



**HAL**  
open science

# A Fast Daylighting Method to Optimize Opening Configurations in Building Design

Eduardo Fernández, Benoit Beckers, Gonzalo Besuievsky

► **To cite this version:**

Eduardo Fernández, Benoit Beckers, Gonzalo Besuievsky. A Fast Daylighting Method to Optimize Opening Configurations in Building Design. *Energy and Buildings*, 2016, 125, pp.205-218. 10.1016/j.enbuild.2016.05.012 . hal-02153323

**HAL Id: hal-02153323**

**<https://univ-pau.hal.science/hal-02153323>**

Submitted on 24 Apr 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A fast daylighting method to optimize opening configurations in building design

Eduardo Fernández<sup>a,\*</sup>, Benoit Beckers<sup>b</sup>, Gonzalo Besuievsky<sup>c</sup>

<sup>a</sup> Centro de Cálculo, Universidad de la República, Uruguay

<sup>b</sup> Roberval Laboratory, Urban Systems Engineering, Université de Technologie de Compiègne, Sorbonne University, France

<sup>c</sup> Geometry and Graphics Group, Universitat de Girona, Spain

Daylighting plays a very important role for energy saving in sustainable building, therefore, setting the optimal shapes and positions of the openings is crucial for daylighting availability. On the other hand, computing daylighting for climate-based data is a time-consuming task involving large data set and is not well suited for optimization approaches. In this paper we propose a new and fast daylighting method that allows to perform opening shape optimizations. The base of our method is to model each element of an opening surface as a pinhole and then formulate a compact irradiance-based representation to ease global illumination calculations. We use the UDI metric to evaluate our method, on an office-based model, for different orientations and different geographical locations, showing that optimal windows shapes can be obtained in short times. Our method also provides an efficient way to analyze the impact of climate-based data on the shape of the openings, as they could be modified interactively.

## 1. Introduction

Configuring the opening shape and position is a crucial element for improving the daylight exploitation, a well known effective means to reduce artificial requirements of buildings. The problem of finding the best opening configuration involves two related tasks: the geometric model optimization and the daylight computation.

Concerning daylight measurements, nowadays there are well established metrics that take into account hourly-data for the whole year, such as the daylight autonomy (DA) [1] or the useful daylight illuminance (UDI) [2]. These metrics replaced successfully the rough approximation of the widely used daylight factor, with more realism. The metrics are known as climate-based, since they consider time-varying daylight illumination for a full year. As output of the computation, they evaluate the percentage in hours that a place can have daylight accessibility. Involving hourly Sun and sky conditions leads to work with a huge dataset, with thousands of skies.

Regarding the problem of finding the optimal geometric model that achieves a given goal, such as maximizing the daylight hours, this problem cannot be solved by standard CAD tools that work on

forward-based strategies. This strategy is unsuitable for optimization problems, where thousands of possible configurations should be tested. The problem should be stated as an inverse problem [3] and formulated as an optimization approach [4]. In the case of optimizing the shape for daylighting intentions, an additional difficulty is that we need to evaluate the whole hourly dataset of the year, at each iteration of the optimization.

One of the most used daylighting method is based on the daylight coefficient (DC) approach, originally proposed by Tregenza and Watters [5]. The concept behind this approach is to divide the sky dome into patches and the contribution of each sky tile is computed at each particular sensor position. Then, for a given configuration, it is possible to compute the total illuminance contribution for different sky conditions, for the whole year dataset. However, if the geometry changes, the DCs may also change and should be re-computed at each optimization step. This drawback discourages the DC method for optimization problems.

In this paper, we propose a new daylighting computation method which is suitable for the optimization of opening shapes. The base of the method is to represent each element of a window as a pinhole approximation. The incoming light that passes through each pinhole of the opening is then modeled using a compact radiosity formulation that allows to isolate the window as a single source contributor. In [6], a pinhole-based radiosity method was presented, but only for static sky and environment maps. The new method can deal with a whole-year dataset, providing fast

---

\* Corresponding author.

daylighting computation and considering full global illumination solutions. An important result obtained in our formulation is that the overall computational cost of the optimization process does not depend on the size of the scene nor on the number of sky tiles. We also show that our approach could work in problems based on the use of daylighting coefficients.

The main contributions of our work is a new formulation of daylighting computation that allows modifying the geometry of the opening. The new formulation enables us to obtain the optimal shape for windows at early stages of the design process. However, here we do not go further with building technological considerations of the result, such as real windows insertions or electric energy requirements, which are out of the scope of this work. Moreover, a complementary contribution is the possibility to analyze the impact of weather data on the geometry at the design phase, a subject not completely addressed before in the literature.

The rest of the paper is divided into six sections. In Section 2 related work for daylighting and inverse lighting problems is reviewed. In Section 3 the pinhole-based radiosity method is reviewed and in Section 4 the extension for computing illuminance from dynamic skies is formulated. Then, our optimization approach is described in Section 5. Our method was tested with a box office (Section 6). Our analysis provides results for different orientations and different geographical locations of this office. Finally, the conclusions and further work are summarized in Section 7.

## 2. Related work

The main related subject to this work are daylighting and the optimization problem, which are reviewed in this section.

### 2.1. Daylighting computation

Considering dynamic daylighting simulation, several metrics based on hourly measured data have been developed. These include daylight autonomy (DA) [1], continuous daylight autonomy (CDA) [1], useful daylight illuminance (UDI) [2], and spatial daylight autonomy (sDA) [7]. The daylight performance assessment of an interior space is computed from annual hourly illuminance values calculated over some sensors, typically at 0.75 m height representing a workplane. DA is the percentage of illuminance values above a minimum desired illuminance, without an upper threshold bound for illuminance. For this reason, it does not capture over-illuminated situations that can produce visual discomfort. CDA improves continuity by giving partial credits to values below the minimum desired illuminance. sDA relates the working space by computing the percentage of area that is above a threshold for at least 50% of the annual evaluated hours. These metrics also have the problem of not taking visual comfort into account. Another frequently used metric is the UDI [2], which is the percentage of illuminance values above a desired minimum, typically 100 lx, and below a desired maximum, typically 2000 lx. Unlike other metrics, UDI captures the daylight sufficiency and visual comfort of a design solution because values above the upper threshold are likely to cause visual discomfort/glare. In Section 6 we use UDI as the daylighting metric to optimize.

Regarding the daylighting computation, one of the most used methods is based on the DC approach originally proposed by Tregenza and Watters [5]. The concept of DCs is to divide the skydome into a set of sky tiles and then calculate the contribution of each sky tile to the total illuminance at various sensor points in a building based on each sensors position and orientation. The total sensor illuminance at a given point is obtained by linear superposition of each DC. Time-varying solar and sky tiles luminances can

be calculated using direct and diffuse irradiances from weather data-files. Working with DCs is a two-step process: first calculating the DCs, then folding them against time-varying luminances. The approach is very efficient for static scenes, but, when the geometry changes, the DCs should be re-computed. This discourages the use of a DC approach for optimization problems. However, it is computationally possible with time consuming executions. Recent approaches following this strategy, and aiming also to link daylighting to energy performance, can be found in [8,9]. In [8], an example for a particular model optimization using DC is presented, whereas in [9] the problem is addressed by studying a few number of configurations, without an optimization process.

Other daylighting methods are focusing on the efficient calculation of complex fenestration systems. In this case, a bidirectional scattering distribution function (BSDF) is used to represent the optical behavior of the opening. The “three-phase simulation method” [10,11] and the “five-phase method” [12] allow computing annual daylight performance for such systems using the RADIANCE package [13], by condensing the computation into several pre-computed matrices. For instance, in the three-phase method, the matrices account for the relation between sky patches and incident opening direction (**D**: daylight matrix), the relation between the incident opening direction and the exiting directions (**T**: transmission matrix) and the relation between the outgoing opening directions to the desired calculation points (**V**: view matrix). The matrix product **VDT** relates the luminance of the sky tiles with the illuminance of interior points of the scene. This strategy allows obtaining in a few seconds the illuminance at the desired points for any changing sky condition, since no rays are casted. It can be used also to optimize the BSDF (as for example the slat angle) by changing only the **T** matrix. However, this approach cannot be used to optimize the shape of the window, because the full computation of **V** and **D** matrices are required at each step of the optimization process, which is expensive in computational resources.

### 2.2. Opening optimization and inverse lighting problems

An early approach to inverse opening shape design was presented by Turre et al. [14]. They considered openings with anisotropic light sources, however, their work did not consider essential global illumination features neither occlusions. A more general solution, where the previous restrictions are overcome, is presented in [15]. This method considers openings composed of a set of small elements as in the present work. It computes the directional incoming light from the sky through parallel projections. Next, at each opening element, a directional and spatial representation is stored by means of anisotropic light sources. These light sources are then used to evaluate their importance for a given indoor lighting intention. The final solution used a ray-based method for global illumination taking several hours to achieve simple shape optimization. No existing lighting coherence was considered to accelerate the computational process.

Global illumination coherence in architectural models can be exploited using a low-rank radiosity (LRR) approach in combination with a meta-heuristic method for optimization [3,16]. In this case, optimal shapes of diffuse skylights can be obtained in minutes. However, the method is restricted to translucent surfaces.

