A Photon Tracing Approach to Solve Inverse Rendering Problems

Ignacio Avas
Centro de Cálculo
Instituto de Computación
Universidad de la República
Montevideo-Uruguay
Email: javas@fing.edu.uy

Eduardo Fernández
Centro de Cálculo
Instituto de Computación
Universidad de la República
Montevideo-Uruguay
Email: eduardof@fing.edu.uy

***Abstract*—Lighting intentions are the goals and constraints that designers like to achieve in a lighting design process. In this context, rendering problems are the kind of problems based on the rendering equation that are proposed to satisfy a set of lighting intentions. These problems are usually expressed as optimization problems.**

In this article is presented a novel method based on photon tracing, the VNS optimization metaheuristic, and the determination of the number of photons needed, allows to handle a wider variety of lighting intentions without incurring in high computational costs. Moreover, the method developed shows to be efficient when the geometry is also a variable in the rendering problem. The techniques explained here could be included in a package used by architects or designers to aid in the lighting design process of architectural environments.

I. Introduction

Rendering problems (RPs) are the kind of problems related to the rendering equation (Eq. 1). These problems include the direct problem -rendering an image given a scene composed of geometry, reflectivity properties, and emitters-, as well as the inverse and optimization problems, where the scene contains some unknowns and their solutions imply the satisfaction of a set of constraints and goals.

To solve inverse or optimization RPs, the brute force approach is not a good strategy. The computational effort often grows exponentially in relation to the number of variables, goals and constraints to be satisfied. In this context, inverse RPs were analyzed in previous work by using radiosity and ray tracing techniques, among others [1], [2]. These techniques are time consuming, may include an expensive pre-computation stage, or only can solve a particular kind of problems efficiently.

In this paper we propose a new approach for solving inverse and optimization RPs based on Photon Tracing (PT) [3], and the use of a metaheuristic to implement the optimization process. The metaheuristic evaluates a set of possible configurations for the unknowns, using PT and statistical tools to determine the value of the optimization function. The PT technique, is the global illumination method used to evaluate the objective function and constraints of each configuration tested by the metaheuristic. In PT, one problem to deal with is its stochastic nature. Each result has an error which is related to the number of photons used in the simulation.

The main contribution of this work is the development of an efficient way to solve inverse and optimization RPs using PT. Specifically the contributions are:

- Efficient resolution of RPs based on a stochastic approach.
- Estimation of the number of photons in each step of the optimization process, to reduce the computational effort.
- Optimization of scenes composed of non-Lambertian surfaces and anisotropic emitters at interactive rates.
- Implementation of a RP solver, that serves as proof of concept for the proposed approach.

The rest of the paper is organized as follows: In Sec. II the main concepts used through this work are presented. Sec. III presents the method developed to solve RPs using PT and optimization techniques. In Sec. IV are explained the details of the RP solver, and in Sec. V the main results of the method are presented. Finally, in Sec. VI the conclusions and further work are summarized.

II. Related Work

According to Kajiya [4], the rendering equation (Eq. 1) generalizes a variety of known rendering algorithms, simply balancing the energy flows from one point of a surface to another.

$$L_o(x' \to x) = L_e(x' \to x) + \int_S f_r(x'' \to x' \to x) L_i(x'' \to x') V(x', x'') G(x', x'') dA'' \quad (1)$$

In this equation, $L_o(x' \to x)$ refers to the radiance ($Wsr^{-1}m^{-2}$) in a surface point x from another point x', $f_r(x'' \to x' \to x)$ is the bidirectional reflectance distribution function (BRDF), i.e. the proportion of light reflected from x'' to x at x', $L_i(x'' \to x')$ is the incident radiance at x' from x'', $V(x', x'')$ is a visibility term which values 1 when x and x' are mutually visible and 0 otherwise, $G(x', x'')$ is a geometry term that relates the distance between x' and x'', their normal's vectors and the directions of the emitted and incident rays, and S is the set of all surface points. More details can be found in Jensen [5].

Eq. 1 can be solved using different approaches. Ritschel et al. [6] establish that the classical approaches to compute (interactive) global illumination are: finite elements (e.g., radiosity) [7], Monte Carlo ray tracing [4], photon mapping [3], instant radiosity [8], many-light-based global illumination [9], point-based global illumination [10], discrete ordinate methods [11], and precomputed radiance transfer [12]. These techniques intend to simulate 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 to deal with the scalability, the amount of bounces, multiple glossy bounces, multiple scattering, and complex light sources.

Photon mapping includes PT as one of its main components. Our proposal uses PT in each step of the optimization process, combined with a statistical method to limit the amount of photons needed, which results in the speed up of the overall process.

A. Inverse and Optimization Rendering Problems

According to Tarantola [13], 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 is direct or 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 and have many or even infinite solutions. They are of interest in a wide range of fields, including lighting engineering and lighting design.

Marschner [14], studied the rendering equation and considered that there are three main kinds of inverse RP: the inverse lighting problems, the inverse reflectometry problems, and the inverse geometry problems. In the inverse lighting problems (related to $L_e(x' \rightarrow x)$ in Eq. 1), the shape and reflection properties of a scene are known, and the problem consists in finding the emission in the scene. In the inverse reflectometry problems the variables of the problem are related to the reflection properties of the surfaces ($f_r(x'' \rightarrow x' \rightarrow x)$ in Eq. 1). Finally, in inverse geometry problems the unknowns are linked to the geometry of the object (mainly $V(x', x'')$ and $G(x', x'')$ terms in Eq. 1). Two important surveys on IRP are [1], [2].

