems to be quite arbitrary, and in the literature several different approaches to this point can be found. It is clear that the best optimization method to use is an open research area. Least squares and constrained least squares methods [28], simulated annealing [63], modified truncated singular value decomposition (MTSVD) [42], are the most common algorithms used for ILP, as can be seen in the work of Patow and Pueyo [37].

Hill climbing [64] and beam search [65] heuristics, have been also applied in inverse lighting problem. Castro et al. [66, 47] explored a wide range of these algorithms to solve optimal economical light positioning. Kawai et al. [45] uses the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [46], a hill-climbing optimization technique, and the constraints are managed with penalty and barrier methods, as mentioned above. In Cassol et al. [67] and in Schneider et al. [68], the scenes are simplified to rectangular spaces, and the inverse problem is solved through the generalized extremal optimization (GEO) approach, a global search metaheuristic [69]. Tena [70], Ferentinos and Albright [71], and Delepoulle et al. [72] proposed population based methods (genetic algorithms). The latter two calculate the light configuration to improve the growth of plants in closed environments.

In our work, we used the variable neighborhood search (VNS) metaheuristic for optimization problems [73] (see Section 4.4.3). We adapt this technique to our illumination problem and show that good optimization results can be achieved.

2.6 The Radiosity Inverse Matrix

The calculation of the radiosity inverse matrix $\mathbf{M} = (\mathbf{I} - \mathbf{RF})^{-1}$ is the main strategy developed in this thesis to speed up the calculation of thousands of radiosity configurations in an ILP. The matrix $\mathbf{M}$ can be transformed, using Neumann series, into a sum of powers of $\mathbf{RF}$. Therefore, the radiosity of the scene B can be calculated through the following sum of terms:

$$B = \mathbf{M}E = (\mathbf{I} - \mathbf{RF})^{-1}E = (\underbrace{\mathbf{I} + \mathbf{RF}}_{\text{Direct Illumination}} + \underbrace{(\mathbf{RF})^2 + (\mathbf{RF})^3 + \ldots}_{\text{Indirect Illumination}})E \tag{2.7}$$

In this series the first term, $\mathbf{I}E$, corresponds to the emission, the second one, $\mathbf{RF}E$, corresponds to the first reflection of the emitted light, the third one, $(\mathbf{RF})^2E$, corresponds to the second reflection, and so on.

The classic way to calculate the radiosity consists in an iterative procedure, expresed in Equation 2.8.

$$B^{(i+1)} = \mathbf{RF}B^{(i)} + E; \quad with \quad B^{(0)} = E \tag{2.8}$$

In this procedure, $B^{(i)}$ represent the result of the truncated series until the i^{th} power. Jacobi and Gauss-Seidel iteration are commonly used in finite elements publications, in hierarchical and non-hierarchical approaches [14].

According to Cohen and Wallace [14], the eigenvalues of $\mathbf{RF}$ are less than one. Also, the spatial coherence of the scene is reflected in the similarity between the rows of $\mathbf{F}$, as is explained above. Therefore, the matrix $\mathbf{RF}$ has a low numerical rank, that is, the number of significant singular values and eigenvalues is small compared to the size of the matrix. The greatest eigenvalues of a Cornell box scene can be appreciated in the dotted line of Figure 2.6.

In the Neumann series, the $\mathbf{RF}$ terms can be substituted by the eigendecomposition $\mathbf{Q\Lambda Q}^{-1}$, where $\mathbf{\Lambda}$ and $\mathbf{Q}$ contain the eigenvalues and eigenvectors of $\mathbf{RF}$, respectively. The substitution results in the Equation 2.9.

$$B = (\mathbf{I} - \mathbf{RF})^{-1}E = \mathbf{Q}(\mathbf{I} + \mathbf{\Lambda} + \mathbf{\Lambda}^2 + \ldots)\mathbf{Q}^{-1}E = \mathbf{Q}(\underbrace{\mathbf{I} + \mathbf{\Lambda}}_{\text{Direct}} + \underbrace{\sum_{i=2}^{\infty} \mathbf{\Lambda}^i}_{\text{Indirect}})\mathbf{Q}^{-1}E \tag{2.9}$$

Figure 2.9 shows the eigenvalues of the first terms in the Newmann series. It can be appreciated that the sum of diagonal matrices representing the indirect illumination goes down faster than the eigenvalues related to direct lighting (see the black dashed line), therefore the indirect illumination has a lower numerical rank. Also, the nonzero elements of the term $\mathbf{\Lambda}^i$ decreases when the value of i increases, meaning that their influence in the final result is also lower than the former terms.

2.6.1 Alternatives to LRR decomposition

As mentioned above, much of the information pertaining to $\mathbf{RF}$ is redundant and can be derived from a relatively small sample of matrix elements. This fact leads us to suppose

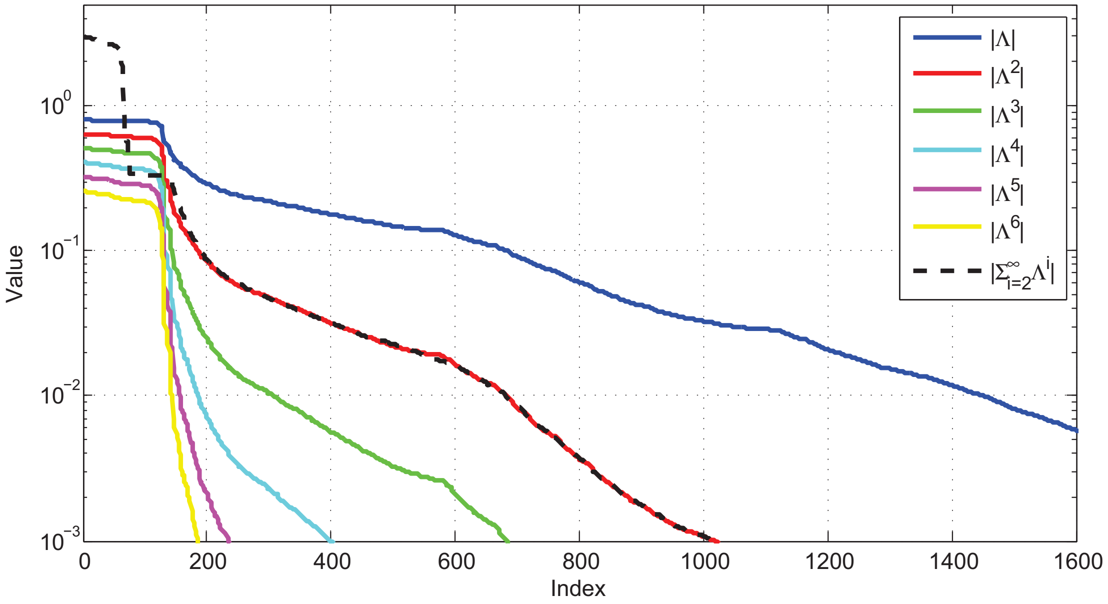


Figure 2.9: First terms in Neumann series and the sum of 3^{rd} and subsequent terms.

that instead of storing and processing a vast array like **RF**, it can be approximated by a product of two low-rank matrices ($\mathbf{RF} = \mathbf{UV}^T$), where **U** and **V** have dimension $n \times k$, with $n \gg k$. The LRR scheme allows us to find the approximated decomposition, but it suffers certain problems. It requires the calculation of all the form factors of the scene, which slows down the whole precomputation process. Another drawback of LRR is that it depends on the existence of two meshes with different granularity (one with n elements and the other with k elements), which determine the size of the matrices and the accuracy of the radiosity calculation.

In the literature, there are not many other approaches of global illumination based on the calculation of the inverse matrix. Kontkanen et al. [33] established an approach where a precomputed Global Transport Operator (GTO) is calculated to express the relationship between incident and outgoing surface radiance. This GTO contains the global interaction of radiance, independent of the particular light emission of the scene, and is calculated through the accumulation of the first terms of a Neumann series. After the computation of each term, all coefficients below a certain threshold are removed. This approach was later used in other works [74, 34].

The numerical linear algebra community has a long history in the development of algorithms for matrix decomposition, and low-rank matrix approximations. The classical approaches, like truncated singular value decomposition (TSVD) and rank-revealing QR decomposition [28], can be prohibitive for large matrices, because their running time are $O(n^2k)$ for square matrices. Other methodologies have been developed recently based on the construction of randomized algorithms that compute partial matrix decompositions. These methods use random sampling to identify a subspace that captures most of the action of a matrix. The input matrix is then compressed to this subspace, and the reduced matrix is manipulated deterministically to obtain the desired low-rank factorization. The works of Halko et al. [75] and Mahoney [76] bring deep insight into these methodologies. All the randomized algorithms presented in the above works require at least one pass over the data (as in LRR), and require $O(n^2 \log k)$ floating-point operations.

Another important result can be appreciated when the matrix to decompose has exactly low rank k. In this case, to find an exact factorization of the matrix it is enough to find a subset of k rows and k columns linearly independent [29]. In this decomposition, called CUR matrix decomposition or pseudo-skeleton approximation method, a given matrix $\mathbf{A}$ of dimension $n \times m$ is decomposed in the product of three matrices $\mathbf{A} = \mathbf{CUR}$, where $\mathbf{C}$ $(n \times k)$ is composed by a subset of columns of $\mathbf{A}$, $\mathbf{R}$ $(k \times m)$ contains a subset of rows of $\mathbf{A}$, and $\mathbf{U}$ $(k \times k)$ is defined as the inverse of the matrix defined by the intersection of $\mathbf{C}$ and $\mathbf{R}$.

When the real rank of the matrix is much greater than k, then the optimal selection of $\mathbf{C}$ and $\mathbf{R}$ matrices over all $\binom{n}{k}$ such matrices is an intractable combinatorial optimization problem [77]. The existing methods try to avoid such brute force methodology with different strategies, always passing one or more times over the data.

Bibliography

[1] T. Ritschel, C. Dachsbacher, T. Grosch, J. Kautz, The state of the art in interactive global illumination, Comput. Graph. Forum 31 (1) (2012) 160–188.

[2] A. S. Glassner, Principles of Digital Image Synthesis, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994.

[3] J. T. Kajiya, The rendering equation, SIGGRAPH Comput. Graph. 20 (4) (1986) 143–150.

[4] C. M. Goral, K. E. Torrance, D. P. Greenberg, B. Battaile, Modeling the interaction of light between diffuse surfaces, in: Proceedings of the 11th annual conference on Computer graphics and interactive techniques, SIGGRAPH '84, ACM, New York, NY, USA, 1984, pp. 213–222.

[5] H. W. Jensen, Global illumination using photon maps, in: Proceedings of the eurographics workshop on Rendering techniques '96, Springer-Verlag, London, UK, UK, 1996, pp. 21–30.

[6] A. Keller, Instant radiosity, in: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1997, pp. 49–56.

[7] B. Walter, S. Fernandez, A. Arbree, K. Bala, M. Donikian, D. P. Greenberg, Lightcuts: a scalable approach to illumination, in: ACM SIGGRAPH 2005 Papers, SIGGRAPH '05, ACM, New York, NY, USA, 2005, pp. 1098–1107.

[8] M. Hašan, F. Pellacini, K. Bala, Matrix row-column sampling for the many-light problem, ACM Trans. Graph. 26 (3).

[9] C. P., Point-based approximate coolor bleeding, Tech. rep., Pixar (2008).

[10] T. Ritschel, T. Engelhardt, T. Grosch, H.-P. Seidel, J. Kautz, C. Dachsbacher, Micro-rendering for scalable, parallel final gathering, in: ACM SIGGRAPH Asia

2009 papers, SIGGRAPH Asia ’09, ACM, New York, NY, USA, 2009, pp. 132:1–132:8.

[11] S. Chandrasekar, Radiative Transfer, Oxford Univ. Press, 1950.

[12] P.-P. Sloan, J. Kautz, J. Snyder, Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments, ACM Trans. Graph. 21 (3) (2002) 527–536.

[13] P. Dutre, K. Bala, P. Bekaert, P. Shirley, Advanced Global Illumination, AK Peters Ltd, 2006.

[14] M. Cohen, J. Wallace, P. Hanrahan, Radiosity and realistic image synthesis, Academic Press Professional, Inc., San Diego, CA, USA, 1993.

[15] J. McCorquodale, Radiosity homework, `http://www.cs.utah.edu/~mcq/radiosity/`, accessed: 2013-10-16.

[16] M. F. Cohen, S. E. Chen, J. R. Wallace, D. P. Greenberg, A progressive refinement approach to fast radiosity image generation, SIGGRAPH Comput. Graph. 22 (4) (1988) 75–84.

[17] P. Hanrahan, D. Salzman, A rapid hierarchical radiosity algorithm, in: Computer Graphics, 1991, pp. 197–206.

[18] B. Smits, J. Arvo, D. Greenberg, A clustering algorithm for radiosity in complex environments, in: Proceedings of the 21st annual conference on Computer graphics and interactive techniques, SIGGRAPH ’94, ACM, New York, NY, USA, 1994, pp. 435–442.

[19] M. Garland, A. Willmott, P. S. Heckbert, Hierarchical face clustering on polygonal surfaces, in: Proceedings of the 2001 symposium on Interactive 3D graphics, I3D ’01, ACM, New York, NY, USA, 2001, pp. 49–58.

[20] Z. Dong, J. Kautz, C. Theobalt, H.-P. Seidel, Interactive global illumination using implicit visibility, in: Proceedings of the 15th Pacific Conference on Computer Graphics and Applications, PG ’07, IEEE Computer Society, Washington, DC, USA, 2007, pp. 77–86.

[21] D. S. Immel, M. F. Cohen, D. P. Greenberg, A radiosity method for non-diffuse environments, SIGGRAPH Comput. Graph. 20 (4) (1986) 133–142.

[22] C. Dachsbacher, M. Stamminger, G. Drettakis, F. Durand, Implicit visibility and antiradiance for interactive global illumination, ACM Trans. Graph. 26 (3).

[23] Q. Meyer, C. Eisenacher, M. Stamminger, C. Dachsbacher, Data-parallel hierarchical link creation for radiosity, in: Proceedings of the 9th Eurographics conference on Parallel Graphics and Visualization, EG PGV’09, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2009, pp. 65–70.

[24] G. V. G. Baranoski, R. Bramley, J. G. Rokne, Eigen-analysis for radiosity systems, in: Proceedings of the Sixth International Conference on Computational Graphics and Visualization Techniques (Compugraphics ’97), Vilamoura, Algarve, Portugal, 1997, pp. 193–201.

[25] I. Ashdown, Eigenvector radiosity, Master's thesis, Department of Computer Science, University of British Columbia, Vancouver, British Columbia (April 2001).

[26] E. Fernández, Low-rank radiosity, in: O. Rodríguez, F. Serón, R. Joan-Arinyo, E. C. J. Madeiras, J. Rodríguez (Eds.), Proceedings of the IV Iberoamerican Symposium in Computer Graphics, Sociedad Venezolana de Computación Gráfica, DJ Editores, C.A., 2009, pp. 55–62.

[27] C. Eckart, G. Young, The approximation of one matrix by another of lower rank, Psychometrika 1 (3) (1936) 211–218.

[28] G. H. Golub, C. F. Van Loan, Matrix computations (3rd ed.), Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[29] S. Goreinov, E. Tyrtyshnikov, N. Zamarashkin, A theory of pseudoskeleton approximations, Linear Algebra and its Applications 261 (1 - 3) (1997) 1 – 21.

[30] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes 3rd Edition: The Art of Scientific Computing, 3rd Edition, Cambridge University Press, New York, NY, USA, 2007.

[31] I. E. Sutherland, R. F. Sproull, R. A. Schumacker, A characterization of ten hidden-surface algorithms, ACM Comput. Surv. 6 (1) (1974) 1–55.

[32] M. Cohen, D. Greenberg, D. Immel, P. Brock, An efficient radiosity approach for realistic image synthesis, IEEE Comput. Graph. Appl. 6 (1986) 26–35.

[33] J. Kontkanen, E. Turquin, N. Holzschuch, F. Sillion, Wavelet radiance transport for interactive indirect lighting, in: W. Heidrich, T. Akenine-Möller (Eds.), Rendering Techniques 2006 (Eurographics Symposium on Rendering), Eurographics, 2006, pp. 161–171.

[34] J. Lehtinen, M. Zwicker, E. Turquin, J. Kontkanen, F. Durand, F. X. Sillion, T. Aila, A meshless hierarchical representation for light transport, in: ACM SIGGRAPH 2008, ACM, New York, NY, USA, 2008, pp. 1–9.

[35] A. Tarantola, Inverse Problem Theory and Methods for Model Parameter Estimation, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2004.

[36] S. R. Marschner, Inverse rendering in computer graphics, Ph.D. thesis, Program of Computer Graphics, Cornell University, Ithaca, NY (1998).

[37] G. Patow, X. Pueyo, A survey of inverse rendering problems, Computer Graphics Forum 22 (4) (2003) 663–688.

[38] G. Patow, X. Pueyo, A survey of inverse surface design from light transport behavior specification., Computer Graphics Forum 24 (4) (2005) 773–789.

[39] R. Ramamoorthi, P. Hanrahan, On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object, J. Opt. Soc. Am. A 18 (10) (2001) 2448–2459.

[40] V. Harutunian, J. C. Morales, J. R. Howell, Radiation exchange within an enclosure of diffuse-gray surfaces: The inverse problem, Inverse Problems in Heat Transfer, ASME/AIChE National Heat Transfer Conference, Portland, USA, 1995.

[41] M. Contensin, Inverse lighting problem in radiosity, Inverse Problems in Engineering 10 (2) (2002) 131–152.

[42] P. C. Hansen, T. Sekii, H. Shibahashi, The modified truncated svd method for regularization in general form, SIAM J. Sci. Stat. Comput. 13 (5) (1992) 1142–1150.

[43] C. Schoeneman, J. Dorsey, B. Smits, J. Arvo, D. Greenberg, Painting with light, in: Proceedings of the 20th annual conference on Computer graphics and interactive techniques, ACM SIGGRAPH 93, ACM, New York, NY, USA, 1993, pp. 143–146.

[44] A. C. Costa, A. A. de Sousa, F. N. Ferreira, Lighting design: A goal based approach using optimisation., in: D. Lischinski, G. W. Larson (Eds.), Rendering Techniques, Springer, 1999, pp. 317–328.

[45] J. K. Kawai, J. S. Painter, M. F. Cohen, Radioptimization - goal based rendering, in: ACM SIGGRAPH 93, Anaheim, CA, 1993, pp. 147–154.

[46] D. Luenberger, Y. Ye, Linear and Nonlinear Programming, International Series in Operations Research & Management Science, Springer, 2008.

[47] F. Castro, E. del Acebo, M. Sbert, Energy-saving light positioning using heuristic search, Engineering Applications of Artificial Intelligence 25 (3) (2012) 566 – 582.

[48] F. Pellacini, F. Battaglia, R. K. Morley, A. Finkelstein, Lighting with paint, ACM Trans. Graph. 26 (2).

[49] S. Gibson, T. Howard, R. Hubbold, Flexible image-based photometric reconstruction using virtual light sources, Computer Graphics Forum 20 (2001) 203–214.

[50] A. Shesh, B. Chen, Crayon lighting: sketch-guided illumination of models, in: Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia, GRAPHITE ’07, ACM, New York, NY, USA, 2007, pp. 95–102.

[51] V. Tourre, J.-Y. Martin, G. Hégron, An inverse daylighting model for caad, in: Proceedings of the 24th Spring Conference on Computer Graphics, SCCG ’08, ACM, New York, NY, USA, 2008, pp. 83–90.

[52] G. Besuievsky, V. Tourre, A daylight simulation method for inverse opening design in buildings, in: O. Rodríguez, F. Serón, R. Joan-Arinyo, E. C. J. Madeiras, J. Rodríguez (Eds.), Proceedings of the IV Iberoamerican Symposium in Computer Graphics, Sociedad Venezolana de Computación Gráfica, DJ Editores, C.A., 2009, pp. 29–46.

[53] Y. Liu, X. Granier, Online tracking of outdoor lighting variations for augmented reality with moving cameras, IEEE Transactions on Visualization and Computer Graphics 18 (4) (2012) 573–580.

[54] D. Arquès, S. Michelin, Proximity radiosity: exploiting coherence to accelerate form factor computations, in: Proceedings of the eurographics workshop on Rendering techniques '96, Springer-Verlag, London, UK, UK, 1996, pp. 143–ff.

[55] J. Vilaplana, X. Pueyo, Multilevel use of coherence for complex radiosity environments, The Journal of Visualization and Computer Animation 5 (3) (1994) 129–141.

[56] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Wiley-Interscience Series in Systems and Optimization, John Wiley & Sons, Chichester, 2001.

[57] C. C. Coello, G. Lamont, D. van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd Edition, Genetic and Evolutionary Computation, Springer, Berlin, Heidelberg, 2007.

[58] E.-G. Talbi, Metaheuristics: From Design to Implementation, Wiley Publishing, 2009.

[59] M. J. Atallah, Handbook of algorithms and theory of computing, CRC Press, 1999.

[60] C. H. Papadimitriou, K. Steiglitz, Combinatorial optimization: algorithms and complexity, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982.

[61] K. Apt, Principles of Constraint Programming, Cambridge University Press, New York, NY, USA, 2003.

[62] H. Kargupta, D. E. Goldberg, Search, blackbox optimization, and sample complexity, in: In R.K. Belew & M. Vose (Eds.) Foundations of Genetic Algorithms 4, Morgan Kaufman, 1997, pp. 291–324.

[63] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[64] S. M. Goldfeld, R. E. Quandt, H. F. Trotter, Maximization by Quadratic Hill-Climbing, Vol. 34, The Econometric Society, 1966.

[65] S. J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 2nd Edition, Pearson Education, 2003.

[66] F. Castro, E. Acebo, M. Sbert, Heuristic-search-based light positioning according to irradiance intervals, in: A. Butz, B. Fisher, M. Christie, A. Krüger, P. Olivier, R. Therón (Eds.), Smart Graphics, Vol. 5531 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 128–139.

[67] F. Cassol, P. S. Schneider, F. H. França, A. J. S. Neto, Multi-objective optimization as a new approach to illumination design of interior spaces, Building and Environment 46 (2) (2011) 331 – 338.

[68] P. S. Schneider, A. C. Mossi, F. H. R. Franca, F. L. de Sousa, A. J. da Silva Neto, Application of inverse analysis to illumination design, Inverse Problems in Science and Engineering 17 (6) (2009) 737–753.

[69] F. L. DE SOUSA, F. M. RAMOS, P. PAGIONE, R. M. GIRARDI, New stochastic algorithm for design optimization, AIAA Journal 41 (9) (2003) 1808–1818.

[70] J. E. Tena, I. Rudomin, An interactive system for solving inverse illumination problems using genetic algorithms, in: Proceedings of Computacion Visual, 1997.

[71] K. Ferentinos, L. Albright, Optimal design of plant lighting system by genetic algorithms, Engineering Applications of Artificial Intelligence 18 (4) (2005) 473 – 484.

[72] S. Delepoulle, C. Renaud, M. Chelle, Improving light position in a growth chamber through the use of a genetic algorithm, in: Artificial Intelligence Techniques for Computer Graphics, Springer, 2008, pp. 67–82.

[73] P. Hansen, N. Mladenovic, Variable neighborhood search: Principles and applications, European Journal of Operational Research 130 (3) (2001) 449–467.

[74] J. Lehtinen, M. Zwicker, J. Kontkanen, E. Turquin, F. Sillion, T. Aila, Meshless finite elements for hierarchical global illumination, Tech. Rep. TML-B7, Helsinki University of Technology (may 2007).

[75] N. Halko, P. G. Martinsson, J. A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, SIAM Rev. 53 (2) (2011) 217–288.

[76] M. W. Mahoney, Randomized algorithms for matrices and data., Foundations and Trends in Machine Learning 3 (2) (2011) 123–224.

[77] M. W. Mahoney, P. Drineas, Cur matrix decompositions for improved data analysis, Proceedings of the National Academy of Sciences 106 (3) (2009) 697–702.

Chapter 3

Low-Rank Radiosity Using Sparse Matrices [1]

Radiosity methods are part of the global illumination techniques, which deal with the problem of generating photorealistic images in 3D scenes with Lambertian surfaces. Low-rank radiosity is a $O(nk)$ method, where n is the number of polygons and k is the rank of the matrix used as a direct transport operator. This method allows calculating, in real-time and with infinite bounces, the illumination of a scene with static geometry and dynamic lighting. In this paper we present a new methodology for low-rank radiosity calculation based on the use of sparse matrices, which significantly reduces the memory storage required and achieves speedup improvements over the original low-rank method. Experimental analysis was performed in both traditional computers and new graphics processing unit architectures.

Keywords: Radiosity, Real-time global illumination

3.1 Introduction

The radiosity method is a pioneering technique for global illumination on scenes with Lambertian surfaces, which has been applied in many areas of design and computer animation [1].

Since the classic formulation of the radiosity technique in the decade of 1980, several iterative resolution methods have been proposed [2]. These methods are not feasible for computing realistic images in real-time (many bounces at 20 fps), due to their iterative nature. After that, several other methods have been proposed to achieve the real-time computation of the global illumination, such as *instant radiosity* [3], *precomputed radiance transfer* [4], *radiance transport* [5, 6] and *photon mapping* [7], which have shown some success using global illumination approximations.

[1] **Published in:** E. Fernández, P. Ezzatti, S. Nesmachnow and G. Besuievsky *Low-rank Radiosity using Sparse Matrices.* Proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP 2012) and International Conference on Information Visualization Theory and Applications (IVAPP 2012), Rome, Italy, 24-26 February, 2012, Pages 260-267.
A shorter version of the paper was published in: E. Fernández, P. Ezzatti and S. Nesmachnow *Improving the Low Rank Radiosity Method Using Sparse Matrices.* Mecánica Computacional Vol XXIX, Buenos Aires, Argentina, 15-18 November 2010, Pages 7053-7066.

By exploiting spatial coherence, the low-rank radiosity method (LRR) [8] allows solving the real-time radiosity problem with infinite bounces, for scenes with static geometry and dynamic lighting, using a low memory storage approach. The original LRR implementation used full matrices, and the promising idea of generating sparse matrices was previously pointed out, but it was not developed.

This work presents a new approach for the utilization of sparse matrices in the LRR method. Specific implementations of algorithms to solve the radiosity problem in both traditional computer architectures (CPU) and modern graphic processing units (GPU) are introduced. The experimental results show that the GPU implementation of the LRR method based on sparse matrices has an acceleration factor greater than four, compared to the full matrix approach of the LRR method. These promising results demonstrate that our approach is an alternative for real-time radiosity applications. Comparing to other strategies, LRR ensures always a full global illumination solution being capable of dealing efficiently with environments with significant indirect illumination contribution.

The rest of the article is organized as follows. Section 3.2 introduces the radiosity problem and the previous work. The low-rank radiosity method is described in Section 3.3. Section 3.4 introduces the utilization of sparse matrices in LRR, and describes the implementations of the proposed algorithms for sparse low-rank radiosity in CPU and GPU. The experimental analysis is reported in Section 3.5. Finally, Section 3.6 summarizes the conclusions of the research and formulates the main lines for future work.

3.2 The Radiosity Problem

This section starts by introducing the radiosity problem formulation. After that, the main proposals found in the literature related with the pre-computed radiosity and radiance transfer problem are reviewed.

3.2.1 Problem Formulation

The radiosity problem is modeled by a Fredholm integral equation, the *radiosity equation* (3.1). In Equation 3.1, $B(x)$ is the radiosity in point x, $E(x)$ is the light emission in x, $\rho(x)$ is the Lambertian diffuse reflectivity of x, and $G(x, x')$ is a geometric factor that determines how much of the radiosity in x' contributes to the radiosity of x [2].

$$B(x) = E(x) + \rho(x) \int_S B(x')G(x, x')dA' \tag{3.1}$$

Any Fredholm-type equation can be transformed into a discrete equation, which implies solving a linear system. In the radiosity problem the linear system is expressed by Equation 3.2, where $\mathbf{I}$ is the identity matrix with dimension $n \times n$ (n is the number of polygons), $\mathbf{R}$ is a diagonal matrix that stores the reflectivity of the polygons, $\mathbf{F}$ is the matrix with the form factors $\mathbf{F}_{ij}$ –which indicate the fraction of light reflected by the polygon i that arrives to the polygon j–, B is a vector with the radiosity value of each patch, and E is the vector with the emission of each patch.

$$(\mathbf{I} - \mathbf{RF})B = E \tag{3.2}$$

This discrete equation can be also formulated through the use of a direct transport operator $\mathbf{T}$, which expresses the direct transmission of light between two surfaces of the scene. Using $\mathbf{T}$, the discrete radiosity equation (Equation 3.2) is expressed as $(\mathbf{I}{-}\mathbf{T})B{=}E$. Moreover, the operator $\mathbf{M} = (\mathbf{I} - \mathbf{T})^{-1}$, is a global transport operator relating the emitted light with the final radiosity of the scene, $B = \mathbf{M}E$. For scenes where the geometry is fixed and only the light output varies, the operators $\mathbf{T}$ and $\mathbf{M}$ are constant.

3.2.2 Related Work

The main class of elements used in light transport includes the polygons, the parametric surfaces [2], and the discrete points [9]. In all these cases it is possible to build operators $\mathbf{T}$ and $\mathbf{M}$ that indicate the transport relationship between any pair of elements.

The $O(n^2)$ memory required to store $\mathbf{T}$ has been clearly posed as a problem in early radiosity papers [10]. Many solutions have been proposed, such as two-level hierarchy [11], hierarchical structures [12], and wavelets [13]. The memory savings are determined by the spatial coherence of the scenes, where it is highly probable that neighboring elements i, j have similar transport values with many other elements k of the scene, that is, $\mathbf{T}_{ik} \approx \mathbf{T}_{jk}$. In this case, these values are considered equal, then, many transport values can be replaced by a single value. This is the main idea behind the hierarchical approximation $H(\mathbf{T})$ of any $\mathbf{T}$ operator, where $H(\mathbf{T})$ is usually stored in $O(n)$ memory. The balance between the error introduced by the method and the memory saved is highly convenient for most scenes. In these works the calculation of B is done by iterative methods, avoiding the need of computing $H(\mathbf{M})$.

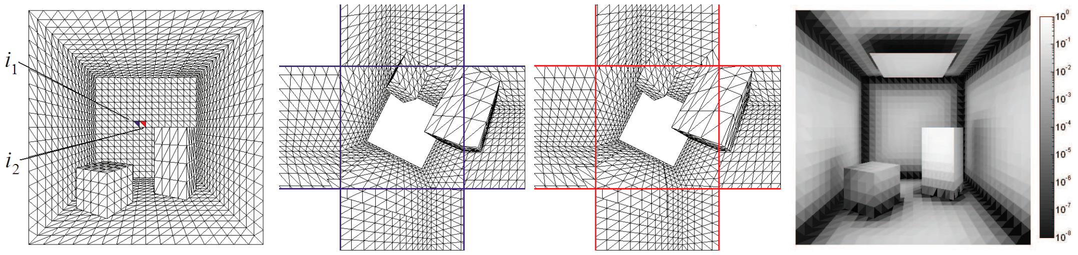
Recently, other kinds of methods that include BRDF surfaces and the calculation of $H(\mathbf{M})$ have been developed. In [5], the BRDF operator ($\mathbf{T}$) is transformed into $H(\mathbf{T})$ through the use of wavelets, and $H(\mathbf{M})$ is calculated using the Neumann series $H(\mathbf{M}) = \mathbf{I} + H(\mathbf{T}) + H(\mathbf{T})^2 + \dots$, by removing all values of $H(\mathbf{T})^n$ smaller than a certain threshold. After calculating the global incident radiance of a scene, the BRDF function is applied again to find an image from a particular point of view. Later, in [6] a meshless hierarchical representation of the objects in the scene, built upon a hierarchy of points, is performed. Each point consists of a pair (position, normal) and has a radius of influence related to the hierarchy level to which it belongs. Using a hierarchical function basis, a sparse matrix $H(\mathbf{T})$ is generated by elimination of all terms below a threshold, and a final sparse operator $H(\mathbf{M} - \mathbf{I})$ is created, using also Neumann series.

Other methods used to achieve real-time global illumination are based on instant radiosity techniques [3] and its variants [14], but these methods compute only a few bounces.

In contrast, the LRR method proposed in this work allow us to solve the radiosity problem with infinite bounces and relatively small computational resources. This result is based on the premise that when $\mathbf{T}$ is a low-rank matrix, the operator $\mathbf{M}$ can be easily calculated, as it is explained in the next section.

3.3 The Low-Rank Radiosity Method

It is very likely that the $\mathbf{RF}$ matrix included in Equation 3.2 has a low numeric rank. This is because each row $(\mathbf{RF})_i$ is computed based on the scene view from the element i, and most pairs of close patches have a very similar view of the scene (Figure 3.1). Then $\mathbf{RF}$ has many similar rows, which results in a reduction of the numerical rank.



(a) Two close patches i_1 and i_2.

(b) Hemicube view from i_1.

(c) Hemicube view from i_2.

(d) Spatial coherence in the scene.

Figure 3.1: Close patches have similar views of the scene and generate similar rows in **RF**. In (d) the spatial coherence of each patch is measured (darker means lower coherence).

The rank reduction of **RF** matrices allows approximating them by the product of two matrices $\mathbf{U}_k\mathbf{V}_k^T$, both with dimension $n\times k$ ($n\gg k$), without losing relevant information about the scene.

3.3.1 Low-Rank Approximation

The product $\mathbf{U}_k\mathbf{V}_k^T$ generates a matrix with dimension $n\times n$ and rank k. The memory space needed to store $\mathbf{U}_k$ and $\mathbf{V}_k^T$ is $O(nk)$, significantly less than the $O(n^2)$ required to store the matrix **RF**, especially when $n\gg k$. This memory saving allows reducing the space needed to store the information for large scenes.

In the *LRR equation* (3.3), **RF** is replaced by the low-rank approximation $\mathbf{U}_k\mathbf{V}_k^T$. The unknown is not longer B, but an approximation $\widetilde{B}$ of the radiosity.

$$\left(\mathbf{I} - \mathbf{U}_k\mathbf{V}_k^T\right)\widetilde{B} = E \tag{3.3}$$

The matrix $\left(\mathbf{I} - \mathbf{U}_k\mathbf{V}_k^T\right)$ is easily invertible using the Sherman-Morrison-Woodbury formula [15]. Equation 3.4 shows the expression for $\widetilde{B}$.

$$\widetilde{B} = E + \mathbf{U}_k\left(\left(\mathbf{I}_k - \mathbf{V}_k^T\mathbf{U}_k\right)^{-1}\left(\mathbf{V}_k^T E\right)\right) \tag{3.4}$$

In order to find $\widetilde{B}$ through Equation 3.4, $O(nk^2)$ operations are required and $O(nk)$ memory is needed.

In scenes with static geometry and dynamic lighting (i.e., only the term E varies in Equations 3.2 to 3.4), part of the computation can be executed before the real-time stage. Equation 3.4 is transformed into 3.5, where the $n\times k$ matrices $\mathbf{Y}_k$, $\mathbf{U}_k$, and $\mathbf{V}_k$ are calculated once. Thus, the real-time stage only has complexity $O(nk)$. This is an auspicious result in order to develop a new methodology for real-time radiosity.

$$\begin{aligned}\widetilde{B} &= E - \mathbf{Y}_k\left(\mathbf{V}_k^T E\right), \\ \mathbf{Y}_k &= -\mathbf{U}_k\left(\mathbf{I}_k - \mathbf{V}_k^T\mathbf{U}_k\right)^{-1}\end{aligned} \tag{3.5}$$

Equation 3.5 can also be formulated as $\widetilde{B} = \mathbf{M}E$, so $\mathbf{M} = \mathbf{I} - \mathbf{Y}_k\mathbf{V}_k^T$ is considered as the global operator that manages the infinite bounces of light in a single operation.

This global operator is computed without the use of Neumann series, as in [5, 6], taking advantage of the fact that **RF** is a low-rank matrix.

3.3.2 $\mathbf{U}_k$ and $\mathbf{V}_k$ Calculation

$\mathbf{U}_k$ and $\mathbf{V}_k$ can be computed using factorization techniques, such as singular value decomposition (SVD or PCA) [15], CUR factorization [16], and discrete transformations based on Fourier and wavelets [17]. SVD is the only method that produces good approximations to **RF** for low values of k, but it has an $O(n^3)$ complexity. The other methods mentioned have lower complexity orders, but they also produce larger errors in the **RF** approximation.

To overcome the weakness of the traditional factorization techniques, the Low-Rank Radiosity (LRR) method proposed in [8] uses the concept of *spatial coherence* [18]. This method generalizes the two-level hierarchy methodology [11], where two meshes with different granularity levels are generated for the scene (a *coarse mesh* with k patches and a *fine mesh* with n elements). Using these two meshes it is possible to build the $n \times k$ matrices $\mathbf{U}_k$ and $\mathbf{V}_k$.

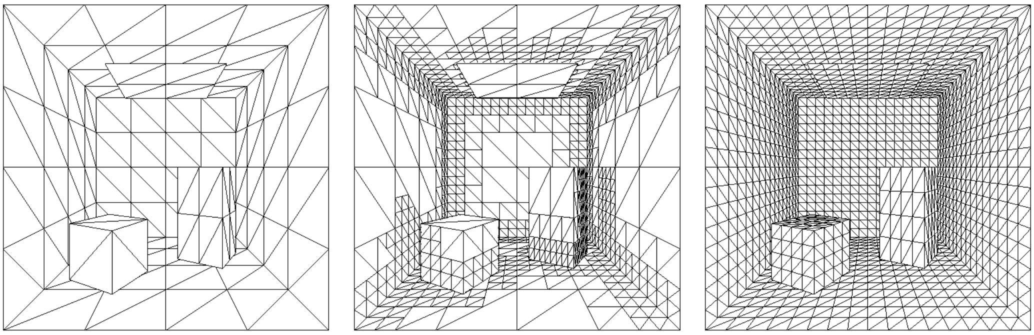
Both, the coarse and the fine meshes, can be uniform or nonuniform. In this paper, we proposed a nonuniform coarse mesh that is adapted to the spatial coherence of the scene. In the resulting algorithm, named *2 Meshes with Spatial Coherence* (2MSC), the fine mesh is given, assuming that inside each element there is spatial coherence, but the coarse mesh is built taking care of the spatial coherence inside each patch.

The 2MSC produces a nonuniform mesh, with smaller patches in those areas with low spatial coherence. Algorithm 1 shows a *top-down* procedure for creating the coarse mesh from a few large patches, using the 2MSC algorithm. An initial set of large patches are inserted into a queue Q (see Figure 3.2 (a)). Every patch is taken from Q, one at a time. All patches without spatial coherence (i.e., when the inner scene views vary more than a specified threshold) are split into several subpatches, which are inserted into Q. The algorithm stops when the queue Q is empty (i.e., when all patches have spatial coherence), or when all those patches without spatial coherence are smaller than a predefined value (A_{min}). The number of patches (k) depends on the value of the parameter A_{min} and also depends on the procedure used to evaluate the spatial coherence. Figure 3.2 (b, c) shows a coarse mesh generated using the 2MSC algorithm and a fine mesh. The discretization of the coarse mesh shown in Figure 3.2 (b) agrees with the colors of spatial coherence shown in Figure 3.1 (d), i.e., the darker the color, the more need to split the surface.

Algorithm 1: 2MSC Algorithm.

```
Store patches $p$ in a queue $Q$.
while ($Q \neq \{\}$) do
    Remove a patch $p$ from $Q$.
    if $\neg$(spatial coherence inside $p$) $\wedge$ ($Area(p) > A_{min}$)
        Divide $p$ into subpatches and store them in $Q$.
    end.
end.
```

The spatial coherence inside a patch p can be accomplished in different ways. A simple algorithm involves dividing p in several subpatches $\{p_1, \ldots, p_s\}$, and then cal-



(a) Original coarse mesh. (b) Coarse mesh after 2MSC. (c) Fine mesh.

Figure 3.2: Meshes used to compute $\mathbf{U}_k$ and $\mathbf{V}_k$.

culate the form factors between each subpatch and the fine mesh version of the scene. This generates a set of vectors $\{v_1, \ldots, v_s\}$, each containing the form factors related to a subpatch. Now we are in condition to measure the euclidean distance between each v_i and the average vector $\sum_{i=1}^{s} v_i/s$. If some of the distances are greater than a specified threshold, then we consider that there is not spatial coherence inside p.

After the scene meshes are defined, the matrices $\mathbf{U}_k$ and $\mathbf{V}_k$ are built using the information from an assignment function $P : e \to p$ that gives, for each element e in the fine mesh, the corresponding patch p in the coarse mesh. The assignment can be defined by inclusion (element e belongs to single patch p) or by proximity (p is the closest patch to element e).

If P is defined by inclusion and all elements belonging to any patch have equal area, the construction of matrices $\mathbf{U}_k$ and $\mathbf{V}_k$ follows Equation 3.6. There, $\mathbf{F}_{ep}$ is the form factor between an element e of the scene and the elements assigned to the patch p, and c_p is the number of elements assigned to p.

$$\begin{gathered}\mathbf{U}_k(e,p) = \mathbf{F}_{ep}/c_p \\ \mathbf{V}_k(e,P(e)) = 1\ ,\ \mathbf{V}_k(e,p) = 0\ \forall p \neq P(e)\end{gathered} \tag{3.6}$$

Other methodologies can be applied to build the matrices $\mathbf{U}_k$ and $\mathbf{V}_k$. Besides using SVD factorization, where the columns of both matrices are orthogonal, the columns of the matrix $\mathbf{U}_k$ generated using Equation 3.6 can also be orthogonalized using QR factorization. After the QR factorization is performed, a new matrix $\mathbf{U}_{QR}$ is generated and $\mathbf{V}_k$ can be replaced by $\mathbf{V}_{QR} = (\mathbf{RF})^T\mathbf{U}_{QR}$. These methodologies have been implemented by the authors to generate $\mathbf{U}_k$ and $\mathbf{V}_k$, but the experiments showed no conclusive results about which is best method. In this paper, the Equations in 3.6 were employed to build the matrices, due to both the simplicity of building a low rank approximation of the $\mathbf{RF}$ matrices and the possibility to build a sparse version of the $\mathbf{V}_k$ matrix, as it is explained in the next section.