Other related works on opening optimization focus on an integrated energy evaluation, which includes artificial lighting and thermal analysis [17,18]. In [17], they focus on the whole facade for optimization, then, only the window-to-wall ratio is used as a parameter for optimization. In [18], a genetic algorithm is used to optimize the modeling of windows as cells. However, they use

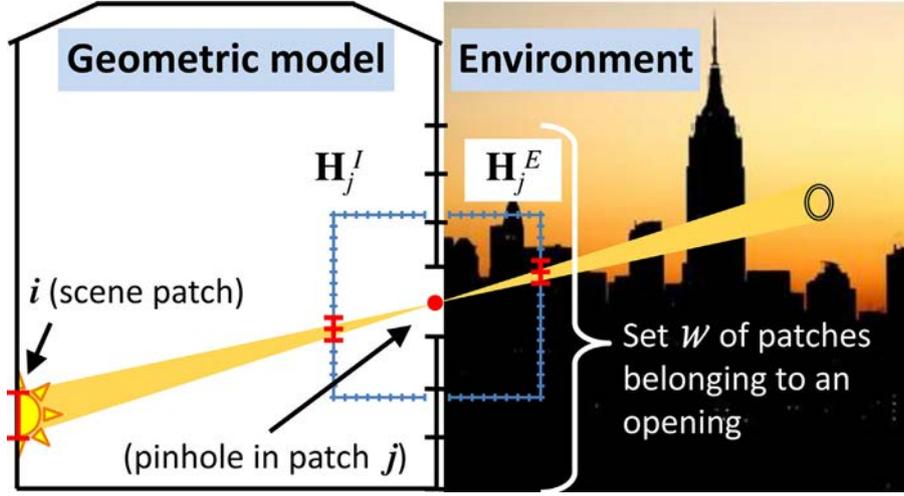


Fig. 1. Pinhole based radiosity and its main elements.

EnergyPlus as the energy evaluation engine, with a rough approximation for daylighting. Our work focuses only on natural lighting analysis with full hourly-year datasets providing a way to obtain optimal window shapes.

### 3. Pinhole based radiosity

The substitution of an opening by a set of pinholes was proposed in [6]. In this proposal, the pinholes are used to model the interaction between the exterior daylighting and the interior surfaces of a scene. The main feature of this so-called “pinhole-based radiosity” (PBR) is that it is a radiosity method that allows to model anisotropic emissions and isotropic reflections (Fig. 1). In this section, we summarize this technique, which is fundamental to introduce the new proposed method.

In PBR, the well-known radiosity equation [19]  $(\mathbf{I} - \mathbf{RF})\mathbf{B} = \mathbf{E}$  is substituted by

$$(\mathbf{I} - \mathbf{RF})\mathbf{B} = \mathbf{E} + \mathbf{GW} \quad (1)$$

where  $\mathbf{I}$  is the identity matrix,  $\mathbf{R}$  is a diagonal matrix containing the reflection index of each patch,  $\mathbf{F}$  is the form-factor matrix,  $\mathbf{B}$  and  $\mathbf{E}$  are vectors containing the radiosity values, and the emission values of all patches, respectively. The new terms are the matrix  $\mathbf{G}$  of dimension  $n \times \bar{w}$  (where  $n$  is the number of patches in the scene and  $\bar{w}$  is the number of patches in the opening), and  $\mathbf{W}$ , which is a binary vector of dimension  $\bar{w}$ , where  $W(j) = 1$  when the patch  $j$  is

open and  $W(j) = 0$  when it is closed. Each element  $\mathbf{G}(i, j)$  contains the radiosity value at the scene patch  $i$ , corresponding to the first bounce of the light coming through patch  $j$ .

When the matrix  $(\mathbf{I} - \mathbf{RF})$  is inverted, it is possible to find  $\mathbf{B}$  (Eq. (2)), for the configuration of pinholes defined in  $\mathbf{W}$ .

$$\mathbf{B} = \mathbf{ME} + \mathbf{NW} \quad (2)$$

where  $\mathbf{M} = (\mathbf{I} - \mathbf{RF})^{-1}$  and  $\mathbf{N} = \mathbf{MG}$ .

The calculation of  $\mathbf{G}(i, j)$  is expressed in Eq. (3). This calculation is based on the approximation of the light passing through a patch emulating a pinhole. Then, we use two opposite hemi-cubes centered in each pinhole  $j$ , to model the light passing through the corresponding patch  $j$ . These hemi-cubes,  $\mathbf{H}_j^I$  and  $\mathbf{H}_j^E$  (Figs. 1 and 2), respectively contain the projection of the interior and exterior views of the scene.

$$\mathbf{G}(i, j) = \frac{A(j)}{A(i)} R(i) \sum_{(u,v) | \mathbf{H}_j^E(u,v)=i} \pi \Delta \mathbf{F}(u, v) \mathbf{H}_j^E(u, v) \quad (3)$$

The hemi-cubes are represented as matrices. Each cell  $\mathbf{H}_j^E(u, v)$  contains the light energy expressed in radiance ( $\text{W}/\text{sr}/\text{m}^2$ ),  $\mathbf{H}_j^E(u, v)$  contains an index that represents the scene patch projected on it,  $\Delta \mathbf{F}(u, v)$  is the delta form-factor of cell  $(u, v)$  [20], and  $\pi$  is needed to transform angular quantities (radiance of  $\mathbf{H}_j^E$ ) into area ones (irradiance ( $\text{W}/\text{m}^2$ ) on the pinhole). Other elements are  $R$  and  $A$ , which

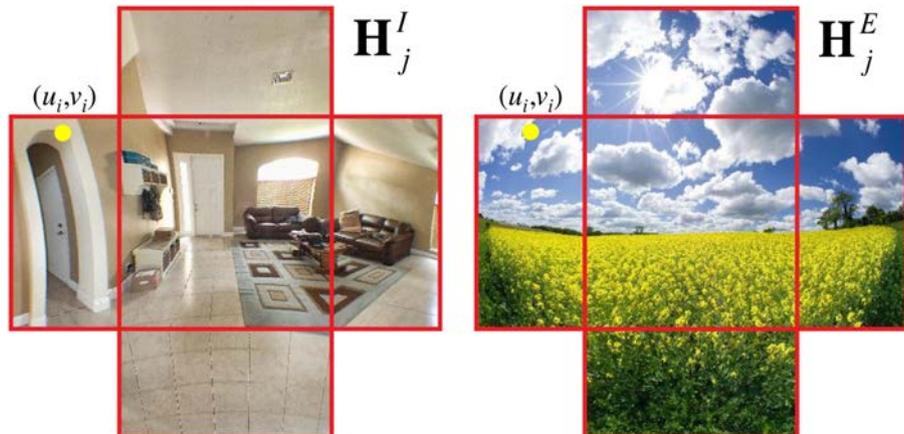


Fig. 2. Internal ( $\mathbf{H}_j^I$ ) and external ( $\mathbf{H}_j^E$ ) hemi-cubes at pinhole  $j$ . Each pixel  $(u_i, v_i)$  of  $\mathbf{H}_j^E$  illuminates the surface represented in pixel  $(u_i, v_i)$  of  $\mathbf{H}_j^I$  through this pinhole.

contains the reflection coefficient of the scene patches and their areas, respectively.

In Eq. (3) the sum calculates the total irradiance that arrives at  $j$  and goes to patch  $i$ . To do this, first the pixels  $(u, v)$  related to patch  $i$  are sought in  $\mathbf{H}_j^i$ . After that, each radiance value  $\mathbf{H}_j^i(u, v)$  is weighted by  $\pi \Delta \mathbf{F}(u, v)$ . This is done to transform the radiance of the sky into the irradiance on patch  $j$ , and because the influence of each pixel on the hemi-cube over the pinhole is dependent on pixel location and orientation [20]. Finally, the result is multiplied by  $(A(j)/A(i))R(i)$ , to transform the irradiance going through patch  $j$  into the radiosity reflected by patch  $i$ .

A hemi-cube structure, that is, a cube sliced in half through a plane parallel to one of its faces, is used just for computational reasons [20], but hemispheres or other surfaces can be used as well.

#### 4. Pinhole based illuminance from the sky

In this work we seek for the calculation of the illuminance at the surfaces instead of their radiosities, therefore, the PBR is modified to represent a pinhole-based illuminance (PBI). In this new approach, Eqs. (1) and (2) are substituted by the following equations:

$$(\mathbf{I} - \mathbf{FR})\mathbf{I} = \mathbf{FE} + \mathbf{G}_I \mathbf{W} \quad (4)$$

$$\mathbf{I}\mathbf{I} = \mathbf{M}_I \mathbf{FE} + \mathbf{N}_I \mathbf{W} \quad (5)$$

where  $\mathbf{M}_I = (\mathbf{I} - \mathbf{FR})^{-1}$  and  $\mathbf{N}_I = \mathbf{M}_I \mathbf{G}_I$ .