An inverse problem can be formulated as an optimization problem. When the problem consists in approximating a model using experimental data, it is solved using a linear or nonlinear least squares method [13]. When the problem includes other goals and constraints to satisfy, more general optimization methods are required [15].

An optimization problem consists in finding the solution that minimizes an objective function from all feasible solutions, which are limited by a set of constraints [15]. For illumination purposes, the objective function and the constraints of the RP are related to lighting intentions. Castro et al. [16] explored a wide range of optimization techniques to solve economically optimal lighting positioning. Other authors [17] [18] simplify the scenes to rectangular spaces, and the optimization RP is solved through a generalized extremal optimization approach.

To speed up the calculations related to the lighting of the scene in architectural models, global illumination coherence is used in [19] with a low-rank radiosity (LRR) approach [20] in combination with a metaheuristic method for optimization. In this case, optimal shapes of diffuse sky lights can be obtained in mere minutes. This method allows to combine lighting contraints and goals. However, the main restriction to this method is that it only considers Lambertian emitters and surfaces.

B. Photon Tracing

Photon tracing (PT) and photon gathering are the two main steps in the photon mapping [3] process, which is a global illumination method based on the use of flux elements called photons. In PT, each photon is emitted from a light source towards the scene, and then it is reflected, transmitted or absorbed by a scene element. After each interaction, the information of the photon is stored in a *Photon Map*. Then, the photon mapping uses the photon gathering step to render the final image. In this step, the information stored is used along with a ray-tracing method to render the final image.

Photon tracing is a highly parallelizable process, because the calculation of each photon interaction with the geometry is independent. Several GPU based approach were used to implement it, mainly based on CUDA [21] and OptiX [22], [23]. During the PT stage, if a photon collides with a surface, and if the diffuse component of the reflection is applied, then the coordinates, incident direction of the collision, and power are stored in a structure called *Photon Map*. In all cases the photon can rebound on the surface, be absorbed by it, or be refracted. If the photon is not absorbed, then a new photon is created and sent from the intersection point toward the scene. The direction of the new photon is randomly determined, depending on the BRDF [24] property of the surface.

In this paper a variation of PT is used. Unlike photon mapping, the first bounce of photons is also stored (in photon mapping, the energy of the first bounce is measured from a ray-tracing process that runs in the gathering stage). All the information needed to evaluate a scene is computed here. The photon map structure is not needed when the sum of the power of the photons arriving to a particular surface gives enough information (see Sec. III). In other cases, such as when the distribution of the light in a surface is needed, a photon map structure can be useful in the calculation of the sample standard deviation or in other statistical methods.

PT is a stochastic process, due to the random nature of the processes implied in the generation and rebound of photons. Two executions of the same algorithm may yield slightly different results. While photon mapping is a biased process, due to the radiance estimation done during the photon gathering process, the PT component has no bias [5]. Therefore,

the average expected value is always correct, even for small samples, and the errors in the result are considered as variance.

III. Stochastic Optimization Using Photon Tracing

An inverse or optimization RP is a problem based on the rendering equation, where a set of optimal configurations of a given scene must be found. The objective of these problems consist in optimizing a function related to some radiometric or photometric unit in a scene, subject to certain set of constraints. This can be mathematically formulated as:

$$\arg\min_{x} f(x) \quad \text{subject to } x \in S \tag{2}$$

where $f{:}S{\to}\mathbb{R}$ represents the objective function to optimize, S contains all feasible configurations of the scene, which are usually defined by a set of constraints, and x is a specific configuration of the scene. For example, in the search of the optimal position of a point light in a ceiling to maximize lighting on a table, S is the set of all the points belonging to the ceiling surface.

In this paper, we use the irradiance E_s ($\mathrm{W/m^2}$) of a surface s as the objective function f. The results could be extended to other units as the radiosity, or the radiance. But irradiance is specially useful in building design, because it is used to measure the amount of light that arrives from the sun, or the light that arrives on an office desk. The equivalent in photometric units is the illuminance, measured in luxes (lx).

The irradiance can be estimated by adding the power of the photons that reached s and dividing it by its area (A_s). An approximation of $E_s(x)$ for N emitted photons ($E_N(x)$) can be formulated as:

$$E_s(x) \approx E_N(x) = \frac{\sum_{p \in P} \Phi_p(x)}{A_s} = \frac{\Phi_s(x)}{A_s} \tag{3}$$

where P is the set of photons that impacted in the surface s, $\Phi_p(x)$ is the the power of each photon, A_s is the area of s, and $\Phi_s(x)$ is the accumulated power arriving on s. Each photon is emitted with a power proportional to Φ_L/N, where Φ_L is the power of the light-source, and N is the number of photons emitted. An increment in N reduces the power of each photon ($\propto N^{-1}$) as well as the confidence interval of $E_s(x)$ ($\propto N^{-1/2}$), but increases the computational effort needed in its calculation.