3.4 Sparse Matrices in LRR

The 2MSC algorithm and the Equations 3.6 generate sparse matrices $\mathbf{V}_k$ with dimension $n{\times}k$, with a unique non-zero entry in each row (corresponding to the $P(e)$ column).

This section introduces an $O(k)$ storage methodology for the $\mathbf{V}_k$ matrix, and an implementation of the LRR method for both CPU and GPU platforms.

3.4.1 Compressing the Information in $\mathbf{V}_k$

The rows of $\mathbf{V}_k$ can be reordered, grouping them by the column where the non-zero value is in (k disjoint groups are created, see Figure 3.3). Using ideas from the compressed storage of sparse matrices [19], an index vector I can be defined. I indicates that the rows of $\mathbf{V}_k$ with a one in the first column are placed from $I_1 = 1$ to I_2-1, that the rows of $\mathbf{V}_k$ with a one in the second column are placed from I_2 to I_3-1, and so on. The $O(k)$ entries of I are enough to define the matrix $\mathbf{V}_k$.

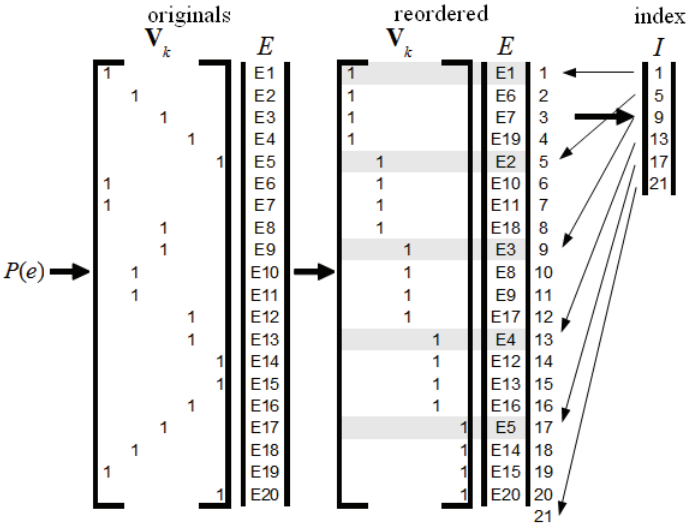


Figure 3.3: Construction of $\mathbf{V}_k$ and I from the function P.

Once the vector I is built and the values in the vector E are reordered accordingly to the rows of $\mathbf{V}_k$, the product $\mathbf{V}_k^T E$ can be formulated as the sum in Equation 3.7. The i-th value of the vector $\mathbf{V}_k^T E$ is the sum of the values of E between the positions I_i and $I_{i+1} - 1$.

$$\left(\mathbf{V}_k^T E\right)_i = \sum_{s=I_i}^{I_{i+1}-1} E_s \text{ , } \forall i, 1 \le i \le k \tag{3.7}$$

The previously presented results allow us to conclude that if $\mathbf{V}_k$ is a sparse matrix with the stated properties, the matrix-vector product $\mathbf{V}_k^T E$ can be computed in $O(n)$ operations (instead of the $O(nk)$ operations needed when $\mathbf{V}_k$ is a full matrix).

3.4.2 Sparse LRR Implementation

In order to validate the theoretical results presented in the previous subsection, two implementations of the sparse LRR method were developed to solve the radiosity problem with a sparse $\mathbf{V}_k$ matrix generated by Equations 3.6. Each implementation uses a different hardware to perform the calculations: one uses a traditional CPU, and the other one uses a GPU.

The LRR implementation in CPU uses the BLAS library [20], and Equation 3.7 is solved sequentially. On the other side, the LRR implementation in GPU uses the CUBLAS implementation of BLAS for CUDA [21], and k threads are used to update the vector $\left(\mathbf{V}_k^T E\right)_i$ in parallel, each thread computing $\sum_{s=I_i}^{I_{i+1}-1} E_s$ for a different i.

Both LRR implementations use the `xGEMV` function from the BLAS library to compute the matrix-vector product $\mathbf{Y}_k^T(\mathbf{V}_k^T E)$. The product $\mathbf{V}_k^T E$ is not explicitly computed, instead, it is calculated using the sum in Equation 3.7. The pseudocode of both implementations for a real-time processing of frames from a graphic application is sketched in Algorithm 2.

Algorithm 2: Sparse LRR

```
{Pre-processing: Compute $\mathbf{Y}_k$ and $I$}
[ONLY CPU] Store $\mathbf{Y}_k$, $I$ into RAM memory
[ONLY GPU] Send and store $\mathbf{Y}_k$, $I$ into GPU memory
for each frame do {in real-time}
    Receive $E$ from a graphic application
    [ONLY GPU] Send and store $E$ into GPU memory
    Compute $X_i = \left(\mathbf{V}_k^T E\right)_i, \forall i, 1 \leq i \leq k$, with $\sum_{s=I_i}^{I_{i+1}-1} E(s)$
    Apply xGEMV to compute $\widetilde{B} = E - \mathbf{Y}_k^T X$
    [ONLY GPU] Send $\widetilde{B}$ to RAM memory
    Store $\widetilde{B}$ into RAM memory
end
```

3.5 Experimental Analysis

The experimental analysis compares the time required to compute $\widetilde{B}$ using the LRR method, based on full and sparse matrices, as well as on CPU and GPU architectures.

3.5.1 Execution Platforms

A first analysis was performed in a Pentium dual-core E5200 (2.50 GHz), 2 GB RAM, with a NVIDIA 9800 GTX+ 512 MB. A second analysis using only sparse matrices and larger scenes was executed in a Tesla Intel Xeon QuadCore E5530 (2.27 GHz), 8 GB RAM, with a NVIDIA c1060 (240 cores at 1.3 GHz) 4 GB.

3.5.2 Test Scenes

A first analysis was performed using a *Cornell box* with several discretizations ($n=\{3456, 13824, 55296, 221184\}$ and $k=\{216, 864, 3456\}$). For the experiments in the Tesla platform, six scenes with dimension ($n\times k$) 1769472×216, 221184×3456, 442368×1728, 884736×864, 1769472×432, and 3538944×216 were created from the same Cornell box scene. Finally, we perform tests on a two-floor patio model with dynamic lighting changes.

3.5.3 Experimental Results

In the following tables, the row categorization puts together all scenes with approximately equal memory consumption. Each category consumes four times more memory than the previous one.

Table 3.1 shows the speedup values (ratio of execution times) obtained when computing $\mathbf{V}_k^T E$ using the sparse LRR implementations in GPU and CPU, and the comparison with the full matrix implementation. The best speedup values are shown in bold font.

The experimental results show that the speedup values were up to **335.50** in CPU, and up to **37.47** in GPU. The best speedup values were obtained for the scene with dimension 13824×3456, taking advantage of the large number of calculations in the sum and the parallel computing strategy in `xGEMV`. Table 3.1 also indicates that the sparse $\mathbf{V}_k^T E$ product is faster in CPU.

dimension	$\frac{\textbf{fullCPU}}{\textbf{sparseCPU}}$	$\frac{\textbf{fullGPU}}{\textbf{sparseGPU}}$	$\frac{\textbf{sparseCPU}}{\textbf{sparseGPU}}$	$\frac{\textbf{fullCPU}}{\textbf{sparseGPU}}$
3456 × 216	15.67	8.00	0.60	9.40
3456 × 864	68.33	9.00	0.60	41.00
13824 × 216	21.67	11.36	0.64	13.93
3456 × 3456	75.75	18.33	**0.67**	50.50
13824 × 864	92.11	13.21	0.64	59.21
55296 × 216	17.42	11.20	0.64	11.20
13824 × 3456	**335.50**	**37.47**	**0.67**	**223.67**
55296 × 864	94.03	13.09	0.65	61.55
221184 × 216	24.01	10.99	0.58	13.84

Table 3.1: Speedup when computing $\mathbf{V}_k^T E$.

Table 3.2 shows the speedup values when computing $\widetilde{B}$. These results demonstrate that the sparse implementation is about two times faster than the full implementation in CPU. Regarding the GPU implementations, the speedup values were between **5.21** and 2.05, and the best results were obtained when $k = 216$. Since 216 is not a multiple of 32, the GPU computing capacities are not fully used [22], and better speedup values shall be expected for other values of k. The computation of $\widetilde{B}$ is faster in GPU, despite that $\mathbf{V}_k^T E$ is faster in CPU.

dimension	$\frac{\textbf{fullCPU}}{\textbf{sparseCPU}}$	$\frac{\textbf{fullGPU}}{\textbf{sparseGPU}}$	$\frac{\textbf{sparseCPU}}{\textbf{sparseGPU}}$	$\frac{\textbf{fullCPU}}{\textbf{sparseGPU}}$
3456 × 216	1.95	4.50	4.40	8.60
3456 × 864	1.69	2.38	6.48	10.93
13824 × 216	1.91	**5.21**	5.50	10.53
3456 × 3456	1.43	2.05	5.70	8.17
13824 × 864	1.88	3.41	9.73	18.32
55296 × 216	1.90	5.04	5.58	10.61
13824 × 3456	1.74	3.59	**11.34**	19.67
55296 × 864	1.95	3.62	10.95	**21.34**
221184 × 216	**2.05**	5.04	5.08	10.43

Table 3.2: Speedup when computing $\widetilde{B}$.

Table 3.3 summarizes the speed of the LRR implementations regarding the number of frames per second (fps) that they are able to compute.

dimension	**CPU**		**GPU**	
	full	**sparse**	**full**	**sparse**
3456 × 216	**814**	**1591**	**1556**	**7000**
3456 × 864	221	372	1014	2414
13824 × 216	196	374	395	2059
3456 × 3456	70	100	278	569
13824 × 864	51	96	273	933
55296 × 216	47	90	100	504
13824 × 3456	14	24	76	275
55296 × 864	12	23	70	252
221184 × 216	11	23	23	116
Average	159	299	421	1569

Table 3.3: Speed (fps) of the LRR implementations.

The results in Table 3.3 show that the minimum fps rate in CPU for the sparse LRR implementation (23 fps) is twice the value of the full LRR implementation from the previous work. In GPU, the minimum speed largely increased from 23 fps to 116 fps with the sparse LRR implementation. For the small scene with dimension 3456×216, a large fps rate of **7000** was achieved in GPU. Overall, the average speed values duplicate in CPU and quadruplicate in GPU, when compared the speed of the full LRR implementation.

3.5.4 Real-Time LRR for Larger Scenes

The computational platform with the NVIDIA 9800 GTX+ GPU does not allow to scale up for computing the radiosity of scenes with a million polygons, due to its memory limitations. In order to show how the hardware evolution helps to solve the radiosity problem, six new larger scenes of up to more than 3.5 million elements were solved in a Tesla GPU platform.

Table 3.4 presents the speed results (in fps) when using the Tesla platform to compute the radiosity for the three largest instances computed with the NVIDIA 9800 GTX+ GPU, and the speedup comparison with the Table 3.3. This analysis compares the capability of computing the radiosity in real-time using the new hardware. The fps rates in Table 3.4 show that a significant speed improvement is obtained with the Tesla platform: it allows performing the radiosity computation in real-time, while achieving speedup values up to **3.20** in CPU and **1.64** in GPU.

Table 3.5 summarizes the speed of the radiosity calculation for six new larger scenes of up to 3.5 million elements in the Tesla platform. The scene with dimension 1769472×216 requires 8 times more memory than the largest scenarios in the previous subsection. Each of the other five scenes requires 16 times more memory than the largest scenarios in the previous subsection.

The results in Table 3.5 demonstrate that the radiosity computation using the sparse LRR method can be performed in real-time in GPU, while none of the studied scenes

dimension	speed (fps)		speedup	
	CPU	GPU	CPU	GPU
13824 × 3456	**74.07**	297.87	3.06	1.09
55296 × 864	73.61	**301.72**	**3.20**	1.20
221184 × 216	72.92	190.74	3.18	**1.64**

Table 3.4: Comparative speed (fps) in the Tesla platform.

dimension	speed in CPU	speed in GPU
1769472 × 216	**8.15**	**24.05**
221184 × 3456	4.94	21.81
442368 × 1728	4.94	20.59
884736 × 864	4.55	18.67
1769472 × 432	4.10	15.45
3538944 × 216	4.00	11.48

Table 3.5: Speed (fps) for larger scenes in the Tesla platform.

can be computed in real-time in CPU. The GPU implementation was able to achieve a reasonable level of speed (around 20 fps) for scenes up to 1.7 million elements. A significant speed reduction was detected when solving the scene with more than 3.5 million elements, suggesting that there is still room to improve the proposed method in order to face even more complex scenarios.

Figure 3.4 shows a wireframe view and three frames of an animated sequence for a patio building model, where the emission lights change for each frame. The complete video can be seen at `www.fing.edu.uy/inco/grupos/cecal/cg/LRRSM.wmv`. In the patio building (Figure 3.4), the dimension ($n \times k$) is equal to 24128×1364 and the frame rate achieved is of 430 fps.

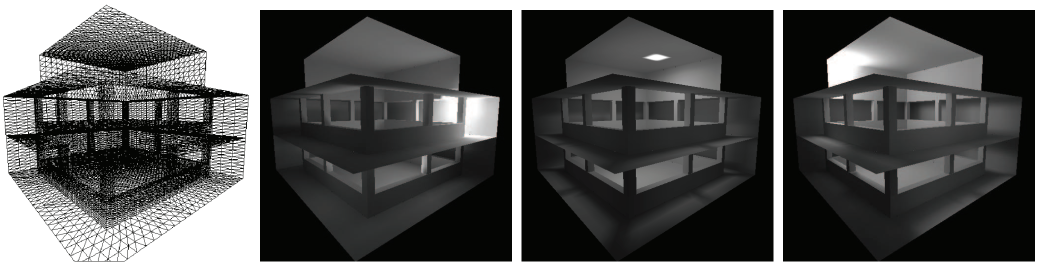


Figure 3.4: Patio building model. A wireframe view, and three different frames.

3.6 Conclusions and Future Work

This work introduces the utilization of sparse matrices in the low-rank radiosity technique. Four implementations of the LRR method (full/sparse matrices on CPU/GPU architectures) have been developed to solve the radiosity problem, and several examples of sparse LRR calculation have been implemented on a powerful GPU platform for greater $n \times k$ scenarios.

The main contribution of this work consists in the inclusion of a sparse matrix implementation into the LRR method. By reordering and grouping the information, $\mathbf{V}_k$ is transformed into an index vector I which requires only $O(k)$ memory to store the same information. As a result, the averaged acceleration factor rises to four in GPU architectures, due to the implementation of $\mathbf{V}_k^T E$ product with $O(n)$ sums.

The proposed method was able to solve the radiosity calculation in real-time for scenes with more than 1.7 million elements in a Tesla GPU platform. This result shows the degree in which the sparse LRR method has a better performance than the full LRR method.

Two main lines of future work can be directly inferred from this paper: to further improve the efficiency of the proposed method, and to develop new ways to build $\mathbf{U}_k\mathbf{V}_k^T$ factorizations. Regarding the first line of work, the speed of the radiosity calculation shall be reduced when using a hybrid strategy of simultaneous computation in CPU and GPU, taking advantage of the fast computation of the product $\mathbf{V}_k^T E$ in CPU. Respecting the second line of work, further research must be performed to know the relation between low-rank approximations of $\mathbf{RF}$ and the relative error of $\widetilde{B}$. Preliminary results have shown that in many situations, SVD is not the optimal factorization for the radiosity problem, because other factorizations (such as those presented in this paper) generate radiosity results with lower relative error.

Finally, another area of future work consists in facing the problem of processing scenes with dynamic geometry in order to extend the applicability of this new technique.

Acknowledgments

This work was partially funded by Programa de Desarrollo de las Ciencias Básicas, Uruguay, and by the TIN2010-20590-C02-02 project from Ministerio de Educación y Ciencia, Spain.

Bibliography

[1] P. Dutre, K. Bala, P. Bekaert, P. Shirley, Advanced Global Illumination, A. K. Peters, Ltd., Natick, MA, USA, 2006.

[2] M. Cohen, J. Wallace, P. Hanrahan, Radiosity and realistic image synthesis, Academic Press Professional, Inc., San Diego, CA, USA, 1993.

[3] A. Keller, Instant radiosity, in: Proceedings of the 24th annual conference on computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1997, pp. 49–56.

[4] P. Sloan, J. Kautz, J. Snyder, Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments, ACM Transactions on Graphics 21 (2002) 527–536.

[5] J. Kontkanen, E. Turquin, N. Holzschuch, F. Sillion, Wavelet radiance transport for interactive indirect lighting, in: W. Heidrich, T. Akenine-Möller (Eds.), Rendering Techniques 2006 (Eurographics Symposium on Rendering), Eurographics, 2006.

[6] J. Lehtinen, M. Zwicker, E. Turquin, J. Kontkanen, F. Durand, F. X. Sillion, T. Aila, A meshless hierarchical representation for light transport, ACM Trans. Graph. 27 (2008) 37:1–37:9.

[7] R. Wang, R. Wang, K. Zhou, M. Pan, H. Bao, An efficient GPU-based approach for interactive global illumination, in: SIGGRAPH International Conference and Exhibition on Computer Graphics and Interactive Techniques, ACM, New York, NY, USA, 2009, pp. 1–8.

[8] E. Fernández, Low-rank radiosity, in: Proceedings IV Iberoamerican Symposium in Computer Graphics, Isla Margarita, Venezuela, 2009, pp. 55–62.

[9] G. E. Fasshauer, Meshfree methods, in: Handbook of Theoretical and Computational Nanotechnology. American Scientific Publishers, 2006, pp. 33–97.

[10] M. F. Cohen, S. E. Chen, J. R. Wallace, D. P. Greenberg, A progressive refinement approach to fast radiosity image generation, SIGGRAPH Comput. Graph. 22 (1988) 75–84.

[11] M. Cohen, D. Greenberg, D. Immel, P. Brock, An efficient radiosity approach for realistic image synthesis, IEEE Comput. Graph. Appl. 6 (1986) 26–35.

[12] P. Hanrahan, D. Salzman, L. Aupperle, A rapid hierarchical radiosity algorithm, SIGGRAPH Comput. Graph. 25 (1991) 197–206.

[13] S. J. Gortler, P. Schröder, M. F. Cohen, P. Hanrahan, Wavelet radiosity, in: Proceedings of the 20th annual conference on Computer graphics and interactive techniques, SIGGRAPH '93, ACM, New York, NY, USA, 1993, pp. 221–230.

[14] S. Laine, H. Saransaari, J. Kontkanen, J. Lehtinen, T. Aila, Incremental instant radiosity for real-time indirect illumination, in: Proceedings of Eurographics Symposium on Rendering 2007, Eurographics Association, 2007, pp. 277–286.

[15] G. Golub, C. V. Loan, Matrix Computations, The Johns Hopkins University Press, 1996.

[16] S. Goreinov, E. Tyrtyshnikov, N. Zamarashkin, A theory of pseudoskeleton approximations, Linear Algebra and its Applications 261 (1997) 1–21.

[17] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, Numerical Recipes: The Art of Scientific Computing, Cambridge University Press, New York, NY, USA, 2007.

[18] I. Sutherland, R. Sproull, R. Schumacker, A characterization of ten hidden-surface algorithms, ACM Computing Surveys 6 (1974) 1–55.

[19] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. V. der Vorst, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition, SIAM, Philadelphia, PA, 1994.

[20] C. L. Lawson, R. J. Hanson, D. R. Kincaid, F. T. Krogh, Basic Linear Algebra Subprograms for Fortran Usage, ACM Transactions on Mathematical Software 5 (3) (1979) 308–323.

[21] NVIDIA, CUDA CUBLAS Library, NVIDIA Corporation, Santa Clara, California, 2008.

[22] S. Barrachina, M. Castillo, F. Igual, R. Mayo, E. Quintana-Orti, Evaluation and tuning of level 3 CUBLAS for graphics processors, in: Workshop on Parallel and Distributed Scientific and Engineering Computing, Miami, EE.UU., 2008, pp. 1–8.

Chapter 4

Inverse Lighting Design for Interior Buildings Integrating Natural and Artificial Sources [1]

In this paper we propose a new method for solving inverse lighting design problems that can include diverse sources such as diffuse roof skylights or artificial light sources. Given a user specification of illumination requirements, our approach provides optimal light source positions as well as optimal shapes for skylight installations in interior architectural models. The well known huge computational effort that involves searching for an optimal solution is tackled by combining two concepts: exploiting the scene coherence to compute global illumination and using a metaheuristic technique for optimization.

Results and analysis show that our method provides both fast and accurate results, making it suitable for lighting design in indoor environments while supporting interactive visualization of global illumination.

Keywords: Lighting Design, Inverse Problem, Global Illumination

4.1 Introduction

Lighting design is an important issue for sustainable buildings which involves both setting natural and artificial lights as well as meeting energy distribution goals. This process requires accurate lighting simulations, which are known to be computationally expensive for a single model and may become prohibitive when directly exploring lighting configurations for several sources. An efficient alternative to this computational task is to use an inverse lighting method. Inverse lighting designates all setting in which, unlike traditional direct calculations, illumination aspects are unknown and must be determined. If we know in advance the desired illumination at some surfaces, an inverse lighting approach can establish the missing parameters (e.g., light position, shape, and power emission, among others).

Providing computationally efficient inverse lighting tools is a challenge. The whole process involves two complex computational tasks: the global illumination simulation

[1] **Published in:** E. Fernández and G. Besuievsky *Inverse lighting design for interior buildings integrating natural and artificial sources*, Computers & Graphics, Volume 36, Issue 8, December 2012, Pages 1096-1108.

and the search for an optimal solution. The first task is crucial for accurate lighting design. Computing the indirect illumination constitutes a simulation of the light transport process through multiple bounces around the environment, and requires dense numerical solutions. For the second task, an optimization process is used to find a solution that fulfills some requirements. The best solution that optimizes a given goal is chosen. The problem is difficult to compute because the solution search space is generally huge.

Inverse lighting is not a new research topic. Several approaches have been proposed based on different motivations, assumptions and optimization strategies. Some of the techniques already explored are genetic algorithms [1], inverse radiosity systems [2, 3] and heuristic methods [4]. Most of these works provide only numerical solutions for each specific kind of environment, showing that an acceptable solution is eventually found. In general, all the techniques are time consuming (requiring minutes or even hours) and are not designed for an interactive design cycle. Another limitation is that no technique to date includes electric sources integrated with natural lighting, as found in real buildings (see Figure 4.1).

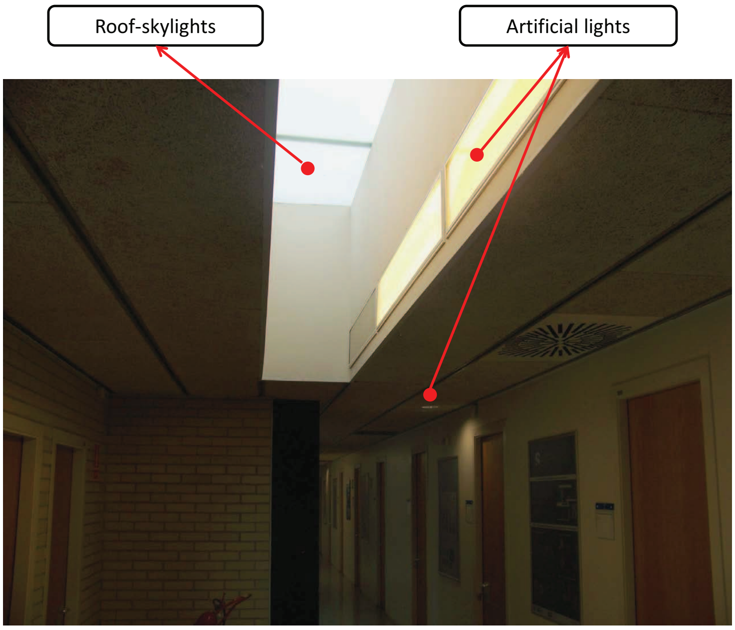


Figure 4.1: Artificial light sources integrated with skylights in a real building.

In this paper we present a novel inverse lighting method that efficiently shortens the execution time and integrates natural and artificial light sources. Given user illumination goals and a set of geometric restrictions and lighting intentions, our system provides optimal light source positions and optimal shapes for skylight installations in interior architectural models. The key element of our method is to exploit the coherence of architectural models to build a global illumination representation that allows designers to explore many solutions efficiently. This functionality is encoded into a low-rank radiosity representation, used as a solver called from an optimization method based on the Variable Neighborhood Search (VNS) method [5].

The main contributions of our approach are as follows:

- We provide both a fast and accurate method for inverse lighting that allows designers to browse solutions in short design-cycle times.

- We provide a flexible lighting design approach in which light sources can be specified at any place in the environment.

- We provide a method that integrates artificial light sources with skylight sources.

The rest of the paper is organized as follows. The following section presents the related work. Section 4.3 provides an introduction to the problem definition. In Section 4.4, the mathematical basis of the problem and of the proposed solution is developed. The main results of five inverse lighting experiments are exposed in Section 4.5. The last section is devoted to the conclusions of this study and future work.

4.2 Related Work

There are two key areas of knowledge related to the inverse lighting problem: numerical optimization and global illumination. On the one hand the huge search space generated by inverse problems must be tackled by heuristic search-based techniques avoiding costly brute-force approaches. On the other hand, lighting simulation is required for each solution found in the search process. In this section we review the main related work on inverse lighting problems.

4.2.1 Inverse Lighting

Direct methods calculate data from a specific configuration of model parameters. In contrast, inverse problems generally infer the properties (or model parameters) of a physical system from observed or desired data. Inverse problems are usually numerically complex and are of interest in a wide range of fields in lighting engineering and lighting design.

One of the first attempts to infer emitter position and shape (parameters), given expressed lighting intentions (desired data), was presented by Schoeneman et al. [6]. These authors introduced the idea of providing an iterative numerical solution to achieve results from a "spray-painting" user-interface description. The interactivity was achieved only for direct illumination. Several works searching for similar goals but driven by different motivations and assumptions were later proposed. A survey of this topic was reported by Patow and Pueyo [7]. Contensin [3] formulated an inverse radiosity method based on a pseudo-inverse analysis of the radiosity matrix. Costa et al. [8] proposed an optimization engine to deal with complex light specifications. Kawai et al. [2] performed the optimization over the intensities and directions of a set of lights as well as surface reflectiveness to best convey the subjective impression of certain scene qualities (e.g., pleasantness or privacy), as expressed by users. Their so-called radioptimization system is a framework that determines optimal setting parameters based on unconstrained optimization techniques used in conjunction with a hierarchical radiosity solver [9]. Castro et al. [4] used heuristic search algorithms combined with linear programming to optimize light positioning with an energy-saving goal. Similar problems were solved with closely related techniques. Pellacini et al. [10] presented an interactive system for computer cinematography that allows users to paint the desired lighting effects in the scene, and a solver provides the corresponding parameters to achieve these effects using a non-linear optimization method. Gibson et al. [11] solved the inverse lighting problem through iteration of virtual light sources in the context of photometric reconstruction data.

Considerably less work has been devoted to optimizing daylight sources, such as openings and skylights. The problem of finding the modeling shapes for lighting goals is more complex for daylight sources than for artificial lights, because in the former case, we are dealing with a dynamic light source. An inverse method for designing opening buildings is presented by Tourre et al. [12] based on the evaluation of potential elements related to an interior lighting intention. In Besuievsky and Tourre [13], the method is extended for accurate sky computation with exterior occlusion. These approaches consider an anisotropic distribution of the light over the potential opening. Our system considers only openings with diffuse filters, such as typical skylight roof installations. However, our method greatly improves both accuracy and speed, also allowing interactive visualization results.

4.2.2 Coherence in Global Illumination

Independent of the strategy used to tackle the inverse lighting problems, the global illumination function must be evaluated many times prior to finding a converged solution. For this purpose an efficient method should be used to compute the solution for each setting of parameter values. However, research on this point has not been emphasized. The use of coherence is crucial to improve the global illumination computation in inverse lighting. Castro et al. [4] improved this known expensive computation by re-using random walk paths. They build and store, in a preprocessing step, an irradiance matrix that allows computing, for each patch of the scene the power contribution of several fixed light points. The main restrictions of this approach are that the authorized light positions must be predefined and that all the light sources are point lights. In our work, we used a low-rank radiosity method (LRR) [14] that allows computers to solve the radiosity equation in real-time with infinite bounces for scenes with dynamic lighting and a relatively small computer storage. With this method no restrictions are required for the light sources: the sources could be anywhere in the scene and area light sources can be allowed, as well. Other approaches, such as the one presented by Kontkanen et al. [15], which uses wavelets to store a pre-computed matrix, could also be used for this purpose. Other fast global illumination techniques, like instant radiosity or VPL-based methods [16], could also be adapted for optimization. However, when multiple bounces are required they are not well suited for exploring many solutions in a short time with changing lighting emission parameters.

4.2.3 Optimization

An optimization problem consists in finding the best solution from all the feasible solutions, which are defined through a set of constraints. For illumination purposes, each constraint is related to a lighting intention for all or part of the scene. Although optimization is a well known topic, there is no computational algorithm that will always provide the global minimum for a general non-linear objective function. Finding the optimal solution by *brute force* is usually not feasible in a reasonable time because of the huge search space of the possible states. Heuristic algorithms avoid visiting the whole search space, by means of designing rules that drive the search towards optimal solutions. There is a large number of heuristic search algorithms in the literature, which can potentially be used to solve lighting problems. Hill climbing [17], beam search [18] and simulated annealing [19] are some of the most commonly used algorithms. Castro et al. [4] explored a wide range of these algorithms to solve optimal economical light positioning. In [20] and [21], the scenes are simplified to rectangular spaces, and the inverse problem is solved through a generalized extremal optimization approach. In our work, we used the VNS metaheuristic for optimization problems. We adapt this technique to our illumination problem and show that good optimization results can be achieved.

4.3 Problem Definition

The main goal of our proposal is to provide a helpful tool for efficient lighting design for both light sources and skylight installations. In this section we describe the lighting design problem formulation and our system proposal.

We consider diffuse environments, that is, all the surfaces have perfectly diffuse materials with no specular component. Area light sources are also considered as Lambertian, with constant emission power. For roof-skylight sources we consider that a diffuse filter with a homogeneous transmittance coefficient is used. This system is one of the most commonly used systems in real buildings. This system provides diffuse and controllable sunlight, which creates a desirable ambient light [22]. These kinds of skylights can also be considered as Lambertian emitters, because they scatter light homogenously. However, the difference between these sources and artificial light sources is that the emission varies over time.

The energy goals are also driven in different ways: for artificial lights, we must aim for the most economical solution, whereas for skylights, in general, we aim to obtain as much light as possible, satisfying geometric restrictions. When designing daylight systems, it is beneficial to consider the solution across all sun positions (time of the day) and climatological conditions (sunny or cloudy). Although this variation is an important issue, for this work, we consider only the average of all of the incoming intensities. However, our mathematical framework can also take into account dynamic daylight changes. A further development of this subject is left for future work.

To specify the design task and find the optimal solution given a 3D interior space to illuminate, first, we must define the variables involved in the optimization, the constraints to satisfy, and the optimization goals.

4.3.1 Optimization Variables

In the lighting design problem, the optimization variable is generally related to the light source specification. The light sources are no longer fixed as in a direct simulation, and optimal parameter values must be found for the emission, position, and shape. In our case, we consider rectangular area light sources with these three variables for optimization (see Figure 4.2).

4.3.2 Constraints

Constraints are the restrictions, either imposed by the light designer or due to constructive building reasons, that must be satisfied in the illumination problem. Two different categories of constraints are considered: geometric restrictions and lighting intentions. For the first category, we consider any imposed restriction that the light source must achieve (such as light size, aspect ratio, in-between distances of sources or symmetries). For example, building constructive constraints regarding the installation

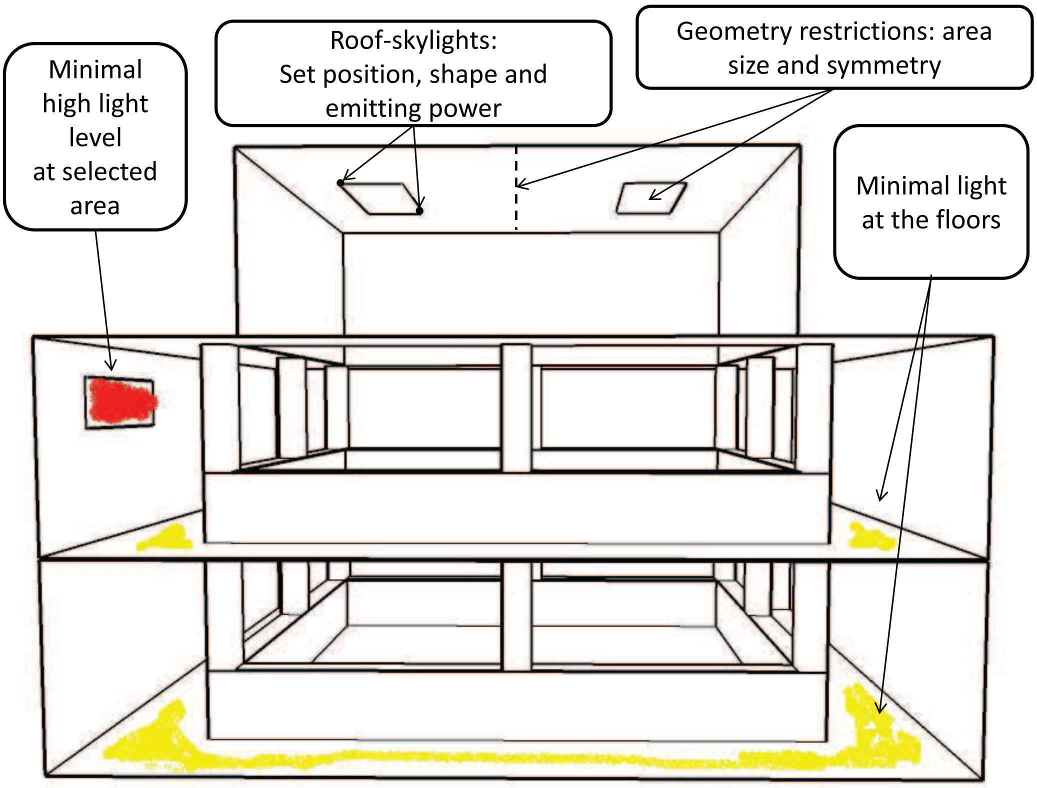

Minimal high light level at selected area
Roof-skylights: Set position, shape and emitting power
Geometry restrictions: area size and symmetry
Minimal light at the floors

Figure 4.2: An example sketch of a design configuration. Skylights with area and symmetry restrictions must be installed in the roof, with the goal of achieving as much light as possible in the scene. Some special places in the wall require a minimum intensity. If a light source is required to achieve all of the restrictions, the system computes the optimal position and emissive power.

of skylight sources in a roof may impose a regular distance between the skylights as well as that they must be aligned along a given axis. For lighting intentions, we consider the constraints that must be satisfied at the surfaces. Intentions are described by inequality constraints that specify the interval of light intensity that each surface is allowed to reflect.

4.3.3 Optimization Goals

Our goal is driven by the intention of optimizing energy consumption. Thus, we used the minimization of energy as an objective function to select the best solutions among the many possible ones. In general, in a lighting-optimization problem such as the one we are describing, there may be an infinite number of possible solutions. We consider energy optimization for our lighting problem in the following way:

- For artificial light sources we will search for the most economical solution, that is, the one that minimizes the power consumption.

- For skylights, where light comes from natural resources, we aim to find the maximal global light-power solution. This way, less artificial light may be needed in the environment.

The pipeline design of our approach is described in Figure 4.3. Given an architectural interior model for lighting design, in which reflectance surfaces are already defined, the user first configures the parameters to find the optimal solution. These parameters specify where to put the light sources as well as the variables to optimize (source position, size and emission), the geometric restrictions and the lighting intention to achieve, and the energy goals to optimize. Before the optimization process begins, we first pre-process the scene to obtain a compact representation of the form-factor matrix of the elements of the scene. For this purpose, we use a low-rank radiosity method. The optimization method works by getting the compact representation found and a design configuration, to obtain a fast result that can be visualized interactively. Regarding the resulting values, the designer can modify the setting parameters to search for a new solution.

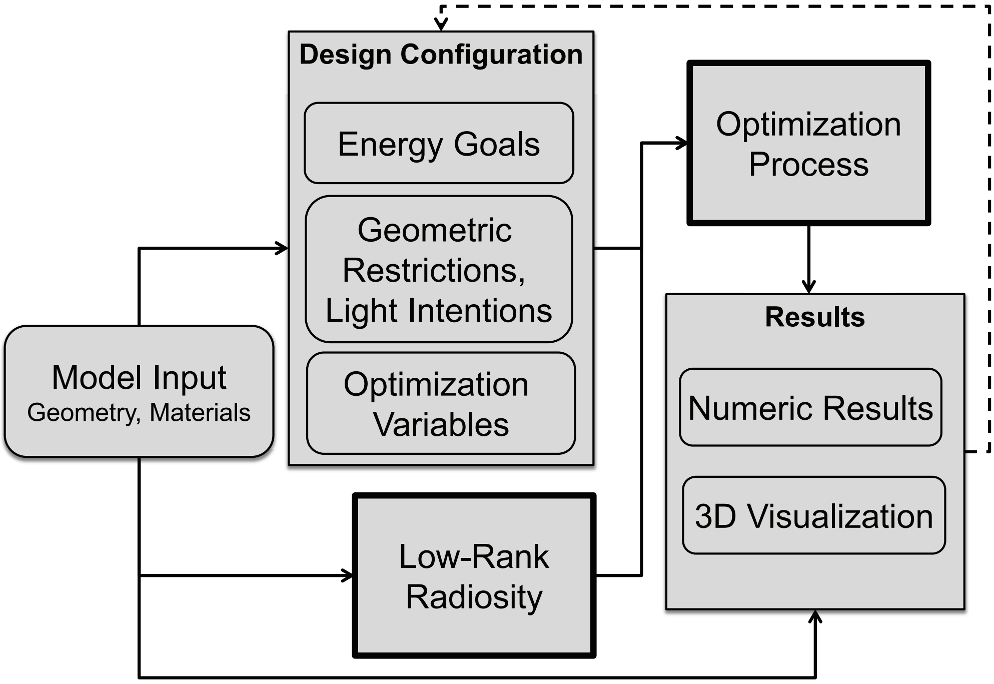


Figure 4.3: Pipeline system.

4.4 Mathematic Foundation

Our solution proposal is based on a two-step approach. In this section we describe details of both steps.

4.4.1 Problem Formulation

We explain here how to model the optimization problem as an unconstrained problem. Further development of this topic was reported by Kawai et al. [2] and Luenberger and

Ye [23].

The goal of the problem can be formulated as the optimization of a function $f(x)$ that usually represents the power emitted by the artificial light sources or the total light power of the scene. There are also constraints related to geometric restrictions and lighting intentions. The general formulation of the optimization problem is expressed in Equation 4.1 where S is the set of feasible values for x.

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } x \in S \end{aligned} \tag{4.1}$$

S can be formulated as a set of functional constraints, $S = \{x : c_i(x) \leq 0, i = 1 \ldots p\}$, where each $c_i(x) \leq 0$ is an inequation that defines a constraint, and p is the number of constraints. This set can be transformed into a continuous penalty function $P(x)$, which "turns on" when a constraint is not satisfied, i.e., $P(x) = 0$ when $x \in S$, and $P(x) > 0$ when $x \notin S$. The main idea behind the penalty method is the conversion of a constrained problem (Equation 4.1) into an unconstrained problem of the form:

$$\text{minimize } f(x) + P(x) \tag{4.2}$$

The penalty function used in this work has the form expressed in Equation 4.3, where *pow* is an even positive integer and W_i is a positive weight associated with each constraint.

$$P(x) = \sum_{i=1}^{p} \{W_i \times (\max[0, c_i(x)])^{pow}\} \tag{4.3}$$

The weights W_i must be adjusted to ensure that the solution of Equation 4.2 will approach the feasible region S.