Now,  $\mathbf{I}$  stands for the global illuminance ( $\text{lx} = \text{lm}/\text{m}^2$ ),  $\mathbf{G}_I \mathbf{W}$  is the direct natural illuminance from the sky (or  $I^{(D,K)}$ ), and  $\mathbf{E}$  is the luminous emittance ( $\text{lx}$ ) from other light sources. The calculation of  $\mathbf{G}_I(i, j)$  is:

$$\mathbf{G}_I(i, j) = \frac{A(j)}{A(i)} \sum_{(u,v) \in \mathbf{H}_j^i(u,v)=i} \pi \Delta \mathbf{F}(u, v) s(k) \quad (6)$$

where  $k = \mathbf{S}_j(u, v)$ .

In this equation,  $\mathbf{S}_j$  is the hemi-cube view of the tiles of the sky as it is seen from pinhole  $j$  (each element of  $\mathbf{S}_j$  has the index of a sky tile), and  $s$  is a vector with the luminance ( $(\text{lm}/\text{sr})/\text{m}^2$ ) of each sky tile. The product between  $s$  and  $\pi \Delta \mathbf{F}(u, v)$  transforms the luminance into illuminance, and the sum gives all the illuminances received by patch  $j$  and going to patch  $i$ . Finally, the result is multiplied by  $A(j)/A(i)$  to find the illuminance on  $i$ . Fig. 3(a) shows a lateral view of a room, with three different opening cases, and Fig. 3(b), (c), and (d) shows the corresponding  $\mathbf{S}_j$  views of the sky and the ground.

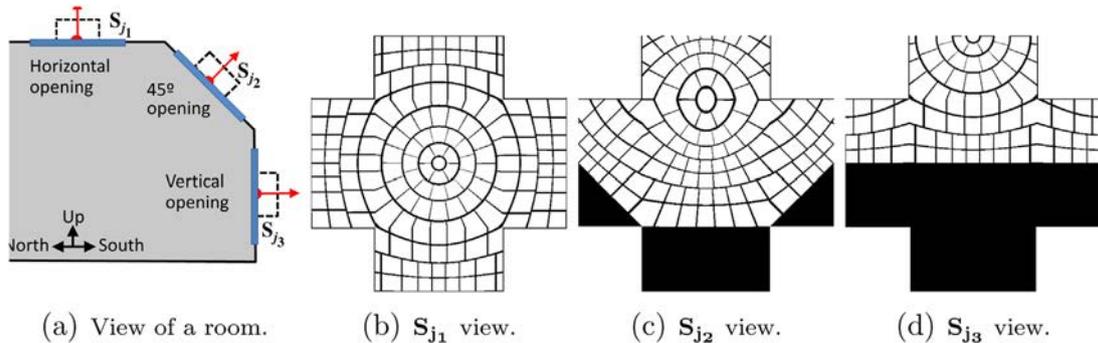


Fig. 3. A room and three hemi-cube views of the sky tiles, one for each opening.

#### 4.1. PBI for many skies

In order to simplify the calculation of  $\mathbf{G}_I$  when the sky  $s$  changes, Eq. (6) can be transformed by grouping the terms of the sum by  $s(k)$ :

$$\begin{aligned} \mathbf{G}_I(i, j) &= \sum_{k=1}^{\bar{s}} \left( \frac{A(j)}{A(i)} \sum_{(u,v) \in \mathbf{H}_j^i(u,v)=i, \mathbf{S}_j(u,v)=k} \pi \Delta \mathbf{F}(u, v) \right) s(k) \\ &= \sum_{k=1}^{\bar{s}} \mathbf{L}(i, j, k) s(k) \end{aligned} \quad (7)$$

In this equation,  $\mathbf{L}$  is a 3-way array or tensor with dimension  $n \times \bar{w} \times \bar{s}$ , where  $\mathbf{L}(i, j, k)$  is the ratio between the direct illuminance ( $\text{lm}/\text{m}^2$ ) received by the scene patch  $i$  and the amount of light ( $\text{lm}/\text{sr}$ ) emitted by the sky tile  $k$  (stored in  $s(k)$ ) passing through patch  $j$ .

A tensor is a multidimensional or  $N$ -way array [21,22]. Thus, a 1-way tensor is a vector, a 2-way tensor is a matrix, and a 3-way tensor is a cube of data. A sparse tensor is a tensor where only a small fraction of the elements are nonzero. In such case, it is more efficient to store just the nonzeros and their indexes.

We can note that  $\mathbf{L}$ , in fact, is a sparse tensor because each patch  $i$  "sees" only a few set of sky tiles through pinhole  $j$ , thus there are few  $\mathbf{L}(i, j, k)$  cells containing nonzero elements. Can be estimated that the memory needed to store  $\mathbf{L}$  has  $O(n\bar{w})$ .

The relation between  $\mathbf{G}_I$ ,  $\mathbf{L}$  and  $s$  of Eq. (7) can be expressed as:

$$\mathbf{G}_I = \mathbf{L} \times_3 s \quad (8)$$

where  $\times_3$  stands for the product between a tensor and a vector in the dimension 3 of the tensor [21].

Following the definition of  $\mathbf{N}_I$  in Eq. (5), it can be deduced that:

$$\mathbf{N}_I = \mathbf{M}_I \mathbf{G}_I = \mathbf{M}_I (\mathbf{L} \times_3 s) = (\mathbf{M}_I \times_1 \mathbf{L}) \times_3 s = \mathbf{Q} \times_3 s \quad (9)$$

where  $\mathbf{Q} = \mathbf{M}_I \times_1 \mathbf{L}$  is a tensor of dimension  $n \times \bar{w} \times \bar{s}$ , and  $\times_1$  specifies a product made along the first dimension of the tensor. Therefore, each cell  $\mathbf{Q}(i, j, k)$  is a number that multiplied by  $s(k)$  gives the global illuminance (direct and indirect illuminance) received by patch  $i$ , of the light emitted by the sky tile  $k$ , and passing through the patch  $j$  of the opening.

The matrix  $\mathbf{Q} \times_2 \mathbf{W}$  allows to transform the luminance of the sky tiles into the illuminance of the scene patches. This representation is somehow similar to the matrix  $\mathbf{VTD}$  introduced in [10]. An important difference is that we can modify the geometry of the window just by setting  $\mathbf{W}$ . To build  $\mathbf{Q}$  from  $\mathbf{VTD}$ , the matrices  $\mathbf{V}$  and  $\mathbf{D}$  should be calculated for each patch of the opening, a computationally expensive task with RADIANCE package [13].

The indirect illuminance is the main reason to consider that  $\mathbf{Q}$  and  $\mathbf{N}_I$  have relatively few zero elements. The light that comes from any sky tile  $k$  and passes through any pinhole  $j$ , can potentially bring indirect illuminance to any patch  $i$  of the scene, with the consequence that  $\mathbf{Q}(i, j, k) \neq 0$ . The amount of memory needed to store  $\mathbf{Q}$  is  $O(n\bar{p}\bar{s})$ , meaning that even for small scenes it is required a large amount of memory. For instance, a scene composed of 10,000 patches, using a sky of 145 tiles and setting the number of pinholes to 200, generates a tensor  $\mathbf{Q}$  of more than 2 Gbytes, when each cell has a double precision number. But currently [2], the illuminance  $I$  is not calculated in all the surfaces but only over a small set of points  $P = p_1, \dots, p_{\bar{p}}$  called sensors. These sensors, that are just light receptors, can be placed on the barycenter of some scene patches or can be added as new transparent patches.

It can be shown that the calculation of  $\mathbf{Q}(P, :, :)$  and  $\mathbf{N}_I(P, :)$  could be directly computed, if the matrix  $\mathbf{M}_I$  is obtained and the rows  $P$  of the sensors are selected:

$$\mathbf{N}_I(P, :) = (\mathbf{M}_I(P, :) \times_1 \mathbf{L}) \times_3 s = \mathbf{Q}(P, :, :) \times_3 s \quad (10)$$

Now, the amounts of memory needed to store  $\mathbf{Q}(P, :, :)$  and  $\mathbf{N}_I(P, :)$  have  $O(\bar{P}\bar{w}\bar{s})$  and  $O(\bar{P}\bar{w})$  respectively. Then, the amount of memory needed is proportional to the amount of sensors and is independent of the number of patches in the scene. For instance,  $\mathbf{Q}(P, :, :)$  needs 2 Mbytes for a scene containing 10 sensors, 145 tiles and 200 pinholes, and  $\mathbf{N}_I(P, :)$  needs only 16 Kbytes for the same scene. Furthermore, in the case of  $\mathbf{N}_I(P, :)$ , the amount of memory needed is also independent of the number of tiles in the sky. These results have practical implications in the calculation of the illuminance for many skies and large scenes.

The main cost in the computational aspects of having a large scene is due to the calculation of the inverse matrix  $\mathbf{M}_I = (\mathbf{I} - \mathbf{FR})^{-1}$ . For large scenes composed of  $n$  patches, its calculation has complexity  $O(n^3)$  and consumes  $O(n^2)$  memory. In spite of that, these computational costs can be reduced using efficient factorization techniques, such as low-rank radiosity [23].