A. Distribution of objective functions

If we run a Monte Carlo simulation based on PT, where $E_s(x)$ is its desired result, the estimation of $E_s(x)$ is guided by the central limit theorem [25], and its confidence interval (CI) satisfies a normal distribution (Fig. 1). Therefore, there is a probability of $\alpha\%$ that $E_s(x)$ is contained in a CI centered on $E_N(x)$ with radius $\sigma_N(E_s(x))z_\alpha$:

$$\mathrm{CI}_\alpha(E_s(x)) = [E_N(x) - \sigma_N(E_s(x))z_\alpha \,,\, E_N(x) + \sigma_N(E_s(x))z_\alpha]$$

where $\sigma_N(E_s(x))$ is the estimated standard deviation of $E_s(x)$ and z_α is the critical value associated to the normal distribution ($z_{90\%}$=1.645, $z_{95\%}$=1.96, $z_{99\%}$=2.575). For instance, the interval $\mathrm{CI}_{95\%}(E_s(x))$ means that we are 95% confident that $E_s(x)$ is inside it. The size of the CI_α, for a given α varies proportionally to $N^{-1/2}$.

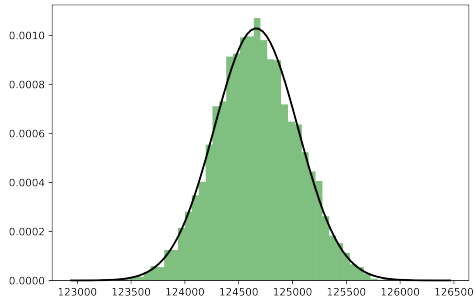


Fig. 1. Histogram of 4000 samples of $E_N(x)$ on a surface in the Hangar Scene (Fig: 7), and the Normal distribution $N(\overline{E_N}(x), \sigma(E_N(x))^2)$

When $\Phi_s(x)/\Phi_L \ll 1$, the path of a photon hits a surface s only once in most cases. Then the standard deviation σ of $E_N(x)$ can be approximated based on the binomial distribution:

$$\sigma_N(E_s(x)) = \frac{\Phi_L}{A_s N}\sqrt{Np(1-p)} = \frac{\Phi_L}{A_s}\sqrt{\frac{p(1-p)}{N}}\,,$$
$$\text{where } p = \frac{\Phi_s}{\Phi_L}$$

When $\Phi_s(x)/\Phi_L$ is larger (it could be even larger than 1 in highly reflective environments), then s could be divided into several smaller patches s' until $\Phi_{s'}(x)/\Phi_L \ll 1$. This strategy allows the use of the binomial distribution in any case.

B. Comparing Configurations

As explained in Sec. III-A the approximation $E_N(x)$ to $E_s(x)$ follows a normal distribution. It is necessary to compare the values of $E_s(x)$ for different configurations x to find the optimum one, during the optimization process. The comparison is based on the sample mean $\bar{\mu}(E_s(x)) = E_N(x)$ and the sample standard deviation $\sigma_N(E_s(x))$, because the exact value $E_s(x)$ is not known.

Now we proceed to define a set of operators ($>_\alpha$, $<_\alpha$, $=_\alpha$, and $\neq_\alpha$), used to compare the irradiances $E_s(x)$ and $E_s(y)$ for two configurations x and y. These operators mean that we are $\alpha\%$ confident that the relation between the irradiances is $>$, $<$, $=$, or $\neq$, respectively. Therefore, given two configurations x and y it is said that:

- $E_s(x) >_\alpha E_s(y)$ (we are $\alpha\%$ confident that $E_s(x) > E_s(y)$) iif the confidence interval of $E_s(x) - E_s(y)$ (i.e. $\mathrm{CI}_\alpha(E_s(x) - E_s(y))$) has only positive values. So:

$$\begin{gathered} E_s(x) >_\alpha E_s(y) \Leftrightarrow \\ \bar{\mu}(E_s(x) - E_s(y)) - \bar{\sigma}(E_s(x) - E_s(y))z_\alpha > 0 \end{gathered} \tag{4}$$

  The approximations $E_N(x)$ and $E_N(y)$ are independent random variables, which implies that:

$$\bar{\mu}(E_s(x) - E_s(y)) = E_N(x) - E_N(y) \tag{5}$$

$$\bar{\sigma}(E_s(x) - E_s(y))^2 = \sigma_N(E_s(x))^2 + \sigma_N(E_s(y))^2 \tag{6}$$

Therefore, combining Eqs. 4, 5, and 6 we can conclude that:

$$E_s(x) >_\alpha E_s(y) \text{ iif} \quad (7)$$
$$E_N(x) - E_N(y) > z_\alpha \sqrt{\sigma_N(E_s(x))^2 + \sigma_N(E_s(y))^2}$$

Using the same reasoning it could be said that:

- $E_s(x) <_\alpha E_s(y)$ iif
$$E_N(y) - E_N(x) > z_\alpha \sqrt{\sigma_N(E_s(x))^2 + \sigma_N(E_s(y))^2}$$
- $E_s(x) \neq_\alpha E_s(y)$ (x and y are *distinguishable*, meaning that we are α% confident that $E_s(x) > E_s(y)$ or $E_s(x) < E_s(y)$) iif:
$$|E_N(x) - E_N(y)| > z_\alpha \sqrt{\sigma_N(E_s(x))^2 + \sigma_N(E_s(y))^2} \quad (8)$$

When both configurations are distinguishable, it can be said that there is a relation of $>_\alpha$ or $<_\alpha$ among them.