In lighting design problems, the evaluation of the optimization goal $f(x)$ and some constraints (which we call $c_i^{Rad}(x)$) requires a radiosity evaluation of the scene. However, there are other constraints that do not depend on radiosity values ($c_i^{NoRad}(x)$), such as constraints that control the shape and location of the light sources. To avoid unnecessary radiosity evaluations, it must first be checked that $c_i^{NoRad}(x) \leq 0, \forall i$. So, the optimization problem expressed in Equation 4.2 can now be formulated as shown in Equation 4.4, where $P^{Rad}(x)$ is a penalty function that includes only the $c_i^{Rad}(x)$ constraints.

$$\begin{aligned} &\text{minimize } f(x) + P^{Rad}(x), \\ &\text{subject to } \{c_i^{NoRad}(x) \leq 0\} \end{aligned} \tag{4.4}$$

4.4.2 Low-Rank Radiosity

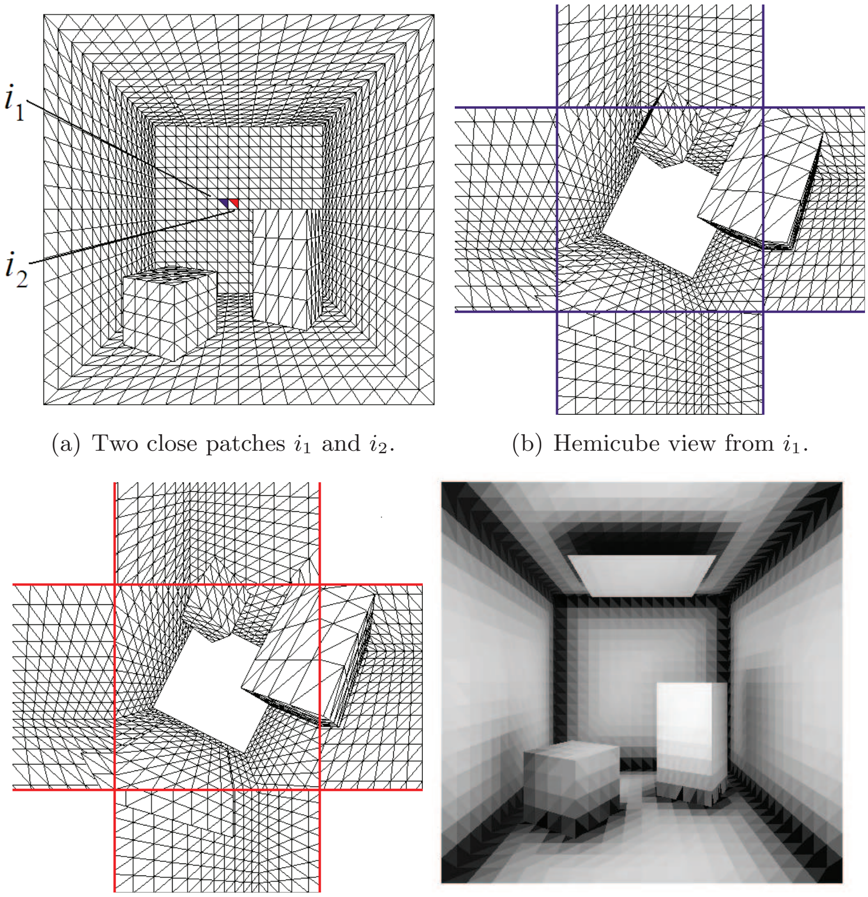
In this section we present a method to efficiently assess the radiosity equation, which is used to solve the optimization problem.

In the discrete radiosity problem, the radiosity of the scene is computed by solving the linear system shown in Equation 4.5.

$$(\mathbf{I} - \mathbf{RF})B = E \tag{4.5}$$

In this equation, $\mathbf{I}$ is the identity matrix with dimension $n \times n$ (n is the number of patches), $\mathbf{R}$ is a diagonal matrix that stores the reflectivity of the patches, $\mathbf{F}$ is the matrix with the form factors F_{ij} (which indicate the fraction of light reflected by the patch i that arrives to the patch j), B is a vector with the radiosity value of each patch, and E is the emission vector of the scene.

It is very likely that the **RF** matrix of Equation 4.5 has a low numeric rank because each row $(\mathbf{RF})_i$ is computed based on the scene view from the same element i. As most close pairs of patches have a very similar view of the scene (see Figure 4.4), **RF** has many similar rows resulting in a reduction of the numerical rank. The rank reduction of **RF** matrices allows for the matrices to be approximated by the product of two matrices $\mathbf{U}_k \mathbf{V}_k^T$, both with dimension $n \times k$ ($n \gg k$), without losing relevant information about the scene.



(a) Two close patches i_1 and i_2.

(b) Hemicube view from i_1.

(c) Hemicube view from i_2.

(d) Spatial coherence in the scene.

Figure 4.4: Close patches present similar views of the scene and generate similar rows in **RF** (see [14]). In (d), the coherence of each patch is measured (a darker color indicates lower coherence).

Low-Rank Approximation

The product $\mathbf{U}_k \mathbf{V}_k^T$ generates a matrix with dimensions $n \times n$ and rank k. The memory requirement of $\mathbf{U}_k$ and $\mathbf{V}_k^T$ is $O(nk)$, significantly less than $O(n^2)$ that would be required to store the matrix **RF**. This reduction is especially appreciated when $n \gg k$, due to the spatial coherence of the scene. In this way, the memory saving allows the information to be stored in the main processor memory, facilitating the work with large scenes.

Replacing **RF** by the low-rank approximation $\mathbf{U}_k\mathbf{V}_k^T$ in Equation 4.5 allows us to obtain the row-rank radiosity (LRR) equation 4.6. The unknown is no longer B but rather an approximation $\widetilde{B}$ of the radiosity.

$$\left(\mathbf{I} - \mathbf{U}_k\mathbf{V}_k^T\right)\widetilde{B} = E \tag{4.6}$$

The matrix $\left(\mathbf{I} - \mathbf{U}_k\mathbf{V}_k^T\right)$ is now invertible using the Sherman-Morrison-Woodbury formula [24].

$$\widetilde{B} = E + \mathbf{U}_k\left(\left(\mathbf{I}_k - \mathbf{V}_k^T\mathbf{U}_k\right)^{-1}\left(\mathbf{V}_k^T E\right)\right) \tag{4.7}$$

To find $\widetilde{B}$ using Equation 4.7, $O(nk^2)$ operations and $O(nk)$ memory are required. For scenes with static geometry and dynamic lighting (i.e., only the independent term E varies in Equations 4.5 to 4.7), part of the computational effort can be pre-computed and stored. Equation 4.7 is now written as follows:

$$\widetilde{B} = E - \mathbf{Y}_k\left(\mathbf{V}_k^T E\right) \quad , \quad \text{where } \mathbf{Y}_k = -\mathbf{U}_k\left(\mathbf{I}_k - \mathbf{V}_k^T\mathbf{U}_k\right)^{-1} \tag{4.8}$$

where $\mathbf{Y}_k$ is a $n \times k$ matrix that is computed once (as well as $\mathbf{U}_k$ and $\mathbf{V}_k$). After $\mathbf{Y}_k$ is found, the radiosity calculation has complexity $O(nk)$. This result is very useful to develop a new methodology for inverse lighting problems with static geometry [14]. Fernández et al. [25] combined Equation 4.8 with GPU architectures to solve the radiosity problem in real time, hundreds of times per second in scenes with tens of thousands of elements.

Another important result is that the LRR methodology is a direct method. Equation 4.8 can also be formulated as $\widetilde{B} = \mathbf{G}E$, so the matrix $\mathbf{G}{=}\mathbf{I}{-}\mathbf{Y}_k\mathbf{V}_k^T$ is a global operator that manages the infinite bounces of light in a single operation. The matrix $\mathbf{G}$ allows finding the radiosity values for any set of p patches ($p < n$) without solving the radiosity for all the patches (as in iterative methodologies like hierarchical radiosity [26]). Only p rows of $\mathbf{G}$ are needed to find the radiosity values (i.e., $\mathbf{G}_p{=}\ \mathbf{I}_p{-}\mathbf{Y}_{k,p}\mathbf{V}_k^T$, where $\mathbf{Y}_{k,p}$ consists of p rows of $\mathbf{Y}_k$ related to the p patches, and the same also applies to $\mathbf{I}_p$). Therefore, the radiosity calculation goes down to $O(n + pk)$ operations.

Using the LRR methodology, the global operator $\mathbf{G}$ is computed while avoiding the iterative utilization of Neumann series, as is employed by Kontkanen et al. [15] and Lehtinen et al. [27], taking advantage of the fact that **RF** is a low-rank matrix.

Direct Radiosity Method in Inverse Lighting Problems

Any optimization algorithm used to solve inverse lighting problems finds thousands of feasible solutions to identify the one that meets all of the constraints and optimizes the objective function. For most of the optimization time, the algorithm is focused on the radiosity calculation; consequently, the speed of the optimization algorithm is mainly related to the speed of the radiosity solver. Our proposal is based on the use of the LRR method as a fast and direct solver.

Besides solving the radiosity method faster when only the radiosity of p patches are needed, the LRR method also can be used to speed up the optimization process of complete scenes with good spatial coherence. Because close patches have similar radiosity properties, the radiosity values of a representative set of patches provide enough scene information for optimization purposes.

Following this approach, an optimal solution found using a subset of patches could be the starting point of another optimization process with a greater set of patches. After the execution of several optimization processes, using an increasing sequence of sets of patches, this multilevel strategy could terminate after considering all the scene patches. Given the right conditions, it is expected that a multilevel strategy like this would converge much faster than only one optimization process with all the patches.

4.4.3 Heuristic Search and Optimization

We developed an algorithm to solve the inverse lighting problem based on the VNS metaheuristic [5]. This methodology is based on the idea of successive explorations of a set of neighborhoods $(N_1(x), N_2(x), \ldots, N_k(x))$. The method explores, either at random or systematically, a set of neighborhoods to obtain different local optima. Each neighborhood has its own local optimum, and it is expected that the global optimum is the same as a local optimum for a given neighborhood (see Figure 4.5).

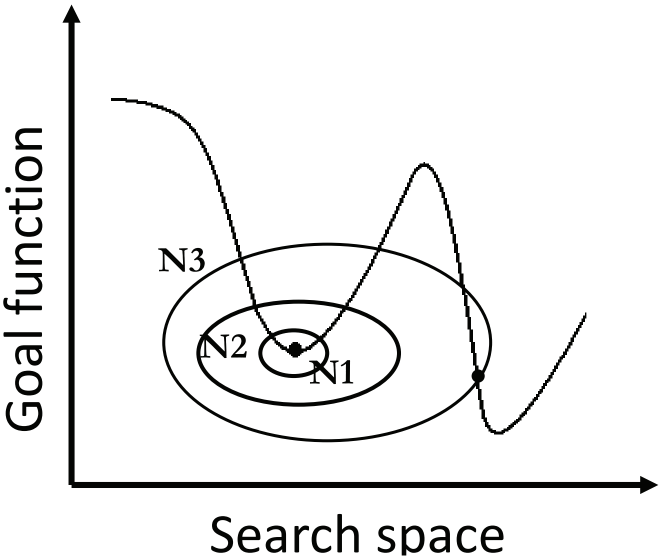


Figure 4.5: Variable neighborhood search using three neighborhoods (N1, N2 and N3). When a local optimum is found (as in N3), the search methodology continues at this point.

The set of neighborhoods is usually nested (i.e., $N_1(x) \subset N_2(x) \subset \ldots \subset N_k(x)$). This strategy will be effective if the different neighborhoods used are complementary in the sense that a local optimum for a neighborhood N_i will not be a local optimum in the neighborhood N_j (when $N_i \subset N_j$).

We used the general pseudocode of the VNS strategy to develop an algorithm for our lighting problem (see Algorithm 3). The algorithm attempts to find x such that the function $f(x) + P^{Rad}(x)$ is minimized, and that all of the $c_i^{NoRad}(x) \leq 0$ constraints are satisfied.

In our implementation, each neighborhood N_k has two parameters, v and r, that define the neighborhood's scope. The parameter v establishes the number of optimiza-

Algorithm 3: Radiosity VNS.

```
Input: Set of neighborhood structures $N_k$ for $k = 1, ..., k_{max}$
Output: The most optimal solution found
$x = x_0$;                                    /* Generate the initial solution */
while not Stopping Criteria do
    $k = 1$;
    while $k <= k_{max}$ do
        Find a better neighbor $x'$ of $x$ in $N_k(x)$ by local-search;
        if $\exists\, x' | \left(c_i^{NoRad}(x') \leq 0\ \forall i\right)$ and $\left(f(x') + P^{Rad}(x') < f(x) + P^{Rad}(x)\right)$ then
            $x = x'$;
            $k = 1$;
        end
        otherwise
            $k = k + 1$;
        endsw
    end
end
```

tion variables that are modified at the same time, and r defines a normalized interval of variation. For instance, $v = 1$ and $r = 0.1$ means that, in each trial of the local search, a variable x_i of x is randomly selected, and the variable's value is substituted by another value chosen from the interval $x_i \pm 0.1(x_i^M - x_i^m)$, where x_i^M and x_i^m are the maximum and minimum values allowed for x_i, respectively. In one step, a variable related to the position of the light is modified; in the next step, the power of the light emission is modified; and so on. Following the parametric neighborhood definition, a grid of neighborhoods can be defined, considering all the combinations of values $v = 1, 2, \ldots, nvar$; $r = 0.1, 0.2, \ldots, 1$. Then, the set of neighborhoods used by the algorithm is a selection -and eventually all- of the neighborhoods defined in the grid.

Using the problem formulation expressed in Equation 4.4, the local-search algorithm first controls that each tested neighbor x' fulfills $c_i^{NoRad}(x') \leq 0\ \forall i$, before the radiosity is evaluated and the value of the goal (i.e., $f(x) + P^{(Rad)}(x)$) is checked. This prior control avoids unnecessary radiosity calculations.

Different stopping criteria can be used, such as the time required to obtain a given target solution, the distance to the target solution or the number of iterations.

Multiobjective Optimization Problem

Many inverse lighting problems can be defined as an optimization problem with multiple objectives such as the minimization of artificial power consumption and maximization of natural light influence. The mathematical definition of a multi-objective optimization problem (MOP) can be stated as follows:

$$\begin{aligned} \text{minimize } F(x) &= (f_1(x), f_2(x), \ldots, f_n(x)) \\ \text{subject to } \quad x &\in S \end{aligned} \tag{4.9}$$

Multi-objective optimization problems and techniques are analyzed by Deb [28] and in Coello et al. [29]. The optimal solution for MOP is not a single solution as for mono-objective optimization problems, but a set of "non-dominated" solutions, that is, solutions for which it is not possible to improve one objective without impairing at least one other objective. This set of optimal solutions is called a Pareto front.

Many methodologies used to solve multi-objective optimization problems are founded on population-based metaheuristics, such as evolutionary algorithms [28, 29]. Other methodologies are based on scalar approaches [30], which transform a multi-objective optimization problem into a set of mono-objective problems. This approach allows using a single solution based on metaheuristics as the VNS. In our work, we implemented the ϵ-constraint method as a scalar approach. The ϵ-constraint method consists in optimizing one selected objective f_k subject to constraints on the other objectives f_j, $j{\in}[1\ldots n]$, $j{\neq}k$ of a MOP (see Equation 4.10). Hence, some objectives are transformed into constraints.

$$\begin{aligned} \text{minimize } f_k(x) \\ \text{subject to } x \in S \\ f_j(x) \leq \epsilon_j, j = 1\ldots n, j \neq k \end{aligned} \tag{4.10}$$

In this equation, each ϵ_j is an upper bound for the objectives. The ϵ-constraint method is run many times, changing the values of ϵ_j. This method generates a set of non-dominated solutions with good diversity properties. It is necessary to know a priori the suitable interval for each ϵ_j value.

4.5 Analysis and Results

To evaluate the fundamental aspects of our method, we built a set of five different experiments. The goal of the first experiment is to evaluate the convergence properties of the proposed algorithm. For this reason, a scenario is built in which the method must converge to a known solution. In another experiment, we check how the spatial coherence of the radiosity solution can help to decrease the convergence time. Next, we test how the algorithm addresses the fulfillment of the constraints (geometric restrictions and lighting intentions). Finally, the scenario of problems with multiple solutions is considered for two cases: a single objective problem with many practical solutions and a MOP study case. These experiments were designed to demonstrate the good properties of the method and to identify further areas for development.

For the daylight intensity, we used average data, not taking into account the date and time of the day or the variation of light intensity throughout the day. Thus, we present tests scenes as architectural prototype models, and we will develop adjustments for real data in a future work.

All the simulations were performed in a Matlab environment on a notebook computer (Intel Core i7-2670QM 2.2 Ghz processor with Turbo Boost up to 3.1 Ghz and 4 GB DDR3 memory).

4.5.1 First Experiment: Convergence

Scene: Patio building. Size $(n{\times}k)$: $24,128{\times}1,508$. The pre-computation of $\mathbf{Y}_k$ and $\mathbf{V}_k$ takes about 12 minutes.

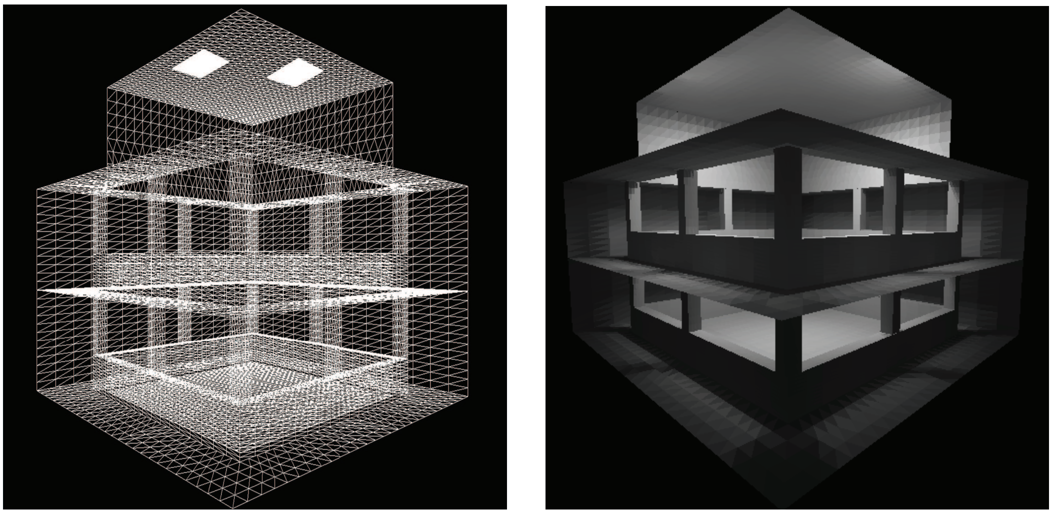
Goal: Find the shape and position of two area light sources such that the reflected radiosity $C(E)$ for emission E is close to a desired reflected radiosity C_{obj}.

$$C(E) = B - E = -\mathbf{Y}_k\mathbf{V}_k^T E$$
$$C_{obj} = B_{obj} - E_{obj} = -\mathbf{Y}_k\mathbf{V}_k^T E_{obj}.$$

Number of constraints: 8
The light emitters must lie in the ceiling.

Variables: 8
Four 2D coordinates that delimit the location of the light sources.

In this experiment, we test the capability of our method to approximate the initial conditions of the scene from a previously known radiosity solution. To build a known solution, we first choose two rectangular light sources at the ceiling of the patio (Figure 4.6(a)), which define E_{obj}. Then, the diffuse reflection of the scene C_{obj} is computed using the LRR method. (Figure 4.6(b)).



(a) Two emitters in the ceiling.

(b) Reflected radiosity in patio.

Figure 4.6: Patio scene.

The algorithm attempts to find the location of two rectangular emitters E in the ceiling, such that the resulting radiosity reflected $C(E)$ minimizes the normalized Euclidean distance (Equation 4.11).

$$\min_E e(E) = \frac{||C(E) - C_{obj}||_2}{||C_{obj}||_2} \tag{4.11}$$

The space of possible solutions is composed of the set of lists of four roof patches that define the diagonal extremes of two skylights. In this example, the roof is composed of 1024 quads, and therefore, the possible solutions can be estimated as $C_4^{1024}/4 \approx 11.4 \times 10^9$ (the division by 4 is due to symmetry considerations). A brute-force exploration of all the solution space is not feasible on desktop computers.

There is a biunivocal relation between the radiosity (B or C) and the emission (E) caused by the existing linear relations expressed in Equations 4.5 and 4.6. If the goal of the problem is to find C close to C_{obj} (and the problem is not ill-conditioned), then E is usually close to E_{obj}. Figure 4.7 shows different runs of the VNS algorithm. Each image is a 2D projection of the ceiling, where E_{obj} is presented in green, and E is presented

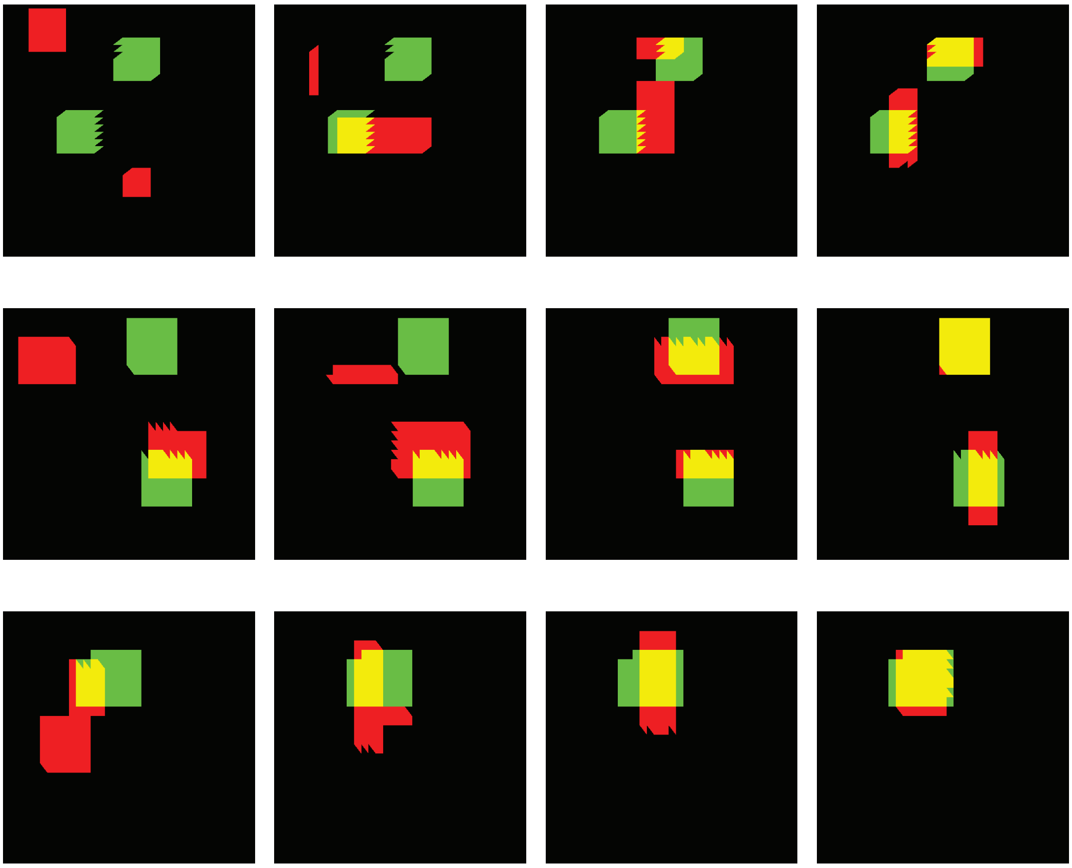


Figure 4.7: Visualization of the emitters in the ceiling in three runs of experiment 1. Each row is a different run (from left to right). In each run, E (in red) is approaching to E_{obj} (in green).

in red. The leftmost column shows the starting configuration for each run, and the rightmost column shows the final result. The center columns show the intermediate results.

Figure 4.8 shows, from left to right, another view of a VNS run (the first row of Figure 4.7). The top row presents the convergence of the reflected radiosity $C(E)$ to the desired radiosity C_{obj} (Figure 4.6 (b)). The bottom row displays the radiosity errors of each $C(E)$ found. Those patches coloured green comply with $C_{obj}{>}C(E)$, and those colored red satisfy that $C_{obj}{<}C(E)$. The gray patches meet $C_{obj}{\approx}C(E)$.

4.5.2 Second Experiment: A Multilevel Method

Considering that the patio scene has good spatial coherence, it can be assumed that the distance $e_p(E)$ is a good approximation of $e(E)$ (Equation 4.12), where p is the number of representative patches, $C_p(E) = -\mathbf{Y}_{k,p}\mathbf{V}_k^T E$, and $C_{p,obj} = -\mathbf{Y}_{k,p}\mathbf{V}_k^T E_{obj}$.

$$\min_E e_p(E) = \frac{||C_p(E) - C_{p,obj}||_2}{||C_{p,obj}||_2} \tag{4.12}$$

For the patio scene, 50,000 radiosity evaluations $C(E)$ take approximately 100 minutes ($O(nk)$), and the computation of $e_p(E)$ with $p{=}\frac{n}{16}$, takes only 11 minutes ($O(n{+}pk)$).

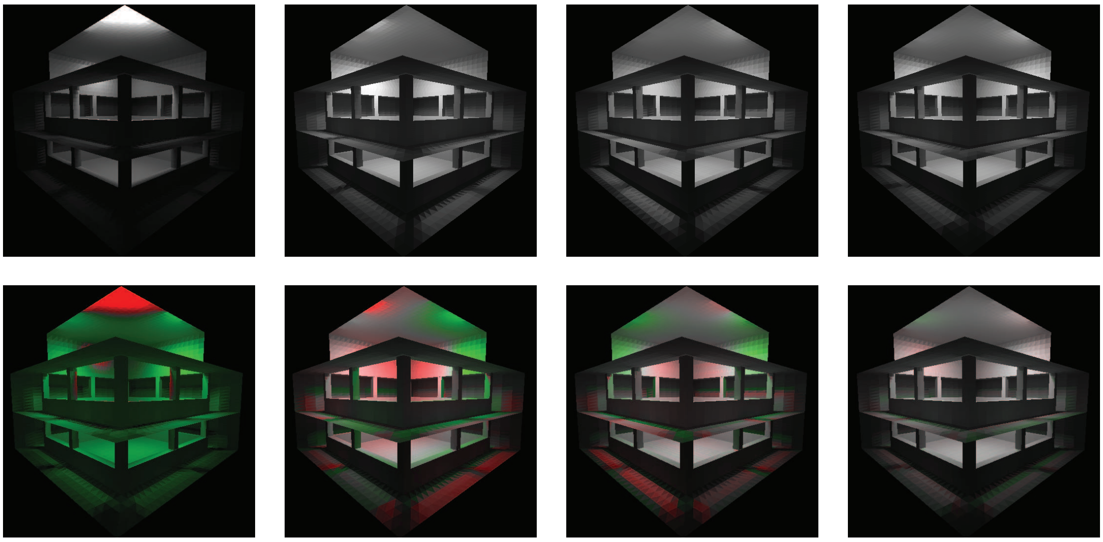


Figure 4.8: Each column corresponds to a stage in the optimization process (left-to-right). Top: the reflected radiosities ($C(E)$). Bottom: The colors show the difference between $C(E)$ and C_{obj}. Green means $C_{obj}{>}C(E)$, and red means $C_{obj}{<}C(E)$.

Given the above results, two sets of experiments were conducted to demonstrate the potential of using spatial coherence properties. In the first experiment, the stop condition of the VNS algorithm is set to $e_p(E_i){<}0.03$, with $p{=}n/16$. After the algorithm stops, the real distance $e(E)$ is calculated. A total of 30 runs were conducted to allow a statistical comparison between both of the distances. The statistical analyses reveal that the ratio $r(E_i){=}e(E_i)/e_p(E_i)$ has a mean $\mu{=}1.24$ and a standard deviation $\sigma{=}0.29$. Assuming that r has a normal distribution, then $\mu{\pm}2\sigma$ is the 95.5% confidence interval. Therefore, it can be concluded that, if $e_p(E){<}0.03$, then $e(E){<}0.03(\mu{+}2\sigma){=}$ $0.03(1.24{+}2 \cdot 0.29){=}0.055$ with a probability of 95.5%.

Figure 4.9 shows the convergence path for five runs of the algorithm, each with a different starting seed. The values of the objective function are shown in blue. The stop conditions of the algorithm are a distance threshold ($e_p(E_i) < 0.03$; the red dotted line in Figure 4.9) and the number of radiosity evaluations ($50,000$). For this scene, we can conclude that the algorithm took, on average, 340 seconds to reach the solution, because the threshold is reached after an average of $26,000$ radiosity evaluations.

The above results motivate a second set of experiments consisting in the implementation of a multilevel method with the aim of accelerating the optimization process. The multilevel algorithm used involves three consecutive optimization processes, where each process is related to a greater set of patches. In addition to all the scene patches, we also consider two subsets of scene patches with $p = n/16$ and $p = n/64$ elements. First, we solve the optimization problem with the smallest set of patches ($p = n/64$). The solution found becomes the starting point for a second run of the optimization algorithm, this time with a mid-size set of patches ($p = n/16$). Finally, a last run including all of the scene patches is performed.

Comparing the result previously obtained with all the scene patches (100 minutes), we now take only 470 seconds for the same total number of radiosity evaluations and the same quality results. Because each optimization algorithm stops when e_p is lower than a certain threshold, the statistical results for the patio scene show that the sequence of op-

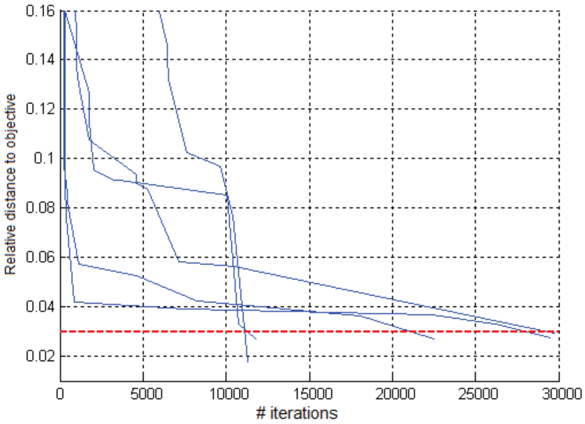


Figure 4.9: Convergence of VNS algorithm for different runs.

timization problems stops after a mean of 28, 500 iterations and 310 seconds. Figure 4.10 shows a sequence of three consecutive runs of the algorithm, following the above scheme. In this example, a solution that considers all the patches is found in 350 seconds. A new processing run starts when the previous run has an error e_p lower than 0.03.

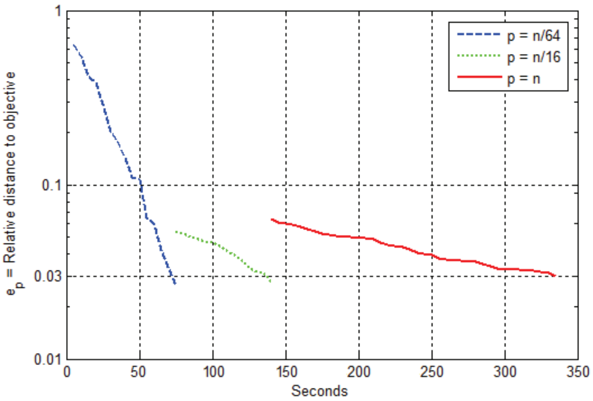


Figure 4.10: Convergence of VNS algorithm (multilevel method).

4.5.3 Third Experiment: Many Constraints

Scene: Patio building. Size ($n \times k$): $24,128 \times 1,508$.

Goal: Maximize the light power in the scene: $\max \sum_{\forall i} C_i A_i$.

Constraints:
- One rectangular light source in the ceiling.
- Reflected radiosity $\forall i$: $C_{min} \leq C_i \leq C_{max}$.
- Bounded area: $A_{min} \leq$ Area of skylight $\leq A_{max}$.

Variables: 4

Two 2D coordinates that define the light shape and position.

This experiment studies the convergence of the VNS algorithm in maximizing the total power reflected from the scene ($\max \sum_{\forall i} C_i A_i$), where C_i and A_i are the radiosity reflected and the area of patch i, respectively. A rectangular skylight must be installed in the ceiling. As explicit constraints, the radiosity value of each patch and the area value of the skylight should be placed within defined ranges.

Figure 4.11 shows the evolution of the algorithm. The red dotted line shows the value of the penalty function $P^{Rad}(x)$ in each iteration. As explained in Section 4, when all the constraints are fulfilled, the penalty function value is 0. The continuous blue line shows the total light power reflected by the scene.

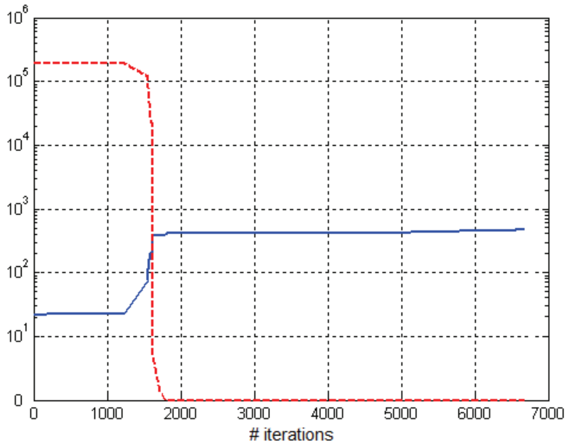


Figure 4.11: Convergence to feasible solutions.

Figure 4.12 shows the evolution of the same problem when a constraint is modified. Now $C_{min} \leq C_i \leq C_{max}/4, \forall i$. Given this configuration of constraints, there is no feasible solution set, and, as shown in the plot, the algorithm also fails to find a feasible solution. Therefore, the designer must change one or more constraints to find a solution that meets all of them.

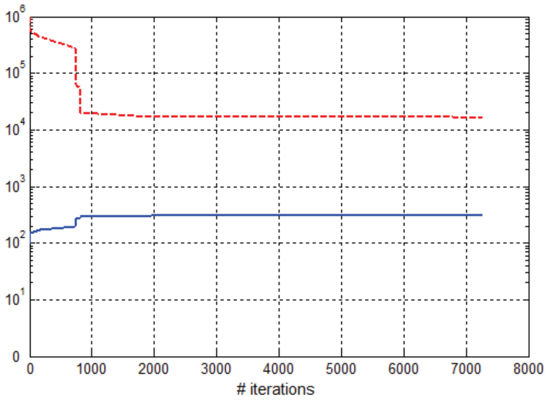


Figure 4.12: Unfeasible convergence.

It is important to determine whether the constraints pursued are possible to fulfill. As shown in the pipeline system (Figure 4.3), the analysis of the results after the optimization process can be used to redefine the specification of the problem with new lighting intentions.

4.5.4 Fourth Experiment: Many Solutions

Scene: Patio building. Size ($n \times k$): $24,128 \times 1,508$.

Goal: Maximize the light power in the scene: $\max \sum_{\forall i} C_i A_i$

Constraints:

- One rectangular light source in the ceiling.
- Reflected radiosity $\forall i$: $C_{min} \leq C_i \leq C_{max}$.
- Bounded area: $A_{min} \leq$ Area of skylight $\leq A_{max}$.
- Filter value $0 \leq f \leq 1$.

Variables: 5

- Two 2D coordinates that define the light shape and position.
- A variable that defines the filter value.

This experiment is similar to the previous one, with an added filter value f to control the skylight emittance ($f \times e_{max}$, where e_{max} is the maximum skylight emittance).

Each run of the algorithm converges to a different solution. In Figure 4.13, the blue asterisks show the pairs of skylight area and filter values found for several runs. The red dotted line shows a hyperbolic curve that almost fits the solutions.

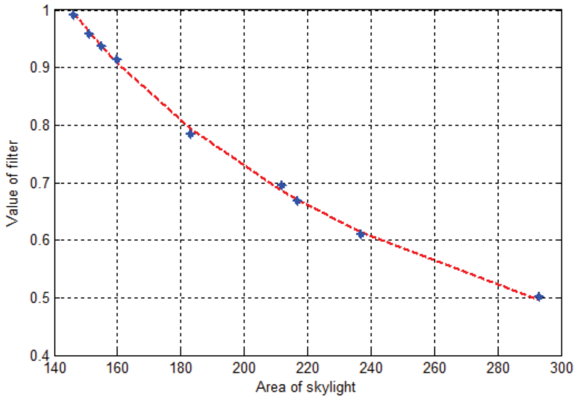


Figure 4.13: Relation between skylight area and the filter value.

The total light power produced by the skylight is almost the same in all of the solutions (skylight-area×filter is almost constant), which intuitively means that the smaller the area of the skylight, the more light must pass through the filter. In this case, the designer must manually identify the best solution from the set of optimal solutions provided by the algorithm.

4.5.5 Fifth Experiment: A Case Study with MOP

First problem:

Scene: Corridor. Size($n \times k$): $16,736 \times 1,046$. The pre-computation of $\mathbf{Y}_k$ and $\mathbf{V}_k$ takes about 8 minutes.

Goal: Maximize the natural light power: max $\sum_{\forall i} C_i A_i$

Constraints:
- Skylights delimited into areas S_1, S_2 and S_3.
- Skylights in S_1 and S_3 must be symmetric.
- The skylight in S_2 must be centered.
- Area of the skylights $\leq A_{max}$.
- Aspect ratio of the skylights ≤ 4.
- Radiosity reflected by the panels $\geq B_{min}$.

Variables: 6

- Two 2D coordinates that defines in S_1 the skylight shape and position of the skylight (the skylight in S_3 is symmetric).
- One 2D coordinate that delimits the position of the skylight in S_2 (centered skylight).

After 10^5 iterations it was not possible to find a feasible solution. The amount of natural light passing through the skylights does not fulfill the constraint for the panels. The designer may relax some of the constraints (i.e., increasing the area of skylights or decreasing the radiosity in the panels) to find feasible solutions. Another course of action would be the addition of small artificial lights near each panel to satisfy all of the predefined constraints (Figure 4.14). In this case, the problem is transformed into a MOP because the minimization of the artificial light power must also be considered.
Second problem (MOP):

Goal 1: Maximize the light power in the scene: max $\sum_{\forall i} C_i A_i$

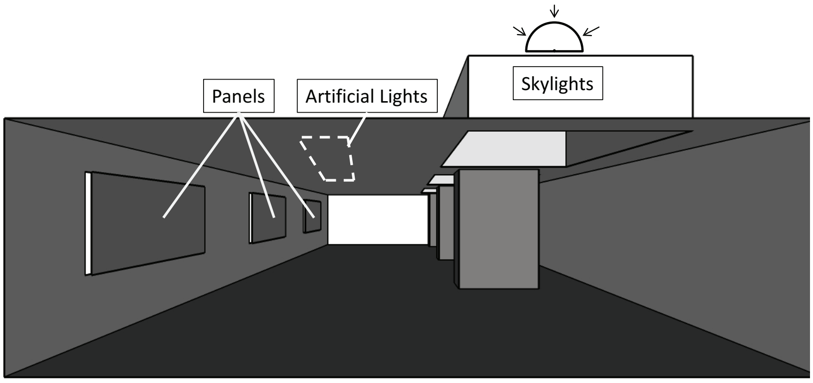
Goal 2: Minimize the artificial light power: min $\sum_{\forall i} E_i A_i$

Constraints:
- The same set included in the first problem.
- Area light sources delimited into A_1, A_2 and A_3.
- Area light sources in A_1 and A_3 must be symmetric.
- The light area source in A_2 must be centered.
- Aspect ratio of each emitter ≤ 10.

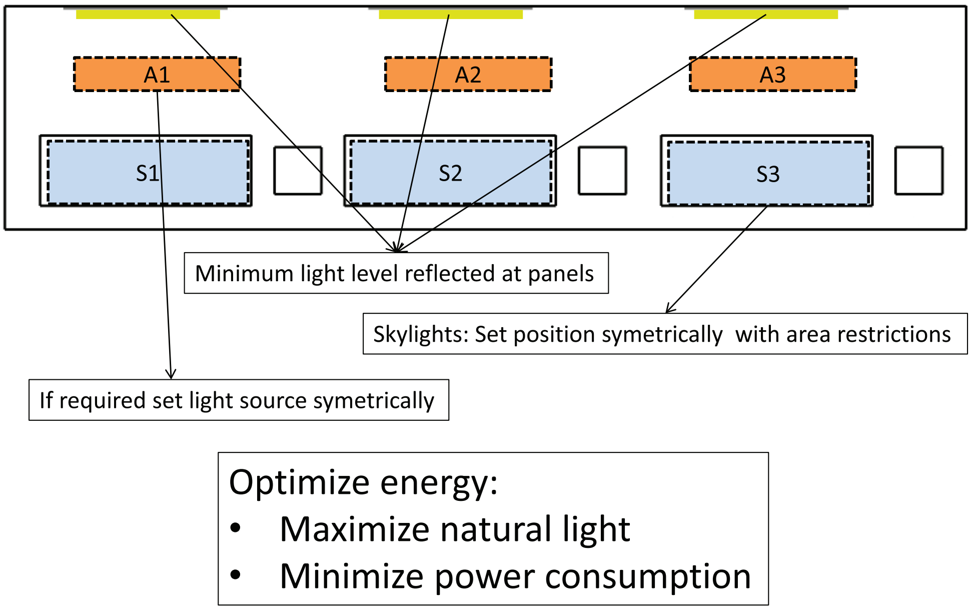
Variables: 12
The same variables (6) used in the previous problem and 6 more related to the artificial light sources.

A multi-objective optimization process must be used to find a Pareto front of non-dominated solutions. In Figure 4.15, the blue '+' are a set of feasible solutions found when running the ϵ-constraint method, and the red '$\circ$' set is their associated Pareto front. The ϵ-constraint method minimizes the artificial light power (Goal 2) when all of the constraints are satisfied and when the total light power is greater than ϵ (a new constraint defined with Goal 1). The variable ϵ takes all the even values between 170 and $260W$, with $20,000$ radiosity evaluations each.