#### 4.2. Daylighting illuminances

In this paper we concentrate our experimental results on the calculation of the daylighting, that is, the illuminance received by the sky and the Sun without the consideration of other light sources (i.e.,  $E=0$  in Eq. (4)). The daylighting can be categorized into direct illuminance  $I^{(D)}$  and indirect illuminance  $I^{(I)}$ , according to the number of bounces. Direct illuminance covers the light arriving directly from a light source, and indirect illuminance refers to the light that arrives after at least one bounce. The addition of both quantities is the global illuminance  $I^{(G)}$ , which is the illuminance accumulated after infinite bounces. These daylight values can be calculated through the following equations:

$$I^{(D)} = \mathbf{G}_I W; \quad I^{(G)} = \mathbf{N}_I W; \quad I^{(I)} = I^{(G)} - I^{(D)} \quad (11)$$

Another categorization can be associated to the kind of source they come from. The daylighting can be calculated considering that the light arrives from the Sun, from the sky, or from both sources (total illuminance). The combination of both categorizations produces another categorization composed of 9 kinds of illuminances that are expressed as:

$$I^{(D,*)} = \mathbf{G}_I^{(*)} W = (\mathbf{L} \times_3 s^{(*)}) W \quad (12)$$

$$I^{(G,*)} = \mathbf{N}_I^{(*)} W = (\mathbf{Q} \times_3 s^{(*)}) W \quad (13)$$

$$I^{(I,*)} = I^{(G,*)} - I^{(D,*)} \quad (14)$$

where  $*$  denotes for  $S, K$  or  $T$  depending if the light source is the Sun, the sky or both ( $T$  for total) sources, respectively. In these equations,  $s^{(S)}$  contains the value of the Sun light in each sky tile,  $s^{(K)}$  contains

the value of the scattered light through the atmosphere in each sky tile, and  $s^{(T)} = s^{(S)} + s^{(K)}$ . This new categorization is used in the results analysis of Section 6.

#### 4.3. Daylight coefficients equivalence

In this section, we show that our methodology could be used to implement tests and to solve inverse lighting problems based on daylight coefficients. Both  $\mathbf{Q}$  and  $W$  can be related to the classical DCs formulation. In this way, this relation could be used to accelerate the calculation of the DCs for any opening configuration.

Following Tregenza and Watters [5], a daylight coefficient  $DC_\alpha(p_i)$  related to a sky tile  $S_\alpha$  is defined as:

$$DC_\alpha(p_i) = \frac{I_\alpha(p_i)}{s(\alpha)\Delta S_\alpha} \quad (15)$$

where  $I_\alpha(p_i)$  is the illuminance  $I$  in luxes at a sensor  $p_i$  caused by the sky tile  $\alpha$ ,  $s(\alpha)$  is the luminous intensity of  $\alpha$  (lm/sr), and  $\Delta S_\alpha$  is the angular size of the sky tile. The total sensor illuminance  $I(p_i)$  is obtained by the linear superposition of all the illuminances  $I_\alpha(p_i)$ .

$$I(p_i) = \sum_{\alpha=1}^{\bar{s}} I_\alpha(p_i) = \sum_{\alpha=1}^{\bar{s}} (DC_\alpha(p_i)\Delta S_\alpha)s(\alpha) \quad (16)$$

Moreover, using Eqs. (5) and (10), and assuming that there are no additional light sources ( $E=0$ ), the calculation of  $I(p_i)$  is reduced to:

$$I(p_i) = \mathbf{N}_I(p_i, :) W = (\mathbf{Q}(p_i, :, :) \times_3 s) W \quad (17)$$

Now, grouping Eq. (17) by  $s(\alpha)$ , it turns into:

$$\begin{aligned} I(p_i) &= (\mathbf{Q}(p_i, :, :) \times_2 W) s \\ &= \sum_{\alpha=1}^{\bar{s}} (\mathbf{Q}(p_i, :, \alpha) \times_2 W) s(\alpha) \end{aligned} \quad (18)$$

Therefore, from the coefficients of Eqs. (16) and (18), the value of  $DC_\alpha(p_i) \forall \alpha$  can be deduced as:

$$DC_{\alpha,W}(p_i) = \frac{\mathbf{Q}(p_i, :, \alpha) \times_2 W}{\Delta S_\alpha} \quad (19)$$

where  $\alpha$  and  $W$  are parameters that modify the DC's. Then, using Eq. (19) it is possible to calculate a matrix  $\mathbf{DC}_W$  of dimension  $\bar{P} \times \bar{s}$ , where  $\mathbf{DC}_W(p_i, \alpha) = DC_{\alpha,W}(p_i)$ . The calculation of  $\mathbf{DC}_W$  is now reduced to a product between a tensor and a vector that has order of complexity  $O(\bar{P}\bar{s}\bar{w})$ .

This result relates our work with DCs, and shows a way to reduce the computational resources needed to calculate the DCs, in comparison to the currently used routines of RADIANCE package [13].

## 5. Implementation

In this section we present practical considerations for implementing an optimization system.

### 5.1. Illuminances in all sensors and for all skies at the same time

Assuming that the set of skies  $S = \{s_1, \dots, s_{\bar{s}}\}$  is related to a set of matrices  $\{\mathbf{N}_{I,s_1}, \dots, \mathbf{N}_{I,s_{\bar{s}}}\}$  and that there are no other light sources

( $E=0$ ), then it is possible to define the global illuminance ( $I_S^{(G)}$ ) on sensors  $P$ , and to build a unique matrix  $\mathbf{N}_{I,S}(P, :)$ :

$$I_S^{(G)}(P) = \mathbf{N}_{I,S}(P, :)W$$

$$\text{where } \mathbf{N}_{I,S}(P, :) = \begin{pmatrix} \mathbf{N}_{I,S_1}(P, :) \\ \vdots \\ \mathbf{N}_{I,S_S}(P, :) \end{pmatrix} \quad (20)$$

In this equation,  $I_S^{(G)}(P)$  contains the global illuminance associated with sensor  $p_i \in P$  and each sky of  $s_j \in S$ . The elements in  $I_S^{(G)}(P)$  can be characterized as:

$$I_S^{(G)}(P) = \left( \underbrace{I_{s_1}^{(G)}(p_1), \dots, I_{s_1}^{(G)}(p_{\bar{P}})}_{I_{s_1}^{(G)T}(P)}, \dots, \underbrace{I_{s_S}^{(G)}(p_1), \dots, I_{s_S}^{(G)}(p_{\bar{P}})}_{I_{s_S}^{(G)T}(P)} \right)^T$$

where  $I_{s_i}^{(G)}(P) = \mathbf{N}_{I,S_i}(P, :)W$  is the global illuminance on sensors  $P$  when the sky is  $s_i$ .

Similarly,  $I_S^{(D)}(P)$  and  $I_S^{(I)}(P)$  can be calculated as:

$$I_S^{(D)}(P) = \mathbf{G}_{I,S}(P, :)W,$$

$$\text{where } \mathbf{G}_{I,S}(P, :) = \begin{pmatrix} \mathbf{G}_{I,S_1}(P, :) \\ \vdots \\ \mathbf{G}_{I,S_S}(P, :) \end{pmatrix} \quad (21)$$

$$I_S^{(I)}(P) = I_S^{(G)}(P) - I_S^{(D)}(P) \quad (22)$$

$$= (\mathbf{N}_{I,S}(P, :) - \mathbf{G}_{I,S}(P, :))W$$

Furthermore, the distinction between sky, Sun, and total illuminance in the direct, indirect, and global illuminance can be calculated if the sky and Sun luminous intensities for each sky tile are available. Then, we place them in the  $s$  term of Eq. (8) and following (Eqs. (10), (20), (21) and (22)). The calculation of Sun illuminance may require the use of a finer mesh [2], affecting the size of  $s$ ,  $\mathbf{L}$  and  $\mathbf{Q}$ , but without effect on the size of  $\mathbf{G}_{I,S}(P, :)$  and  $\mathbf{N}_{I,S}(P, :)$ .

## 5.2. UDI formulation

Eqs. (20)–(22) are a previous step to the calculation of UDI. We base the optimization tests on UDI, but our methodology could also be used on any other daylighting metric. Following [2], the proportion of skies  $S$  that fulfills the three indicators of UDI (exceeded, fell-short, and achieved) are calculated respectively as:

$$P_{I>2k} = \frac{\sum_{i=1}^{\bar{S}} (\max(I_{s_i}(P)) > 2000)}{\bar{S}} \quad (23)$$

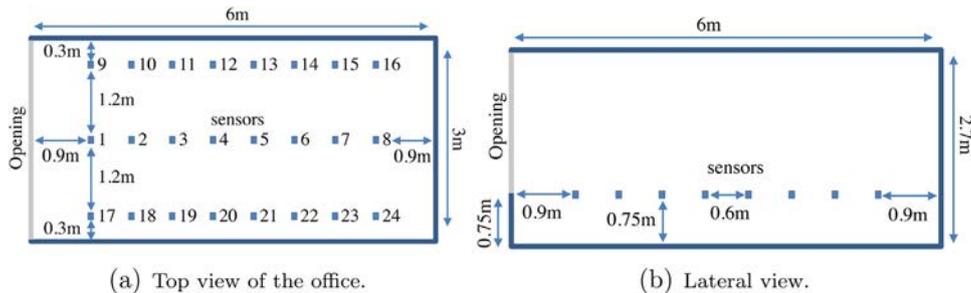


Fig. 4. Nabil & Mardaljevic office [2] with 24 sensors, distributed in three rows.