- $E_s(x) =_\alpha E_s(y)$ (x, y are *indistinguishable*, or we are not α% confident that $E_s(x) > E_s(y)$ or that $E_s(y) > E_s(x)$) iif:
$$|E_N(x) - E_N(y)| < z_\alpha \sqrt{\sigma_N(E_s(x))^2 + \sigma_N(E_s(y))^2} \quad (9)$$

The above operators are based on statistics, so the relations proposed can generate false solutions (e.g., it may happen that $E_s(x) >_\alpha E_s(y)$ but $E_s(x) < E_s(y)$).

Given 2 configurations x and y such that $E_N(x) =_\alpha E_N(y)$, the operator $=_\alpha$ should be transformed into $\neq_\alpha$ (i.e. $<_\alpha$ or $>_\alpha$) for optimization purposes, by increasing the value of N. This increment reduces the size of $\bar{\sigma}$ proportionally to $N^{-1/2}$, allowing to transform Eq. 9 into Eq. 8. But, to reduce in $^1/_2$ the length of a CI an increase of $4\times$ in the number of photons is required. Since the move from $=_\alpha$ to $\neq_\alpha$ may imply a very large number of photons, this slows down the PT performance in each step of the optimization process. Moreover, this computational effort could be useless because both configurations are far from the optimum, and both will be erased in future optimization steps. Therefore, we have to administrate efficiently the computational effort, avoiding to distinguish all early $=_\alpha$ problems. To gain efficiency in the use of the computational resources, we define (see Sec. III-C) an *Indistinguishable Set of Optimum Configurations* (ISOC), and the *Superposition Interval* associated to it (SIISOC), which are used to simplify the management of the mentioned set.

C. Optimal Configuration Set

Eq. 9 defines when two configurations are indistinguishable for a given N. Now, we define that an indistinguishable set of configurations (ISC) fulfills that

$$\forall x, y \in \text{ISC} : E_s(x) =_\alpha E_s(y)$$

When the set contains the optimum configurations it is called ISOC. Therefore, the objective of an optimization technique with stochastic functions is to define an ISOC, instead of finding a single optimal configuration. For the case in which the optimum is a maximum, ISOC meets the following criteria:

$$\forall x, y \in \text{ISOC} : E_s(x) =_\alpha E_s(y) \quad (10)$$
$$\forall z \notin \text{ISOC}\ \exists x \in \text{ISOC} : E_s(x) >_\alpha E_s(z)$$

1) Superposition Interval in an ISOC: Given two configurations x and y, their confidence intervals intersect each other when the distance between their sample means is smaller than the sum of their radius:

$$\text{CI}(E_s(x)) \cap \text{CI}(E_s(y)) \neq \emptyset \Leftrightarrow \quad (11)$$
$$|E_N(x) - E_N(y)| < (\sigma_N(E_s(x)) + \sigma_N(E_s(y)))\, z_\alpha$$

If $E_s(x) =_\alpha E_s(y)$ then the intersection of their confidence intervals is not empty. This happens because:

$$\sqrt{\sigma_N(E_s(x))^2 + \sigma_N(E_s(y))^2} \leq \sigma_N(E_s(x)) + \sigma_N(E_s(y))$$

and therefore, Eq. 11 is met when Eq. 9 is satisfied.

$$E_s(x) =_\alpha E_s(y) \Rightarrow \text{CI}(E_s(x)) \cap \text{CI}(E_s(y)) \neq \emptyset \quad (12)$$

On the other hand, the reciprocal is not true: Eq. 11 can be fulfilled while Eq. 9 is not satisfied.

A generalization of the above property is that the intersection of all confidence intervals in an ISOC is not empty:

$$\text{SIISOC} = \bigcap_{\forall x \in \text{ISOC}} \text{CI}(E_s(x)) \neq \emptyset$$

This interval is called *superposition interval in an ISOC* (SIISOC). To show its existence, *first* we define an interval whose extremes are the maximum of minimums of all CIs (*maxmin*) and the minimum of maximums of all CIs (*minmax*).

$$maxmin = \max_{x \in \text{ISOC}} \{\ \min(\text{CI}(E_s(x)))\ \}$$

$$minmax = \min_{y \in \text{ISOC}} \{\ \max(\text{CI}(E_s(y))\ \}$$

A *second* concept to show is that $maxmin < minmax$ (Fig. 2). This happens because, when both numbers belong to the same CI, the inequality is satisfied by construction (left case in Fig. 2). When both numbers belong to two different CIs, these CIs must intersect each other (Eq. 12), therefore, the only possibility is that $maxmin < minmax$ (middle and right cases in Fig. 2). A *third* interesting concept is that all the CIs of the ISOC, include the $[maxmin\ minmax]$ interval: $\forall z \in \text{ISOC}$,

$$\min(\text{CI}(E_s(z)) \leq maxmin < minmax \leq \max(\text{CI}(E_s(z))$$

These inequalities are true because of the definition of *maxmin* and *minmax*, and because $\min\{\text{CI}(E_s(z))\} \leq maxmin$ and $minmax \leq \max\{\text{CI}(E_s(z))\}$. Therefore, $[maxmin\ minmax] \subset$ SIISOC is satisfied. *Finally*, by construction, each extreme of the interval $[maxmin\ minmax]$ is also an extreme of at least one CI belonging to the ISOC, then the SIISOC can not be larger than $[maxmin\ minmax]$. Therefore:

$$[maxmin\ minmax] = \text{SIISOC}$$

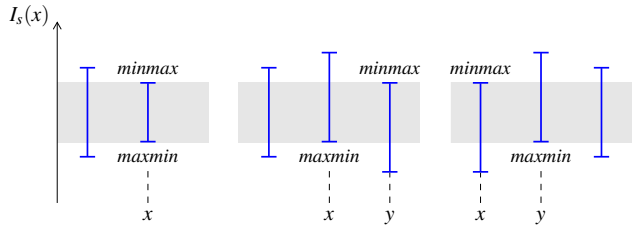


Fig. 2. Three possible states in SIISOC construction given an ISOC.