Besides the use of a method, such as the ϵ-constraint method, to find the Pareto front, another approach consists in solving the optimization problem using a procedure



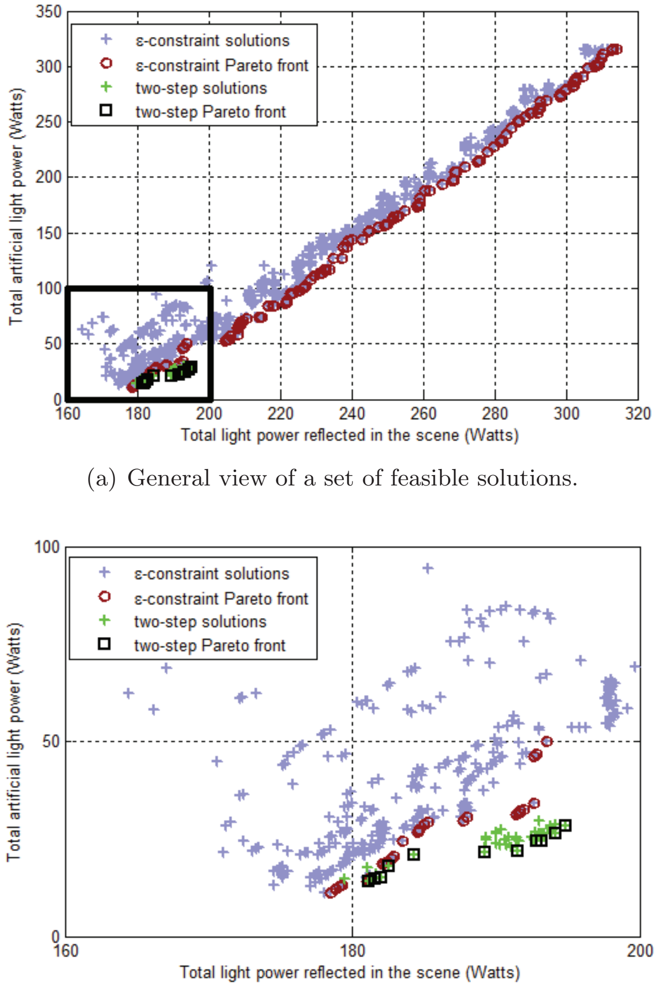
(a) Perspective view of the corridor scene.


A1
A2
A3
S1
S2
S3
Minimum light level reflected at panels
Skylights: Set position symetrically with area restrictions
If required set light source symetrically
Optimize energy:
• Maximize natural light
• Minimize power consumption

(b) Configuration scheme.

Figure 4.14: Corridor scene as a MOP.

that follows the designer's intentions. For instance, if the designer wants to maximize the natural light that comes through the skylights, and the use of artificial light is only used to complete the illumination needed in the panels, then a good approach consists of a two-step process. First, an optimization problem is solved involving only the variables related to the skylights, ignoring all the constraints associated with the panels and also ignoring all the variables and constraints related to the artificial emitters. The goal of this problem is to maximize the total power of the natural light reflected in the scene (see Figure 4.16 (a)). In a second step, another optimization problem is solved, ignoring all the variables and constraints related to the skylights, involving only the variables related to the artificial emitters and considering the panel's constraints. The goal of this problem is to minimize the light power of the artificial emitters (see Figure 4.16 (b)).



(a) General view of a set of feasible solutions.

(b) Detail view of the area bounded in (a).

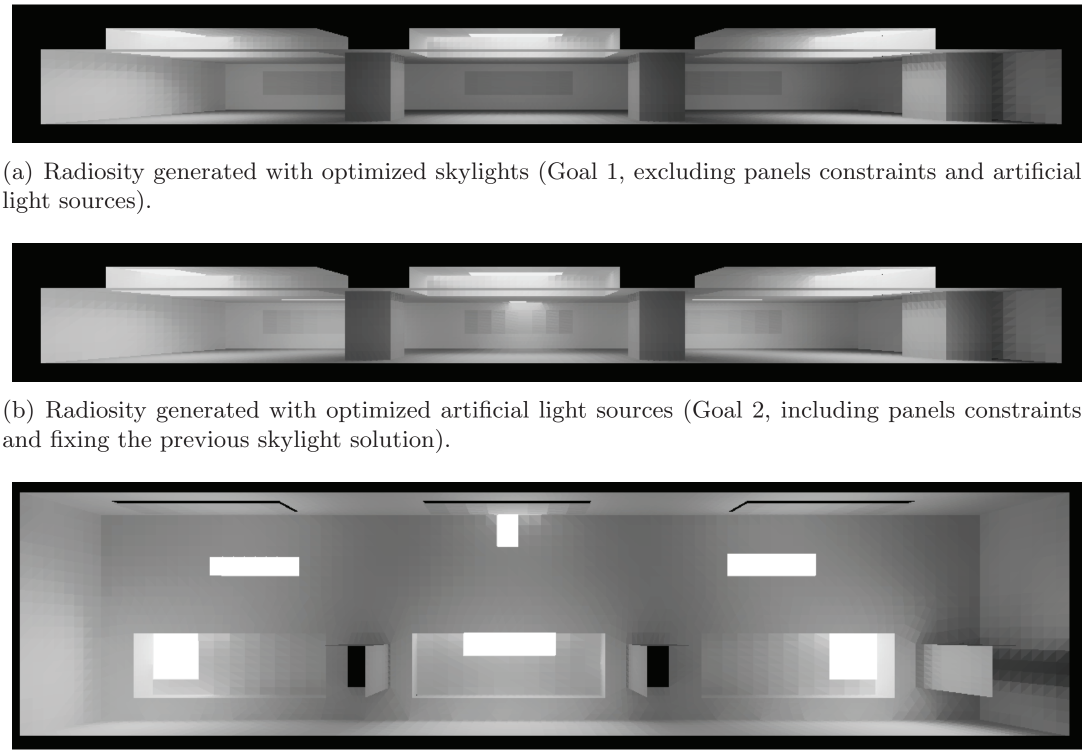
Figure 4.15: Feasible solutions found in a ϵ-constraint process (blue '+') and its corresponding Pareto front (red 'o'). Additionally, solutions found from a two-step process (green '+') and the associated Pareto front (black '□').

The position and shape of the skylights and emitters determined using the two-step process are shown in Figure 4.16 (c).

Figure 4.15 shows the solutions found by the two-step process. The green '+' are the solutions found by this process, and the black '□' set is their corresponding Pareto front. The two-step process was executed 50 times, with 20,000 iterations each.

Many of the 50 solutions found using the two-step method are better than (i.e., not dominated by) the ϵ-constraint Pareto front solutions, but the two-step method is concentrated in one extreme of the Pareto front. These results show that the Pareto front found contains rather than good solutions and good diversity and also show that the two-step method is a very effective approach to find solutions that meet specific design goals.

Close Pareto front solutions can be caused by very different light source configurations. In Figure 4.15 (b), the two-step Pareto front is located in a narrow range of



(a) Radiosity generated with optimized skylights (Goal 1, excluding panels constraints and artificial light sources).

(b) Radiosity generated with optimized artificial light sources (Goal 2, including panels constraints and fixing the previous skylight solution).

(c) Top view of the scene with the positioning solution of all the area sources.

Figure 4.16: Solutions of the two-step optimization process for the corridor scene.

powers, and Experiment 4 shows that solutions with almost the same light power have different light-source configurations. Therefore, a designer must check all the solutions to decide which one is the most convenient.

4.5.6 Summary Discussion

Numerically comparing our method with other previous inverse lighting techniques is a difficult task because each method provides different conditions. However, we can claim that we provide a fast solver for moderately complex environments with Lambertian surfaces. Using the acceleration technique described in our second experiment, we can obtain reliable results in only a few minutes, which means an improvement from recent works [4, 20]. Moreover, our system provides interactive radiosity visualization that can help in making design decisions.

The low-rank methodology, as a direct method, allows finding the radiosity of p patches in $O(n + pk)$. This is faster than other iterative methods (like hierarchical radiosity) where the radiosity of all the scene must be solved, even if the algorithm optimizes the radiosity value of a single patch.

The low-rank methodology allows considering only diffuse scene surfaces, which is not true in many interior settings. To overcome this limitation, a method for computing the entire light transport, including glossy surfaces, could be analyzed and tested.

The reverse-engineering process applied to design, which computes the initial conditions to achieve a lighting effect, is not a simple issue. One important subject is the search for feasible solutions, where the methodology can provide sufficient assistance.

Besides searching for a solution, our method also provides much more information about the range of possibilities that may occur when expressing a design intention. This information will allow an interactive use of the design cycle, in which the designer can explore multiple possibilities. Further extending this concept, our system provides a MOP solver that finds a Pareto front of solutions. Given this set of optimal solutions, designers must choose the ones that match their design ideas.

4.6 Conclusions and Future Work

We developed a novel technique of inverse lighting, combining the use of an optimization metaheuristic with the LRR technique as a radiosity solver. The paper addressed the problem of optimization with constraints, using the penalty method approach. A radiosity engine based on LRR methodology was used, taking into consideration the spatial coherence of the scene. Also, as LRR is a direct methodology, a multilevel scheme was tested, showing promising results. Finally, the case of MOP was analyzed, developing two methods based on VNS metaheuristic: ϵ-constraint, when the goal consists in finding a Pareto front, and the two-step optimization process, when the goal is to maximize the natural light provided by skylights and minimize the power of artificial light.

Regarding future work, one objective consists in improving the technique using a CPU-GPU architecture in order to speed up the VNS and LRR calculation times. It is also necessary to explore the multilevel methodology deeper, to transform the basic scheme implemented into a more robust algorithm. Related to the emitters, further work is needed to include anisotropic light sources and temporal and climate variation of the daylight sources. Light sources with anisotropic distribution, such as spot lights, may also be included in our method. This could be accomplished through a graphics hardware mapping step converting the projected light into light emitting patches at each iteration of the optimization engine. However, this requires an analysis of non-linear parameters, as either the cosine dispersion exponent or the focus direction, can affect performance.

Another line of future work is the implementation of a MOP solver using a population-based metaheuristic like genetic algorithms. Following Talbi [30], this kind of metaheuristic allows a better diversification in the whole search space than single-solution based metaheuristic. Finally, the development of real examples and working experiences with designers are needed. This line of work will transform the code implemented into a design tool useful for architects and interior designers.

Acknowledgments

This work was partially funded by Programa de Desarrollo de las Ciencias Básicas (Uruguay), by Asociación Universitaria Iberoamericana de Postgrado, by Comisión Sectorial de Investigación Científica de la Universidad de la República (Uruguay), and by grant TIN2010-20590-C02-02 from Ministerio de Ciencia e Innovación (Spain).

Bibliography

[1] J. E. Tena, I. Rudomin, An interactive system for solving inverse illumination problems using genetic algorithms, in: Proceedings of computación visual, 1997.

[2] J. K. Kawai, J. S. Painter, M. F. Cohen, Radioptimization - goal based rendering, in: ACM SIGGRAPH 93, Anaheim, CA, 1993, pp. 147–154.

[3] M. Contensin, Inverse lighting problem in radiosity, Inverse Problems in Engineering 10 (2) (2002) 131–152.

[4] F. Castro, E. del Acebo, M. Sbert, Energy-saving light positioning using heuristic search, Engineering Applications of Artificial Intelligence 25 (3) (2012) 566 – 582.

[5] P. Hansen, N. Mladenovic, Variable neighborhood search: Principles and applications, European Journal of Operational Research 130 (3) (2001) 449–467.

[6] C. Schoeneman, J. Dorsey, B. Smits, J. Arvo, D. Greenberg, Painting with light, in: Proceedings of the 20th annual conference on computer graphics and interactive techniques, ACM SIGGRAPH 93, ACM, New York, NY, USA, 1993, pp. 143–146.

[7] G. Patow, X. Pueyo, A survey of inverse rendering problems, Computer Graphics Forum 22 (4) (2003) 663–688.

[8] A. C. Costa, A. A. de Sousa, F. N. Ferreira, Lighting design: a goal based approach using optimisation., in: D. Lischinski, G. W. Larson (Eds.), Rendering techniques, Springer, 1999, pp. 317–328.

[9] P. Hanrahan, D. Salzman, A rapid hierarchical radiosity algorithm, in: Computer graphics, 1991, pp. 197–206.

[10] F. Pellacini, F. Battaglia, R. K. Morley, A. Finkelstein, Lighting with paint, ACM Trans. Graph. 26 (2).

[11] S. Gibson, T. Howard, R. Hubbold, Flexible image-based photometric reconstruction using virtual light sources, Computer Graphics Forum 20 (2001) 203–214.

[12] V. Tourre, J.-Y. Martin, G. Hégron, An inverse daylighting model for caad, in: Proceedings of the 24th spring conference on computer graphics, SCCG '08, ACM, New York, NY, USA, 2008, pp. 83–90.

[13] G. Besuievsky, V. Tourre, A daylight simulation method for inverse opening design in buildings, in: O. Rodríguez, F. Serón, R. Joan-Arinyo, E. C. J. Madeiras, J. Rodríguez (Eds.), Proceedings of the IV iberoamerican symposium in computer graphics, Sociedad Venezolana de Computación Gráfica, DJ Editores, C.A., 2009, pp. 29–46.

[14] E. Fernández, Low-rank radiosity, in: O. Rodríguez, F. Serón, R. Joan-Arinyo, E. C. J. Madeiras, J. Rodríguez (Eds.), Proceedings of the IV iberoamerican symposium in computer graphics, Sociedad Venezolana de Computación Gráfica, DJ Editores, C.A., 2009, pp. 55–62.

[15] J. Kontkanen, E. Turquin, N. Holzschuch, F. Sillion, Wavelet radiance transport for interactive indirect lighting, in: W. Heidrich, T. Akenine-Möller (Eds.), Rendering techniques 2006 (Eurographics symposium on rendering), Eurographics, 2006, pp. 161–171.

[16] J. Křivánek, M. Hašan, A. Arbree, C. Dachsbacher, A. Keller, B. Walter, Optimizing realistic rendering with many-light methods, in: ACM SIGGRAPH 2012 courses, SIGGRAPH '12, ACM, New York, NY, USA, 2012.

[17] S. M. Goldfeld, R. E. Quandt, H. F. Trotter, Maximization by Quadratic Hill-Climbing, Vol. 34, The Econometric Society, 1966.

[18] S. J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 2nd Edition, Pearson Education, 2003.

[19] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[20] F. Cassol, P. S. Schneider, F. H. França, A. J. S. Neto, Multi-objective optimization as a new approach to illumination design of interior spaces, Building and Environment 46 (2) (2011) 331 – 338.

[21] P. S. Schneider, A. C. Mossi, F. H. R. Franca, F. L. de Sousa, A. J. da Silva Neto, Application of inverse analysis to illumination design, Inverse Problems in Science and Engineering 17 (6) (2009) 737–753.

[22] S. Russell, The architecture of light: architectural lighting design, concepts and techniques: a textbook of procedures and practices for the architect, interior designer and lighting designer, Conceptnine, 2008.

[23] D. Luenberger, Y. Ye, Linear and Nonlinear Programming, International Series in Operations Research & Management Science, Springer, 2008.

[24] G. H. Golub, C. F. Van Loan, Matrix computations (3rd ed.), Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[25] E. Fernández, P. Ezzatti, S. Nesmachnow, G. Besuievsky, Low-rank radiosity using sparse matrices, in: Proceedings of GRAPP2012, 2012, pp. 260–267.

[26] M. Cohen, J. Wallace, P. Hanrahan, Radiosity and realistic image synthesis, Academic Press Professional, Inc., San Diego, CA, USA, 1993.

[27] J. Lehtinen, M. Zwicker, E. Turquin, J. Kontkanen, F. Durand, F. X. Sillion, T. Aila, A meshless hierarchical representation for light transport, in: ACM SIGGRAPH 2008, ACM, New York, NY, USA, 2008, pp. 1–9.

[28] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Wiley-Interscience Series in Systems and Optimization, John Wiley & Sons, Chichester, 2001.

[29] C. C. Coello, G. Lamont, D. van Veldhuizen, Evolutionary algorithms for solving multi-objective problems, 2nd Edition, Genetic and evolutionary computation, Springer, Berlin, Heidelberg, 2007.

[30] E.-G. Talbi, Metaheuristics: From Design to Implementation, Wiley Publishing, 2009.

Chapter 5

From Lighting Intention to Light Filters [1]

Lighting intentions are the goals and constraints that designers would like to achieve in an illumination design process. Light filters can be used as an architectural element to obtain such intentions by inserting them into walls or roofs. Defining correctly the filter shape is a challenge. In this work, we present a novel method to design optimal filters from a given lighting intention. Our methodology, which could be completely integrated within a computer aided architectural design framework, is based on a global illumination and optimization approach. Our test results show that optimal filter shapes can be obtained in a short time.

Keywords: Lighting intention, Inverse lighting, CAD, Building envelope

5.1 Introduction

As light reveals architecture, the natural lighting is a critical feature in architecture design. For the designer, the problem comes from the management of the natural lighting during the design process. Some design approaches have focused on keeping the intention of the designer as the project goes on. As an example, the design by intent approach is based on a lighting description, expressed by texts or by drawings, taking into account the user comfort or the energy savings. In general, the whole process to design an architectural illumination space may achieve several steps that go from revealing the general use of the space to the special light accentuation that must be obtained [1].

The goal of the Computer Aided Architectural Design (CAAD) developers is to keep the mentioned intents as a guideline through the design process by a connection between the lighting description and the geometric or optic properties of the building. In this work, we propose to link the designer's lighting concepts and the building matter through the natural lighting simulation used in an inverse approach. Through our approach architects and engineers could set up constraints parameters about lighting needs, and an inverse lighting model optimizes some of the geometric properties of the

[1] **Published in:** V. Tourre, E. Fernández and G. Besuievsky *From Lighting Intention to Light Filters*. Proceedings of the International Conference on Cleantech for Smart Cities & Buildings from Nano to Urban Scale (CISBAT 2013), EPFL, Lausanne, Switzerland, 3-6 September, 2013, Pages 1181-1186.

built spaces. Previous work proposing solutions for this problem can be found in [2], and an approach with a case-study application can be found in [3, 4].

In this paper, we present a methodology for computing optimal filters from a given lighting intention. An architectural real-case example is shown in Figure 5.1. The paradigm of our method is to focus on potential light filtering areas such as walls, openings or architectural elements. The key of our method is to find a compact representation of the lighting equations derived, so we can apply an inverse lighting approach to solve the design constraint problems. The main contribution of our work is to provide a flexible lighting design approach intended for searching both, fast and accurate solutions to the inverse lighting problem. The method is presented as a framework approach that could be extended for many lighting filtering purposes. We present here the results of its application to the design of filters, using a pattern that the designer provides as a drawing.

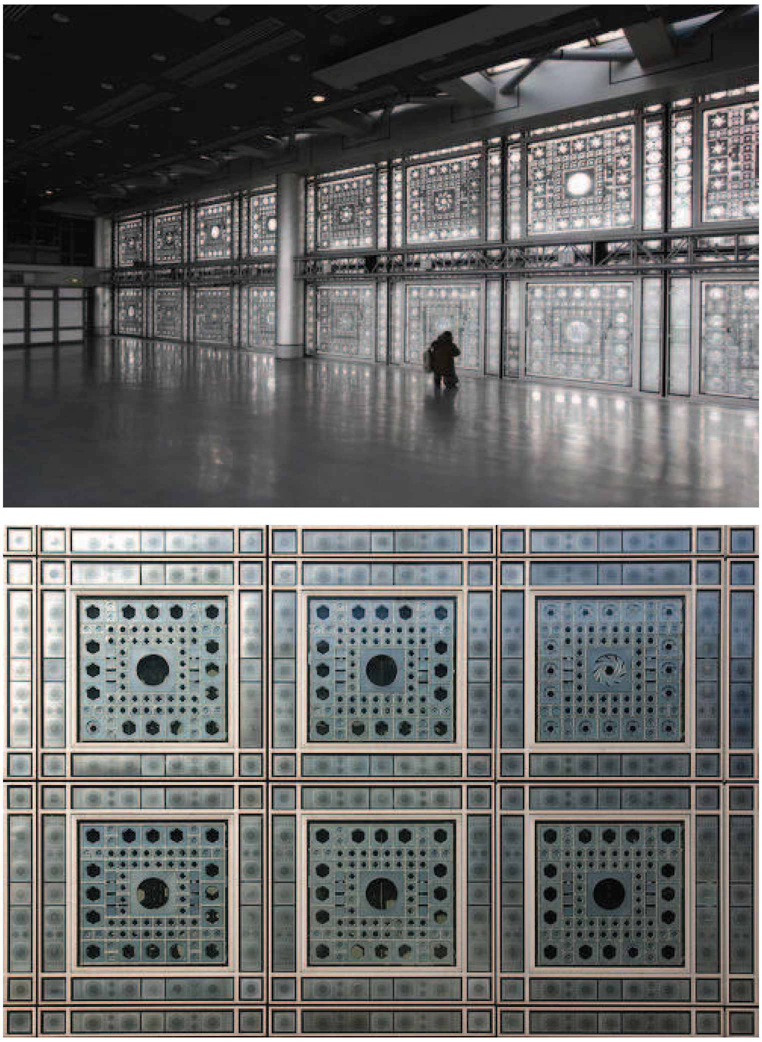


Figure 5.1: Lighting filter examples in modern architecture. Institute du monde Arabe (Paris), by Jean Nouvel. The holes of the filter (right) can vary to achieve a desired intention.

5.2 Related Work

Our work deals with an inverse lighting problem. Direct methods calculate data from a specific configuration of model parameters. In contrast, inverse problems generally infer the properties of a physical system from desired data. Inverse problems are usually numerically complex and are of interest in a wide range of fields in lighting engineering and lighting design.

In our work we focus on inverse problems considering global illumination solution, for such purpose we use the radiosity computation. In this context, several works driven by different motivations and assumptions were proposed [5, 6, 7]. We based our solution on work done in [7], where the low-rank radiosity method is used to solve the inverse lighting problem integrating skylights and artificial Lambertian sources.

5.3 Problem Definition

The main goal of our proposal is to provide a tool for efficient lighting design of filters. We consider diffuse environments, that is, all the surfaces have perfectly diffuse materials with no specular component. For skylight filters we also consider that a diffuser with a homogeneous transmittance coefficient is used. This kind of diffusers are currently used systems in real buildings. They provide diffuse and controllable sunlight, which creates a desirable ambient light [1]. These kinds of skylights, can also be considered as Lambertian emitters, because they scatter light homogenously. The filters we use in this work are defined as holes of this kind of emitters.

5.3.1 Lighting intention

Lighting intentions can be defined as the goals and constraints that designers must provide to achieve a desired illumination. Regarding the reference focus and motivation, they could be classified in different ways by:

- Lighting representation (incoming or reflected light).
- The set of surfaces considered (the overall scene or a specific set of surfaces).
- The kind of target goal to achieve (optimize specific lighting levels or contrast between surfaces, or satisfy several constraints).
- The light transport level (only direct lighting or global illumination).

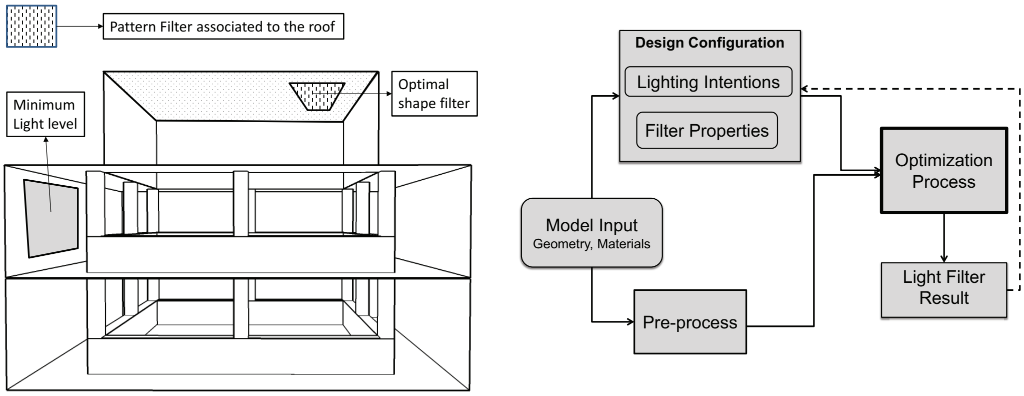
Designers have freedom to set as many lighting intention as necessary. Possible examples are: “Guarantee a minimum of irradiance in a wall” or “Maximize the incoming light into the room”. An example of restriction may be the size of the opening or any other architectural constraint (see Figure 5.2(a)). In an inverse lighting problem we must obtain the “best” lighting position that also satisfies both the constraints and the lighting intentions. Usually, a goal related to the energy-consumption should be minimized.

5.3.2 Filter Design

In order to specify the filter design we use a pattern drawing. First, the designer has to decide which is the wall or roof where to install the filters. Then, a pattern is associated. The designer has complete freedom to use any drawing. This drawing will be mapped to the associated surface as Lambertian holes emitters in the constructive surface. Of course they may be many ways for expressing a possible shape of a filter, we just decide to use this simple one in order to test our optimization engine.

5.3.3 System Overview

The pipeline design of our approach is described in Figure 5.2(b). Given an architectural interior model for lighting design, the user first configures both the parameters of the filter and lighting intentions. These parameters specify where to install the filters as well as the pattern they may have. In the example of Figure 5.2(a), a homogeneous filter with holes was used with a maximum rectangular area to provide the opening. Before the optimization process begins, we first pre-process the scene to obtain a compact representation of the form-factor matrix of the elements of the scene. For this purpose, we use a low-rank radiosity method. The optimization method works by getting the compact representation found and a design configuration, to obtain efficiently the optimal shape. The results can be visualized interactively and the direct lighting computation can be assured. The designing process may iterate if the designer wants to change or explore other solutions by modifying parameters.



(a) Design example scene.

(b) System overview.

Figure 5.2: From a given design specification and filter pattern (a), our pipeline uses an optimization process to compute the optimal filter shape (b).

5.4 Optimization Process

In this section we summarize our mathematical method to optimize the filter shape. The method is based on direct computation through the use of the low-rank radiosity method and the same inverse lighting method as in [7].

5.4.1 Direct Radiosity Computation

In the discrete radiosity formulation, the radiosity of the scene is computed by solving a linear system that can be expressed as:

$$(\mathbf{I} - \mathbf{RF})B = E \tag{5.1}$$

where $\mathbf{I}$ is the identity matrix with dimension $n{\times}n$ (n is the number of patches or polygons), $\mathbf{R}$ is a diagonal matrix that stores the reflectivity index of the patches, $\mathbf{F}$ is the form factor matrix, B is a vector with the radiosity value of each patch, and E is the emission vector. B and E are measured in W/m^2.

Regardless of the optimization method used, the radiosity must be evaluated many thousand times before finding a converged solution. For this purpose an efficient method should be used. The low-rank radiosity method [8], offers the advantage of obtaining directly the radiosity values through an inverse matrix approximation:

$$\widetilde{B} = \widetilde{\mathbf{M}}_B E \tag{5.2}$$

where $\widetilde{\mathbf{M}}_B$ is an approximation of $(\mathbf{I} - \mathbf{RF})^{-1}$. The detailed process of how to obtain this matrix is described in [8]. The key of the method is to explore the coherence in the scene to reduce the numerical rank of the form factor matrix $\mathbf{F}$. This matrix can be approximated by the product of two matrices with dimension $n{\times}k$ ($n{\gg}k$), without loss of relevant information. The memory requirement for both matrices is $O(nk)$, which is significantly less than $O(n^2)$ required to store $\mathbf{F}$ of Equation 5.1. This memory saving allows storing the information in the main processor memory for scenes with several hundred thousand patches.

For static scenes with dynamic lighting, the required matrices are computed once, therefore, the calculation of $\widetilde{B}$ has complexity $O(nk)$. This result allows performing many radiosity calculations per second, which motivates its use for inverse problems.

5.4.2 Optimization Algorithms

The lighting intentions and the objective, can be formulated as an optimization problem. The problem consists of finding the emission configuration E that optimizes a goal function $f(E)$, subject to a set of constraints $G(E)$ that must be satisfied:

$$\min\ f(E), \text{subject to } G(E) \tag{5.3}$$

Although optimization is a well known topic, there is no computational algorithm that provides the global optimum for a general non-linear objective function. Finding the optimal solution by *brute force* is usually not feasible in a reasonable time because of the huge search space of the possible states. Heuristic algorithms avoid visiting the whole search space, by means of designing rules that drive the search towards optimal solutions. There is a large number of heuristic search algorithms in the literature, which can potentially be used to solve lighting problems [9, 10].

In our work, we used the Variable Neighborhood Search (VNS) method [11]. This methodology is based on the idea of successive explorations of a set of neighborhoods. The method explores, either at random or systematically, a set of neighborhoods to obtain different local optima. Each neighborhood has its own local optimum, and it is expected that the global optimum is the same as a local optimum for a given neighborhood. We adapted the mentioned algorithm to find the optimal filter shape (Figure 5.3). We used a closed polygon (for example a rectangle) as a clipping figure to find the best holes configuration that would achieve the intentions. The algorithm shakes over the extremes of the polygon, that are used as variables in the optimization algorithm (see Figure 5.4).

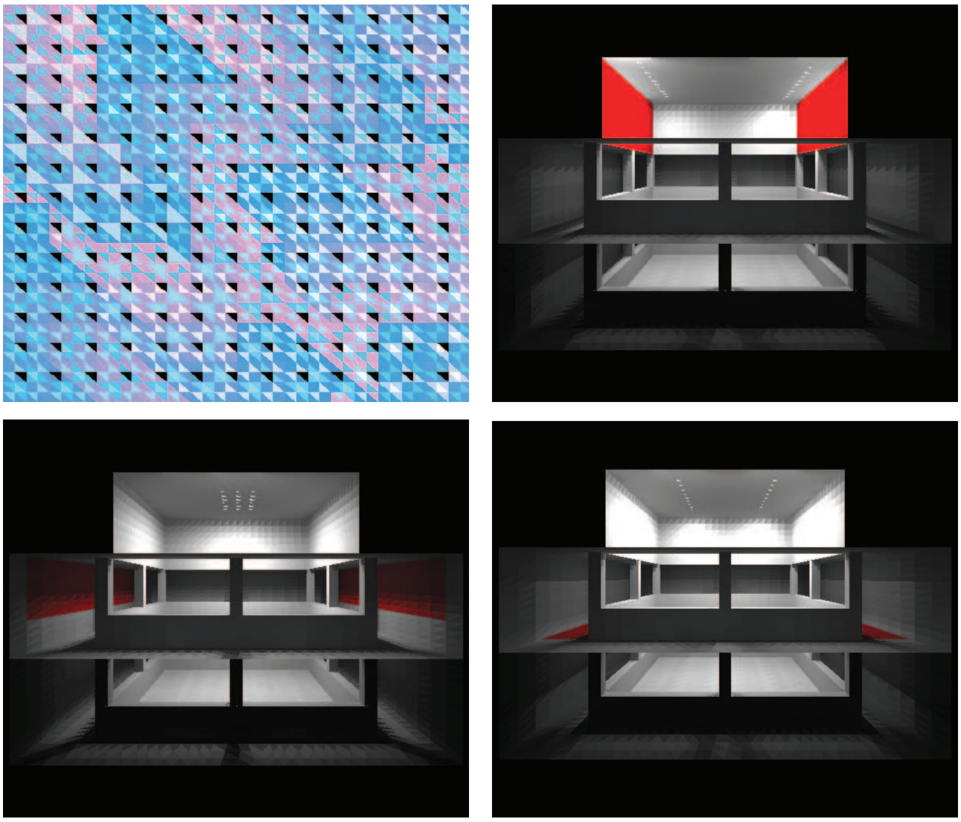


Figure 5.3: Filter optimization for an homogeneous filter (left), the black triangles represent the pattern and the rest of triangles, the associated mesh. Three different lighting intentions (form left to right: lateral upper walls, first floor walls and first floor) were designed to obtain the corresponding filters.

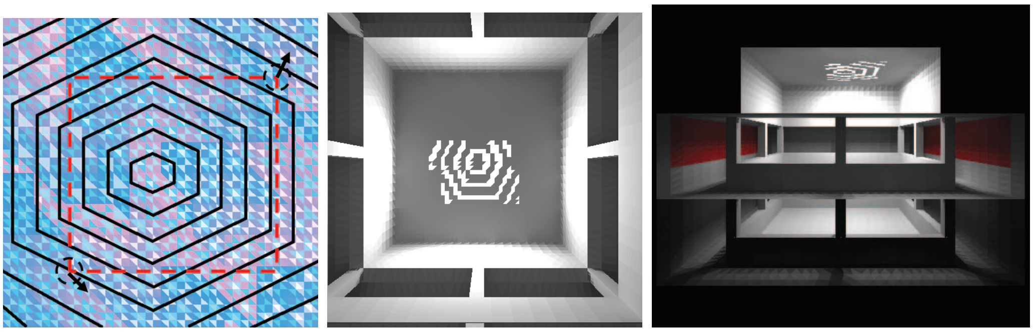


Figure 5.4: Filter optimization using an hexagonal pattern filter (left) and the resulted filter (middle) for a given lighting intention at the lateral (in red) walls (right).

5.5 Results and Conclussions

We implemented the proposed method and tested it with different pattern filters and lighting intentions. We chose a patio scene, where filters should be installed in the roof (see Figures 5.3 and 5.4). All simulations were performed in a Matlab environment using a standard PC (Intel Core i7 2.2 Ghz processor and 8 GB memory).

Scene : Patio. Size ($n \times k$): $23,744 \times 1,484$.

Goal : Maximize power light in the marked areas

Constraints : The light emitters must lie in the ceiling, and
$B_{min} \leq B(P) \leq B_{max}, \forall P \in$ red surface
$A_{min} \leq Area_{Emitter} \leq A_{max}$

Variables : 4 or 8, depending on the number of light sources (Two 2D-coordinates determine each source position)

Processing time :
Pre-processing: The low-rank pre-computation takes 18 minutes.
Optimization: 570 seconds for $100,000$ radiosity evaluations

We presented a new method for designing optimal filters from a given lighting intention. We believe that it could be a useful method for opening design, where singular installations in buildings are unpredictable. Our test results show that optimal filter shapes, using a global illumination approach, can be obtained in a short time. This considerable time reduction could promote its use for interactive applications and its inclusion in standard illumination design tools, as for example [12]. For future work, we would like to extend our model to include time-dynamic variable holes as in Figure 5.1.

Acknowledgments

This work was partially funded by the TIN2010-20590-C02-02 project from Ministerio de Ciencia e Innovación, Spain, and LiPaD project (PICS 5786) from CNRS, France.

Bibliography

[1] S. Russell, The Architecture of Light: Architectural Lighting Design, Concepts and Techniques: a Textbook of Procedures and Practices for the Architect, Interior Designer and Lighting Designer, Conceptnine, 2008.

[2] G. Besuievsky, V. Tourre, A daylight simulation method for inverse opening design in buildings, in: Proceedings of the IV Iberoamerican Symposium in Computer Graphics, 2009, pp. 29–46.

[3] V. Tourre, F. Miguet, Lighting intention materialization with a light-based parametric design model, International Journal of Architectural Computing 08 (4) (2010) 507–524.

[4] F. Imbert, K. Frost, A. Fisher, A. Witt, V. Tourre, B. Koren, Concurrent geometric, structural and environmental design: Louvre abu dhabi, in: Advances in Architectural Geometry 2012, 2013, pp. 77–90.

[5] M. Contensin, Inverse lighting problem in radiosity, Inverse Problems in Engineering 10 (2) (2002) 131–152.

[6] J. K. Kawai, J. S. Painter, M. F. Cohen, Radioptimization - goal based rendering, in: ACM SIGGRAPH 93, Anaheim, CA, 1993, pp. 147–154.

[7] E. Fernández, G. Besuievsky, Inverse lighting design for interior buildings integrating natural and artificial sources, Computers & Graphics 36 (8) (2012) 1096–1108.

[8] E. Fernández, Low-rank radiosity, in: Proceedings of the IV Iberoamerican Symposium in Computer Graphics, 2009, pp. 55–62.

[9] D. Luenberger, Y. Ye, Linear and Nonlinear Programming, International Series in Operations Research & Management Science, Springer, 2008.

[10] E.-G. Talbi, Metaheuristics: From Design to Implementation, Wiley Publishing, 2009.

[11] P. Hansen, N. Mladenovic, Variable neighborhood search: Principles and applications, European Journal of Operational Research 130 (3) (2001) 449–467.

[12] DIALux, Dial gmbh, www.dial.de (Nov 2013).

Chapter 6

Efficient Inverse Lighting: A Statistical Approach [1]

Given a scene to illuminate satisfying a specific set of lighting intentions, the inverse lighting techniques allow us to obtain the unknown light source parameters, such as light position or flux emission. This paper introduces a new inverse lighting approach that uses the radiosity mean and variance to define lighting intentions of a scene. It is shown that these statistical parameters can be obtained without the previous calculation of the radiosity of the scene. Avoiding the explicit computation of the illumination of the scene results in a drastic reduction of the time required for the inverse process. This approach also provides a methodology that transforms a current set of lighting intentions into a single lighting intention with statistical parameters. The tests show that the processing time to solve the inverse problem can be reduced to a few seconds in most cases, improving the state of the art.

Keywords: Lighting Design; Optimization; Statistics

6.1 Introduction

Lighting intentions (LI) are the goals and constraints that designers would like to achieve in an illumination design process. Given an architectural interior space to illuminate, this process requires several steps that go from revealing the general use of the space, to the specific light accentuation that must be obtained [1]. Concerning the LI, they can be defined in different ways according to the reference observation (incoming light or reflected light) or to the global feature of the intention (local subset of surfaces or the overall scene). Once the LI are specified, the designer must provide the parameter setting that fulfills the desired intentions. The main features, such as the position of the lights, their shape and their spectral power, must be provided.

Finding efficiently all lighting parameters is a challenge since the space of possibilities

[1] **Published in:** E. Fernández and G. Besuievsky *Efficient Inverse Lighting: A Statistical Approach*, Automation in Construction, Volume 37, January 2014, Pages 48-57.
A shorter version of the paper was published in: E. Fernández and G. Besuievsky *Statistical Inverse Lighting*. Proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP) and International Conference on Information Visualization Theory and Applications (IVAPP), Barcelona, Spain, 21-24 February, 2013, Pages 185-190.

is huge. The current approach to improve the solution is to treat this issue as an optimization problem. As in other inverse problems, the initial data conditions must be determined from the model parameters. In the case of inverse lighting problems (ILP) the unknown illumination settings must be completely determined from the LI.

The whole ILP process involves two large computational tasks: an optimization process for the search of an optimal solution, and the global illumination process. Both tasks are strongly related because the first one is an iterative process that calls the second one in each iteration. The optimization process is used to find a solution that satisfies the set of LI, that minimizes a given goal, whereas the global illumination process consists in the simulation of the light transport process through multiple bounces. The time consumed in the ILP process is proportional to the number of calls to the illumination process.

Concerning the most recent work on ILP [2, 3, 4, 5] which considers global illumination, the main drawback is that they require several minutes or hours for even simple scenarios, limiting their use in interactive applications. This work focuses on the reduction of the global illumination processing time.

This paper presents the statistical inverse lighting (SIL) approach, a new ILP method that greatly improves both timing and memory of previous work. Our method exploits coherence of architectural interior spaces to build a compact representation, which can be efficiently used to explore many lighting solutions. The main contribution of our methodology is the introduction of a new mathematical technique that uses the statistical mean and variance of the radiosity of a scene as LI. Specifically, the designed LI are transformed into other intentions represented by statistical parameters. The innovative approach provides a mechanism that allows calculating this new LI, avoiding the explicit computation of the global illumination.

Experiments were performed with different LI and showed that in most cases the ILP processing time can be shorten to only a few seconds, a significant improvement concerning the state of the art. We believe that this time reduction could be a step forward in the integration of ILP methods in interactive professional illumination design tools, as for example Dialux [6] or Relux [7].

Section 6.2 reviews the previous work and state the problem formulation. The new approach is introduced in Section 6.3, and a proposal algorithm for automatic parameter determination is presented in Section 6.4. Test results are shown in Section 6.5 and conclusions are summarized in Section 6.6.

6.2 Problem Definition and Related Work

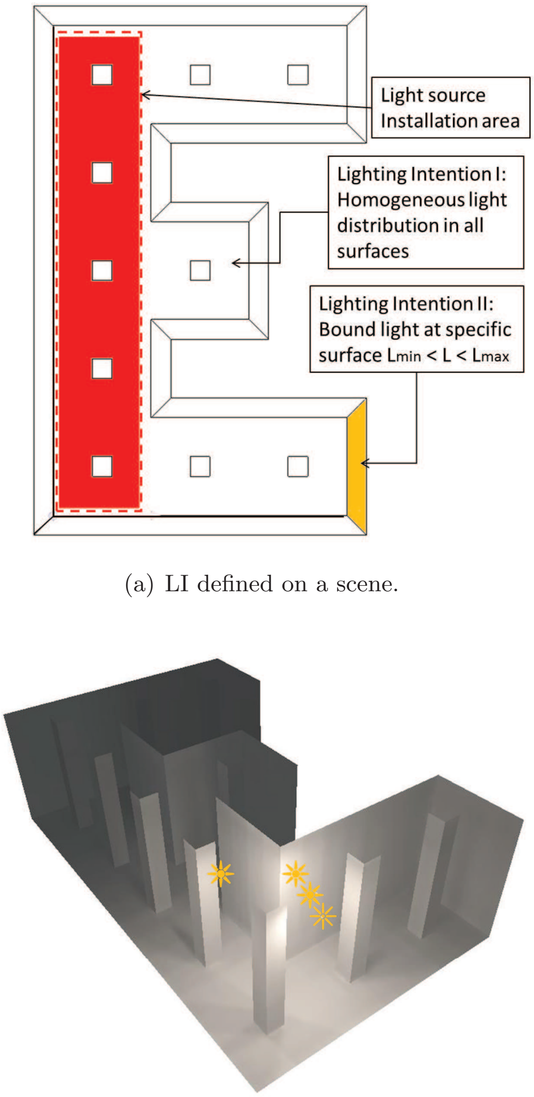
This section provides the background required for our method. First, the LI and the general considerations taken in ILP problems are defined. Then, the ILP in the context of an optimization problem is defined, and the low-rank radiosity is introduced as a methodology to obtain the inverse of the radiosity matrix. The relevant related work is described at the end of the section.