$$P_{I<100} = \frac{\sum_{i=1}^{\bar{S}} (\min(I_{s_i}(P)) < 100)}{\bar{S}} \quad (24)$$

$$P_{100<I<2k} = \frac{\sum_{i=1}^{\bar{S}} ((\min(I_{s_i}(P)) \geq 100) \wedge (\max(I_{s_i}(P)) \leq 2000))}{\bar{S}} \quad (25)$$

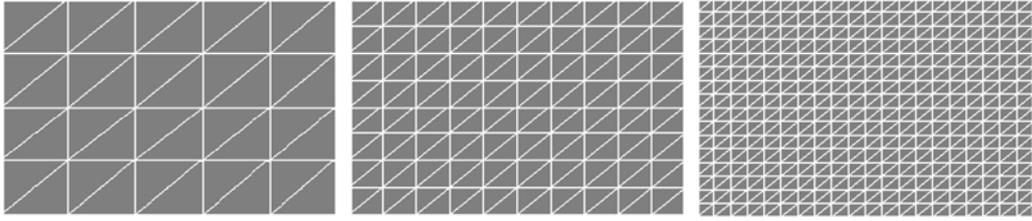
In the equations above, the addends of each sum are Boolean expression, whereas the operators  $>$ ,  $<$ , and  $\wedge$  return 1 when they are satisfied and 0 otherwise. The sums accumulate the amount of hours where the Boolean expressions are satisfied, i.e., the amount of hours where at least one sensor receives more than 2000 lx (Eq. (23)), the amount of hours where at least one sensor receives less than 100 lx (Eq. (24)), and the amount of hours where the illuminances in all sensors are between 100 lx and 2000 lx (Eq. (25)).

Due to space constraints, in Section 6 we only present results of Eq. (25). In this equation, we removed the division by  $S$  to show the total number of hours along the year where the UDI indicator is satisfied.

## 5.3. The optimization process: VNS and rectangles

We use variable neighborhood search (VNS) [24] metaheuristic to solve the inverse daylighting problem. This methodology is based on the idea of successive explorations of a set of neighborhoods ( $N_1(x)$ ,  $N_2(x)$ , ...,  $N_k(x)$ ). 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. The set of neighborhoods is usually nested (the neighborhood  $N_{i+1}(x)$  is included in  $N_i(x)$ ). To address the inclusion of constraints into the optimization problem, we use the penalty approach [25] which modifies the function to optimize through the consideration of the constraints.

As we mentioned in Section 2.2, our approach is focused only on daylighting. In Eqs. (20)–(22), the configuration of an opening is defined by the selection of 0's and 1's in  $W$ . Then, the search of an optimal opening configuration is a combinatorial problem, where it is needed to add several constraints to define the shape and other desired properties for the openings. To avoid such kind of problems, our proposal is based on the use of a set of rectangles to define the shape of the opening (Fig. 8). Each rectangle is defined by two opposite corners, and all the patches that partially or totally fall inside a rectangle are considered as part of the opening (see all gray triangles in Fig. 8). The rectangles are built over walls and roofs previously set to have openings installed. Each corner of a rectangle opening can be represented as a 2D vector. We associate each patch of those surfaces to a cell in  $W$ . Then, each cell of  $W$  is set as "open" if its corresponding patch is partially or totally inside of the rectangle, and as "close" otherwise. Working with corner bounds has the advantage that reduces the potential number of variables from hundreds (patches) to tens or less (corners). Also, it assures the existence of large sets of connected patches.



(a) Window with 40 patches. (b) Window with 160 patches. (c) Window with 640 patches.

Fig. 5. Three possible configurations of the window.

## 6. Results

The results of the presented set of experiments exemplify the practical use of the method. The simulations were conducted on a desktop computer, with Intel quad-core i7 processor and 8 Gbytes RAM. The code was implemented in MATLAB [26], and the tensors in the MATLAB Tensor Toolbox Version 2.6 [27].

The geometric model used is the same as in [2] (Fig. 4). This office is a box of  $6\text{ m} \times 3\text{ m} \times 2.7\text{ m}$  size, with an opening in one of its walls. The opening covers the entire wall except a lower section of  $0.75\text{ m}$  high. The wall of the office has 620 patches, and the opening is modeled with 3 different configurations (Section 6.1). The sensors are installed at  $0.75\text{ m}$  from the floor. The window glass has a transmittance of  $0.76$ . The reflectivities of the walls, ceiling, and floor are  $0.7$ ,  $0.8$ , and  $0.2$ , respectively.

Hourly sky and Sun conditions were derived from the direct normal and diffuse horizontal irradiation data, extracted from Test Reference Year data [28].

### 6.1. Pinholes in the window

Here we analyze the performance of the pinhole method in daylighting computation, for modeling the opening of the office. In order to evaluate the amount of pinholes needed for realistic

modeling, we tested three configurations of the opening with 40, 160 and 640 patches (Fig. 5).

Each pinhole is positioned at the barycenter of a triangular element; therefore, the 3 meshes are composed of 40, 160 and 640 pinholes, respectively. For each mesh, we calculate the illuminance distribution (ID) in the sensors of the central line (sensors 1–8 of Fig. 4). The diagrams of Fig. 6 show, along the sensors line, the annual distribution of the illuminances measured in percentage of the office hours ( $9:00\text{--}18:00$ ). They are also clearly showing the UDI availability along the central line (percentage of hours between the two thick separation lines). In this situation, UDI is increasing significantly from the front to the back of the room. These results are computed in Gatwick-UK, and the window is oriented to the South.

The three diagrams of Fig. 6 are quite similar, especially 6(b) and (c), where the differences are lower than 5%. They are also similar to the ID diagram of [2], although the weather data and the number of sensors are not exactly the same.

The required amounts of memory to store  $\mathbf{Q}(P, :, :)$  are 224 KB, 897 KB and 6.0 MB for 8 sensors and when the number of pinholes is 40, 160, and 640, respectively. The largest matrix  $\mathbf{N}_{I,S}(P, :)$  occurs when the opening has 640 pinholes. This matrix has dimension  $(365 * 10 * 8) \times 640$  and needs a memory of 150 MB. Its calculation takes about 80 s applying Eq. (5), assuming that  $\mathbf{F}$ ,  $\mathbf{R}$ ,  $\mathbf{s}$ ,  $\mathbf{S}_j$  and  $\mathbf{H}_j^l$  are

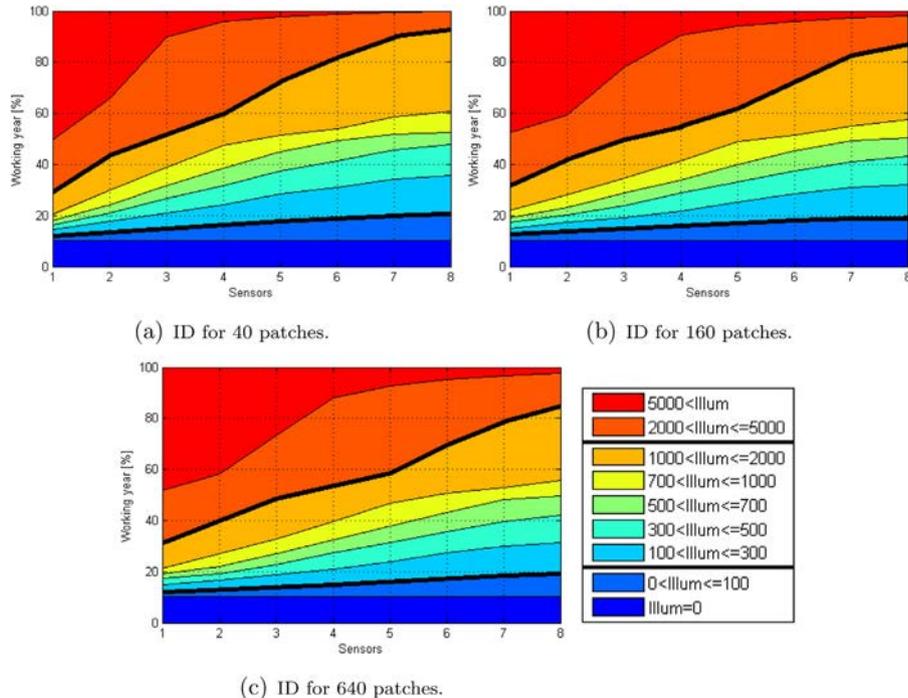


Fig. 6. ID for the central sensors, with three meshes of the window during working hours ( $9:00\text{--}18:00$ ) in a year, in Gatwick, UK.