2) Using SIISOC in a rendering problem: The objective of the optimization process associated to any not-direct RP is to find an ISOC. In each iteration of the optimization process, a new configuration x' is evaluated and compared to the ISOC. Many x' are associated with larger confidence intervals of $E_s(x')$ than at least one element x of the ISOC (in our case $E_s(x') <_\alpha E_s(x)$), but eventually some x' has better confidence interval ($E_s(x') >_\alpha E_s(x)$), which forces to build a new ISOC. To do an efficient comparison of x' against the ISOC, we propose to use the SIISOC interval. For instance, if $\mathrm{CI}(E_s(x'))$ does not intersect the SIISOC and has worse values than the SIISOC, then x' should be discarded (Sec. III-C1). But, if $\mathrm{CI}(E_s(x'))$ has better values than the SIISOC, then both the ISOC and the SIISOC should be modified.

There are three possible states to take into consideration:

1) $\max \mathrm{CI}(E_s(x')) < maxmin$ (see x'_1 in Fig. 3). This means that the CI of $E_s(x')$ is "below" the SIISOC, so there is a $x \in$ ISOC such that $E_s(x')$ is farther to the objective than $E_s(x)$, and therefore x' is discarded.
2) $\mathrm{CI}(E_s(x')) \cap \mathrm{SIISOC} \neq \emptyset$ (see x'_2 in Fig. 3). In this case it is needed to check $\forall x \in \mathrm{SIISOC}$ that $E_s(x) =_\alpha E_s(x')$ is satisfied. There are also three cases to consider:
   a) If $\exists x \in \mathrm{ISOC}$: $E_s(x') <_\alpha E_s(x)$, then x' does not belong to the ISOC.
   b) If $\exists x \in \mathrm{ISOC}$: $E_s(x) <_\alpha E_s(x')$, then x must be erased from the ISOC and x' included in the ISOC.
   c) If $\forall x \in \mathrm{ISOC}$: $E_s(x) =_\alpha E_s(x')$, then x belongs to the ISOC.

   The cases 2a and 2b do not happen simultaneously. In an ISOC there are not a x and y configurations such that $E_s(x) <_\alpha E_s(x') <_\alpha E_s(y)$. This would mean that $E_s(x) <_\alpha E_s(y)$. Then, x and y can not belong to the same ISOC.
3) $maxmin < \min \mathrm{CI}(E_s(x'))$ (see x'_3 in Fig. 3). This means that the CI of $E_s(x')$ is "above" the SIISOC, and that $E_s(x) <_\alpha E_s(x')$ for at least one x belonging to the ISOC. Therefore, x' now belongs to the ISOC. All other $x \in$ ISOC must satisfy that $E_s(x') =_\alpha E_s(x)$, otherwise they are discarded. For instance, in Fig. 3 the new ISOC is formed by $\{x_4, x'_3\}$.

The distinction between these three states would reduce the evaluation time of a new configuration x'. This is specially true when its CI is worse than the SIISOC (state 1). In state 2, the comparison with the ISOC finishes when a $x \in$ ISOC such that $E_s(x') <_\alpha E_s(x)$ is found.

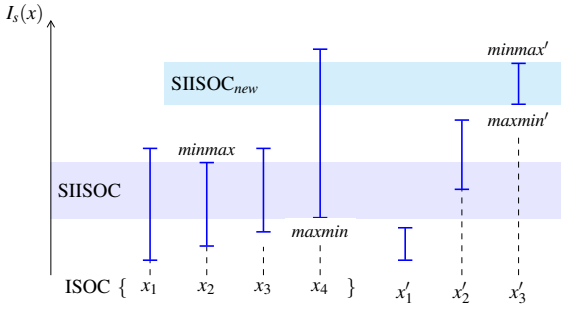


Fig. 3. ISOC, SIISOC, and the three possible states considered when SIISOC is used in inverse RPs (x'_1, x'_2, and x'_3)

IV. Optimization Process

This section illustrates the optimization process used to find the indistinguishable set of optimal configurations (ISOC). The process is iterative, as shown in Fig. 4. A configuration x is applied to the scene and then $E_s(x)$ is evaluated using PT, and used as the optimization function. The process stops after a number of iterations given. Such number ought be adjusted taking into consideration the scene complexity, as well as the characteristics of the optimization variables. The setting of x implies that the scene objects, the lights and/or material parameters are modified following certain rules or constraints. This modification in the scene is done through OptiX using fast hierarchical data structures (in this work we use a variant of the BVH called TRBVH [26]), as well as modifying the scene material properties, which does not require the recalculation of the data structure. At each iteration there is an ISOC, which is refined by adding and removing elements, following the SIISOC strategy (Sec. III-C2).