6.2.1 Lighting intentions

LI can be defined as the goals and constraints that designers must provide to achieve a desired illumination. Regarding the reference focus and motivation, they could be classified in different ways by:

- Lighting representation (incoming or reflected light).
- The set of surfaces considered (the overall scene or a specific set of surfaces).
- The kind of target goal to achieve (optimize specific lighting levels or contrast between surfaces, or satisfy several constraints).
- The light transport level (only direct lighting or global illumination).

Designers have freedom to set as many LI as necessary. Possible examples are: "Guarantee a minimum of irradiance on a panel" or "Distribute the light homogeneously over all surfaces of the scene" (see Figure 6.1). In the ILP the "best" lighting position


Light source Installation area
Lighting Intention I: Homogeneous light distribution in all surfaces
Lighting Intention II: Bound light at specific surface Lmin < L < Lmax

(a) LI defined on a scene.

(b) Final configuration of lights.

Figure 6.1: The ILP process. For a set of LI provided by the designer (a) the best solution for positioning four light sources is obtained, which fulfills all the constraints (b).

that also satisfies the constraint intentions must be obtained. Usually, a goal related to the energy-consumption should be minimized.

6.2.2 Problem Considerations

The present work formulates the problem assuming the following considerations:

- All surfaces of the environments have perfectly diffuse materials with no specular component.
- Emitters are approximated as point light sources. Their optimal position must be obtained. However, this is not a restriction of our method, which could also operate with area light sources.
- The environment is static, that is, surfaces can not change their position. The change of position of the emitters is implemented by modifying the emission property of previously defined patches.
- The use of indirect illumination is emphasized, a radiosity algorithm for computing the global illumination is used.

In the discrete radiosity formulation [8], the radiosity of the scene is computed by solving a linear system that can be expressed as:

$$(\mathbf{I} - \mathbf{RF})B = E \tag{6.1}$$

where $\mathbf{I}$ is the identity matrix with dimension $n \times n$ (n is the number of patches or polygons), $\mathbf{R}$ is a diagonal matrix that stores the reflectivity index of the patches, $\mathbf{F}$ is the form factor matrix, B is a vector with the radiosity value of each patch, and E is the emission vector. B and E are measured in W/m^2.

6.2.3 Optimization Problem

The set of LI presented in Section 6.2.1 can be formulated as an optimization problem [9]. The problem consists of finding the emission configuration E that optimizes a goal function $f(E)$, subject to a set of constraints $G(E)$ that must be satisfied:

$$\min f(E), \text{subject to } G(E) \tag{6.2}$$

The radiosity values B are not expressed in Equation 6.2 because they implicitly depend on E.

Although optimization is a well known topic, there is no computational algorithm that provides the global optimum for a general non-linear objective function. Finding the optimal solution by *brute force* is usually not feasible in a reasonable time because of the huge search space of the possible states. Heuristic algorithms avoid visiting the whole search space, by means of designing rules that drive the search towards optimal solutions. There are a large number of heuristic search algorithms in the literature, which can potentially be used to solve lighting problems [9, 10, 11]. Castro et al. [5] explore a wide range of algorithms to solve optimal economical light positioning. Cassol et al. [4] and Schneider et al. [3], work with rectangular simplified scenes, solving the ILP through a generalized extremal optimization approach. In the present work, the Variable Neighborhood Search (VNS) method [12, 13] is used.

6.2.4 Low-Rank Radiosity

Regardless of the optimization method used, the global illumination function must be evaluated many times before finding a converged solution. For this purpose an efficient method should be used to compute the radiosity B given the emission E. A method to efficiently assess the radiosity in Equation 6.1 is introduced. This method will be used later to speed up the calculation of statistical parameters.

It is very likely that the $\mathbf{F}$ matrix in Equation 6.1 has a low numeric rank. This happens because each row $\mathbf{F}(i,:)$ is computed based on the scene view from the geometrical patch i. As close patches have a similar view of the scene, then $\mathbf{F}$ also has similar rows, resulting in the reduction of its numerical rank. References about low-rank properties of radiosity and radiance matrices can be found in Baranoski et al. [14], Ashdown [15], Hasan et al. [16], and Fernández [17].

The low numeric rank of $\mathbf{F}$ allows its approximation by the product of two matrices ($\mathbf{U}\mathbf{V}^T$), both with dimension $n{\times}k$ ($n{\gg}k$), without loss of relevant information. The memory requirement for both matrices is $O(nk)$, which is significantly less than $O(n^2)$ required to store $\mathbf{F}$. This memory saving allows storing $\mathbf{U}$ and $\mathbf{V}^T$ in the main processor memory of desktop computers for scenes with several hundred thousand patches.

As shown by Fernández [17], the matrix $\mathbf{F}$ can be replaced by the low-rank approximation $\mathbf{U}\mathbf{V}^T$ in Equation 6.1, where $\mathbf{U}$ is a dense $n \times k$ matrix and $\mathbf{V}$ is a $n \times k$ sparse matrix with n nonzero elements. This substitution allows us to obtain the low-rank radiosity (LRR):

$$\left(\mathbf{I} - \mathbf{R}\mathbf{U}\mathbf{V}^T\right)\widetilde{B} = E \tag{6.3}$$

$\widetilde{B}$ can be calculated using a non-iterative procedure, by Sherman-Morrison-Woodbury formula [18]:

$$\widetilde{B} = \left(\mathbf{I} - \mathbf{Y}\mathbf{V}^T\right)E = \widetilde{\mathbf{M}}_B E, \tag{6.4}$$

$$\text{where } \mathbf{Y} = -\mathbf{R}\mathbf{U}\left(\mathbf{I}_{k\times k} - \mathbf{V}^T\mathbf{R}\mathbf{U}\right)^{-1}$$

The matrix $\mathbf{Y}$ is $n{\times}k$, and the inverse matrix included in its formulation is $k{\times}k$. The matrix $\widetilde{\mathbf{M}}$ is the inverse of $\left(\mathbf{I} - \mathbf{R}\mathbf{U}\mathbf{V}^T\right)$ and is also an approximation of $(\mathbf{I} - \mathbf{R}\mathbf{F})^{-1}$. In static scenes with dynamic lighting, the matrices $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{Y}$ are computed once; therefore the calculation of $\widetilde{B}$ has complexity $O(nk)$. This result allows us to perform many radiosity calculations per second, and was used by Fernández and Besuievsky [13] to accelerate ILP algorithms.

6.2.5 Related Work

In the context of radiosity, several works driven by different motivations and assumptions were proposed. Contensin [2] formulated an Inverse Radiosity method based on a pseudo-inverse of the radiosity matrix, whereas Costa et al. [19] proposed an optimization engine to deal with complex light specifications. Kawai et al. [20] performed the optimization over the intensities and directions of a set of lights, in order to best convey the subjective impression of certain scene qualities, as expressed by users. Their so-called Radioptimization system is a framework that determines optimal setting parameters based on unconstrained optimization techniques, used in conjunction with a hierarchical radiosity solver [21]. Castro et al. [5] developed a heuristic search algorithm, which is combined with linear programming to optimize light positioning with an energy-saving goal. Finally, Fernández and Besuievsky [13] used the low-rank radiosity method to build an ILP solution integrating skylights and artificial Lambertian sources.

The main drawback of previous work, as was stated before, is that they are time consuming to be integrated in an interactive design cycle of illumination.

6.3 A Statistical Based Approach of LI

In statistics [22], the mean (μ) and the standard deviation (σ), as well as other parameters, are used to describe properties about a set of numerical data. The mean describes the central tendency of the data and the standard deviation is a measure of their dispersion. In this section we borrow μ and σ concepts to describe LI, taking advantage of their mathematical properties. It is shown that μ and σ can be used as LI and the proposal's advantage is evaluated. In particular, it is demonstrated that $\mu(B(\mathbf{s}))$ and $\sigma(B(\mathbf{s}))$ can be obtained without the previous radiosity $B(\mathbf{s})$ calculation for a scene or a set of surfaces $\mathbf{s}$. This results in a significant advantage, since the global illumination computation is the most expensive part of the ILP process. Furthermore, it is shown that μ and σ can be used to substitute similar lighting constraints by a single and faster-to-check constraint.

6.3.1 μ and σ as LI

Following basic statistics concepts, $B(\mathbf{s})$ can be defined as the population sample of a variable B, and $\mu(B(\mathbf{s}))$ and $\sigma(B(\mathbf{s}))$ are its sample mean and standard deviation, respectively. These parameters can be used to describe LI. Figure 6.2 shows the optimization process. At each step of the process, the illumination of the scene is computed, and a new set of $B(\mathbf{s})$ values are used to update μ and σ. These statistical parameters may correspond to either the entire scene or a particular surface. Once these parameters are obtained, the LI are checked. The optimization algorithm establishes states for the position and intensity, which are used to obtain the new solution. The process stops when all constraints are fulfilled and when no better solution is found for the goals. Also the process may stop when it is not possible to find a solution that satisfies all the constraints.

Figure 6.3 shows an example with five different configurations of lights. It can be observed there that, the more homogeneously illuminated is the scene, the lower the value of $\sigma(B(\mathbf{s}))$.

6.3.2 Efficient Computation of $\mu(B(\mathbf{s}))$

Usually the sample mean of any population is computed after a sample of the population is obtained. The formula of μ can be expressed as a weighted sum of the collected data. This section shows that it is possible to get the mean of the radiosity values of a scene $B(\mathbf{s})$ without the explicit calculation of their radiosity values.

The radiosity of a surface $\mathbf{s}$ ($B_{\mathbf{s}}$) is a measure of the flux density of the reflected light. It is the weighted mean of the flux density values for all patches p that com-

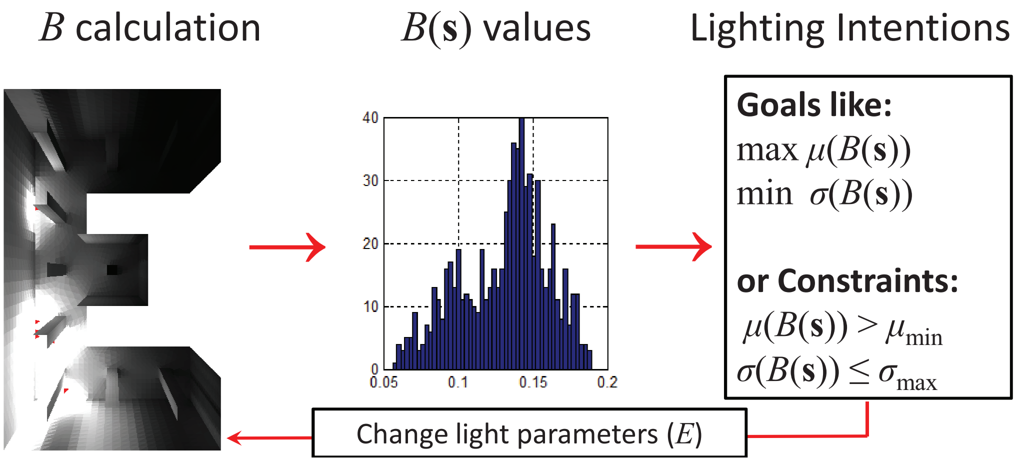

B calculation
$B(\mathbf{s})$ values
Lighting Intentions
40
30
20
10
0
0.05
0.1
0.15
0.2
Goals like:
max $\mu(B(\mathbf{s}))$
min $\sigma(B(\mathbf{s}))$
or Constraints:
$\mu(B(\mathbf{s})) > \mu_{\min}$
$\sigma(B(\mathbf{s})) \leq \sigma_{\max}$
Change light parameters (E)

Figure 6.2: Optimization process. Given a tentative lighting configuration (left), LI constraints are evaluated through an iterative process (right).

pose $\mathbf{s}$ (Equation 6.5), where each weight $w_A(p)$ is proportional to the area of p.

$$B_{\mathbf{s}} = \mu_{w_A}(B(\mathbf{s})) = \sum_{p \in s} w_A(p)B(p) \tag{6.5}$$

$$\text{where } \sum_{p \in s} w_A(p) = 1$$

$B(\mathbf{s})$ is defined as the vector that contains all $B(p)$ values ($p \in \mathbf{s}$), and $\mu_{w_A}(B(\mathbf{s}))$ as their weighted mean. Equation 6.5 can also be written using the following notation:

$$\mu_{w_A}(B(\mathbf{s})) = \left(\frac{\mathbf{1}_{\mathbf{s}}^T \mathbf{W}_A}{\sum_{p \in s} \mathbf{W}_A(p,p)} \right) B \tag{6.6}$$

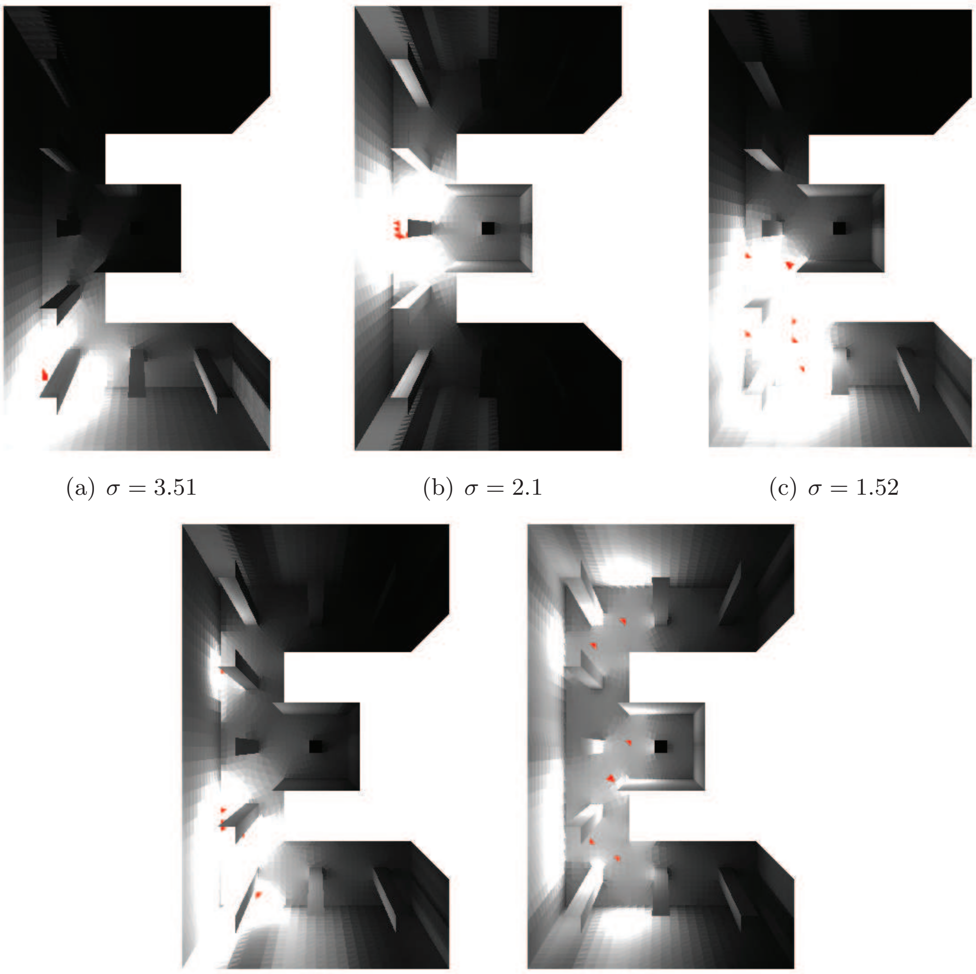
where $\mathbf{1}_{\mathbf{s}}^T$ is a row vector with only ones in the cells associated to the $\mathbf{s}$ patches, $\mathbf{W}_A$ is a diagonal matrix with the weights proportional to the areas of the patches, and B is a vector with the radiosity values of the scene. As exposed in Section 6.2.4, B can be substituted by the low-rank approximation $\widetilde{B} = \widetilde{\mathbf{M}}E$:

$$\mu_{w_A}(\widetilde{B}(\mathbf{s})) = \left(\frac{\mathbf{1}_{\mathbf{s}}^T \mathbf{W}_A \widetilde{\mathbf{M}}}{\sum_{p \in s} \mathbf{W}_A(p,p)} \right) E \tag{6.7}$$

In Equation 6.7, the expression between parenthesis is a $1 \times n$ row vector. Furthermore, using the expanded version of $\widetilde{B}(\mathbf{s})$ (see Equation 6.4), $\mu_{w_A}(\widetilde{B}(\mathbf{s}))$ can be formulated as:

$$\mu_{w_A}(\widetilde{B}(\mathbf{s})) = \left(\frac{\left(\left(\mathbf{1}_{\mathbf{s}}^T \mathbf{W}_A \right) (-\mathbf{Y}) \right) \mathbf{V}^T}{\sum_{p \in s} \mathbf{W}_A(p,p)} \right) E \tag{6.8}$$

The identity matrix defined in Equation 6.4 is not included in the definition of Equation 6.8, to avoid the inclusion of the light emission in the calculation of the mean of the



(a) $\sigma = 3.51$ (b) $\sigma = 2.1$ (c) $\sigma = 1.52$

(d) $\sigma = 1.03$ (e) $\sigma = 0.43$

Figure 6.3: Different radiosity distributions, in a scene composed of 24736 patches. Their σ values are indicated in each case. Red dots indicate the source's position.

reflected radiosity. The expression between parenthesis in Equation 6.8 is also a $1 \times n$ vector. So, in both cases the computation of $\mu_{w_A}(\widetilde{B}(\mathbf{s}))$ is reduced to a dot product with complexity $O(n)$. Therefore, the μ calculation of the radiosity values in $\mathbf{s}$ can be done avoiding the previous calculation of $\widetilde{B}(\mathbf{s})$.

6.3.3 Efficient Computation of $\sigma(B(\mathbf{s}))$

The standard deviation is also usually computed after the population data is collected. This parameter can be expressed as a weighted measure of the Euclidean distance between the collected data and its mean. If $B(\mathbf{s})$ is defined as a linear combination of variables, an alternative formulation can be created by using a covariance matrix. In Equation 6.9, the vector $\widetilde{B}(\mathbf{s})$ is expressed as a linear combination of k column vectors

$\mathbf{Y}(\mathbf{s}, i), i = 1 \ldots k$.

$$\begin{aligned}\widetilde{B}(\mathbf{s}) &= \widetilde{\mathbf{M}}(\mathbf{s}, :)E = -\mathbf{Y}(\mathbf{s}, :)(\mathbf{V}^T E) \\ &= \sum_{i=1}^{k} -\mathbf{Y}(\mathbf{s}, i)(\mathbf{V}^T E)_i\end{aligned} \tag{6.9}$$

Each of these vectors has dimension $|\mathbf{s}| \times 1$. The expression $(\mathbf{V}^T E)_i$ refers to the i^{th} value in vector $(\mathbf{V}^T E)$. Considering each vector $\mathbf{Y}(\mathbf{s}, i)$ as a sample of an i^{th} statistical variable, it is possible to formulate the following expression for σ [22]:

$$\sigma(\widetilde{B}(\mathbf{s}))^2 = (\mathbf{V}^T E)^T \mathtt{cov}_{w_A}(\mathbf{Y}(\mathbf{s}, :))(\mathbf{V}^T E), \tag{6.10}$$

where $\mathtt{cov}_{w_A}(\mathbf{Y}(\mathbf{s}, :))$ refers to the $k \times k$ sample covariance matrix built with the k vectors $\mathbf{Y}(\mathbf{s}, i)$. The right side of Equation 6.10 is a vector×matrix×vector product. The covariance matrix is built using a weighted approach:

$$\begin{aligned}&\mathtt{cov}_{w_A}(\mathbf{Y}(\mathbf{s}, :))_{i,j} = \\ &\sum_{k \in \mathbf{s}} w_A(k)(\mathbf{Y}(k, i) - \mu_{w_A}(\mathbf{Y}(\mathbf{s}, i)))(\mathbf{Y}(k, j) - \mu_{w_A}(\mathbf{Y}(\mathbf{s}, j)))\end{aligned} \tag{6.11}$$

where $\mathtt{cov}_{w_A}(\mathbf{Y}(\mathbf{s}, :))_{i,j}$ is the element (i, j) of the matrix $\mathtt{cov}_{w_A}(\mathbf{Y}(\mathbf{s}, :))$ and contains the covariance between the i^{th} and j^{th} variables defined in $\mathbf{Y}(\mathbf{s}, :)$. The term $\mu_{w_A}(\mathbf{Y}(\mathbf{s}, i))$ is the weighted sampled mean of $\mathbf{Y}(\mathbf{s}, i)$.

Computing $\sigma(\widetilde{B}(\mathbf{s}))$ has complexity $O(n + e^2)$, where e is the amount of nonzero elements in vector $\mathbf{V}^T E$. This can be deduced from the fact that $\mathbf{V}^T E$ can be computed with $O(n)$ operations due to the sparsity of $\mathbf{V}$. The term e^2 corresponds to the amount of non-zero floating point products included in the vector×matrix×vector operations. Considering that usually $e \ll k \ll n$, it can be said that the complexity of σ calculation is approximately linear.

The memory required to obtain $\sigma(\widetilde{B}(\mathbf{s}))$ is $O(n + k^2)$ because $\mathbf{V}$ is a sparse matrix with only n nonzero elements, and the covariance matrix has dimension $k \times k$.

6.3.4 Chebyshev-Based Constraint

The results obtained in the previous section can also be used to simplify the LI. Equation 6.12 shows an optimization problem where there is a LI G that depends on the radiosity values B

$$[E, \Phi] = Opt \begin{cases} \Phi = \min_E f(E) \\ \text{subject to} \begin{cases} B = \mathbf{M}E \\ G(B) \end{cases} \end{cases} \tag{6.12}$$

This optimization problem involves the search of an optimal configuration of emitters E and includes a $f(E)$ optimization goal, the computation of $B(E)$ using a radiosity engine, and the satisfaction of a set of constraints $G(B)$. A typical $G(B)$ consists in bounding the radiosity values of a surface $\mathbf{s}$ and can be expressed as:

$$G(B) : \{c_{min} \leq B(p) \quad \forall p \in \mathbf{s}\}, \tag{6.13}$$

where c_{min} is the lower bound of the radiosity values.

An optimization algorithm would have to solve, at each iteration, the radiosity equation to check that all B values in $\mathbf{s}$ are bigger than c_{min}. Instead, a more efficient approach based on Chebyshev's inequality [22], which avoids the calculation of all B values in each iteration, is proposed. The Chebyshev's inequality establishes that the probability of finding $\widetilde{B}(p)$ values outside the $\mu \pm \alpha\sigma$ interval is not greater than $1/\alpha^2$, regardless the distribution of $\widetilde{B}(p)$. The Chebyshev's inequality is used to build a probabilistic formulation where the initial set of constraints $G(B)$ is substituted by a new Chebyshev-based constraint $G_{Ch}(E, \alpha)$ (Equation 6.14), based on the use of α, μ, and σ values.

$$\begin{aligned} G_{Ch}(E,\alpha) : c_{min} \leq \mu_{w_A}\big(\widetilde{B}(\mathbf{s})\big) - \alpha\sigma_{w_A}\big(\widetilde{B}(\mathbf{s})\big) \\ \text{where } \alpha \geq 0 \end{aligned} \tag{6.14}$$

The equivalent to Equation 6.14 for the upper bound c_{max} would be:

$$\begin{aligned} &c_{max} \geq \mu_{w_A}\big(\widetilde{B}(\mathbf{s})\big) + \alpha\sigma_{w_A}\big(\widetilde{B}(\mathbf{s})\big) \\ &\text{where } \alpha \geq 0 \end{aligned}$$

Equation 6.15 shows the new Chebyshev's optimization problem Opt_{Ch}, which includes the constraint G_{Ch}. The problem now consists on finding an optimal E_{Ch}, which minimizes $f(E)$ subject to $G_{Ch}(E, \alpha)$.

$$[E_{Ch}(\alpha), \Phi_{Ch}(\alpha)] = Opt_{Ch}(\alpha) \begin{cases} \Phi_{Ch} = \min_E f(E) \\ \text{s.t. } G_{Ch}(E,\alpha) \end{cases} \tag{6.15}$$

Figure 6.4 shows an example of possible distribution of the $\widetilde{B}(\mathbf{s})$ values. Most $\widetilde{B}$ values must lie inside the $\mu \pm \alpha\sigma$ interval, which does not prevent that some $\widetilde{B}$ values could be lower than c_{min}.

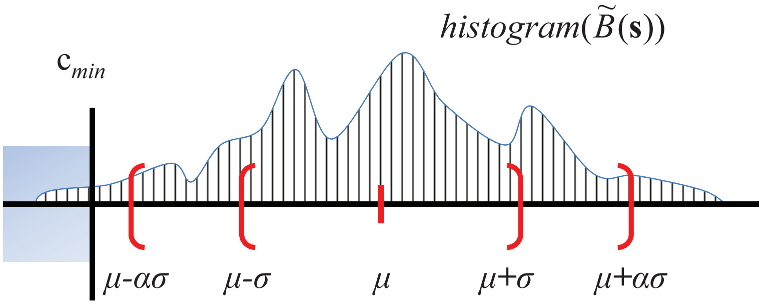


Figure 6.4: Histogram of the $\widetilde{B}(\mathbf{s})$ values. μ and σ values characterize their dispersion.

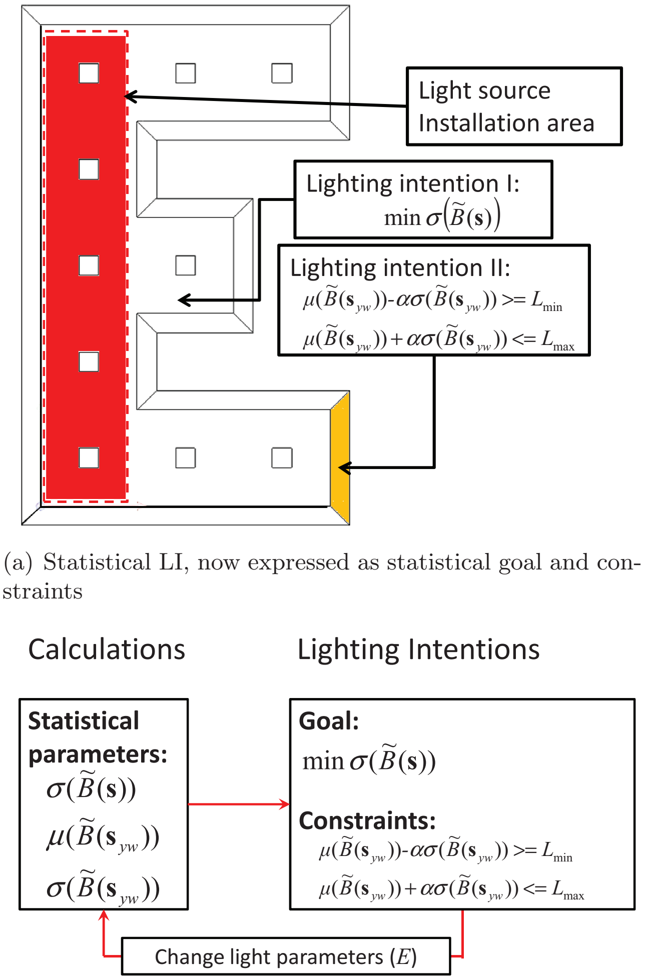
The probability bound in Chebyshev's inequality is the worst probability value. Usually, for lighting distributions in typical architecture scenarios, the probability bound may be much lower than $1/\alpha^2$. For instance, in the test performed in Section 6.5.3, a α value of 3 satisfies the full compliance of the $G(\widetilde{B})$ constraints.

6.3.5 Statistical Based Optimization

To the extent that all LI based on radiosity are replaced by other LI using μ and σ, the complexity of each iteration in the optimization process is reduced from $O(nk)$ to $O(n+e^2)$, and the memory required from $O(nk)$ to $O(n+k^2)$ (with $n \gg k \gg e$). The complexity reduction results in a great acceleration, as is analyzed in Section 6.5.5.

Figure 6.5(a) presents the new statistical transformation for the example shown in Figure 6.1(a). The goal and the constraints now are represented by using parameters μ and σ. The goal optimizes the $\sigma(\widetilde{B}(\mathbf{s}))$ value, and the original constraints are substituted by two Chebyshev-based constraints.

Figure 6.5(b) presents the new optimization process, where in each iteration there is no calculation of the radiosity of the scene. In this figure, $\mathbf{s}$ represents all patches in the scene, and $\mathbf{s}_{yw}$ corresponds to the patches of the yellow-colored wall. The following section presents a methodology to find appropriate values for α, where $E_{Ch}(\alpha)$ satisfies all $G(\widetilde{B}(E_{Ch}(\alpha))$ constraints.


Light source Installation area
Lighting intention I: $\min \sigma(\widetilde{B}(\mathbf{s}))$
Lighting intention II:
$\mu(\widetilde{B}(\mathbf{s}_{yw}))\text{-}\alpha\sigma(\widetilde{B}(\mathbf{s}_{yw})) >= L_{\min}$
$\mu(\widetilde{B}(\mathbf{s}_{yw}))+\alpha\sigma(\widetilde{B}(\mathbf{s}_{yw})) <= L_{\max}$
Calculations
Lighting Intentions
Statistical parameters:
$\sigma(\widetilde{B}(\mathbf{s}))$
$\mu(\widetilde{B}(\mathbf{s}_{yw}))$
$\sigma(\widetilde{B}(\mathbf{s}_{yw}))$
Goal:
$\min \sigma(\widetilde{B}(\mathbf{s}))$
Constraints:
$\mu(\widetilde{B}(\mathbf{s}_{yw}))\text{-}\alpha\sigma(\widetilde{B}(\mathbf{s}_{yw})) >= L_{\min}$
$\mu(\widetilde{B}(\mathbf{s}_{yw}))+\alpha\sigma(\widetilde{B}(\mathbf{s}_{yw})) <= L_{\max}$
Change light parameters (E)

(a) Statistical LI, now expressed as statistical goal and constraints

(b) The new statistical ILP process, which avoids the radiosity calculation at each step of the optimization process.

Figure 6.5: New LI and new ILP optimization process.

6.4 α value determination

The α values can be estimated by means of a trial and error method. However, a mechanical determination may lead to a more efficient approach. An iterative approach is proposed to find a solution of the Opt_{Ch} problem which is also a good solution for the Opt problem (i.e., an $E_{Ch}(\alpha)$ solution that satisfies $G(\widetilde{B}(E_{Ch}))$ constraints, and where $\Phi_{Ch}(\alpha)$ would be close to Φ). The strategy implemented minimizes the amount of radiosity calculations of the scene.

6.4.1 Considerations

Figure 6.6 shows the main elements to be considered when searching for an appropriate α_{Sol} value. The possible α values are always positive, and can be grouped into three regions. In region I the $E_{Ch}(\alpha)$ solutions do not satisfy $G(\widetilde{B}(E_{Ch}(\alpha)))$ constraints, and in region III there are no solutions to the $Opt_{Ch}(\alpha)$ problem. The possible valid α are in region II, where $E_{Ch}(\alpha)$ solutions satisfy the $G(\widetilde{B}(E_{Ch}(\alpha)))$ constraints. This paper assumes that the three regions always exist, but an implementation of the algorithm must consider also borderline cases and pathological situations.

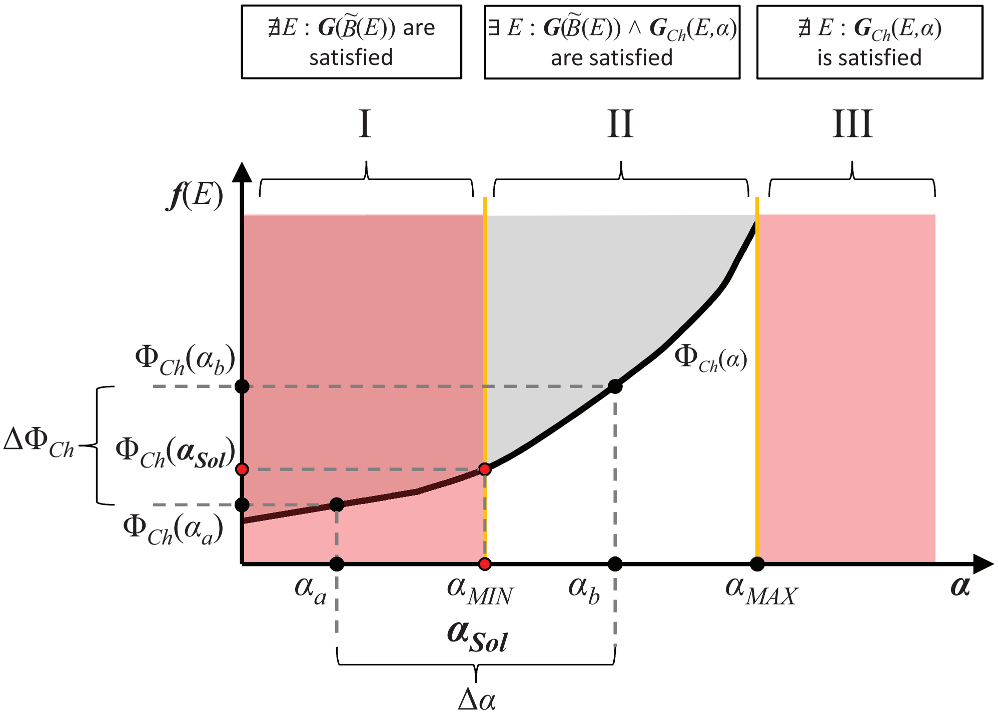
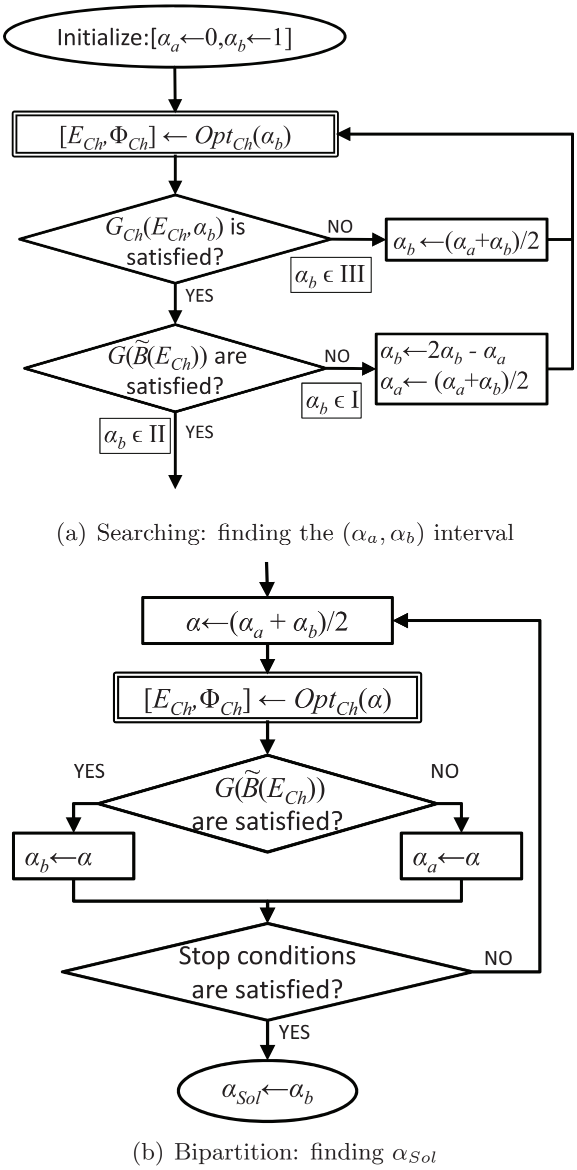

$\nexists E : G(\widetilde{B}(E))$ are satisfied
$\exists E : G(\widetilde{B}(E)) \wedge G_{Ch}(E,\alpha)$ are satisfied
$\nexists E : G_{Ch}(E,\alpha)$ is satisfied
I
II
III
$f(E)$
$\Phi_{Ch}(\alpha_b)$
$\Phi_{Ch}(\alpha)$
$\Delta\Phi_{Ch}$
$\Phi_{Ch}(\alpha_{Sol})$
$\Phi_{Ch}(\alpha_a)$
α_a
α_{MIN}
α_b
α_{MAX}
α
α_{Sol}
$\Delta\alpha$

Figure 6.6: Elements to be considered by the algorithm for "α-determination".

Figure 6.6 also shows the $\Phi_{Ch}(\alpha)$ curve, defined by the optimum Φ_{Ch} value associated to each α value. This curve has nonnegative slope, because when the value of α becomes smaller, the $G_{Ch}(E, \alpha)$ constraint is relaxed, expanding the set of feasible E solutions, and making the optimum Φ_{Ch} value lower or equal than before. Therefore, the best α candidate that minimizes $\Phi_{Ch}(\alpha)$ in region II is α_{MIN}, the smallest value of this region. So our strategy aims to find this value.

6.4.2 Algorithm

An algorithm based on a bipartition strategy is presented. The strategy seeks for the shortest interval (α_a, α_b), such that α_a is in region I, and α_b is in region II (see Figure 6.6). The final α_b value found would be the lowest α found belonging to region II. Then, α_b can be considered as α_{MIN}, the solution we are looking for. Figure 6.7 shows the flowchart of the algorithm, which is divided in two steps.


Initialize:[$\alpha_a \leftarrow 0, \alpha_b \leftarrow 1$]
$[E_{Ch}, \Phi_{Ch}] \leftarrow Opt_{Ch}(\alpha_b)$
$G_{Ch}(E_{Ch}, \alpha_b)$ is satisfied?
NO
$\alpha_b \leftarrow (\alpha_a + \alpha_b)/2$
$\alpha_b \in$ III
YES
$G(\widetilde{B}(E_{Ch}))$ are satisfied?
NO
$\alpha_b \leftarrow 2\alpha_b$ - α_a
$\alpha_a \leftarrow (\alpha_a + \alpha_b)/2$
$\alpha_b \in$ I
$\alpha_b \in$ II
YES
$\alpha \leftarrow (\alpha_a + \alpha_b)/2$
$[E_{Ch}, \Phi_{Ch}] \leftarrow Opt_{Ch}(\alpha)$
YES
$G(\widetilde{B}(E_{Ch}))$ are satisfied?
NO
$\alpha_b \leftarrow \alpha$
$\alpha_a \leftarrow \alpha$
Stop conditions are satisfied?
NO
YES
$\alpha_{Sol} \leftarrow \alpha_b$

(a) Searching: finding the (α_a, α_b) interval

(b) Bipartition: finding α_{Sol}

Figure 6.7: Algorithm for α-determination, which finds the minimum α value where $G(\widetilde{B}(E_{Ch}(\alpha)))$ is satisfied.

The first procedure step (see Figure 6.7(a)) searches for an initial (α_a, α_b) interval, such that the first extreme lies in region I and the second one in region II. These properties ensure that α_{MIN} will be inside the interval. α_a is initialized in zero, assuring that it is in region I. When α_b belongs to III, a new α_b value is generated in the middle point of the previous (α_a, α_b) interval. When α_b belongs to I, the interval is shifted to the right, placing α_a where it was α_b. The evaluation of the second conditional $(G(\widetilde{B}(E_{Ch})))$ implies the execution of the radiosity algorithm to find the $\widetilde{B}(E_{Ch})$ values. This first part of the algorithm ends when α_b belongs to region II.

The second part of the algorithm (Figure 6.7 (b)) is a bipartition algorithm that halves the size of the (α_a, α_b) interval in each iteration. The middle α value is obtained and the conditional establishes which of both halves is the selected interval. Stop conditions could be set using the thresholds for the number of iterations, the $\Delta\alpha$ value, or the $\Delta\Phi_{Ch}$ value.

6.5 Test Results

In order to evaluate the proposed method, five experiments were built, some of them composed by several tests. The first one shows the use of σ as a LI. The second experiment analyzes the optimization of μ and σ values, whereas the third one focuses on the Chebyshev-based constraints stated in Section 6.3.4. The fourth experiment shows results of the algorithm presented in Section 6.4.2. Finally, a performance comparison of the proposed method with other state of the art methods is given.

All simulations were performed in a Matlab environment using a standard PC (Intel Core i7 2.2 Ghz processor and 4 GB memory). Our optimization engine is based on the Variable Neighborhood Search (VNS) method [12]. VNS is a single-solution based metaheuristic that approximates global optimum solutions. However, the technique presented here has no consideration of the optimization engine used. We believe that comparable results could be achieved with other optimization engines.

6.5.1 Dispersion

This experiment shows the use of σ to manage the distribution of light in the scene.

Scene: "E-shape" corridor. Size ($n\times k$): $(24,736\times1,546)$.

LI Goal: Illuminate the scene as homogeneously as possible positioning six diffuse light sources.

Statistical Goal Equivalent: Minimization of the standard deviation of the reflected radiosity in the whole scene $\mathbf{s}$, through the positioning of the light sources: $\texttt{min}\ \sigma(B(\mathbf{s}))$.

LI Constraints:

- Six emitters.
- Emitters location is constrained to a specific area (the red-colored rectangle in Figure 6.1).
- All emitters have the same radiosity value.

Variables: 12

- Six 2D coordinates that determine the position of the light sources.

Experiment Details:

- The precomputation of $\mathbf{Y}$, $\mathbf{V}$, and $\mathtt{cov}(\mathbf{Y})$ matrices, needed to apply Equation 6.10, took 12 min.
- The VNS algorithm is executed 20 times to obtain information about the stability and distribution of the solution.
- σ is evaluated 25000 times in each execution.

Results:

- Processing time: 10 s per execution.
- σ values for all executions: $\sigma_{\text{min}} = 0.425$, $\sigma_{\text{max}} = 0.442$ and $\sigma_{\text{mean}} = 0.428$. The standard deviation (σ_{sd}) for all executions is 0.0035.
- Image result: Figure 6.3(e).