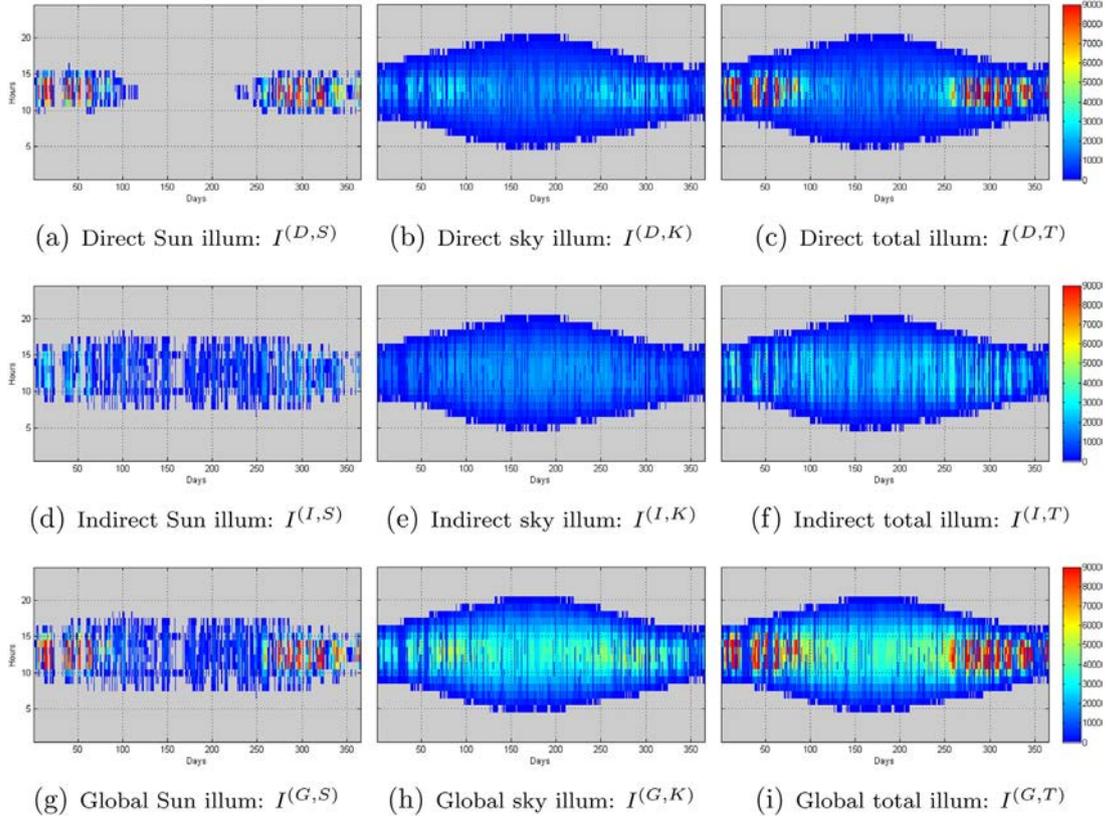


Fig. 7. Direct, indirect and global illuminances from the sky and the Sun at sensor 4 in Gatwick-UK.

available. After  $\mathbf{N}_{I,S}(P, :)$  is built, the calculation of an ID diagram for any opening takes only 0.1 s.

For the other experiments, we use a mesh always composed of 640 elements that allows a better definition of the optimal openings shape. However, if, due to constraints in memory or time, a rough approximation of the shape is preferred, a smaller amount of pinholes may bring comparable results.

## 6.2. Direct and indirect illuminances

This experiment consists in the calculation of the direct and indirect sunlight and skylight illuminances arriving at sensor 4 (see Fig. 4). We apply Eqs. (20)–(22), assuming that the window is fully open. As we do the calculation only for one sensor and 24 h per day,  $\mathbf{N}_{I,S}(P, :)$  has dimension  $(365 * 24 * 1) \times 640$  and requires 44 MB of storage. The results are shown in Fig. 7, where each dot in the images represents an hour in the year, the columns represent

the days (365 columns) and each row represents 1 h. The results are expressed in luxes. Fig. 7(a)–(c) shows the direct illuminance from the Sun, the sky, and the sum of both (total illuminance). Fig. 7(d)–(f) shows the indirect illuminance (the internally reflected component) from the Sun, sky, and total. Finally, Fig. 7(g)–(i) shows the sum of direct and indirect illuminance (global illuminance) of the Sun, sky and total.

## 6.3. Optimization: problem definition and settings

The optimization process is based on the pinhole method, and the shortcuts described in Section 5. The problem to solve is the maximization of the UDI hours in the office. We have to define some constants and to set some parameters that maximize the impact of the solution achieved. The constants are: the number of pinholes (640), the interval of hours (all working hours – 9:00 to 18:00 – in a year), and the number of iterations (50,000). For this setting,

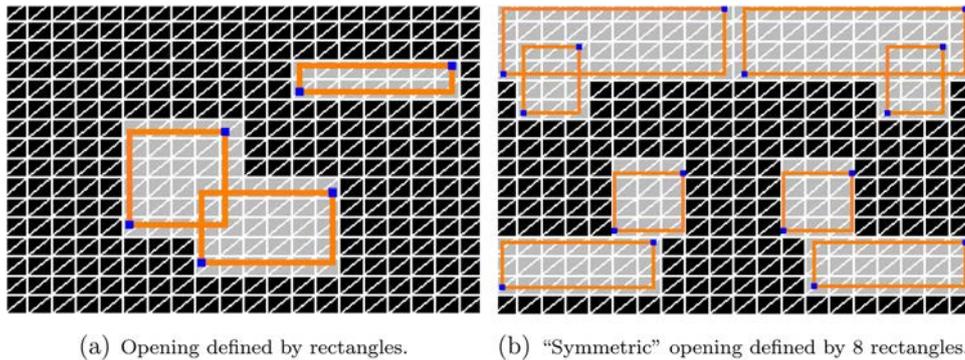


Fig. 8. Valid configurations of the opening. The union of interior patches of the rectangle is considered as the opening shape.

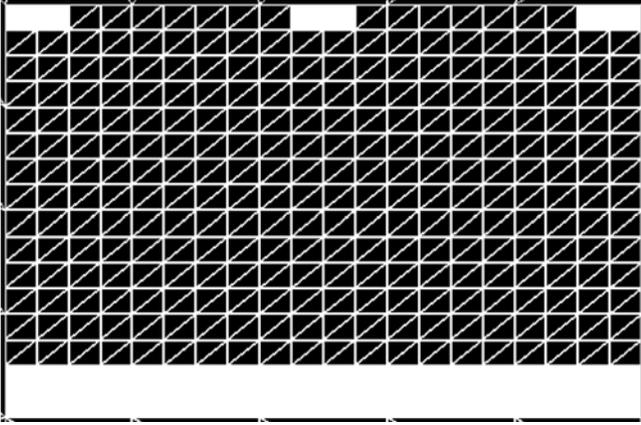


Fig. 9. Optimal opening shape for {24,8,S,UK} settings, the UDI obtained is 1442 h, and the glazing ratio obtained is 4.7%.

each optimization process takes about 1.5 h (about 0.1 s per opening configuration).

The optimization parameters are the number of sensors and the number of rectangles. To reduce the number of variables, the rectangles are related to each other by an axial vertical symmetry constraint (Fig. 8(b)). This reduction has a direct impact on the computational costs involved in the optimization process.

We perform an initial study of the optimal shape with the number and distribution of sensors (1, 8, or 24) corresponding to the scheme of Fig. 4, and with 1, 2, 4, or 8 rectangles.

To label each optimization solution, we define a vector notation containing 4 parameters: {#sensors, #rectangles, orientation, location}. In addition to the number of sensors and rectangles, the other parameters are: the orientation (S, E, N, and W), and the geographical location, which has been selected in a range of latitudes (Quito-Ecuador [0.15 S, 78.48 W], Kharga-Egypt [25.45 N, 30.53 E], Barcelona-Spain [41.28 N, 2.07 E], Gatwick-UK [51.15 N, 0.18 W], and Kiruna-Sweden [67.82 N, 20.33 E], which are codified as EC, EG, ES, UK, and SE respectively).

Our analysis of the optimal results starts by defining one configuration of parameters as a reference result, and it continues by studying results achieved when 1 or 2 parameters are modified.

### 6.3.1. Reference solution

As a reference solution, we use the opening (three small rectangles on the top and a large one on the bottom of Fig. 9) that maximizes amount of working hours where the UDI is achieved, for 24 sensors and 8 rectangles, when the opening is oriented to the South in Gatwick-UK. This configuration is encoded as {24,8,S,UK}.

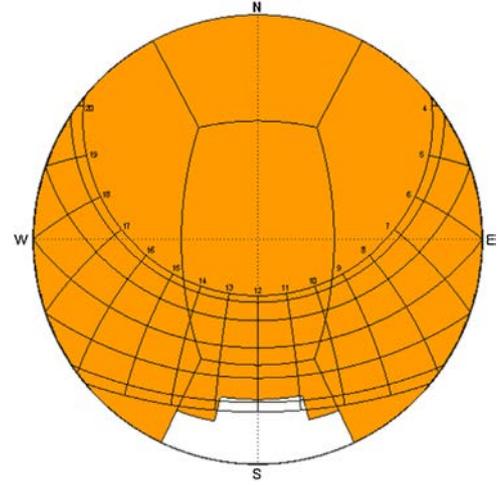


Fig. 11. Solar-path diagram for the solution obtained in 10(a) centered at sensor 4. If the middle-step is removed, UDI decreases of 87 h.