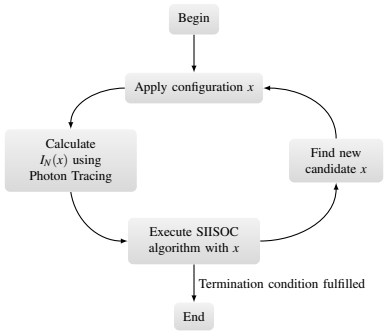


Fig. 4. Optimization process

The algorithm implemented follows the concepts introduced in Sec. III, using OptiX to implement PT and VNS metaheuristic [27] to guide the optimization process[1]. The program (RPsolver) consists of two parts: a main optimization code based on VNS and used to find the ISOC, and a PT library which is applied to each scene configuration x to calculate the $\mathrm{CI}(E_s(x))$. The PT engine is based on the work

[1]The source code of the implementation is available at https://github.com/igui/RPSolver

of Pedersen [28]. RPsolver implements a number of shortcuts in the process such as:

- The dynamical adjustment of the number of photons launched in the PT stage depending on the configuration, the available GPU memory, and the scene complexity.
- The use of a discrete solution space, to avoid the generation of a large ISOC due to the definition of many close and indistinguishable ($=_\alpha$) configurations. This shortcut speeds up the overall process.
- The use of a tabu search metaheuristic [29] for caching of configuration values, and to avoid the evaluation of previously visited configurations.

A. Sample size determination

When comparing two configurations it could be useful to consider that a small value of N often is enough to distinguish ($\neq_\alpha$) their irradiances. Since calculating $E_N(x)$ is $O(N)$, using a small initial sample of photons n can be enough to compare a candidate to the SIISOC. When a candidate configuration x is evaluated, our algorithm starts with an initial sample size n ($n \ll N$). If the value of $E_n(x)$ intersects or is better than the SIISOC x, then a more accurate estimation with N samples is performed, and the comparison algorithm is executed again. Alternatively, the algorithm can start calculating only $E_n(x)$ with for a small n and, after the optimization process ends, $E_N(x)$ is computed to all elements in the ISOC, to filter the real solutions. The algorithm can be configured to use these or other approaches, depending on the lighting intention as well as the geometry of the scene. Selecting an appropriate sample size n can help improve the execution time. A small value of n results in a very inaccurate value of $E_n(x)$ and a large set of ISOC solutions. On the other hand, a large value of n can be harmful because it takes considerable time for computing solutions that are far from the final ISOC. The values n and N are also influenced by the available memory needed to store the photon map.

B. Domain discretization

To reduce the number of configurations in the ISOC, and to prevent the output of many similar configurations, the domain is transformed into a finite set of elements. Here we assume the well continuity and spatial coherence of the irradiance function $E_N(x)$, which means that close configurations x will lead to close irradiance values $E_N(x)$. Following this line of thought, we assume that $E_s(x)$ is indistinguishable from all other configurations x' belonging to the same cell of the grid ($E_s(x) =_\alpha E_s(x')$, when x and X' belong to the same cell, and for a given sample size N). In this sense, the domain is mapped into a kD grid $G : M_1 \times \ldots \times M_k \rightarrow \mathbb{R}$.

A tabu search metaheuristic is applied in the algorithm, where the value of $I_N(x^*)$ is computed only if x^* is the first candidate explored in the cell $G(x^*)$; each subsequent candidate falling in that cell will use the cached value of $I_N(x^*)$. When selecting the grid size, it should be taken into consideration that a coarse mesh can lead to low quality solutions, and that a fine mesh may imply many iterations of the optimization process.

V. Results

The objective of this section is to show the main results when the technique is used to solve some kinds of RPs using relatively complex scenes. Three sample problems are analyzed and different performance metrics are shown. We focus in the measurement of the performance of the algorithm for different scenes.

A. Test hardware

RPsolver was tested in a desktop Computer. Table I specifies the hardware details.

CPU Model	Intel i5 3470
CPU Frequency	3200MHz
RAM Memory	8192MB
GPU Model	GeForce GTX 760
GPU Memory	2048MB
GPU Frequency	980Mhz
GPU Core count	1152
GPU Flops	2258 GFLOPS

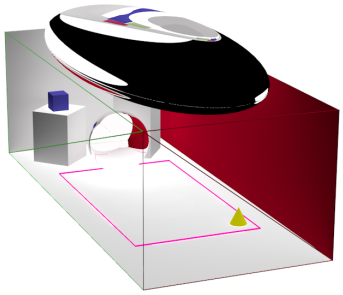
TABLE I
Test hardware specification

B. Test problems

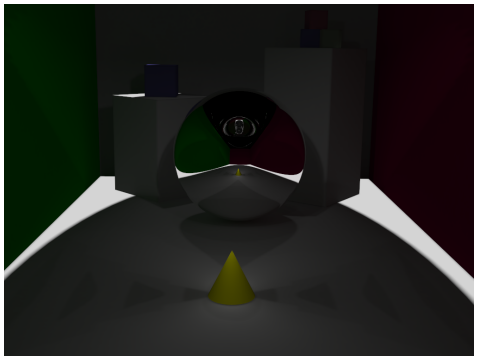
Three scenes are used, each one with a different lighting intention. The objective to achieve is to maximize the irradiance E on a surface s when the scene has the configuration $x \in S$. Beyond this, many other problems could be solved. Up to 4 bounces are allowed. For all scenes, the value of N -maximum number of emitted photons-, is 2.66×10^6, but the value of n -minimum number of emitted photons- varies with the scene. The algorithm stops after 1000 iterations.