Results shown in Figures 6.3(b), (c), and (d) were computed by adding a lower bound constraint to the objective function ($\sigma(B(\mathbf{s})) \geq \sigma_{\text{min}}$). Figure 6.3(a) shows the result of maximizing the dispersion ($\mathtt{max}\ \sigma(B(\mathbf{s}))$) as a goal. For this case, it can be seen that all light sources concentrated in a close position.

6.5.2 Statistical tools for LI

This experiment analyzes the convenience of using μ and σ to satisfy a wide variety of LI. The same scene and experiment details as in the previous experiment are used. Four different categories of LI were tested on a specific wall of the scene ($\mathbf{s}_{yw}$ is associated to the set of patches that defines the “yellow-colored wall” in Figure 6.1(a)).

⊕ Illuminate

LI Goal: Illuminate the wall as much as possible positioning six diffuse light sources.

Statistical Equivalent Goal: $\mathtt{maximize}\ \mu(B(\mathbf{s}_{yw}))$.

Results:

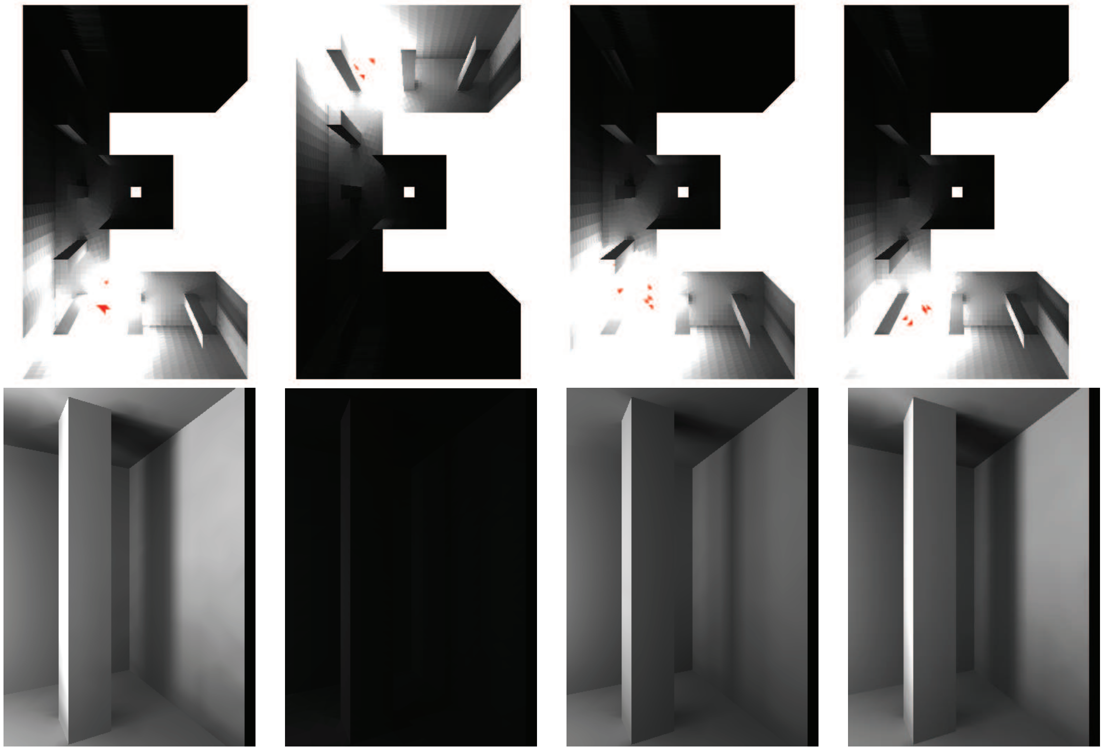
- Processing time: 10.2s per execution.
- μ values for all executions: $\mu_{\text{min}} = 0.409$, $\mu_{max} = 0.426$, $\mu_{\text{mean}} = 0.420$, and $\mu_{\text{sd}} = 0.0048$.
- Image result: Figure 6.8(a).

⊕ Overshadow

LI Goal: Overshadow the wall as much as possible; six diffuse light sources are needed to be positioned to illuminate the rest of the scene.

Statistical Equivalent Goal: $\mathtt{min}\ \mu(B(\mathbf{s}_{yw}))$.

Results:



(a) `max` μ

(b) `min` μ

(c) `min` σ subject to $0.30 \leq \mu(B) \leq 0.35$

(d) `max` σ subject to $0.30 \leq \mu(B) \leq 0.35$

Figure 6.8: Four LI configurations applied to the yellow-colored wall in Figure 6.1(a).

- Processing time: 11.2s per execution.
- μ values: $\mu_{\text{min}} = 7.20\times10^{-4}$, $\mu_{max} = 7.64\times10^{-4}$, $\mu_{\text{mean}} = 7.30\times10^{-4}$, and $\mu_{\text{sd}} = 1.43\times10^{-5}$.
- Image result: Figure 6.8(b).

⊕ Homogenize

LI Goal: Disseminate homogeneously, as much as possible, the light over the focus wall, also satisfying a set of bounding constraints.

Statistical Equivalent Goal: `min` $\sigma(B(\mathbf{s}_{yw}))$.

LI Constraints: $0.30 \leq \mu(B) \leq 0.35$.

Results:

- Processing time: 11.4s per execution.
- σ values for all executions: $\sigma_{\text{min}} = 0.066$, $\sigma_{\text{max}} = 0.072$, $\sigma_{\text{mean}} = 0.068$, and $\sigma_{\text{sd}} = 0.0019$.
- Image result: Figure 6.8(c).

⊕ Vary

LI Goal: Obtain the highest variation in the wall illumination, satisfying a set of constraints.

Statistical Equivalent Goal: `max` $\sigma(B(\mathbf{s}_{yw}))$.

LI Constraints: $0.30 \leq \mu(B) \leq 0.35$.

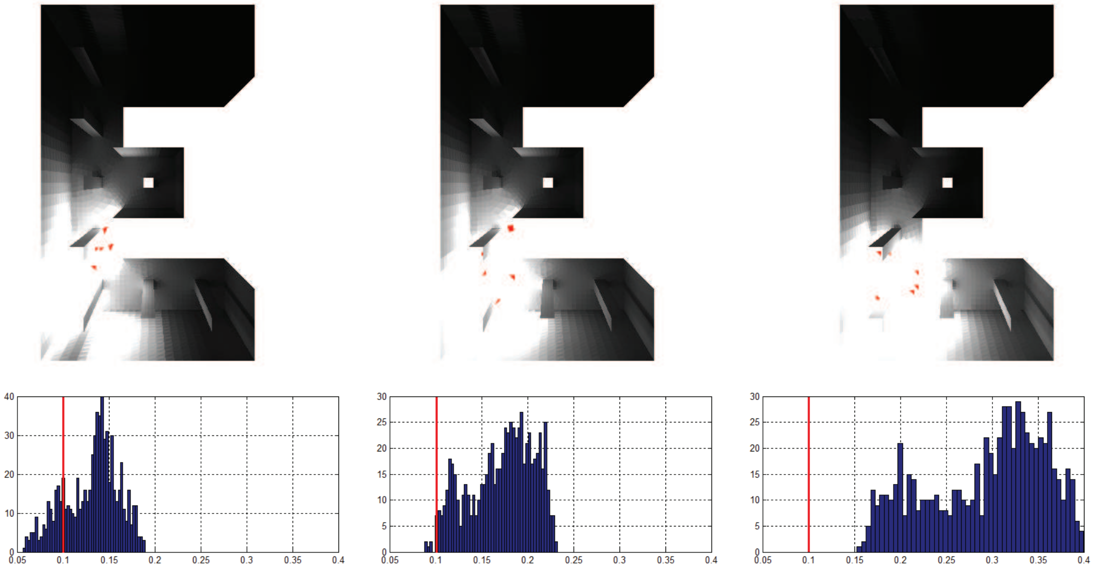
Results:

- Processing time: 11s per execution.
- σ values for all executions: $\sigma_{\text{min}} = 0.159$, $\sigma_{\text{max}} = 0.175$, $\sigma_{\text{mean}} = 0.165$, and $\sigma_{\text{sd}} = 0.0044$.
- Image result: Figure 6.8(d).

6.5.3 Evaluation of Chebyshev-Based Constraints

This experiment substitutes a set of constraints $G(B)$ $(0.1 \leq B(p), \forall p \in \mathbf{s}_{yw})$ by a Chebyshev constraint $G_{Ch}(E, \alpha)$ $(0.1 \leq \mu - \alpha\sigma)$. The optimization goal consists in the minimization of the function $\mu(B(\mathbf{s}_{yw}))$.

The parameter α is tested for three values: $\alpha = 1$, $\alpha = 2$ and $\alpha = 3$ (see Figure 6.9). According to Chebyshev inequality, when $\alpha = 3$, it means that up to 11% $(1/\alpha^2)$ of all patches could have values below the bound (0.1). However, our test shows that in fact, none of the 704 patches is below the bound for this case.



(a) $\alpha = 1$. Final values: μ=0.13 and σ=0.026.

(b) $\alpha = 2$. Final values: μ=0.17 and σ=0.034.

(c) $\alpha = 3$. Final values: μ=0.30 and σ=0.063.

Figure 6.9: Substitution of the constraints to a Chebyshev-based constraint with different values of α.

6.5.4 Finding the Best α Value

In this section two experiments are performed. The first experiment consists in finding the best α value for an ILP and the second one consists on drawing a $\Phi_{Ch}(\alpha)$ curve, similar to the curve shown in Figure 6.6. Both experiments are based on the problem stated in Section 6.5.3, and the $G(B)$ constraints are substituted by a $G_{Ch}(E, \alpha)$ constraint. The algorithm proposed in Section 6.4.2 for the determination of the best α value is used.

In Figure 6.10(a), the α value obtained in each iteration, used to solve the $Opt_{Ch}(\alpha)$ problem (red stars), the α_a value (blue square), and the α_b value (black circle) are shown. The first three iterations correspond to the first part of the algorithm, and the following eight iterations correspond to the bipartition algorithm. The resultant final interval was $(2.24, 2.25)$. As $\alpha_b = 2.25$ is the smallest α value known in region II, then this value is chosen as the α_{Sol} value. The execution of the algorithm takes about 120 s.

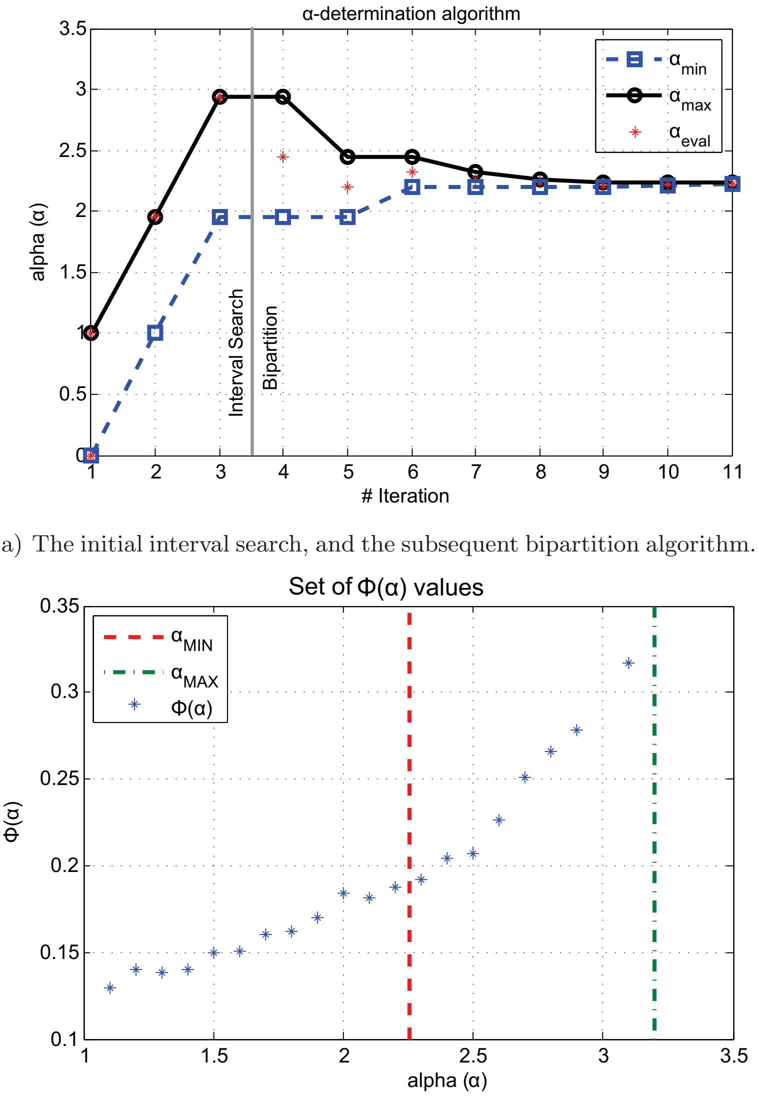
This experiment is performed 20 times in order to analyze the accuracy of the result, obtaining a good stability (standard deviation of 0.22). Moreover, the worst $\Phi_{Ch}(\alpha_{Sol})$ value within the 20 results is 0.214, a lower value than the 0.3 value found in Section 6.5.3.

The result of the second experiment is shown in Figure 6.10(b). The $Opt_{Ch}(\alpha)$ problem is solved for several α values. The α_{MIN} corresponds to the value found in the first experiment ($\alpha = 2.25$), and α_{MAX} corresponds to the lowest value found for which the $Opt_{Ch}(\alpha)$ problem has no feasible solution ($\alpha = 3.4$).

6.5.5 Performance Comparison

In this section is discussed the performance of our method in comparison against other state-of-the-art methods. For the tested scene, using statistical LI, our method took around 12 s to obtain a good solution. In terms of absolute timing, this result means an improvement regarding previous work [4, 5, 13], where reported times range from minutes to hours for more simple scenes and constraints. However, a rigorous comparison is difficult to carry out because of the different assumptions in the problem formulation. For instance, compared to [4], the main assumption differences can be mentioned. First, they only performed tests with a box-shaped scene without occlusion while the present approach uses a more complex scene. They also performed their tests with more optimization variable (27 against 12 in our case). Finally, another difference is that they use a multi-objective technique. Cassol et al. reported 0.022 s per evaluation step, whereas in our test 0.0004 s per evaluation was obtained.

A more feasible comparison can be done through the implementation of the algorithm exposed in [13]. In this work, the VNS method is used as the optimization technique and, in each iteration, the low rank radiosity technique calculates the radiosity values for all patches on the yellow wall. The original problem (Section 6.5.3) is solved, which includes the $G(B)$ constraints. The solution found, after 25000 iterations, is similar to the solution found in Section 6.5.4. The execution time took about 830 s (0.033 s per evaluation step), so the algorithm for α-determination presents a speed-up of about seven.



(a) The initial interval search, and the subsequent bipartition algorithm.

(b) Blue stars correspond to $\Phi(\alpha)$ values.

Figure 6.10: α-determination experiment results.

6.6 Conclusions and Future Work

A new methodology is presented to achieve lighting intentions for inverse lighting problems. The innovative approach introduces the use of the statistical parameters μ and σ as LI. It is shown that the computation can be efficiently performed obtaining a complexity of $O(n)$ and $O(n + e^2)$, respectively. The effectiveness obtained is based on a low-rank approach of the radiosity equation. These results allow us to perform thousands of global illumination evaluations on a standard computer, reducing drastically the overall optimization time required to solve inverse lighting problems. Based on these results and the Chebyshev's inequality, the substitution of the set of constraints $G(B)$ of the radiosity by a single constraint $G_{Ch}(E, \alpha)$ that depends only on the parameters μ, σ, and α is proposed. Finally, a procedure to obtain the optimum α parameter such that $G(\widetilde{B}(E_{Ch}(\alpha)))$ is satisfied and $\Phi_{Ch}(\alpha)$ is minimized is proposed.

We believe that our proposal could open a new avenue searching for optimal ILP solutions. The promising time reduction results could lead to the application of the method in more complex scenes and with more elaborated LI. Concerning further work, more effort could be focused on automatic transformation of expressive lighting intention into mathematical parameters, as for example μ, σ and α. This would approach the method to designer tools.

Acknowledgments

This work was partially funded by the TIN2010-20590-C02-02 project from Ministerio de Ciencia e Innovación, Spain, by program 720 from Universidad de la República, Uruguay, and by Programa de Desarrollo de las Ciencias Básicas, Uruguay.

Bibliography

[1] S. Russell, The Architecture of Light: Architectural Lighting Design, Concepts and Techniques: a Textbook of Procedures and Practices for the Architect, Interior Designer and Lighting Designer, Conceptnine, 2008.

[2] M. Contensin, Inverse lighting problem in radiosity, Inverse Problems in Engineering 10 (2) (2002) 131–152.

[3] P. S. Schneider, A. C. Mossi, F. H. R. Franca, F. L. de Sousa, A. J. da Silva Neto, Application of inverse analysis to illumination design, Inverse Problems in Science and Engineering 17 (6) (2009) 737–753.

[4] F. Cassol, P. S. Schneider, F. H. França, A. J. S. Neto, Multi-objective optimization as a new approach to illumination design of interior spaces, Building and Environment 46 (2) (2011) 331 – 338.

[5] F. Castro, E. del Acebo, M. Sbert, Energy-saving light positioning using heuristic search, Engineering Applications of Artificial Intelligence 25 (3) (2012) 566 – 582.

[6] DIALux, Dial gmbh, www.dial.de (Nov 2013).

[7] Relux, Relux informatik ag, www.relux.biz (Nov 2013).

[8] M. Cohen, J. Wallace, P. Hanrahan, Radiosity and realistic image synthesis, Academic Press Professional, Inc., San Diego, CA, USA, 1993.

[9] D. Luenberger, Y. Ye, Linear and Nonlinear Programming, International Series in Operations Research & Management Science, Springer, 2008.

[10] E.-G. Talbi, Metaheuristics: From Design to Implementation, Wiley Publishing, 2009.

[11] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Wiley-Interscience Series in Systems and Optimization, John Wiley & Sons, Chichester, 2001.

[12] P. Hansen, N. Mladenovic, Variable neighborhood search: Principles and applications, European Journal of Operational Research 130 (3) (2001) 449–467.

[13] E. Fernández, G. Besuievsky, Inverse lighting design for interior buildings integrating natural and artificial sources, Computers & Graphics 36 (8) (2012) 1096–1108.

[14] G. V. G. Baranoski, R. Bramley, J. G. Rokne, Eigen-analysis for radiosity systems, in: Proceedings of the Sixth International Conference on Computational Graphics and Visualization Techniques (Compugraphics '97), Vilamoura, Algarve, Portugal, 1997, pp. 193–201.

[15] I. Ashdown, Eigenvector radiosity, Master's thesis, Department of Computer Science, University of British Columbia, Vancouver, British Columbia (April 2001).

[16] M. Hasan, F. Pellacini, K. Bala, Matrix row-column sampling for the many-light problem, ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007) 26 (3).

[17] E. Fernández, Low-rank radiosity, in: O. Rodríguez, F. Serón, R. Joan-Arinyo, E. C. J. Madeiras, J. Rodríguez (Eds.), Proceedings of the IV Iberoamerican Symposium in Computer Graphics, Sociedad Venezolana de Computación Gráfica, DJ Editores, C.A., 2009, pp. 55–62.

[18] G. H. Golub, C. F. Van Loan, Matrix computations (3rd ed.), Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[19] A. C. Costa, A. A. de Sousa, F. N. Ferreira, Lighting design: A goal based approach using optimisation., in: D. Lischinski, G. W. Larson (Eds.), Rendering Techniques, Springer, 1999, pp. 317–328.

[20] J. K. Kawai, J. S. Painter, M. F. Cohen, Radioptimization - goal based rendering, in: ACM SIGGRAPH 93, Anaheim, CA, 1993, pp. 147–154.

[21] P. Hanrahan, D. Salzman, A rapid hierarchical radiosity algorithm, in: Computer Graphics, 1991, pp. 197–206.

[22] G. Canavos, Applied probability and statistical methods, Little, Brown, 1984.

Chapter 7

A Sample-Based Method for Computing the Radiosity Inverse Matrix [1]

The radiosity problem can be expressed as a linear system, where the light transport interactions of all patches of the scene are considered. Due to the amount of computation required to solve the system, the whole matrix is rarely computed and iterative methods are used instead. In this paper we introduce a new algorithm to obtain an approximation of the radiosity inverse matrix. The method is based on the calculation of a random sample of rows of the form factor matrix.

The availability of this matrix allows us to reduce the radiosity calculation costs, speeding up the radiosity process. This is useful in applications where the radiosity equation must be solved thousands of times for different light configurations. We apply it to solve inverse lighting problems, in scenes up to 170K patches. The optimization process used finds optimal solutions in nearly interactive times, which improves on previous work.

Keywords: Radiosity; Inverse Lighting Problems; Inverse Matrix Approximation

7.1 Introduction

The radiosity method is a global illumination solution that is extensively used in rendering applications and lighting design. The basic algorithm is based on the finite element approach, assuming that the light is reflected diffusely by the surfaces. In the classical approach, the explicit computation of the radiosity matrix is avoided due to the huge computational cost. One of the early formulations proposed is the progressive refinement solution [1], where a selection of form factors is computed, belonging to the most significant patches that must "shoot" their accumulated radiosity. Another important contribution is the hierarchical radiosity method [2], which reduces the $O(n^2)$ form factor calculations to $O(n)$, for a scene composed of n patches. These methods were successful at computing the global illumination of a scene. Further developments

[1] **Accepted by Computers & Graphics:** E. Fernández and G. Besuievsky *A Sample-Based Method for Computing the Radiosity Inverse Matrix.*

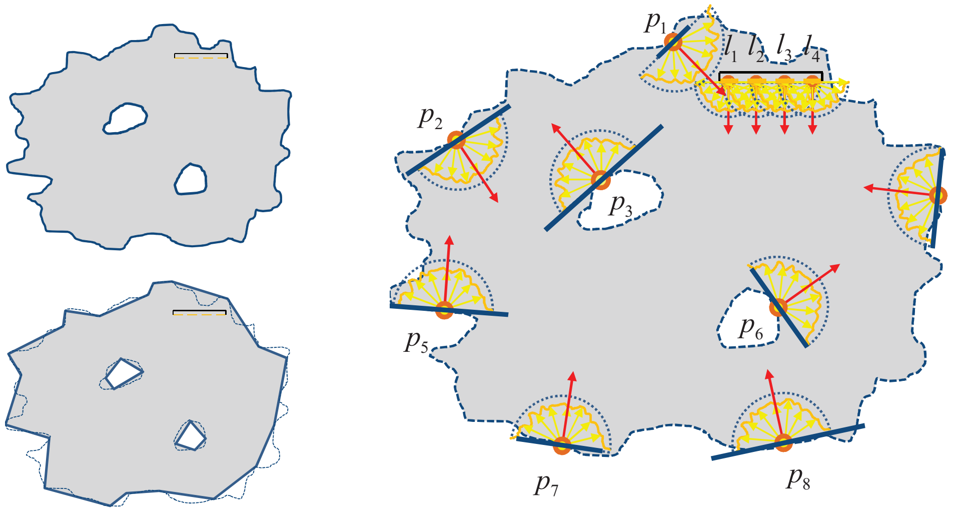
resulted in interactive techniques that are currently used in architectural lighting design applications and in visualization software.

In most cases, neither the full radiosity matrix nor its inverse are required to compute a 3D scene, and a few number of iterations is enough to approximate the radiosity solution. However, the inverse of the radiosity matrix, expressed as $\mathbf{M}=(\mathbf{I}-\mathbf{RF})^{-1}$ [1], is useful when the illumination of the scene must be calculated a huge number of times. In this equation, $\mathbf{I}$ is the identity matrix, $\mathbf{R}$ is a diagonal matrix containing the surface reflectivities, and $\mathbf{F}$ is the form factor matrix.

One kind of problems that requires a huge number of global illumination calculations is the inverse lighting problem (ILP). Unlike traditional direct global illumination calculations, in ILP the surfaces have lighting goals and constraints to satisfy, and the shape, position and power of emitters are the variables of the problem. We follow the notion that ILPs are formulated for static geometries, as laid down by Marschner [3].

These problems could be formulated as optimization problems, which usually are implemented as iterative algorithms where at least one radiosity calculation is performed in each iteration. Solving ILP is a challenge since the global illumination computation should be solved for thousands of possible configurations of the emitters. One of the limitations observed in previous approaches of ILP is that they are only feasible for managing simple scenes.

We present results obtained from scenes composed up to 170K triangles. As happens with any 3D model, the fine grain scene can be simplified into a rough approximation using standard polygon reduction techniques or clustering (Figure 7.1(a)). Instead of doing that, our proposal is based on the random selection of a small set of patches $\mathcal{P}$, and the calculation of the form factors between those patches and all the scene patches. Our method generates a "sample-based scene", or $\mathcal{S}_{\mathcal{P}}$ (Figure 7.1(b)).



(a) Fine grain scene $\mathcal{S}$ (above) and coarse grain scene (below).

(b) A sample-based scene ($\mathcal{S}_{\mathcal{P}}$).

Figure 7.1: Three different representations of a 2D projected scene.

This method facilitates the calculation of an approximate representation of **M**. A small selection of patches provides enough information to calculate a good approximation of the radiosity values. This is mainly caused by the spatial coherence implicit in the radiosity method.

If we were to build **M** based on a rough approximation of the scene, there would be three key issues to solve. One issue is how to simplify the geometry when we try to limit the error of the global illumination of the scene. A second issue is the difficulty in reducing the number of patches to a few thousands, when the scene contains many arcs, windows, statues, furniture and other architectural elements. Finally, another issue is the mapping of the solution into the original model. However, the use of a sample-based scene avoids the treatment of the key issues, facilitates the exploration of many possible configuration of $\mathcal{S}_{\mathcal{P}}$, and makes possible to estimate the error of the radiosity values.

Figure 7.1(b) shows a schematic representation of the information managed. In this example, a sample-scene composed of eight patches is taken and the form factors between the sample and the rest of the scene are computed to obtain the rows of **F** (Figure 7.2). Based on the symmetry of **AF**, where **A** is a diagonal matrix containing the area values, the columns of **F** can be calculated.

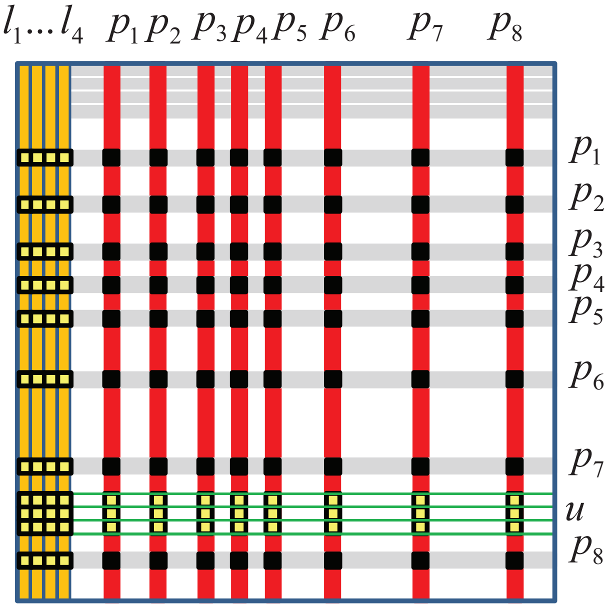


Figure 7.2: Sample rows/columns of **F**.

The information extracted from the sample is combined through the use of operators that process the light interactions in the scene. In Section 7.3 a generic definition of each operator is presented, and in Section 7.4 a specific definition for them is proposed.

The main contributions of this paper are:

- A description of the radiosity inverse matrix **M** using operators and a sample-based scene.
- A matrix-based implementation of the operators.
- A method to estimate the amount of patches in the sample-based scene.

- A significant speed up in the resolution of inverse lighting problems for medium size models.

The rest of the paper is divided into six sections. In Section 7.2 we describe previous work. Sections 7.3 and 7.4 are dedicated to explain the details of the method proposed. In the following section we analize the error estimation and an estimation of the sample size. Section 7.6 describes the experimental results, and finally the conclusions and further work are summarized in Section 7.7.

Table 7.1: Symbol notation and meaning.

L	set of all potential light emitters
l_i	emitter i
p_i	patch i
$\mathcal{P}$	$\{p_1 \cdots p_{\lvert\mathcal{P}\rvert}\}$; sample of $\mathcal{S}$; $\mathcal{P}\subset\mathcal{S}$
$\mathcal{P}_L$	$\{l_1 \cdots l_{\lvert\mathcal{P}_L\rvert}\}$; sample of L; $\mathcal{P}_L\subseteq L$
$\mathcal{S}$	set of all scene patches
$\mathcal{S}_\mathcal{P}$	sample-based scene
u	surface; $u\subseteq\mathcal{S}$
n	$\lvert\mathcal{S}\rvert$; number of patches in $\mathcal{S}$
$\mathbf{M}_\mathcal{S}$	radiosity operator for the scene $\mathcal{S}$
$\mathbf{P}_{u_2\Leftarrow u_1}$	projection operator from u_1 to u_2
$\mathbf{S}_{b\Leftarrow a}$	selection operator from a to b
$A_\mathcal{S}$	area of the scene $\mathcal{S}$
B	radiosity in the entire scene
$\widetilde{B}$	approximation of the radiosity values
B_u	total radiosity in surface u
B_u^D	direct radiosity in surface u
B_u^I	indirect radiosity in surface u
$\phi^{(\eta)}$	$\{\widetilde{B}^{I(1)} \cdots \widetilde{B}^{I(\eta)}\}$; sample of $\widetilde{B}^I$
E	emission
ε, $\mathcal{E}$	error
$r_{\mathit{ff}}(i)$	$(\sum$ all $\mathit{ff})/(\sum$ sampled $\mathit{ff})$, $\forall\ \mathit{ff} \in \mathbf{F}(i,:)$
$\mathbf{A}$	diagonal matrix with area values of $\mathcal{S}$
$\mathbf{A}_{\mathcal{S}_\mathcal{P}}$	diagonal matrix with area values of $\mathcal{S}_\mathcal{P}$
$\mathbf{F}$	form factors matrix
$\mathbf{F}_{\mathcal{S}_\mathcal{P}}$	form factors matrix of $\mathcal{S}_\mathcal{P}$
$\mathbf{I}$	identity matrix
$\mathbf{M}$	inverse of the radiosity matrix
$\widetilde{\mathbf{M}}$	low-rank approximation to $\mathbf{M}$
$\mathbf{R}$	diagonal matrix with reflectivity indexes
$\mathbf{Y}$, $\mathbf{V}$	components of $\widetilde{\mathbf{M}}$ ($n\times k$ matrices, $n\gg k$)
μ	mean
$\bar{\mu}_A$	sample mean, weighted by the areas
σ	standard deviation
$\bar{\sigma}$	sample standard deviation
LTI	length of the tolerance interval

7.2 Related Work

In this section we summarize previous work in factorization techniques, a brief analysis is presented about the connection between our technique and previous global illumination approaches, and the previous uses of the inverse matrices in ILP are exposed.

7.2.1 Matrix approximation by Factorization Techniques

The factorization of a matrix $\mathbf{A}_{m\times n}$ consists in the construction of a pair of matrices $\mathbf{B}_{m\times k}$, and $\mathbf{C}_{k\times n}$ such that $\mathbf{A}{\approx}\mathbf{BC}$. The term k is the numerical rank of the approximation [4]. The factorization techniques that use a small amount of data play an important role in several areas of scientific computing and data analysis. Matrices can be huge, their elements can be missing or inaccurate, or the data transfer can be a bottle neck difficult to overcome.

Several methods have been developed for matrix approximations: sparsification, column selection methods, and approximation by submatrices, among others [5]. In the approximation by submatrices, Goreinov et al. have developed the CUR decomposition. In this decomposition a matrix $\mathbf{A}$ is approximated by a product $\mathbf{CUR}$. where $\mathbf{C}$ and $\mathbf{R}$ denote column and row submatrices of $\mathbf{A}$, and $\mathbf{U}$ is a small square matrix [6].

Factorization techniques have also been applied to solve radiosity equations. Wang et al. [7] presented the kernel Nyström method, based on CUR decomposition and a kernel method, to reconstruct the light transport matrix from a small set of images. A similar goal is pursued by O'Toole and Kutulakos [8], using Krylov subspace methods. In [9], the full matrix computation of $(\mathbf{I}-\mathbf{RF})^{-1}$ is avoided through a low-rank radiosity method.

It is very likely that the $\mathbf{F}$ matrix of the radiosity equation has a low numerical rank. This happens because each row $\mathbf{F}(p,:)$ is computed based on the scene view from the geometrical patch p. As close patches have a similar view of the scene, $\mathbf{F}$ also has similar rows, resulting in the reduction of its numerical rank. References about low-rank properties of radiosity and radiance matrices can be found in Baranoski et al. [10], Ashdown [11], Hasan et al. [12], and Fernández [9].

A matrix $\mathbf{F}$ with low numerical rank can be approximated by the product of two matrices $(\mathbf{UV}^T)$, both matrices with dimension $n{\times}k$ $(n{\gg}k)$, without the loss of relevant information. The memory requirement for both matrices is $O(nk)$, which is significantly less than $O(n^2)$ required to store $\mathbf{F}$. This memory saving allows to store $\mathbf{U}$ and $\mathbf{V}^T$ in the main processor memory of desktop computers for scenes with several hundred thousand patches. In Fernández [9], the calculation of $\mathbf{U}$ and $\mathbf{V}$ is implemented through the execution of an hemicube algorithm n times, thus slowing the precomputation process $(O(n^2))$. Also, the construction of two meshes with different granularity is required (one mesh containing k patches and another with n patches).

Another step in the radiosity calculation is the calculation of the inverse of the radiosity matrix, which allows to simplify the radiosity calculation to a matrix·vector product [9]:

$$B = \mathbf{M}E = (\mathbf{I}-\mathbf{RF})^{-1}E \;\approx\; (\mathbf{I}-\mathbf{UV}^T)^{-1}E = (\mathbf{I}+\mathbf{YV}^T)E = \widetilde{\mathbf{M}}E = \widetilde{B} \qquad (7.1)$$

In Equation 7.1, matrix $\mathbf{Y}$ has dimension $n{\times}k$ and is calculated through the use of the Sherman-Morrison-Woodbury formula [4], in $O(nk+k^3)$ operations.

7.2.2 Global Illumination

Our method is also related to other global illumination approaches that obtain a good approximated solution from a structured selection of samples of the scene, like instant radiosity [13], precomputed light paths [14], or generic many lights methods [15]. Sloan et al. [16] precompute a radiance transfer process based on spherical harmonics, which includes a clustered principal component analysis to compress the volume of information generated.

In our method, it is used a starting set of samples but the light paths are not explicitly followed. We use these samples as a rough approximation of the scene, to build a compressed version of the radiosity equation. We are not focused on obtaining a realistic rendering but to obtain an inverse matrix approximation for forward uses.

7.2.3 The inverse matrix computation for ILP

Matrix inversion is one of the strategies used to accelerate the inverse lighting process based on radiosity. The inverse matrix could be used to explore each step of an optimization algorithm, for static scenes where their parameters are the positioning of the light sources, and their emitting power.

Different algorithms were proposed to build inverse matrices. Contensin uses the pseudo-inverse method to solve the system [17]. In [18], the truncated singular value decomposition (TSVD) is used for regularization and resolution of the system. In both cases, the complexity of each algorithm is $O(n^2k)$ and the memory requirements are $O(n^2)$, where n is the number of patches. In [19] the low-rank radiosity method developed in [9] is applied, allowing to process one hundred solutions per second on a twenty thousand patch scene. Fernández and Besuievsky [20] also used a low-rank approach, but instead of computing the full radiosity solution at each step they introduce statistical operators that can be used to optimize the illumination. In this case the complexity is reduced to almost $O(n)$, speeding up about seven times the computation of their previous work. However, one drawback of all these inverse lighting techniques is that they are not well-suited for large scenes.

7.3 Sample Based Radiosity

In this section we start explaining some introductory concepts, and continue defining three basic operators: *selection*, *projection*, and *radiosity operator*. Finally, these operators are used to define an approximation of the radiosity inverse matrix. The implementation details for each operator are explained in Section 7.4.

7.3.1 Sample of F

Figure 7.1(b) shows the information extracted from a given scene $\mathcal{S}$ to build $\widetilde{\mathbf{M}}$, a low-rank approximation to $\mathbf{M}$. Another representation of the information extracted is described in Figure 7.2. In both representations we can see that: a sample $\mathcal{P}$=$\{p_1 \dots p_8\}$ of $\mathcal{S}$ and a sample $\mathcal{P}_L$=$\{l_1 \dots l_4\}$ of the potential emission area L are taken. As can be observed below in Equation 7.3 and Figure 7.4, L and $\mathcal{P}_L$ must be defined in the precomputation step to allow the calculation of the direct radiosity B^D. If $\widetilde{\mathbf{M}}$ is only used in the calculation of the indirect radiosity, then the information of L is not needed during the precomputation step.

Using the scene information extracted from the view of each patch of $\mathcal{P}$, the radiosity matrix of a sample-based scene $\mathcal{S}_\mathcal{P}$ is built. The patches in $\mathcal{S}_\mathcal{P}$ have the same position and orientation as the $\mathcal{P}$ patches, but different area size. The symbol u represents a surface whose radiosity values have to be computed. This surface could cover the whole scene ($u \subseteq \mathcal{S}$).

The information provided by $\mathcal{P}$ and $\mathcal{P}_L$ is shown in Figure 7.2. The highlighted cells in the intersection of the $\mathcal{P}_L$ columns and the $\mathcal{S}_\mathcal{P}$ rows, are used to calculate direct lighting to $\mathcal{S}_\mathcal{P}$, and the marked cells in $\mathcal{P}$ columns are used to calculate the global illumination in $\mathcal{S}_\mathcal{P}$. Finally, the highlighted cells on u rows are used to calculate the radiosity of u. Figure 7.3 shows a Cornell box with a possible $\mathcal{P}$ set.

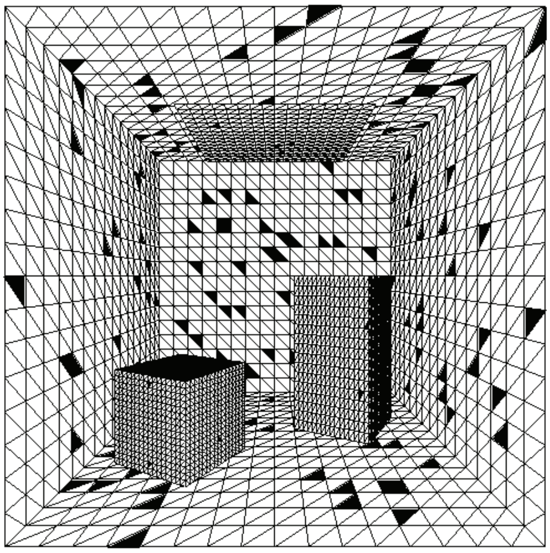


Figure 7.3: $\mathcal{P}$ containing 200 patches in a Cornell box.

The use of a subset of columns and rows resembles some aspects of the CUR method mentioned in Section 7.2, but our approach takes also in consideration the properties of the matrices involved in the radiosity equation.

7.3.2 Sample-Based Scene Radiosity

A *sample-based scene* $\mathcal{S}_\mathcal{P}$ can be seen as a simplification of $\mathcal{S}$. We propose to build the radiosity equation of $\mathcal{S}_\mathcal{P}$ (see Section 7.4.2) bypassing the geometric modeling stage. The equation is built from the information samples extracted from the original scene, wishing to contain all the relevant information of $\mathcal{S}$.

The light emitted by L is projected into $\mathcal{S}_\mathcal{P}$, and the radiosity equation is solved to find the radiosity in $\mathcal{S}_\mathcal{P}$. Then, the light in $\mathcal{S}_\mathcal{P}$ is projected into u obtaining its indirect radiosity (see Figures 7.4(b), (c), and (d)).

The direct radiosity of u can be obtained using shadow mapping or another GPU-based technique for light projection. In our process, the direct light is projected from $\mathcal{P}_L$ into u (Figure 7.4(e)) through the use of the $\mathcal{P}_L$ columns of $\mathbf{F}$. The final radiosity value in u is composed of the sum of direct plus indirect lighting (Figure 7.4(f)).