The matrix  $N_{I,S}(P, :)$  has dimension  $(365 * 10 * 24) \times 640$  and is stored in 449 MB of memory. Its calculation takes about 110 s. The optimal UDI value found for the reference configuration is 1442 h, corresponding to  $1442/3650 = 39.5\%$  of the working hours. The glazing ratio (GR) obtained, measured as the ratio between the surface of the opening and the surface of the floor in the office, is 4.7%. The solution found is composed of a small opening area in the bottom part of the window, and three even smaller areas at the top of the window. A first observation about this solution is that those small openings could cause discomfort in the working areas due to the direct sunlight incidence, but it seems that this problem is not detected by the sensors, or perhaps is compensated by the benefits of the other lights.

The gain obtained with this result can be compared to a completely open window for the same setting, which results in an UDI of 450 h. Therefore, the solution of Fig. 9 has a gain of about 220% over the original configuration.

### 6.3.2. Amount of sensors

The amount and distribution of sensors have a great influence in the final solution. We calculate the optimal solution for two other sensor configurations, different from the reference one with 24 sensors. The first configuration is composed only of the sensor number 4 (see Fig. 4), and the second one is composed of the 8 sensors that belong to the central row. The shapes vary notably, as well as the amount of UDI hours. These shapes seem to avoid the direct illuminance of the sensors by the Sun. Also, these results follow the intuitive idea that the addition of sensors reduces the amount of UDI hours satisfied. When only one sensor

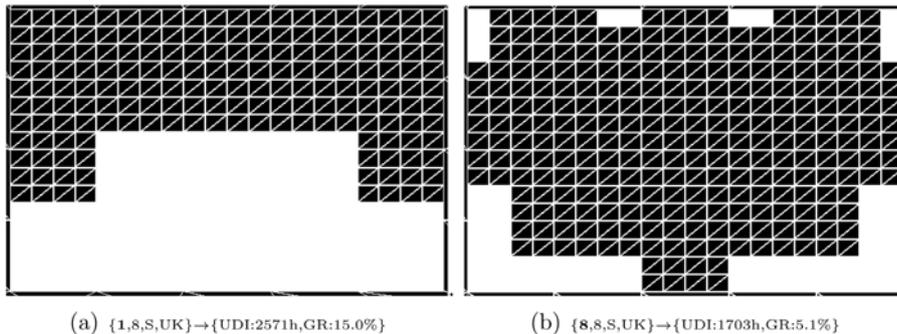


Fig. 10. Sensor 4 UDI=2571 h (a), and 8 sensors on the central line UDI=1703 (b).

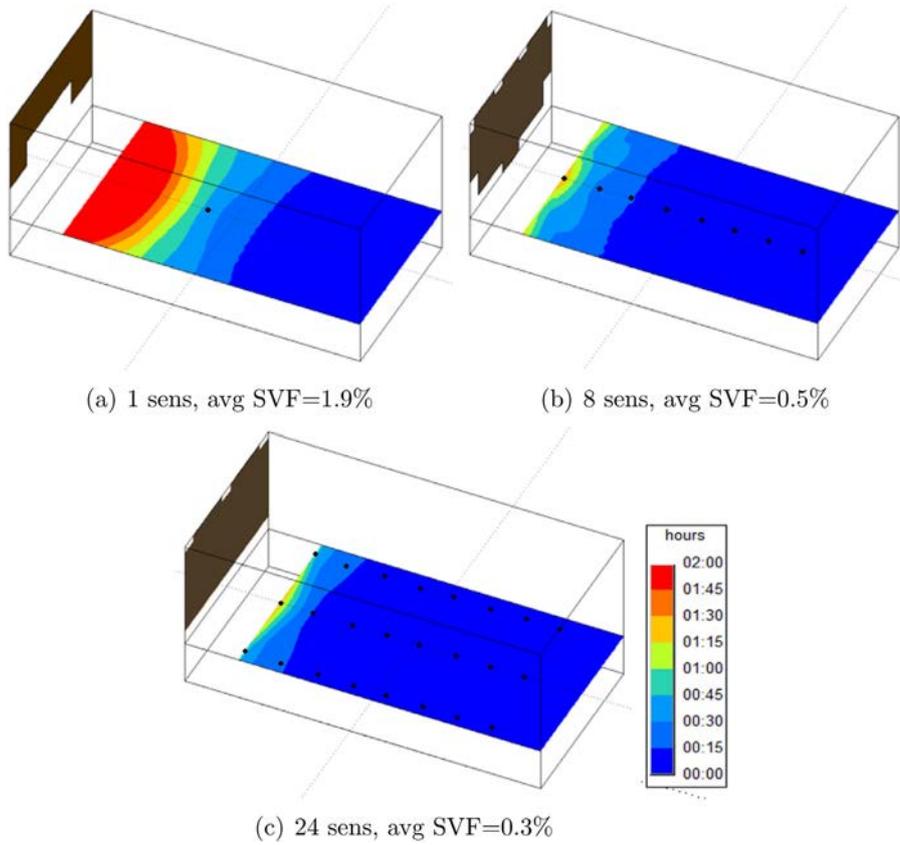


Fig. 12. Average amount of hours per day for the optimal solution computed with clear sky for the solutions with different amount of sensors.

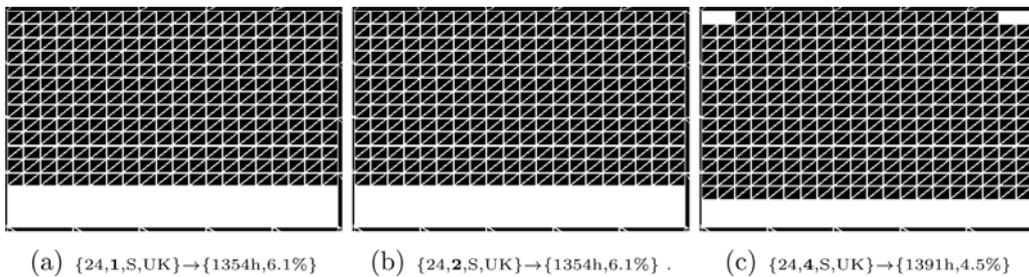


Fig. 13. Solutions for 1, 2 and 4 rectangles.

is used, UDI = 2571 h (GR = 15.0%), whereas when we add the other 7 sensors, the value obtained of UDI is 1703 h (GR = 5.1%), which is higher than the UDI value obtained for 24 sensors (UDI = 1442 h, and GR = 4.7%). The results show that the solution is sensitive to the amount (and position) of the sensors. We also observed that the glazing ratio diminishes when the number of sensors increases, and consequently, the average sky view factor (SVF) in the plane of the sensors is also reduced. Moreover, the computational costs of the irradiances  $I_{\zeta}^{(*)}(P)$  are increasing proportionally to the number of sensors. Therefore, there is a trade-off between the precision of the optimal UDI values found and the computational costs involved.

Fig. 11 shows the stereographic diagram for the office centered at sensor number 4 with the window shape of Fig. 10(a) (white geometry). Regarding the Sun path for that location, it can be observed that the solution allows Sun rays entering in the office for about 2 h at noon during two months of the year because of this “protrusion-step” window shape. To analyze the impact of this

shape, we also compute the UDI without the step (we eliminate the superior rectangle of the opening formed by  $4 \times 12 \times 2$  patches). For this window, we obtain an UDI of 2484 h, that is, 87 h less than the optimal value, confirming the importance of the irregular shape obtained. To improve the interpretation of the solutions using different number of sensors, we also compute the day-averaged amount of sunny hours in each situation using a simple clear-sky model. This evaluation is performed with Heliodon 2 package [29]. Fig. 12 shows the amount of sunny hours computed on a grid situated in the sensors plane using only one sensor. The average number of hours per day at that point is less than 45 min, and it is reduced for the increasing number of sensors. The most noticeable observation is that, for this particular example, only the three points closer to the window are contributing to the UDI value. Another conclusion is that for 1, 8 and 24 sensors, we observe respectively the decreasing of the mean SVF (1.9%, 0.5%, 0.3%), the number of UDI hours computed before as the objective function of the optimization procedure (2371 h, 1703 h, 1442 h) and the

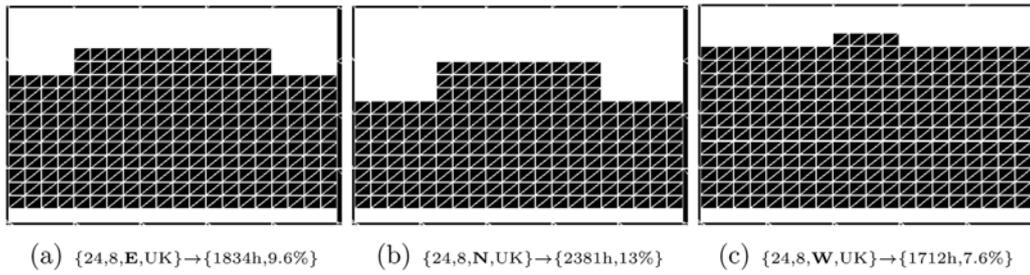


Fig. 14. Solutions for East, North and West orientations.