1) Cone in Cornell Box: The first problem consists in finding the best position of a cone in a *Cornell Box* (Fig. 5) such that the irradiance on the cone surface is maximized. The Cornell box used in this problem contains non-diffuse surfaces represented by a kind of magnifying glass and a reflective sphere, and a vertical directional light. The lens concentrates the light in the middle of the scene, generating a caustic. Both the lens and the sphere are faceted. The scene has a total of 32548 polygons. The value of n is 12544. For this problem the algorithm refine the ISOC after the optimization process stop, using N photons for the estimation of the irradiance values.

In this problem the geometry of the scene is modified at each step. The starting position of the cone is in the corner of the box and is always at floor level. The optimal position of the cone is expected to be somewhere below the central point of the lens, where it concentrates light. After executing the RPsolver, the solutions found seem to match the expectations, placing the cone under the center of the lens.



(a) General view.



(b) A solution.

Fig. 5. Cornell Box.

2) Conference Room: The second scene is a meeting room composed of 644508 triangles, which includes textures in the blackboard and in the floor (Fig. 6a). The test problem consists in finding the optimal two light positions for illuminating a podium near the blackboard. The light sources are two squares that can be located anywhere in the ceiling. The emitters sends light equally distributed in every direction (an isotropic light source). The initial position of the light sources is in the opposite corner of the room. The optimization function is the irradiance of the podium, based on the (x,y) position of the light sources. The lights are expected to move towards the podium, finding a solution somewhere above it. The value of n is 9216. For this problem the algorithm is configured to reevaluate the configuration using N photons, before it is added to the ISOC as a new solution.

After running the program, the ISOC contains solutions as shown in Fig. 6b, where the light sources are located directly above the podium as expected. The target function is relatively simple and rises near the podium, as is expected.

3) Hangar: The last scene shows a hangar consisting of 161988 triangles, with two lights on the ceiling that simulate neon tubes. The lighting intention is to maximize the irradiance of a glass box which can be placed anywhere in the floor. The walls and the floor are symmetrical about the x axis. The scene presents a texture mapping on the floor and non-diffuse surfaces like the glass box. Fig. 7 displays of the scene.

While the Cornell Box and the Conference Room have one solution, the Hangar problem has at least two optimal solutions, corresponding to the box situated right below one of each lights. The value of n is 30976. For this problem the algorithm is configured to refine ISOC at the end using N photons.

(a) General view.

(b) A solution.

Fig. 6. Conference Room.

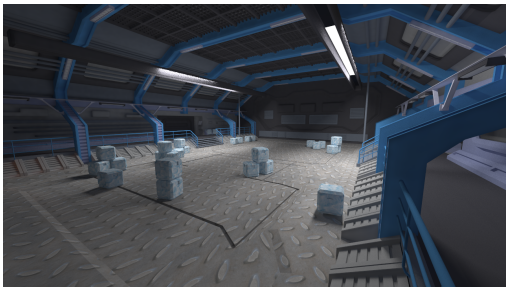


Fig. 7. Hangar.

C. Performance analysis

The performance of the RPsolver was tested by running each test ten times. The Table II shows the time per iteration for each scene using, in average, a small number of photons per iteration, as described in Sec. IV-A, and the time using always N photons.

The time per iteration increments in a sub-linear rate in relation to the number of polygons, probably due to the use of acceleration structures (TRBVH) in each PT step, which allow efficient ray-polygon intersection operations [30].

From the analysis of Table II, can be stated that the component that imply larger execution times is the PT process. PT process depends directly on the number of photons. A conservative selection of the number of photons, where N photons are always used just in case, produces a great slowdown of the algorithm. This is shown in the last three rows of the table. When a small number of photons is used in many iterations, they produce rough, yet useful estimate $E_s(x)$, as well as large speed up values.

	Cornell Box	Hangar	Confer. Room
Reconstruction of acceleration structures	3.07	7.55	9.55
Avge. Photon Tracing ($N \gg n$)	11.47	67.79	156.37
Calculate Hit Count	0.40	1.07	1.93
Other	1.23	1.38	1.36
Total time using n, N ; $N \gg n$	16.27	77.79	169.21
Total time with static N	760.45	1564.10	1682.15
Speedup	43.74	20.11	9.94

TABLE II
TIME PER ITERATION PER SCENE, EXPRESSED IN MS.

VI. CONCLUSIONS AND FUTURE WORK

In this work, a set of new techniques are presented to solve rendering problems. These techniques are based on PT, and the use of statistical concepts like the confidence interval. The concepts of ISOC and SIISOC are presented. The geometry of the scene was optimized (Cornell Box), as well as the position of anisotropic (Sponza) and isotropic (Conference Room) lighting. It is possible to work with scenes composed of hundreds of thousands of elements.

Future work involves replacing the VNS optimization metaheuristics with other metaheuristics like population-based ones [31] that incentives finding more diverse solutions. Using several initial points can further improve the optimization speed. Another line of future work is the use of this technique to solve more complex problems, like the design of optimal interior shapes and luminaries, considering lighting goals, reflectometry parameters, and different kind of constraints.