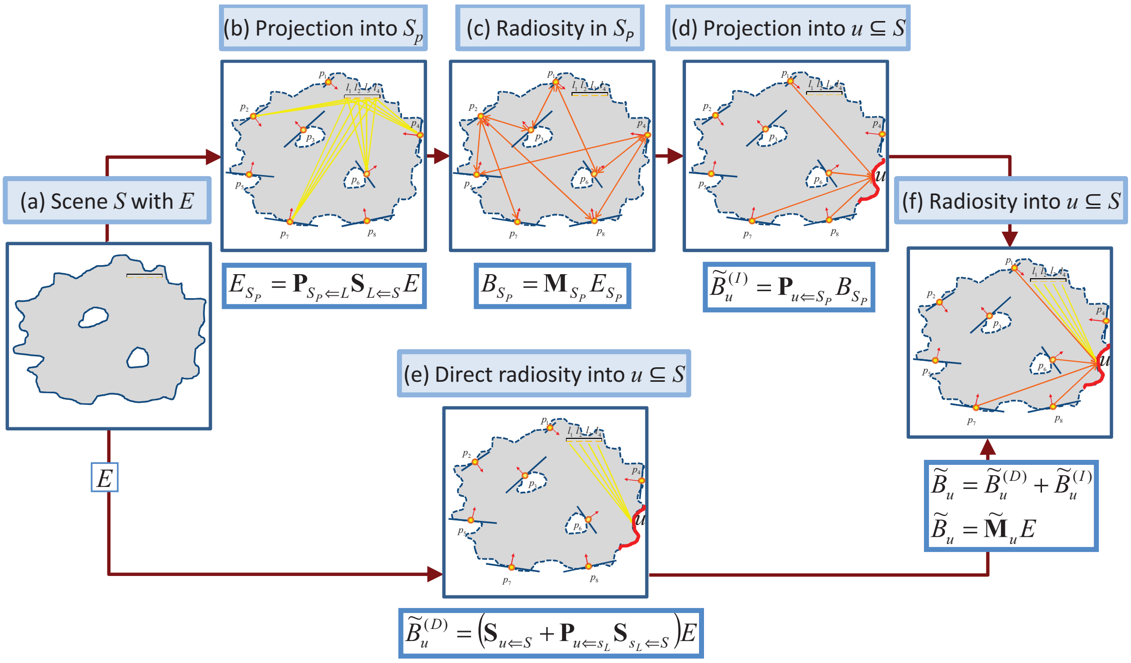

(b) Projection into S_p
(c) Radiosity in S_p
(d) Projection into $u \subseteq S$
(a) Scene S with E
(f) Radiosity into $u \subseteq S$
$E_{S_P} = \mathbf{P}_{S_P \Leftarrow L}\mathbf{S}_{L \Leftarrow S}E$
$B_{S_P} = \mathbf{M}_{S_P}E_{S_P}$
$\widetilde{B}_u^{(I)} = \mathbf{P}_{u \Leftarrow S_P}B_{S_P}$
(e) Direct radiosity into $u \subseteq S$
E
$\widetilde{B}_u = \widetilde{B}_u^{(D)} + \widetilde{B}_u^{(I)}$
$\widetilde{B}_u = \widetilde{\mathbf{M}}_u E$
$\widetilde{B}_u^{(D)} = (\mathbf{S}_{u \Leftarrow S} + \mathbf{P}_{u \Leftarrow s_L}\mathbf{S}_{s_L \Leftarrow S})E$
u

Figure 7.4: Overview of the sample based Radiosity approach.

7.3.3 Operators

The radiosity calculation of u is modeled through the use of operators that shape the selection of patches, the projection of the light between surfaces and the radiosity calculation.

Selection ($\mathbf{S}_{b \Leftarrow a}$): Given a vector a containing light information, $\mathbf{S}_{b \Leftarrow a}$ generates a smaller vector b with selected values of a. The objective of this operator is to reduce the amount of information used. Any sample extraction method or clustering technique of the vector a may be a valid selection criterion.

Projection ($\mathbf{P}_{u_2 \Leftarrow u_1}$): Given a vector with the radiosity values of a surface u_1, this operator calculates a new vector containing the radiosity in u_2 of the first bounce of the light radiated by u_1.

Radiosity Operator ($\mathbf{M}_{Sc}$): Given a vector with emission values in a scene Sc, this operator returns a vector with the radiosity solution in Sc.

These operators can be built in many ways. Each one could be associated to a matrix, but also an algorithmic approach could be formulated for them. The radiosity operator could be based on any radiosity solver, and the projection operator could be implemented with a ray tracer.

7.3.4 A Low-Rank Approximation of M

If the operators defined above are built as matrices, they can be used to calculate a low-rank approximation of **M**. Based on Figure 7.4 (from (b) to (d)), the calculation of the approximated indirect radiosity in u could be expressed as Equation 7.2:

$$\widetilde{B}_u^I = \mathbf{P}_{u \Leftarrow \mathcal{S}_\mathcal{P}} \mathbf{M}_{\mathcal{S}_\mathcal{P}} \mathbf{P}_{\mathcal{S}_\mathcal{P} \Leftarrow L} \mathbf{S}_{L \Leftarrow \mathcal{S}} E \tag{7.2}$$

where E is the emission of the scene. Also, the approximated direct illumination (Figure 7.4(e)) is calculated using Equation 7.3:

$$\widetilde{B}_u^D = (\mathbf{S}_{u \Leftarrow \mathcal{S}} + \mathbf{P}_{u \Leftarrow \mathcal{P}_L} \mathbf{S}_{\mathcal{P}_L \Leftarrow \mathcal{S}}) E \tag{7.3}$$

Combining Equations 7.2 and 7.3 we obtain:

$$\widetilde{B}_u = \widetilde{B}_u^D + \widetilde{B}_u^I = (\mathbf{S}_{u \Leftarrow \mathcal{S}} + \mathbf{P}_{u \Leftarrow \mathcal{P}_L} \mathbf{S}_{\mathcal{P}_L \Leftarrow \mathcal{S}} + \mathbf{P}_{u \Leftarrow \mathcal{S}_\mathcal{P}} \mathbf{M}_{\mathcal{S}_\mathcal{P}} \mathbf{P}_{\mathcal{S}_\mathcal{P} \Leftarrow L} \mathbf{S}_{L \Leftarrow \mathcal{S}}) \, E \tag{7.4}$$

which also can be expressed as:

$$\widetilde{B}_u = \left(\mathbf{S}_{u \Leftarrow \mathcal{S}} + \mathbf{Y}\mathbf{V}^T\right) E = \widetilde{\mathbf{M}} E \tag{7.5}$$

where $\mathbf{Y}$ (of dimension $|u| \times (|\mathcal{P}_L| + |\mathcal{P}|)$) and $\mathbf{V}$ (of dimension $(|\mathcal{P}_L| + |\mathcal{P}|) \times n$) are composed of operators:

$$\mathbf{Y} = \Big(\mathbf{P}_{u \Leftarrow \mathcal{P}_L} | \mathbf{P}_{u \Leftarrow \mathcal{S}_\mathcal{P}}\Big) \quad ; \quad \mathbf{V}^T = \left(\frac{\mathbf{S}_{\mathcal{P}_L \Leftarrow \mathcal{S}}}{\mathbf{M}_{\mathcal{S}_\mathcal{P}} \mathbf{P}_{\mathcal{S}_\mathcal{P} \Leftarrow L} \mathbf{S}_{L \Leftarrow \mathcal{S}}}\right)$$

Once $\mathbf{Y}$ and $\mathbf{V}^T$ are built, $\widetilde{B}_u$ can be computed with complexity $O(|u|(|\mathcal{P}_L| + |\mathcal{P}|))$. Also it should be noted that, when u=$\mathcal{S}$, the $\mathbf{S}_{u \Leftarrow \mathcal{S}}$ operator is the identity matrix and Equation 7.5 is similar to Equation 7.1.

The above formulation is useful when $\mathcal{P}_L \subset L$ and we want to use L for indirect lighting. For practical purposes could be suitable to reduce L to its sample $\mathcal{P}_L$, so L=$\mathcal{P}_L$. In this case $\mathbf{S}_{\mathcal{P}_L \Leftarrow \mathcal{S}}$ and $\mathbf{S}_{L \Leftarrow \mathcal{S}}$ are the same operator, so Equation 7.4 can be transformed into Equation 7.6. $\mathbf{Y}$ is now a $|u| \times |L|$ matrix and $\mathbf{V}^T$=$\mathbf{S}_{L \Leftarrow \mathcal{S}}$ is a sparse $|L| \times n$ matrix. Now, the calculation of $\widetilde{B}_u$ has complexity $O(|u||L|)$.

$$\widetilde{B}_u = (\mathbf{S}_{u \Leftarrow \mathcal{S}} + \underbrace{(\mathbf{P}_{u \Leftarrow L} + \mathbf{P}_{u \Leftarrow \mathcal{S}_\mathcal{P}} \mathbf{M}_{\mathcal{S}_\mathcal{P}} \mathbf{P}_{\mathcal{S}_\mathcal{P} \Leftarrow L})}_{\mathbf{Y}} \underbrace{\mathbf{S}_{L \Leftarrow \mathcal{S}}}_{\mathbf{V}^T}) E \tag{7.6}$$

7.4 Building Operators as Matrices

In the construction of the matrices, the information used is composed by the reflectivity indices and the areas of all the scene patches, the set of potential emitters, and the rows of $\mathbf{F}$ associated to $\mathcal{P}$ and $\mathcal{P}_L$ (Figure 7.2). The complexity of the matrix generation has $O(|\mathcal{P}|^3 + |u|(|\mathcal{P}_L| + |\mathcal{P}|))$, where $O(|\mathcal{P}|^3)$ is derived from the calculation of $\mathbf{M}_{\mathcal{S}_\mathcal{P}}$, with $|\mathcal{P}| \ll n$, and the complexities $O(|u||\mathcal{P}_L|)$ and $O(|u||\mathcal{P}|)$ are derived from the construction of the projection operators. As $\mathcal{P}$ is small, $O(|\mathcal{P}|^3)$ has not a great influence in the overall calculation time.

7.4.1 Previous Considerations

We base our reasoning in the assumption that the following set of premises is satisfied:

Selection of Patches: Patches of $\mathcal{S}$ are selected randomly and are grouped into $\mathcal{P}$. A similar process is executed in L to build $\mathcal{P}_L$.

The probability to choose a patch is proportional to its area size. The set $\mathcal{S}$ is a sample without replacement, meaning that each patch is selected at most one time.

Form Factor Simplification: The form factor between a differential area di and the patch j, is equal to the form factor between i and j (Equation 7.7).

$$\mathbf{F}(di, j) = \mathbf{F}(i, j) \tag{7.7}$$

The form factor value $\mathbf{F}(i, j)$ is independent of the area of patch i. This property is approximately satisfied when the distance between patches is large compared to their size [21]. This is one of the foundations of form factors calculation using hemicubes.

Radiosity Sample: The mean radiosity coming from $\mathcal{P}$ that arrives to $i{\in}\mathcal{P}$ per unit of form factor, is equal to the mean radiosity coming from $\mathcal{S}$ that arrives to $i{\in}\mathcal{P}$ per unit of form factor (Equation 7.8).

$$\frac{\sum_{k\in\mathcal{P}} \mathbf{F}(i,k)B(k)}{\sum_{k\in\mathcal{P}} \mathbf{F}(i,k)} = \frac{\sum_{j\in\mathcal{S}} \mathbf{F}(i,j)B(j)}{\sum_{j\in\mathcal{S}} \mathbf{F}(i,j)} + \varepsilon(i) \quad \forall i \in \mathcal{P} \tag{7.8}$$

This equation expresses the connection between the mean of a sample and the mean of a population. The error term $\varepsilon(i)$ in Equation 7.8 is usually unknown. Following the central limit theorem (CLT) [22], $\varepsilon(i)$ has a normal distribution with $\mu(\varepsilon(i)){=}0$ and $\sigma(\varepsilon(i)){=}\sigma(\widetilde{B})/\sqrt{|\mathcal{P}|}$, $\forall i{\in}\mathcal{P}$. In Section 7.5 we study the influence of $\varepsilon(i)$ on $\widetilde{B}$.

Symmetry of $\mathbf{A}_{\mathcal{S}_\mathcal{P}}\mathbf{F}_{\mathcal{S}_\mathcal{P}}$ In this product, $\mathbf{A}_{\mathcal{S}_\mathcal{P}}$ is a diagonal matrix containing the area of $\mathcal{S}_\mathcal{P}$ patches.

$$\mathbf{A}_{\mathcal{S}_\mathcal{P}}(i,i)\mathbf{F}_{\mathcal{S}_\mathcal{P}}(i,j) = \mathbf{A}_{\mathcal{S}_\mathcal{P}}(j,j)\mathbf{F}_{\mathcal{S}_\mathcal{P}}(j,i) \quad \forall i,j \in \mathcal{S}_\mathcal{P} \tag{7.9}$$

This property must be fulfilled by any form factor matrix.

Invariance of the Scene Area: The area of $\mathcal{S}$ must be equal to the area of the sample-based scene $\mathcal{S}_\mathcal{P}$.

$$A_{\mathcal{S}_\mathcal{P}} = \sum_{\forall i\in\mathcal{S}_\mathcal{P}} A_{\mathcal{S}_\mathcal{P}}(i) = \sum_{\forall j\in\mathcal{S}} A(j) = A_\mathcal{S} \tag{7.10}$$

This property is useful to model the energy flux between $\mathcal{S}$ and $\mathcal{S}_\mathcal{P}$.

7.4.2 Computing $\mathbf{M}_{\mathcal{S}_\mathcal{P}}$ Operator

The radiosity of any patch i belonging to the sample $\mathcal{P}$ must fulfill [1]:

$$B(i) = E(i) + R(i) \sum_{\forall j \in \mathcal{S}} \mathbf{F}(i,j)B(j) \qquad \forall i \in \mathcal{P}$$

When this equation is combined with Equation 7.8 (ignoring $\varepsilon(i)$), we obtain Equation 7.11:

$$\begin{aligned} B(i) &= E(i) + R(i) r_{\mathit{ff}}(i) \sum_{\forall k \in \mathcal{P}} \mathbf{F}(i,k)B(k) \ \ \forall i \in \mathcal{P} \qquad (7.11)\\ \text{where } r_{\mathit{ff}}(i) &= \frac{\sum_{\forall j \in \mathcal{S}} \mathbf{F}(i,j)}{\sum_{\forall k \in \mathcal{P}} \mathbf{F}(i,k)} \end{aligned}$$

In Equation 7.11 the values of B are limited to those belonging to $\mathcal{P}$. This equation can be expressed also as:

$$\begin{gathered} (\mathbf{I} - \mathbf{R}_\mathcal{P} \mathcal{F}_\mathcal{P}) B_\mathcal{P} = E_\mathcal{P} \qquad (7.12)\\ \text{where } \mathcal{F}_\mathcal{P}(i,k) = r_{\mathit{ff}}(i)\mathbf{F}(i,k) \qquad \forall i,k \in \mathcal{P} \end{gathered}$$

This equation looks similar to the well-known radiosity equation. $\mathcal{F}_\mathcal{P}$ is a square matrix with nonnegative values where the sum of elements in each row is not greater than one. Also, Equation 7.12 allows to obtain the radiosity solutions for $\mathcal{P}$. For this reason, we interpret Equation 7.12 as the radiosity equation for a sample-based scene $\mathcal{S}_\mathcal{P}$, where their patches satisfy the form factor values defined in $\mathcal{F}_\mathcal{P}$. Hence, we redefine Equation 7.12 as:

$$\begin{gathered} (\mathbf{I} - \mathbf{R}_{\mathcal{S}_\mathcal{P}} \mathbf{F}_{\mathcal{S}_\mathcal{P}}) B_{\mathcal{S}_\mathcal{P}} = E_{\mathcal{S}_\mathcal{P}}, \qquad (7.13)\\ \text{where: } B_{\mathcal{S}_\mathcal{P}} = B_\mathcal{P},\ E_{\mathcal{S}_\mathcal{P}} = E_\mathcal{P},\ \mathbf{R}_{\mathcal{S}_\mathcal{P}} = \mathbf{R}_\mathcal{P},\ \mathbf{F}_{\mathcal{S}_\mathcal{P}} = \mathcal{F}_\mathcal{P} \end{gathered}$$

If $\mathcal{P}$ is small (a few thousand patches), then $\mathbf{F}_{\mathcal{S}_\mathcal{P}}$ is small enough to allow the calculation of $\mathbf{M}_{\mathcal{S}_\mathcal{P}}$ in a standard PC (Equation 7.14).

$$\mathbf{M}_{\mathcal{S}_\mathcal{P}} = (\mathbf{I} - \mathbf{R}_{\mathcal{S}_\mathcal{P}} \mathbf{F}_{\mathcal{S}_\mathcal{P}})^{-1} \qquad (7.14)$$

7.4.3 The Area of $\mathcal{S}_\mathcal{P}$ Patches ($\mathbf{A}_{\mathcal{S}_\mathcal{P}}$)

Given the Equations 7.9, 7.12, and 7.13, it is possible to say that:

$$A_{\mathcal{S}_\mathcal{P}}(i) r_{\mathit{ff}}(i) \mathbf{F}(i,j) = A_{\mathcal{S}_\mathcal{P}}(j) r_{\mathit{ff}}(j) \mathbf{F}(j,i) \qquad \forall i,j \in \mathcal{S}_\mathcal{P}$$

Considering $A_{\mathcal{S}_\mathcal{P}}(i){=}a_i A(i)\ \forall i{\in}\mathcal{S}_\mathcal{P}$, the above equation is now expressed as:

$$a_i A(i) r_{\mathit{ff}}(i) \mathbf{F}(i,j) = a_j A(j) r_{\mathit{ff}}(j) \mathbf{F}(j,i) \qquad \forall i,j \in \mathcal{S}_\mathcal{P}$$

but, as $A(i)\mathbf{F}(i,j){=}A(j)\mathbf{F}(j,i)\ \forall i,j{\in}\mathcal{P}$ then:

$$a_i r_{\mathit{ff}}(i) = a_j r_{\mathit{ff}}(j) \qquad \forall i,j \in \mathcal{P}$$

then the product $a_i r_{ff}(i)=\mathcal{K}$ has a constant value independent of i. If $\mathcal{K}$ is known, the area of the $\mathcal{S}_\mathcal{P}$ patches can be computed as:

$$A_{\mathcal{S}_\mathcal{P}}(i) = a_i A(i) = \mathcal{K}\frac{A(i)}{r_{ff(i)}} \qquad \forall i \in \mathcal{S}_\mathcal{P} \tag{7.15}$$

To find $\mathcal{K}$, we combine Equations 7.10 and 7.15:

$$A_\mathcal{S} = A_{\mathcal{S}_\mathcal{P}} = \sum_{i\in\mathcal{S}_\mathcal{P}} A_{\mathcal{S}_\mathcal{P}}(i) = \mathcal{K}\sum_{i\in\mathcal{P}} \frac{A(i)}{r_{ff(i)}}$$

so, the value of $\mathcal{K}$ is calculated:

$$\mathcal{K} = \left(\frac{A_\mathcal{S}}{\sum_{j\in\mathcal{P}} \frac{A(j)}{r_{ff(j)}}}\right)$$

Finally, substituting $\mathcal{K}$ in Equation 7.15, $A_{\mathcal{S}_\mathcal{P}}(i)$ is calculated:

$$A_{\mathcal{S}_\mathcal{P}}(i) = \left(\frac{A_\mathcal{S}}{\sum_{j\in\mathcal{P}} \frac{A(j)}{r_{ff(j)}}}\right)\frac{A(i)}{r_{ff(i)}} \qquad \forall i \in \mathcal{P} \tag{7.16}$$

7.4.4 Computing P Operators

The projection operators used in Equation 7.4 are: $\mathbf{P}_{u\Leftarrow\mathcal{P}_L}$, $\mathbf{P}_{\mathcal{S}_\mathcal{P}\Leftarrow L}$, and $\mathbf{P}_{u\Leftarrow\mathcal{S}_\mathcal{P}}$.

$\mathbf{P}_{u\Leftarrow\mathcal{P}_L}$ is a light projection between a source and destination, both belonging to $\mathcal{S}$. Its matrix representation is expressed in Equation 7.17, where A_L is the total area of emission and $A_{\mathcal{P}_L}$ is the area of the emission sample $\mathcal{P}_L$. When $L=\mathcal{P}_L$, then $A_L/A_{\mathcal{P}_L}=1$.

$$\mathbf{P}_{u\Leftarrow\mathcal{P}_L} = \mathbf{RF}(u,\mathcal{P}_L)\left(\frac{A_L}{A_{\mathcal{P}_L}}\right) \tag{7.17}$$

$\mathbf{P}_{\mathcal{S}_\mathcal{P}\Leftarrow L}$ is a light projection that goes from $L\subseteq\mathcal{S}$ to $\mathcal{S}_\mathcal{P}$ (Equation 7.18). Due to Equation 7.7 it is assumed that $\mathbf{F}(\mathcal{P},L)=\mathbf{F}(\mathcal{S}_\mathcal{P},L)$, because the area of the patches in $\mathcal{S}_\mathcal{P}$ does not affect the result.

$$\mathbf{P}_{\mathcal{S}_\mathcal{P}\Leftarrow L} = \mathbf{RF}(\mathcal{P},L) \tag{7.18}$$

$\mathbf{P}_{u\Leftarrow\mathcal{S}_\mathcal{P}}$ is a light projection from $\mathcal{S}_\mathcal{P}$ to $u\subseteq\mathcal{S}$. Given Equation 7.7 and due to the symmetry of $\mathbf{AF}$, the projection can be represented as:

$$\mathbf{P}_{u\Leftarrow\mathcal{S}_\mathcal{P}} = \mathbf{A}_u^{-1}\mathbf{RF}(\mathcal{P},u)^T\mathbf{A}_{\mathcal{S}_\mathcal{P}} \tag{7.19}$$

7.4.5 How to Calculate $\widetilde{\mathbf{M}}$

To calculate $\widetilde{\mathbf{M}}$, first we apply the concepts developed above in this section.

1. Select $\mathcal{P}$ and $\mathcal{P}_L$: see Section 7.4.1.
2. Compute the rows $\{\mathcal{P}, \mathcal{P}_L\}$ of $\mathbf{F}$: hemicube.
3. Calculate $r_{f\!f}(i)\ \forall i{\in}\mathcal{P}$: Equation 7.11.
4. Calculate $\mathcal{F}_{\mathcal{P}}$: Equation 7.12.
5. Calculate $\mathbf{M}_{\mathcal{S}_{\mathcal{P}}}$: Equations 7.13 and 7.14.
6. Calculate $A_{\mathcal{S}_{\mathcal{P}}}(i)\ \forall i{\in}\mathcal{P}$: Equation 7.16.
7. Calculate $\mathbf{P}$ operators : Equations 7.17, 7.18, and 7.19.

Applying the equations developed in Section 7.3.4, now it is possible to build the matrices $\mathbf{Y}$ and $\mathbf{V}^T$ needed to calculate $\widetilde{\mathbf{M}}$. The matrices can vary depending of the kind of radiosity B to be computed:

$\widetilde{B}_u^I$ **:** Equation 7.2, where $\mathbf{Y}{=}\mathbf{P}_{u\Leftarrow\mathcal{S}_{\mathcal{P}}}$, $\mathbf{V}^T{=}\mathbf{M}_{\mathcal{S}_{\mathcal{P}}}\mathbf{P}_{\mathcal{S}_{\mathcal{P}}\Leftarrow L}\mathbf{S}_{L\Leftarrow\mathcal{S}}$;
then $\widetilde{B}_u^I{=}\widetilde{\mathbf{M}}E{=}\mathbf{Y}(\mathbf{V}^T E)$.

$\widetilde{B}_u^D$ **:** Equation 7.3, where $\mathbf{Y}{=}\mathbf{P}_{u\Leftarrow\mathcal{P}_L}$, $\mathbf{V}^T{=}\mathbf{S}_{\mathcal{P}_L\Leftarrow\mathcal{S}}$;
then $\widetilde{B}_u^D{=}\widetilde{\mathbf{M}}E{=}\mathbf{S}_{u\Leftarrow\mathcal{S}}E + \mathbf{Y}(\mathbf{V}^T E)$.

$\widetilde{B}_u$ when $\mathcal{P}_L{\subset}L$ **:** Equation 7.4 (or 7.5) is used to build $\mathbf{U}$, $\mathbf{V}^T$;
then $\widetilde{B}_u{=}\widetilde{\mathbf{M}}E{=}\mathbf{S}_{u\Leftarrow\mathcal{S}}E + \mathbf{Y}(\mathbf{V}^T E)$.

$\widetilde{B}_u$ when $\mathcal{P}_L{=}L$ **:** Equation 7.6 is used to build $\mathbf{U}$, $\mathbf{V}^T$;
then $\widetilde{B}_u{=}\widetilde{\mathbf{M}}E{=}\mathbf{S}_{u\Leftarrow\mathcal{S}}E + \mathbf{Y}(\mathbf{V}^T E)$.

The operator $\mathbf{S}_{b\Leftarrow a}$ can be modeled as a sparse matrix, but also could be efficiently programmed, by taking a subset of a to build b. To compute the rows of $\mathbf{F}$, we use a hemicube algorithm [21] but other techniques, like Monte Carlo ray tracing [1], could be applied. The matrix $\widetilde{\mathbf{M}}$ is never built, because it consumes $O(n^2)$ amount of memory. Instead we operate with its components to calculate the scene radiosity. The formula $\mathbf{Y}(\mathbf{V}^T E)$ expresses the efficient way to do the calculation: first the product $\mathbf{V}^T E$ is performed, then its result is multiplied by $\mathbf{Y}$.

7.5 Error Estimation

7.5.1 Error of the Operators

The projections $\mathbf{P}_{u\Leftarrow\mathcal{S}_{\mathcal{P}}}$ and $\mathbf{P}_{u\Leftarrow\mathcal{P}_L}$ are random functions related to the size of $\mathcal{P}$ and $\mathcal{P}_L$. Each operator projects light from a random sample, producing a Monte-Carlo integration process. So, the standard deviation of their errors scales with the inverse of the square root of the size of each sample [23]:

$$\sigma\left(\|\mathcal{E}_{\mathbf{P}_{u\Leftarrow\mathcal{S}_{\mathcal{P}}}}\|\right) \propto 1/\sqrt{|\mathcal{P}|}\ \ ;\ \ \sigma\left(\|\mathcal{E}_{\mathbf{P}_{u\Leftarrow\mathcal{P}_L}}\|\right) \propto 1/\sqrt{|\mathcal{P}_L|}$$

The errors of $\mathbf{P}_{\mathcal{S_P}\Leftarrow L}$, $\mathbf{S}_{L\Leftarrow\mathcal{S}}$ and $\mathbf{S}_{\mathcal{P}_L\Leftarrow\mathcal{S}}$ are caused by projection artifacts and selection issues. These errors can be reduced independently of the samples size, so they are ignored.

The estimation of $\mathcal{E}_{\mathbf{M}_{\mathcal{S_P}}}$ is influenced by $\varepsilon(i)$ (Equation 7.8), which follows the CLT. When $\varepsilon(i)$ is taken into consideration, then, Equation 7.11 is transformed into:

$$B(i) = E(i) + R(i)r_{\mathit{ff}}(i)\sum_{k\in\mathcal{P}}\mathbf{F}(i,k)B(k) + \left(R(i)\sum_{j\in\mathcal{S}}\mathbf{F}(i,j)\right)\varepsilon(i)$$

$\forall\ i \in \mathcal{P}$, and Equation 7.13 is transformed into:

$$B_{\mathcal{S_P}} = \mathbf{M}_{\mathcal{S_P}}E_{\mathcal{S_P}} + \mathcal{R}\varepsilon = \mathbf{M}_{\mathcal{S_P}}E_{\mathcal{S_P}} + \mathcal{E}_{\mathbf{M}_{\mathcal{S_P}}}$$

where $\mathcal{R}$ is a diagonal matrix which meets $\mathcal{R}(i,i){=}R(i)\sum_{j\in\mathcal{S}}\mathbf{F}(i,j)$, and ε is a vector composed of the $\varepsilon(i)$ values. As $\mathcal{E}_{\mathbf{M}_{\mathcal{S_P}}}$ is a linear combination of the ε terms, it also scales following Equation 7.20.

$$\sigma(\|\mathcal{E}_{\mathbf{M}_{\mathcal{S_P}}}\|) \propto 1/\sqrt{|\mathcal{P}|} \tag{7.20}$$

7.5.2 Errors in B^I, B^D and B

The errors of the operators influence the $\widetilde{B}^I$ result. When Equation 7.2 is expanded to include the first order error terms, then:

$$B^I = \widetilde{B}^I + \mathcal{E}_{\widetilde{B}^I} \approx \mathbf{P}_{u\Leftarrow\mathcal{S_P}}\mathbf{M}_{\mathcal{S_P}}\mathbf{P}_{\mathcal{S_P}\Leftarrow L}\mathbf{S}_{L\Leftarrow\mathcal{S}}E + \left(\mathbf{P}_{u\Leftarrow\mathcal{S_P}}\mathcal{E}_{\mathbf{M}_{\mathcal{S_P}}} + \mathcal{E}_{\mathbf{P}_{u\Leftarrow\mathcal{S_P}}}\right)$$

The $\mathcal{E}_{\widetilde{B}^I}$ term is approximated by a linear combination of $\mathcal{E}_{\mathbf{M}_{\mathcal{S_P}}}$ and $\mathcal{E}_{\mathbf{P}_{u\Leftarrow\mathcal{S_P}}}$, then it also scales proportionally to $1/\sqrt{|\mathcal{P}|}$:

$$\sigma(\|\mathcal{E}_{\widetilde{B}^I}\|) \propto 1/\sqrt{|\mathcal{P}|} \tag{7.21}$$

$\widetilde{B}^D$ is linearly influenced by $\mathbf{P}_{u\Leftarrow\mathcal{P}_L}$ (Equation 7.3), hence $\sigma\left(\|\mathcal{E}_{\widetilde{B}^D}\|\right){\propto}1/\sqrt{|\mathcal{P}_L|}$. The variables $\mathcal{E}_{\widetilde{B}^I}$ and $\mathcal{E}_{\widetilde{B}^D}$ are uncorrelated because $\mathcal{E}_{\widetilde{B}^D}$ depends on $\mathcal{P}_L$ and $\mathcal{E}_{\widetilde{B}^I}$ depends on $\mathcal{P}$, which are independent sets. Therefore, $\sigma(\|\mathcal{E}_{\widetilde{B}}\|)$ fulfills Equation 7.22.

$$\sigma^2\left(\|\mathcal{E}_{\widetilde{B}}\|\right) = \sigma^2\left(\|\mathcal{E}_{\widetilde{B}^I}\|\right) + \sigma^2\left(\|\mathcal{E}_{\widetilde{B}^D}\|\right) \tag{7.22}$$

When $\mathcal{P}_L$ covers all the emission area L, then $\mathcal{E}_{\mathbf{P}_{u\Leftarrow\mathcal{P}_L}}$ is ignored, which results in $\mathcal{E}_{\widetilde{B}}{=}\mathcal{E}_{\widetilde{B}^I}$.

7.5.3 Estimation of $|\mathcal{P}|$ for the Calculation of $\widetilde{B}^I$

We focus our analysis on $\widetilde{B}^I$, because we consider it the main asset of this method. Following Equation 7.21 and assuming that $\widetilde{B}^I$ follows a normal distribution, it is possible to estimate the value $|\mathcal{P}_2|$ needed to bound the error of $\widetilde{B}_{|\mathcal{P}_2|}$. This estimation is based on the previous calculation of a set of radiosity solutions, $\phi^{(\eta)}_{|\mathcal{P}_1|}{=}\{\widetilde{B}^{I(1)}_{|\mathcal{P}_1|}\cdots\widetilde{B}^{I(\eta)}_{|\mathcal{P}_1|}\}$, for a $|\mathcal{P}_1|$ value, where η is the cardinal number of the set.

We base our reasoning in the concept of tolerance interval [22], a statistical interval where, with some confidence level γ, a specified proportion d of a sampled population falls. More specifically we are interested in the length of the tolerance interval (LTI) of $\widetilde{B}^I$. Equation 7.23 is an estimation of the LTI value for a patch p, and a size $|\mathcal{P}|$ of the scene sample.

$$\mathrm{LTI}_{\eta,d,\gamma}(p, |\mathcal{P}|) = 2\bar{\sigma}\left(\phi^{(\eta)}_{|\mathcal{P}|}(p)\right) k(\eta, d, \gamma) \tag{7.23}$$

In Equation 7.23, $\bar{\sigma}$ is the sample standard deviation, and $\phi^{(\eta)}_{|\mathcal{P}|}(p)$ is the set of radiosity values found for patch p. The parameter k is tabulated in some books and papers, for the case in which a population follows a normal distribution and the exact μ and σ values are unknown [22, 24].

Due to the linear relation between LTI and $\bar{\sigma}$, LTI is proportional to $1/\sqrt{|\mathcal{P}|}$, so having the LTI for any given $|\mathcal{P}_1|$, it is possible to find the $|\mathcal{P}_2|$ value such that $\mathrm{LTI}_{\eta,d,\gamma}(p, |\mathcal{P}_2|)$ fulfills a desired value (Equation 7.24).

$$|\mathcal{P}_2| = \left(\frac{\mathrm{LTI}_{\eta,d,\gamma}\left(p, |\mathcal{P}_1|\right)}{\mathrm{LTI}_{\eta,d,\gamma}\left(p, |\mathcal{P}_2|\right)}\right)^2 |\mathcal{P}_1| \tag{7.24}$$

In this equation, $\mathrm{LTI}_{\eta,d,\gamma}(p, |\mathcal{P}_1|)$ is calculated using Equation 7.23, and $\mathrm{LTI}_{\eta,d,\gamma}(p, |\mathcal{P}_2|)$ is the desired LTI value. Equation 7.24 can be used to estimate the size of a sample-based scene, if previously a set $\phi^{(\eta)}_{|\mathcal{P}_1|}$ (for a small $|\mathcal{P}_1|$ value) is computed.

7.6 Experimental Results

In this section we present three sets of experiments. The first one studies the main aspects of the error of B. In the second experiment we analyze how accurate the estimation of $|\mathcal{P}|$ is for a given length tolerance interval. Finally, $\widetilde{\mathbf{M}}$ is used to solve an inverse lighting problem.

All simulations were performed in a Matlab environment using a standard PC (Intel Core i7 2.2 Ghz processor and 8 GB memory).

7.6.1 Study of the Error

In this experiment we show results to analyze the error expressed in Equation 7.21. Our first goal is to check the proportionality between $\sigma(\widetilde{B}^I_{|\mathcal{P}|})$ and $1/\sqrt{|\mathcal{P}|}$. For each $\mathcal{P}$ size, a set $\phi^{(100)}_{|\mathcal{P}|}$ is calculated. After that, the weighted sample mean of the standard deviation of each patch is calculated: $\bar{\mu}_A(\bar{\sigma}(\phi^{(100)}_{|\mathcal{P}|}))$. The sample mean $\bar{\mu}_A$, considers the area of each patch as the weight parameter. We use a Cornell box, composed of 10240 patches (Figure 7.3).

In Figure 7.5, the red dotted line is a scaled inverse of the square root of $|\mathcal{P}|$, and the blue stars are the experimental result of $\bar{\mu}_A(\bar{\sigma}(\phi^{(100)}_{|\mathcal{P}|}))$ for several $|\mathcal{P}|$ values.

Figure 7.6 shows the estimations of μ and σ values for the Cornell box after 500 calculations of $\widetilde{B}$. The $\mu(\widetilde{B}^I_{20})$ estimation is very close to the reference (Figures 7.6(a) to (c)). In Figures 7.6(d) to (f), the $\sigma(\widetilde{B}^I_{20})$, $\sigma(\widetilde{B}^I_{80})$ and $\sigma(\widetilde{B}^I_{320})$ estimations are shown, scaled up seven times, showing that $\sigma(\widetilde{B}^I_{|\mathcal{P}|})$ is smaller the higher the $|\mathcal{P}|$ is.

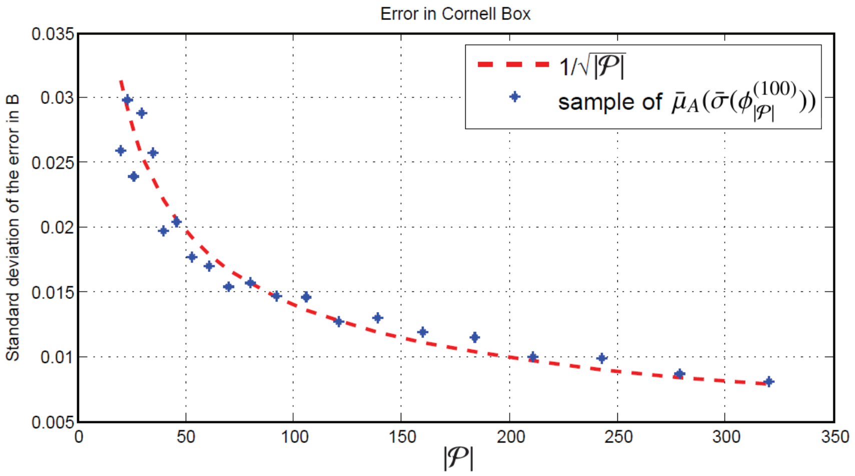
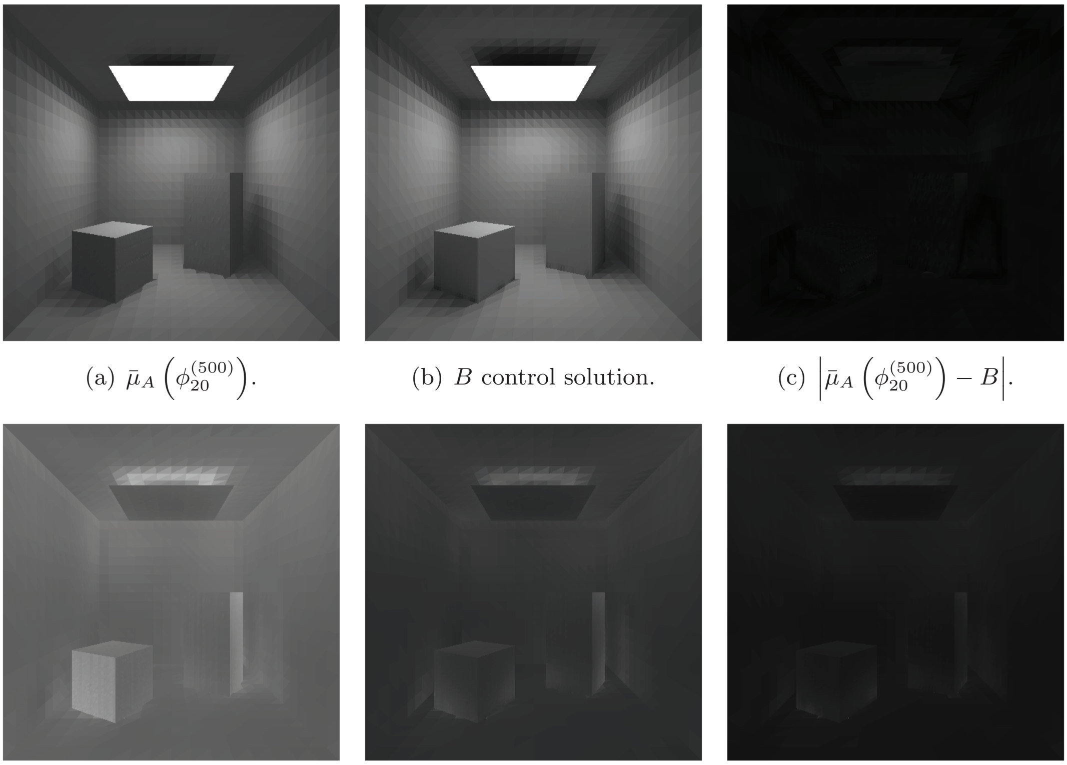


Figure 7.5: $\bar{\mu}_A(\bar{\sigma}(\phi^{(100)}_{|\mathcal{P}|}))$ is estimated (blue stars) for different $|\mathcal{P}|$ values.


(a) $\bar{\mu}_A\left(\phi^{(500)}_{20}\right)$.

(b) B control solution.

(c) $\left|\bar{\mu}_A\left(\phi^{(500)}_{20}\right) - B\right|$.

(d) $\bar{\sigma}(\phi^{(500)}_{20})$.

(e) $\bar{\sigma}(\phi^{(500)}_{80})$.

(f) $\bar{\sigma}(\phi^{(500)}_{320})$.

Figure 7.6: The sample mean and standard deviation of $\widetilde{B}$ for 500 radiosity executions.

Figure 7.7 shows the histograms of $\widetilde{B}^I_{|\mathcal{P}|}$ for two individual patches. The histograms are close to the Gaussian bell for larger $|\mathcal{P}|$ values, $\mu(\widetilde{B}^I_{|\mathcal{P}|})$ varies very little for any $|\mathcal{P}|$ size, and $\sigma(\widetilde{B}^I_{|\mathcal{P}|})$ is roughly halved when $|\mathcal{P}|$ increases four times, confirming the

Equation 7.21.

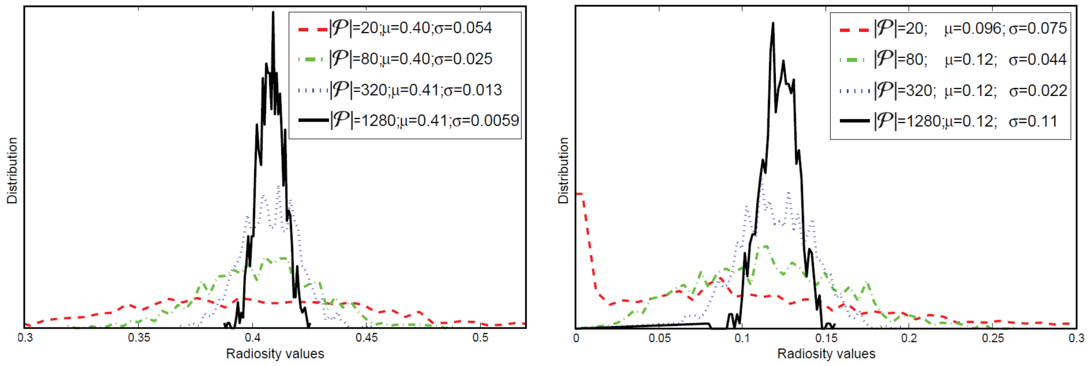
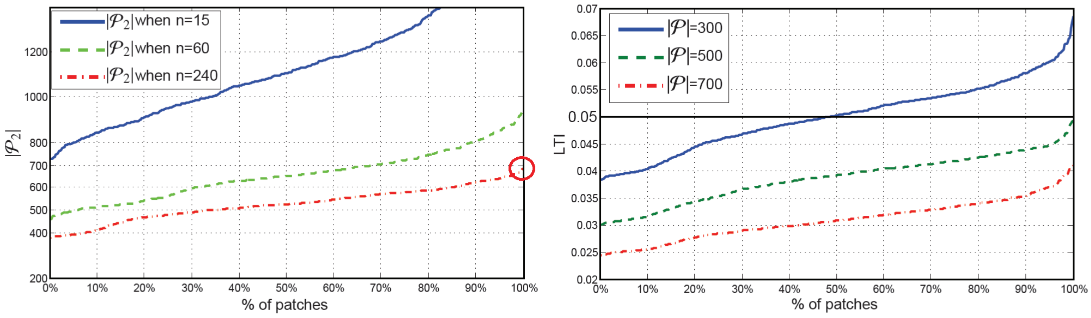


Figure 7.7: Histograms of $\widetilde{B}^I_{|\mathcal{P}|}$ for two different patches when $|\mathcal{P}|$=20, 80, 320, and 1280.

The Lilliefors test [25] is used to test the null hypothesis for a sample extracted from a normally distributed population. This test can not reject the normal distribution hypothesis in 80% of the Cornell box patches at the 5% significance level, when $|\mathcal{P}|$=80.

7.6.2 Estimation of $|\mathcal{P}|$

Based on the Cornell box scene, we estimate the $\mathcal{P}_2$ size for a LTI value using Equation 7.24. After that, we confirm that the estimated values are an upper bound of the real values. Figure 7.8(a) presents several estimations of $|\mathcal{P}_2|$. The tolerance interval of $\widetilde{B}^I$ has a length (LTI) of 0.05, the value of $|\mathcal{P}_1|$=20, and η=15, 60, and 240. We choose d=γ=0.95. Figure 7.8(b) shows the real LTI for all $\widetilde{B}_{|\mathcal{P}|}$, with d=0.95, and $|\mathcal{P}|$=300, 500, and 700. The calculations are based on 200 radiosity results for each $|\mathcal{P}|$ value. The values of $k(\eta, d, \gamma)$ for η=15, 60, and 240 (and d=γ=0.95) are 2.95, 2.33, and 2.12 respectively [22].



(a) $|\mathcal{P}_2|$ value, for $|\mathcal{P}_1|$=20, d=γ=0.95, LTI=0.05.

(b) Real LTI($\widetilde{B}_{|\mathcal{P}|}$) when d=0.95.

Figure 7.8: Estimation of $|\mathcal{P}|$, and real LTI values for different $|\mathcal{P}|$.

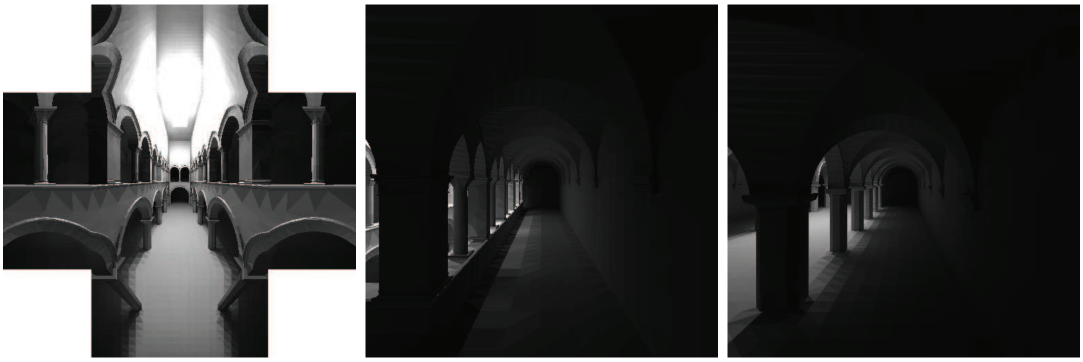
The values estimated in Figure 7.8(a) are an upper bound of the real values of Figure 7.8(b). The minimum estimate of $|\mathcal{P}|$ that fulfills that 100% of the patches meets LTI$\leq$0.05 is of about 700 (red circle in Figure 7.8(a)), and the real best $|\mathcal{P}|$ value is close to 500.

7.6.3 Inverse Lighting Problem for Larger Scenes

ILP is usually restricted to simple scenes. In this section we show the capability of our method to solve ILPs for more complex scenes. We build an experiment for a modified Sponza atrium with a roof in its patio, and another experiment on a museum building.

Sponza Atrium

This scene is composed of 79230 patches and we worked with $|\mathcal{P}|$=2000. The precomputation step, that includes a non optimized hemicube algorithm to compute form factors in general environments [21], takes about three minutes. Figure 7.9 shows three views of a radiosity solution. Analyzing $\widetilde{B}$ after 1000 executions with d=γ=0.95, it was obtained that LTI($\widetilde{B}_{2000}$)$\leq$0.05 for 90% and LTI($\widetilde{B}_{2000}$)$\leq$0.1 for 98% of all the scene patches.



(a) Hemicube view. (b) Upper corridor. (c) Lower corridor.

Figure 7.9: Three views of a $\widetilde{B}_{2000}$ solution for the Sponza atrium.

We solve two different ILPs. The first one is the maximization of the radiosity of a wall w ($\widetilde{B}(w)$) in the Sponza atrium, and the goal of the second ILP is to minimize the standard deviation of $\widetilde{B}(w)$ (see Figure 7.10).

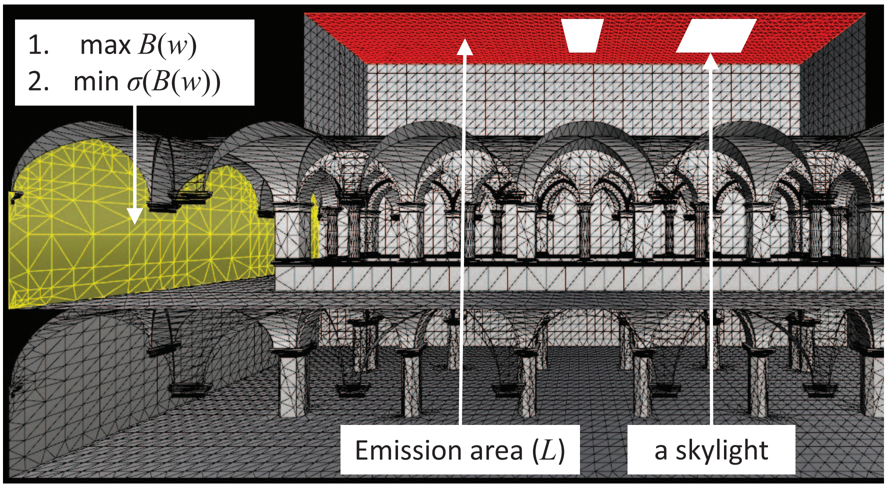


Figure 7.10: Scheme of the ILP in the Sponza atrium.

For both problems, the variables are the position and the shape of two rectangular skylights in the Sponza roof. The area of each skylight is bounded by maximum and minimum values. Finding the optimal rectangular areas is a combinatorial problem. This ILP could be formulated as an optimization problem [26], which is approximately solved by any global optimization solver. We use the variable neighborhood search (VNS) method [27], previously tested in ILPs [19]. In each iteration of the VNS method, a new configuration of the skylights is tested, forcing the calculation of the radiosity in the wall, in order to verify the goal. The radiosity calculation is the most computationally demanding subroutine in each iteration of VNS.

Equation 7.6 is used to evaluate the radiosity calculations, because $\mathcal{P}_L$ covers all the ceiling surface L. The calculation of $\widetilde{B}_{2000}$ for the whole scene takes about 0.07s (14 iterations per second), which allows to evaluate 25000 skylight configurations in about 30 minutes.

Additionally, Equation 7.6 allows to restrict the radiosity calculation to the selected wall, which speed up the optimization process. This improvement increases the speed to about 270 iterations per second, thus allowing to calculate 25000 iterations in about 90s. This improvement in performance is related to the small number of patches in w. When there are goals or constraints involving the whole scene, this kind of optimization can not be applied.

Other shortcuts can also be applied to speed up the calculation time. Graphics cards, for instance, allows to reduce by a factor of five the matrix·vector product used in the radiosity computation [28]. But this technique does not reduce the amount of operations performed by the ILP solver.

From the calculation of the matrix $\widetilde{\mathbf{M}}$, another methods can be applied [20] based on the use of $\bar{\mu}_A(\widetilde{B}(w))$ and $\bar{\sigma}(\widetilde{B}(w))$ operators to define the goals of both problems, for any $w \subseteq \mathcal{S}$. For the first problem defined, the maximization of the radiosity in the wall w is equivalent to maximize $\bar{\mu}_A(\widetilde{B}(w))$. Following this approach, the mean of the radiosity values for any surface could be expressed as the scalar product defined in Equation 7.25.

$$\bar{\mu}_A(\widetilde{B}(w)) = \frac{\mathbf{1}_w \mathbf{A} \widetilde{B}}{\sum_{i \in w} A_i} = \left(\frac{\mathbf{1}_w \mathbf{A} \widetilde{\mathbf{M}}}{\sum_{i \in w} A_i} \right) E = vE \tag{7.25}$$

In this equation, $\mathbf{1}_w$ is a row vector with ones only in w, and vE is the scalar product. The complexity of the vE product is $O(|\mathcal{P}_L|)$, so it is independent to the size of w.

In the vector v, each element $v(i)$ is the ratio between the light power received by w from patch i and the light power emitted by i, that is, the contribution of the emitter to the region of most interest. This is closely related to the concept of "importance" [29].

For the second problem defined, the minimization of $\bar{\sigma}(\widetilde{B}(w))$, we measure the dispersion of the radiosity values in w. The covariance matrix is used, and each column of $\mathbf{Y}(w,:)$ is considered as a sample of a statistical variable (Equation 7.26), following [20]
.

$$\bar{\sigma}(\widetilde{B}(w)) = \sqrt{(\mathbf{V}^T E)^T \mathrm{cov}(\mathbf{Y}(w,:))(\mathbf{V}^T E)} \tag{7.26}$$

In this equation, $\mathbf{Y}$ and $\mathbf{V}$ are calculated through Equation 7.6, and $\mathrm{cov}_{w_A}(\mathbf{Y}(w,:))$ is the $|L| \times |L|$ covariance matrix defined from the columns of $\mathbf{Y}(w,:)$. For static scenes, the covariance matrix is built only once. In Equation 7.6, $\mathbf{V}^T$ is defined as $\mathbf{S}_{L \Leftarrow \mathcal{S}}$, which can be defined as a sparse matrix (one nonzero element per row), so the product $\mathbf{V}^T E$ is reduced to $O(|L|)$ operations. Also, the product of the covariance matrix to both $\mathbf{V}^T E$ is a $O(e^2)$ operation, where e is the amount of emitters turned on [20].

Our optimization tests for the first problem based on Equation 7.25, iterates 830 times per second, totalizing 25000 iterations in about 30s. Also, the tests for the second problem based on Equation 7.26, iterates 400 times per second, totalizing 25000 iterations in about 60s.

In Figure 7.11 there is a set of solutions provided by the ILP solver. Figure 7.11(a) and (b) correspond to the first and second problem defined above, respectively. As the solution of the second problem is darker than desired, then we add a constraint (μ>0.085) and the result is shown in Figure 7.11(c). Finally, we propose to maximize the standard deviation of $\widetilde{B}(w)$ (Figure 7.11(d)).

Museum Building

This scene is composed of 168160 patches, where we defined $|\mathcal{P}|$=2000. The precomputation step takes about five minutes. There is a mezzanine that partially shadows a yellow sculpture (Figure 7.12).

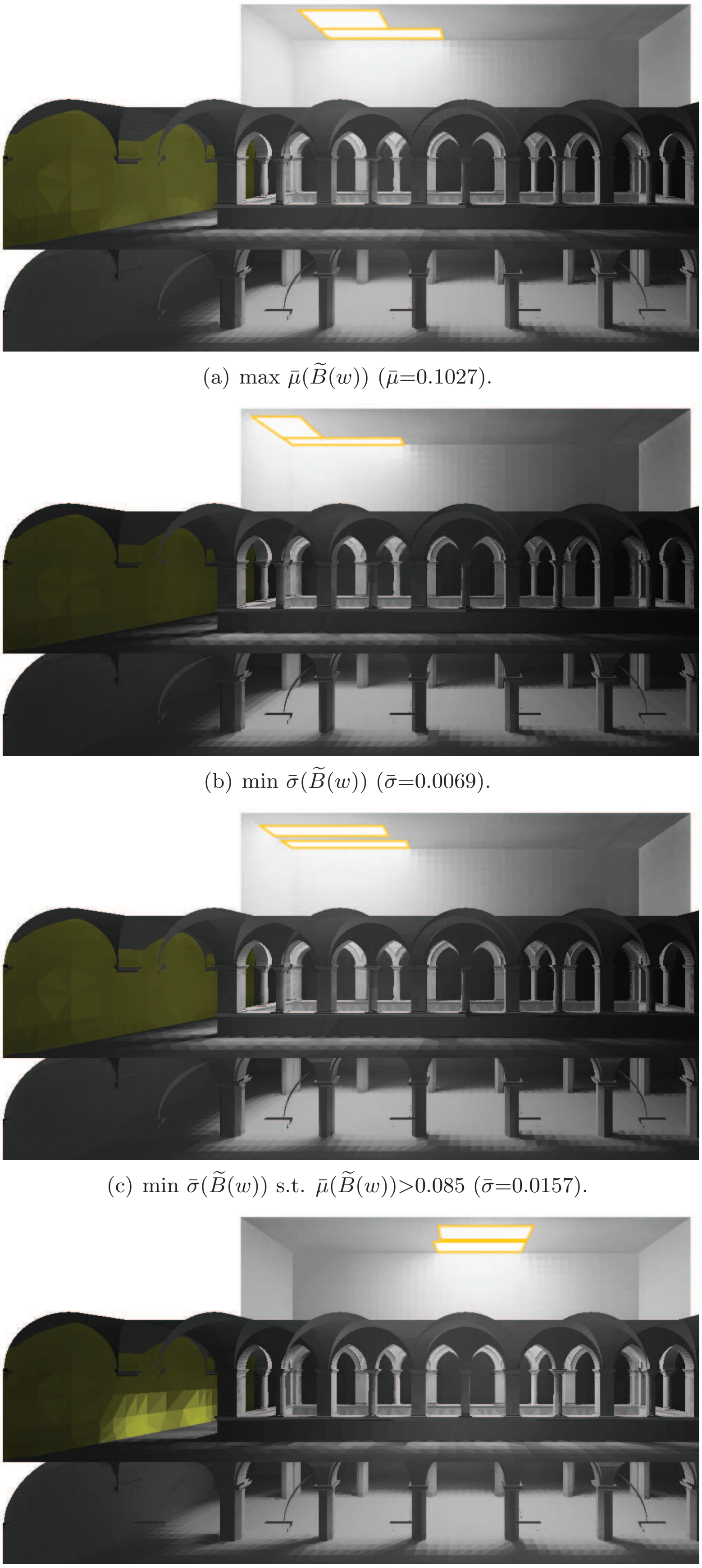
In this experiment we tried three different goals related to the standard deviation of B in the yellow sculpture y. The first goal is to minimize $\sigma(B(y))$ (Figure 7.13(a)), and the second goal is to maximize $\sigma(B(y))$ (Figure 7.13(b)). We can observe in the figures that σ is directly related to μ (more dispersion is associated to more intensity), then the third goal tested is the minimization of the coefficient of variation (σ/μ), which is a normalized measure of dispersion [22] (Figure 7.13(c)). All the above problems were solved in about 95s, after 25000 iterations (260 iterations per second).

7.6.4 Discussion of Experimental Results

We analize the usefulness of $\widetilde{\mathbf{M}}$ in different scenarios. All the radiosity values shown in Figures 7.11, 7.12 and 7.13, were calculated through $\widetilde{\mathbf{M}}E$. While the error may be estimated and bounded, as is shown in Sections 7.6.1 and 7.6.2, we still can observe visually two main kinds of errors: errors due to the discretization level of the mesh, and errors due to $\widetilde{\mathbf{M}}$ (e.g., there is some "noise" in the radiosity values of the Sponza wall). Both errors are important to the subjective quality of rendering, but not so much important in ILP. These errors can be reduced by the construction of meshes with greater n, and by the increment of $|\mathcal{P}|$.

The memory size of $\mathbf{Y}$ matrix in the museum scene is about 1.2Gbytes (with single-precision floating-point numbers). Through Equation 7.6, the size of $\mathbf{Y}$ is independent of $|\mathcal{P}|$, but linearly dependent to $|u|$ (in this case $|u|$=n). Then, scenes with an order of magnitude larger produce matrices with the same order of size increment. Therefore, the amount of memory required makes this method noncompetitive for rendering, compared to currently used real-time global-illumination techniques, which render multiple bounce radiosity in scenes composed of millions of polygons [30].

In ILP, the calculation of the radiosity with an infinite amount of bounces is useful to model real world problems. The results achieved in this work solving ILP are in some cases two orders of magnitud better than previous work [26, 31, 32], in the total number of patches or the runtime employed. The comparison is difficult to carry out because of the different assumptions in the problem formulation. The most noticeable differences are centered in the size and complexity of the geometric model, and the time taken to solve the inverse problem. A more feasible comparison can be done with [19] and [20]. In these cases, the low-rank approach to $\mathbf{RF}$ is performed by a $O(n^2)$ method [9], compared



(a) max $\bar{\mu}(\widetilde{B}(w))$ ($\bar{\mu}$=0.1027).

(b) min $\bar{\sigma}(\widetilde{B}(w))$ ($\bar{\sigma}$=0.0069).

(c) min $\bar{\sigma}(\widetilde{B}(w))$ s.t. $\bar{\mu}(\widetilde{B}(w))>0.085$ ($\bar{\sigma}$=0.0157).

(d) max $\bar{\sigma}(\widetilde{B}(w))$ ($\bar{\sigma}$=0.0627).

Figure 7.11: A $\widetilde{B}_{2000}$ representation of the Sponza atrium showing the solutions for several ILPs.

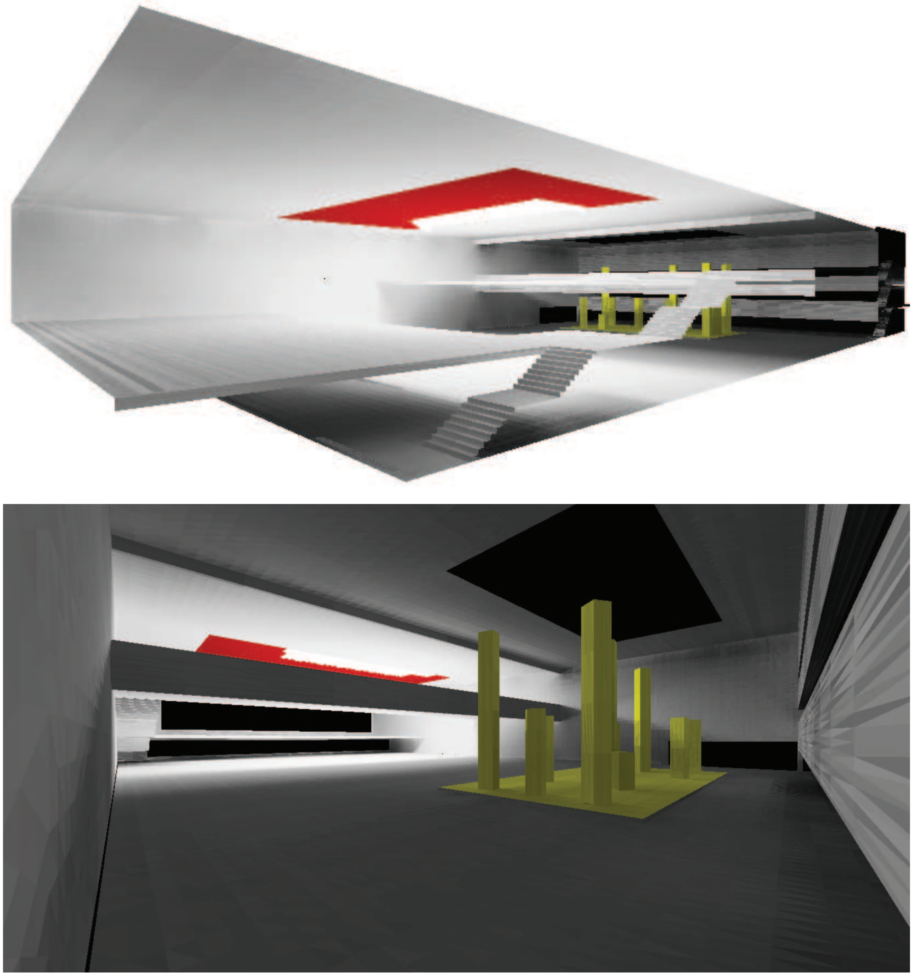
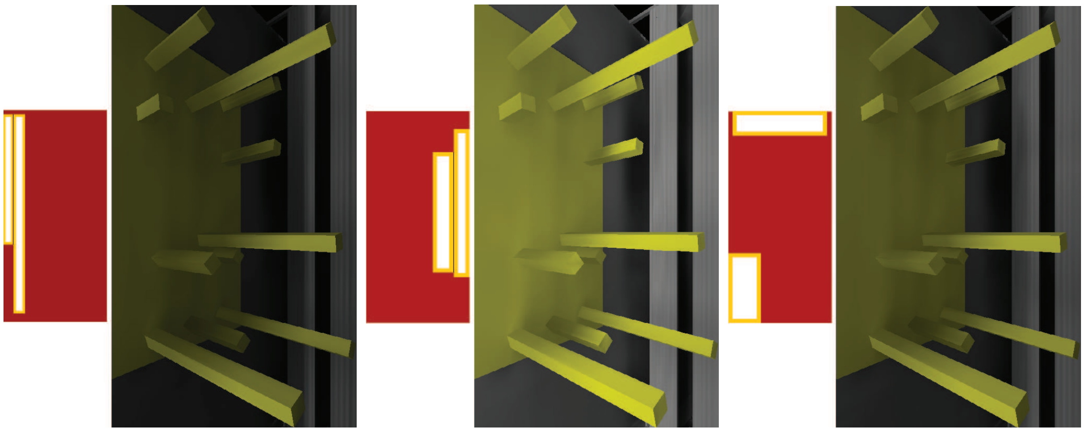


Figure 7.12: Museum guilding. The yellow “sculpture” is the object to be illuminated. The potential emitters are painted in red.



(a) min $\bar{\sigma}(B(y))$ ($\bar{\sigma}$=0.0170). (b) max $\bar{\sigma}(B(y))$ ($\bar{\sigma}$=0.0490). (c) min $\frac{\bar{\sigma}(B(y))}{\bar{\mu}(B(y))}$ $\left(\frac{\bar{\sigma}}{\bar{\mu}} = 0.4645\right)$.

Figure 7.13: Distribution of emitters in the potential emitters area (red rectangles) and the resulting illumination in the sculpture.

to the $O(n|\mathcal{P}|)$ complexity of our proposal, meaning an speed up of $O(n/|\mathcal{P}|)$. Another advantage is that our method avoids the construction of a coarse mesh composed of k patches for the same scene.

7.7 Conclusions and Further Work

A method for approximating the inverse radiosity matrix $\widetilde{\mathbf{M}}$ is presented. The construction of $\widetilde{\mathbf{M}}$ is based in the use of three operators, defined generically. A particular implementation is provided for each one. We showed that the error of the indirect radiosity is proportional to $1/\sqrt{|\mathcal{P}|}$. A set of experiments confirms these results, showing that the distribution of the radiosity error is approximated to a Gaussian bell curve. Based on these results, the size of the sample can be estimated for a given set of parameters.

Several inverse lighting problems based on the Sponza atrium and a museum building were designed and solved, as examples of application. The availability of $\widetilde{\mathbf{M}}$ allows us to know the global illumination influence of the light emitted by j in i, that is $\widetilde{\mathbf{M}}(i,j)$, which reduces the amount of radiosity calculations involved in ILP. Also, $\widetilde{\mathbf{M}}$ allows to find statistical operators, reducing the calculations even more. Depending on the ILP and the scene, the optimization times varies between 30s and 95s for 25000 radiosity calculations, in scenes up to 170K patches.

As future work, variance reduction techniques would allow to construct even smaller sample-based scenes for the same LTI value. Also, TSVD techniques could be applied to reduce the size of the operators $\mathbf{P}$ and $\mathbf{M}_{\mathcal{S}_\mathcal{P}}$. GPU techniques could be used to reduce the construction time of $\widetilde{\mathbf{M}}$ matrices, and to solve ILPs even faster. Finally, our method could be integrated as part of a lighting design tool, to solve ILPs.

7.8 Acknowledgements

This work was partially funded by the TIN2010-20590-C02-02 project from Ministerio de Ciencia e Innovación, Spain, and by Programa de Desarrollo de las Ciencias Básicas, Uruguay.

Bibliography

[1] M. Cohen, J. Wallace, P. Hanrahan, Radiosity and realistic image synthesis, Academic Press Professional, Inc., San Diego, CA, USA, 1993.

[2] P. Hanrahan, D. Salzman, A rapid hierarchical radiosity algorithm, in: Computer Graphics, 1991, pp. 197–206.

[3] S. R. Marschner, Inverse rendering for computer graphics, Ph.D. thesis, Ithaca, NY, USA (1998).

[4] G. H. Golub, C. F. Van Loan, Matrix computations (3rd ed.), Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[5] N. Halko, P. G. Martinsson, J. A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, SIAM Rev. 53 (2) (2011) 217–288.

[6] S. A. Goreinov, E. E. Tyrtyshnikov, N. L. Zamarashkin, A theory of pseudo–skeleton approximations, Linear Algebra Appl. 261 (1997) 1–21.

[7] J. Wang, Y. Dong, X. Tong, Z. Lin, B. Guo, Kernel Nyström method for light transport, ACM Trans. Graph. 28 (3) (2009) 29:1–29:10.

[8] M. O'Toole, K. N. Kutulakos, Optical computing for fast light transport analysis, ACM Trans. Graph. 29 (6) (2010) 1–12.

[9] E. Fernández, Low-rank radiosity, in: O. Rodríguez, F. Serón, R. Joan-Arinyo, E. C. J. Madeiras, J. Rodríguez (Eds.), Proceedings of the IV Iberoamerican Symposium in Computer Graphics, Sociedad Venezolana de Computación Gráfica, DJ Editores, C.A., 2009, pp. 55–62.

[10] G. V. G. Baranoski, R. Bramley, J. G. Rokne, Eigen-analysis for radiosity systems, in: Proceedings of the Sixth International Conference on Computational Graphics and Visualization Techniques (Compugraphics '97), Vilamoura, Algarve, Portugal, 1997, pp. 193–201.

[11] I. Ashdown, Eigenvector radiosity, Master's thesis, Department of Computer Science, University of British Columbia, Vancouver, British Columbia (April 2001).

[12] M. Hasan, F. Pellacini, K. Bala, Matrix row-column sampling for the many-light problem, ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007) 26 (3).

[13] A. Keller, Instant radiosity, in: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1997, pp. 49–56.

[14] L. Szécsi, L. Szirmay-Kalos, M. Sbert, Light animation with precomputed light paths on the gpu, in: Proceedings of Graphics Interface 2006, GI '06, Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 2006, pp. 187–194.

[15] C. Dachsbacher, J. Křivánek, M. Hašan, A. Arbree, B. Walter, J. Novák, Scalable realistic rendering with many-light methods, Computer Graphics Forum 32 (1) (2013) 23–38.

[16] P.-P. Sloan, J. Hall, J. Hart, J. Snyder, Clustered principal components for precomputed radiance transfer, ACM Trans. Graph. 22 (3) (2003) 382–391.

[17] M. Contensin, Inverse lighting problem in radiosity, Inverse Problems in Engineering 10 (2) (2002) 131–152.

[18] P. S. Schneider, A. C. Mossi, F. H. R. Franca, F. L. de Sousa, A. J. da Silva Neto, Application of inverse analysis to illumination design, Inverse Problems in Science and Engineering 17 (6) (2009) 737–753.

[19] E. Fernández, G. Besuievsky, Inverse lighting design for interior buildings integrating natural and artificial sources, Computers & Graphics 36 (8) (2012) 1096–1108.

[20] E. Fernández, G. Besuievsky, Efficient inverse lighting: A statistical approach, Automation in Construction 37 (0) (2014) 48 – 57.

[21] M. F. Cohen, D. P. Greenberg, The hemi-cube: a radiosity solution for complex environments, SIGGRAPH Comput. Graph. 19 (3) (1985) 31–40.

[22] G. Canavos, Applied probability and statistical methods, Little, Brown, 1984.

[23] G. S. Fishman, Monte Carlo: Concepts, algorithms, and applications, 6th Edition, Springer series in operations research, Springer, New York, NY [u.a.], 2004.

[24] W. G. Howe, Two-sided tolerance limits for normal populations, some improvements, Journal of the American Statistical Association 64 (326) (1969) 610–620.

[25] H. W. Lilliefors, On the Kolmogorov-Smirnov test for normality with mean and variance unknown, Journal of the American Statistical Association 62 (318) (1967) 399–402.

[26] J. K. Kawai, J. S. Painter, M. F. Cohen, Radioptimization - goal based rendering, in: ACM SIGGRAPH 93, Anaheim, CA, 1993, pp. 147–154.

[27] P. Hansen, N. Mladenovic, Variable neighborhood search: Principles and applications, European Journal of Operational Research 130 (3) (2001) 449–467.

[28] E. Fernández, P. Ezzatti, S. Nesmachnow, G. Besuievsky, Low-rank radiosity using sparse matrices, in: Proceedings of GRAPP2012, 2012, pp. 260–267.

[29] P. H. Christensen, Adjoints and importance in rendering: An overview, IEEE Transactions on Visualization and Computer Graphics 9 (3) (2003) 329–340.

[30] J. Křivánek, M. Fajardo, P. H. Christensen, E. Tabellion, M. Bunnell, D. Larsson, A. Kaplanyan, Global illumination across industries, in: ACM SIGGRAPH 2010 Courses, SIGGRAPH '10, ACM, New York, NY, USA, 2010.

[31] F. Cassol, P. S. Schneider, F. H. França, A. J. S. Neto, Multi-objective optimization as a new approach to illumination design of interior spaces, Building and Environment 46 (2) (2011) 331 – 338.

[32] F. Castro, E. del Acebo, M. Sbert, Energy-saving light positioning using heuristic search, Engineering Applications of Artificial Intelligence 25 (3) (2012) 566 – 582.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

In this work has been developed several techniques to reduce the computational effort in the inverse lighting process. These contributions are described as improvements to the ILP solver presented in Figure 1.1, even if they may be used in other contexts.

The first practical results are related to the utilization of the GPU architecture to speed up the matrix-vector product, which is used in the calculation of the radiosity through the LRR method. The GPU hardware, and the transformation of $\mathbf{V}_k$ into an index vector, helps to speed-up the radiosity calculation from 8.6 to 21.3 times, in comparison to the CPU timing (for the tested hardware). When the scene has several hundred thousand elements, and it is possible to build a coarse version of it with a few thousand patches, then it is possible to calculate the global radiosity of the scene, with infinite bounces, with a rate of 20 times per second.

Another important result consists in the use of the Variable Neighborhood Search (VNS) method as the optimization heuristic, combined with the use of the penalty approach to include the constraints, and the LRR method as a radiosity engine. This combination of techniques allows us to solve single and multi-objective optimization problem (MOP), mixing natural and artificial lighting goals and constraints. With the aim of reducing the execution time, a multilevel procedure was tested. This procedure is based on the definition of a hierarchy of scene samples, and the calculation of the ILP, one level at a time, beginning in the sample with fewer elements. The set of ILP tested is solved in 5 minutes at a rate of 90 radiosity calculations per second, in a low-end CPU. This method is also used to design optimal filters given a set of lighting intentions, achieving very good results.

The above results, specially those related to the multilevel method, lead us to think in the relevance of the radiosity calculation time in the overall time of the ILP resolution. The calculation of μ and σ for any surface of the scene, can be performed easily through the use of LRR. The efficiency achieved allows the testing of thousands of configurations per second on a standard computer, reducing drastically the overall time required to solve ILP. Also, it was implemented an algorithm that facilitates the substitution of a set of constraints by a single constraint built on the Chebyshev's inequality.

Finally, it is presented a new method for the calculation of the radiosity inverse matrix ($\widetilde{\mathbf{M}}$), based on the use of a randomized algorithm, which takes a small set of columns of the form factor matrix. The amount of columns needed to achieve the error pursued is estimated. This methodology was tested on scenes up to 170K patches, and the $\widetilde{\mathbf{M}}$ found was used in several ILP, solving them in an average time of 70 seconds after 25000 evaluations.

All the results presented here, could be accelerated through the calculation in GPU of the radiosity values. Instead of that we decided to accelerate the ILP pipeline, based exclusively on the simplification of the algorithms involved. The implementation of the VNS metaheuristic was used in all the ILP presented in this thesis, and provides very good results. We consider that the use of μ and σ in the LI, would be extremely useful to simplify the ILP calculation. Finally, all efficiency gains achieved in the optimization stage were overshadowed by the modest times, $O(n^2)$, found in the pre-computation stage. Therefore, the proposal of a sample based method, $O(nk)$, for the calculation of the inverse matrix, increases the usability of the developed methods in medium size scenes.

8.2 Future Work

This section is divided in two parts. The first one is related to the published papers, and the second one is related to medium term goals and other further works.

8.2.1 Future Work Related to Published Papers

Although some results have been fulfilled, the work could be extended in different ways.

Regarding the use of GPU on LRR calculation, it seems that an hybrid strategy of simultaneous computation in CPU and GPU would allow a reduction in the time employed in the radiosity calculation. Another important point to work on, is the inclusion of these GPU based methodologies to speed-up the amount of iterations per second performed by the optimization solver.

Related to the optimization techniques, it is important to consider the use of population based metaheuristic, like genetic algorithms, to solve single and multi-objective optimization problem. These metaheuristics allow a better diversification in the whole search space, than single-solution based metaheuristic, like VNS [1].

The results achieved when μ and σ are used in the LI, could inspire the development of heuristics to transform the descriptive intentions (Figure 1.2) into a mathematical equivalent. This would allow to achieve interactive times for a wider variety of situations, and would ease the development of inverse design modules in CAD systems.

Considering the sample-based construction of the radiosity inverse matrix, variance reduction techniques should be studied to allow the construction of even smaller sample scenes from the same length of the tolerance interval. Also, the use of truncated singular value decomposition or rank-revealing QR techniques would allow to reduce the rank of the matrix approximation. The search for an appropriate combination of these techniques seems to be an interesting area of research, that would allow to solve ILP for larger scenes and with less computational resources.

Finally, the development of real examples and working experiences with designers are needed. This line of work will transform the code implemented into a design tool useful for architects and interior designers.

8.2.2 Further Works

ILP for Openings not Covered by Lambertian diffusers

In this thesis, the emitters are Lambertians diffusers. This kind of emitter illuminates in all directions following the cosine law. A window with transparent glass is not Lambertian, because the incoming light in each direction depends on the configuration of external elements (sky, sun, clouds, other buildings). Despite that, it is possible to model the radiosity of the scene, with a variation of the radiosity equation.

As an approach to the problem, any pair of patches $\vec{ij}$ defines an intensity of light $E_{\vec{ij}}$. In $\vec{ij}$, i is part of a opening and j is an element of the scene (Figure 8.1). Then, it is possible to define a radiosity equation that includes G_i (the first reflection of incoming light through patch i), and E (the artificial emitters).

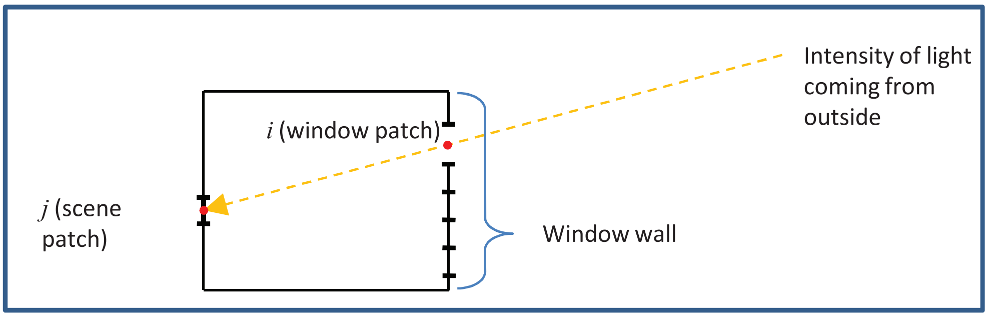


Figure 8.1: Each pair of patches $\vec{ij}$ defines the incoming light.

$$(\mathbf{I} - \mathbf{RF})B = \mathbf{R}\begin{pmatrix} F_{1i}E_{\vec{i1}} \\ \vdots \\ F_{ni}E_{\vec{in}} \end{pmatrix} + E = G_i + E$$

In this equation, each form factor F_{ji} establishes the visibility between j and i.

The vector G_i could be transformed into a matrix $\mathbf{G}$ to include all potential elements of the opening (one column per element). Then the above equation is transformed into:

$$(\mathbf{I} - \mathbf{RF})B = \mathbf{G}W + E \tag{8.1}$$

where W is a vector that establishes which are the patches that form part of the opening. Consequently, each time an opening is defined, $\mathbf{G}W + E$ is transformed into a vector, and the Equation 8.1 has become a radiosity equation. Therefore, all the methods developed in this thesis could include this procedure without any additional effort.

Changing the reflection index of the surfaces

In this thesis the properties of the surfaces, as the reflectance, are fixed. The optimization is centered in the best configuration for the emission patches.

A first consideration is that a modification in the reflection factor of any surface, has a non-linear effect in the final radiosity values of the scene. This happens because a modification of this kind implies a change in the $\mathbf{R}$ matrix, which alter the exponential terms in the Neumann series (see Equation 2.7).

Another important concept is that a modification in the reflection factor of a single patch, may involve a change in all the elements of the inverse of the radiosity matrix $\mathbf{M}$. This implies a complexity of at least $O(n^2)$, or $O(nk)$ in case of the low-rank approximation $\widetilde{\mathbf{M}}$.

Matrix-Vector Vs. Matrix-Matrix Product in GPU

In this thesis, the calculation of the radiosity values is reduced to a matrix-vector product $B = \mathbf{M}E$. The simultaneous calculation of the radiosity for m configurations of the emitters, could be done through the use of a matrix $\mathbf{E} = [E_1, E_2, \ldots, E_m]$. Therefore the simultaneous execution of m matrix-vector products could be transformed into a single matrix-matrix product $\mathbf{B} = \mathbf{ME}$, where $\mathbf{B} = [B_1, B_2, \ldots, B_m]$. Both strategies have the same amount of floating point operations, but the matrix-vector product is slower because it is bandwith bound (the transfer rate between memory and processors slows the overall process). This happen both in CPU and GPU [2] architectures.

The above findings lead us to think that, better time results can be achieved through an strategy that favours simultaneous testing of many emitter configurations, as happens in population based metaheuristics.

Flux and Flux density Goals and Constraints

A designer might not be interested only in the flux density (W/m^2) measured as radiosity. Often there are goals and constraints related to lighting flux, measured in W, like the emitted flux, the reflected flux or the incoming flux. Also, the designer could be interested in other kinds of flux density such as the incoming flux density (irradiance), the indirect irradiance, or the reflected radiosity, among others. In Table 8.1 there is a list of "flux", "flux density" and "importance" [3] quantities and their respective equations.

Table 8.1: Lighting-Representation Equations

Representation (L_r)	**Equation**
Flux $[W]$	
Reflected Flux (L_{Fr})	$(\mathbf{I}-\mathbf{RF})\mathbf{A}^{-1}L_{Fr}=\mathbf{RF}E$
Emitted Flux (L_{Fe})	$L_{Fe}=\mathbf{A}E$
Incoming Flux (L_{Fi})	$(\mathbf{I}-\mathbf{FR})\mathbf{A}^{-1}L_{Fi}=\mathbf{F}E$
Flux density $[W/m^2]$	
Irradiance (L_I)	$(\mathbf{I}-\mathbf{FR})L_I=\mathbf{F}E$
Direct Irradiance (L_{Id})	$L_{Id}=\mathbf{F}E$
Indirect Irradiance (L_{Ii})	$(\mathbf{I}-\mathbf{FR})L_{Ii}=\mathbf{FRF}E$
Radiosity (L_B)	$(\mathbf{I}-\mathbf{RF})L_B=E$
Emitted Radiosity (L_E)	$L_E=\mathbf{I}E$
Direct Radiosity (L_{Bd})	$L_{Bd}=\mathbf{RF}E$
Reflected Radiosity (L_{Br})	$(\mathbf{I}-\mathbf{RF})L_{Br}=\mathbf{RF}E$
Importance (L_{Im})	$(\mathbf{I}-\mathbf{RF})^T L_{Im}=R$

It is important to highlight that all of these quantities are related to equations similar to the radiosity equation. Therefore, a possible avenue of work would be to apply the same strategies (low-rank, random based decomposition of the inverse) to the matrices involved, and the use of the same ILP strategies developed in this thesis.

Final Remarks

Much work must be accomplished prior to the inclusion of a complete set of inverse tools in any CAD software. The inclusion of glossy surfaces requires the use of other global illumination engines. The geometry as a design variable must be treated in a very clever way, to transform the designer intentions into computationally tractable problems. Therefore, creative work must be done in new ILP solvers, and in the translation of the descriptive LI into mathematical problems solvable with personal computers at interactive speeds.

Bibliography

[1] E.-G. Talbi, Metaheuristics: From Design to Implementation, Wiley Publishing, 2009.

[2] V. Volkov, J. W. Demmel, Benchmarking gpus to tune dense linear algebra, in: Proceedings of the 2008 ACM/IEEE conference on Supercomputing, SC '08, IEEE Press, Piscataway, NJ, USA, 2008, pp. 31:1–31:11.

[3] B. E. Smits, J. R. Arvo, D. H. Salesin, An importance-driven radiosity algorithm, in: Proceedings of the 19th annual conference on Computer graphics and interactive techniques, SIGGRAPH '92, ACM, New York, NY, USA, 1992, pp. 273–282.