VII. ACKNOWLEDGMENTS

This work was partially supported by project FSE_1_2017_144731 of the Agencia Nacional de Investigación e Innovación, Pedeciba Informática, and Universidad de la República.

REFERENCES

[1] G. Patow and X. Pueyo, "A survey of inverse rendering problems," *Computer Graphics Forum*, vol. 22, no. 4, pp. 663–688, 2003.

[2] ——, "A survey of inverse surface design from light transport behavior specification." *Computer Graphics Forum*, vol. 24, no. 4, pp. 773–789, 2005.

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

[4] J. T. Kajiya, "The rendering equation," *SIGGRAPH Comput. Graph.*, vol. 20, no. 4, pp. 143–150, Aug. 1986.

[5] H. W. Jensen, *Realistic Image Synthesis Using Photon Mapping*. Natick, MA, USA: A. K. Peters, Ltd., 2001.

[6] T. Ritschel, C. Dachsbacher, T. Grosch, and J. Kautz, "The state of the art in interactive global illumination," *Comput. Graph. Forum*, vol. 31, no. 1, pp. 160–188, Feb. 2012.

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

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

[9] M. Hašan, F. Pellacini, and K. Bala, "Matrix row-column sampling for the many-light problem," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007.

[10] T. Ritschel, T. Engelhardt, T. Grosch, H.-P. Seidel, J. Kautz, and C. Dachsbacher, "Micro-rendering for scalable, parallel final gathering," in *ACM SIGGRAPH Asia 2009 papers*, ser. SIGGRAPH Asia '09. New York, NY, USA: ACM, 2009, pp. 132:1–132:8.

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

[12] P.-P. Sloan, J. Kautz, and J. Snyder, "Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 527–536, Jul. 2002.

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

[14] S. R. Marschner, "Inverse rendering in computer graphics," Ph.D. dissertation, Program of Computer Graphics, Cornell University, Ithaca, NY, 1998.

[15] D. G. Luenberger and Y. Ye, *Linear and Non Linear Programming*, 3rd ed., ser. International Series in Operations Research & Management Science. Springer, 2008.

[16] F. Castro, E. del Acebo, and M. Sbert, "Energy-saving light positioning using heuristic search," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 3, pp. 566 – 582, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S095219761100220X

[17] F. Cassol, P. S. Schneider, F. H. R. Franca, and A. J. S. Neto, "Multi-objective optimization as a new approach to illumination design of interior spaces," *Building and Environment*, vol. 46, no. 2, pp. 331–338, 2011.

[18] P. S. Schneider, A. C. Mossi, F. H. R. Franca, F. L. de Sousa, and A. J. da Silva Neto, "Application of inverse analysis to illumination design," *Inverse Problems in Science and Engineering*, vol. 17, no. 6, pp. 737–753, 2009.

[19] E. Fernández and G. Besuievsky, "Inverse lighting design for interior buildings integrating natural and artificial sources," *Computers & Graphics*, vol. 36, no. 8, p. 1096–1108, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0097849312001550

[20] E. Fernández, "Low-rank radiosity," in *Proceedings of the IV Iberoamerican symposium in computer graphics. Sociedad Venezolana de Computación Gráfica*, 2009, pp. 55–62.

[21] D. B. Kirk and W.-m. W. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach*, 1st ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2010.

[22] *NVIDIA OptiX Ray Tracing Engine Programming Guide*, NVIDIA Corporation, january 2014.

[23] S. G. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, D. McAllister, M. McGuire, K. Morley, A. Robison, and M. Stich, "Optix: A general purpose ray tracing engine," in *ACM SIGGRAPH 2010 Papers*, ser. SIGGRAPH '10. New York, NY, USA: ACM, 2010, pp. 66:1–66:13. [Online]. Available: http://doi.acm.org/10.1145/1833349.1778803

[24] G. J. Ward, "Measuring and modeling anisotropic reflection," *SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 265–272, Jul. 1992. [Online]. Available: http://doi.acm.org/10.1145/142920.134078

[25] G. C. Canavos, *Applied probability and statistical methods:*. Little, Brown, 1984. [Online]. Available: https://books.google.com.uy/books?id=Vz3vAAAAMAAJ

[26] T. Karras and T. Aila, "Fast parallel construction of high-quality bounding volume hierarchies," in *High-Performance Graphics*, Anaheim, CA, USA, Jul. 2013.

[27] N. Mladenović and P. Hansen, "Variable neighborhood search," *Computers and Operations Research*, vol. 24, no. 11, pp. 1097–1100, Nov. 1997.

[28] S. A. Pedersen, "Progressive photon mapping on GPUs," Master of Science in Computer Science, Norwegian University of Science and Technology, Trondheim, 2013.

[29] F. Glover and M. Laguna, "Tabu search," in *Handbook of Combinatorial Optimization*, P. M. Pardalos, D.-Z. Du, and R. L. Graham, Eds. Springer New York, 2013, pp. 3261–3362.

[30] NVIDIA Corporation, *CUDA C Programming Guide*, NVIDIA Corporation, Mar. 2015.

[31] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "An overview of population-based algorithms for multi-objective optimisation," *International Journal of Systems Science*, vol. 46, no. 9, pp. 1572–1599, 2015.