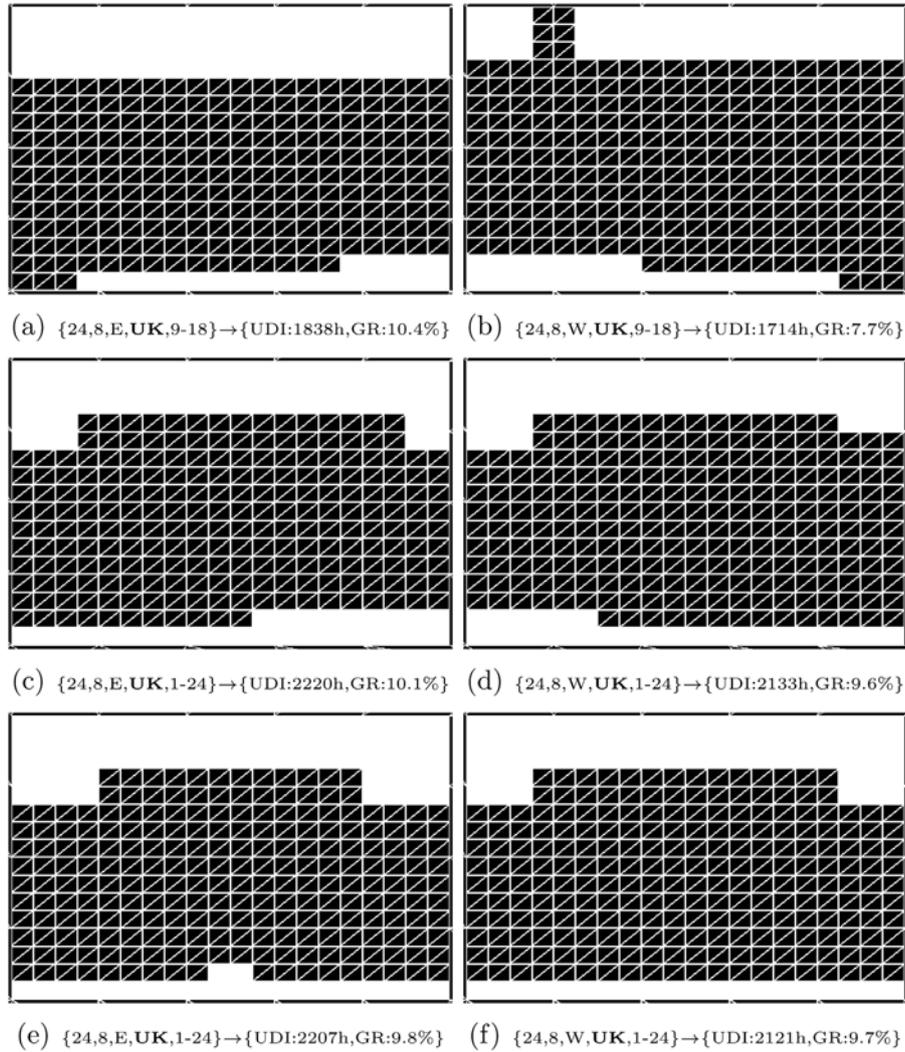


Fig. 15. East and West non-symmetric solutions for office hours (a)-(b), and for 24 h (c)-(d). East and West symmetric solutions for 24 h (e)-(f).

glazing ratios characterizing the openings (15%, 5.1%, 4.7%). As also observed in Fig. 12(b) and (c), these two last situations are very close to each other, but in the last one we see clearly the contribution of the two lateral sensors.

### 6.3.3. Amount of rectangles

Here, we analyze the influence of the number of rectangles in the optimal shape and in the UDI value.

In most of the current lighting problems, the designer is looking for a rectangular opening that maximizes some lighting parameter, in our case the UDI value. Moreover, when the designer is looking

for the optimal shape of the opening, a combination of rectangles may approximate that shape. We found that the use of several rectangles is a useful strategy to seek a first approximation of the optimal shape.

After obtaining the optimal shape for 1, 2, 4 (Fig. 13), and 8 (Fig. 9) rectangles, we see that, compared to the reference solution, all the solutions have quite a similar geometry (Fig. 13(a) and (b) are identical). However, the UDI values are sensible to that variation. For instance, there is a distance of only 4 patches between the reference solution and Fig. 13(c) ( $\Delta GR = 0.2\%$ ), but the UDI distance ( $\Delta UDI$ ) is of 51 h.

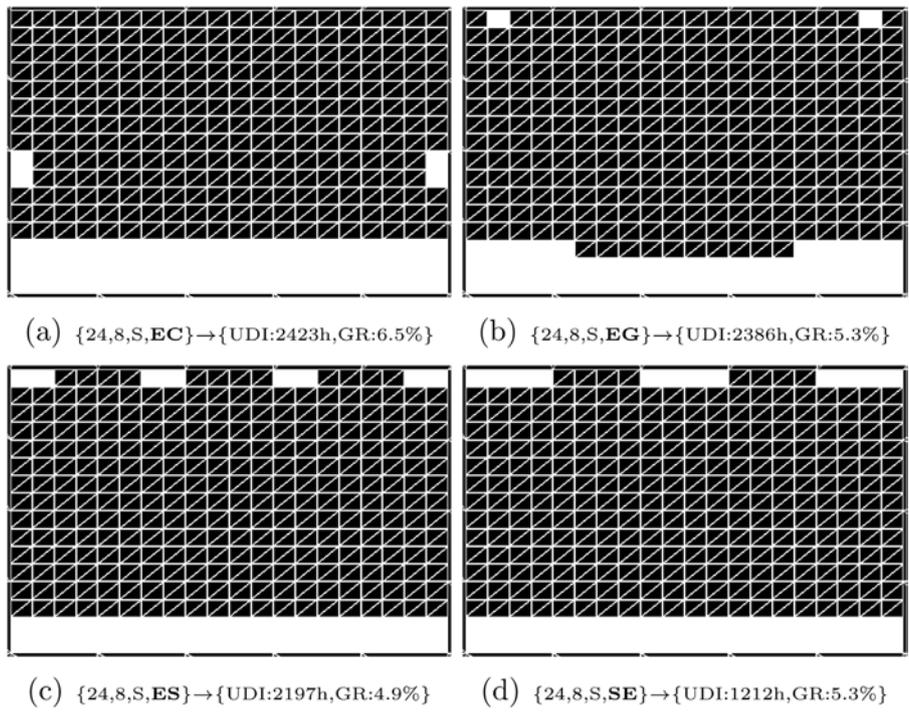


Fig. 16. Solutions for different geographic locations.

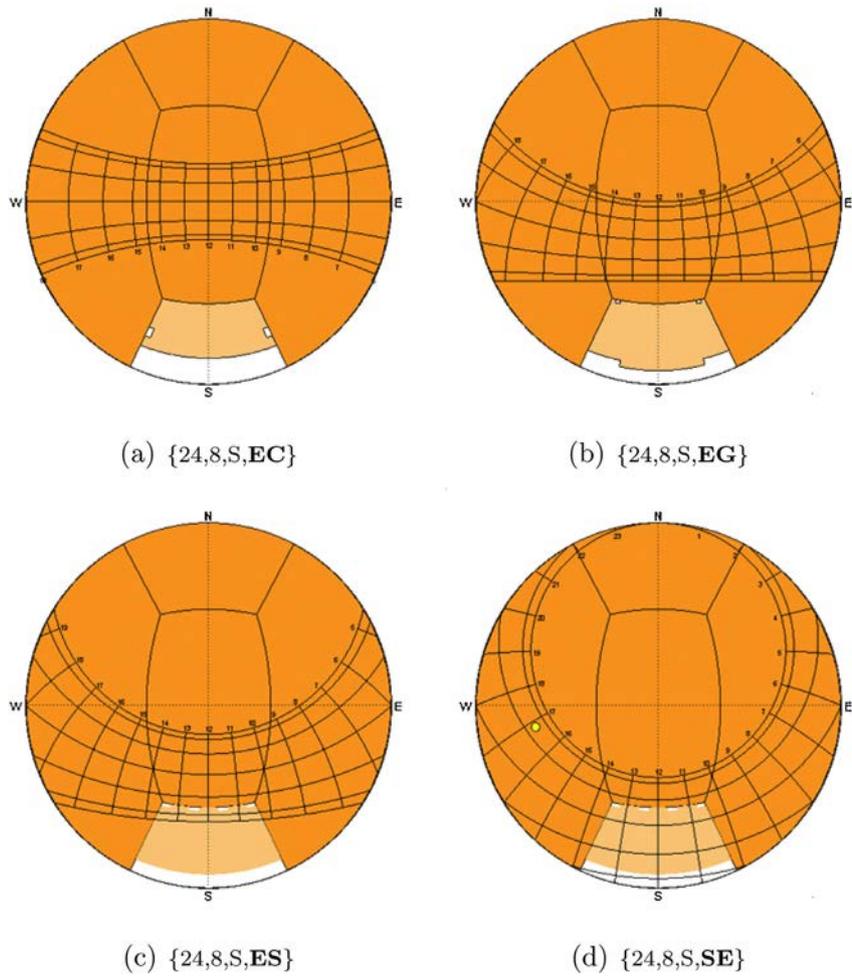


Fig. 17. Solar diagrams for the different geographic locations studied.

## 6.4. Optimization tests

Considering the previous results, we decide to use 24 sensors and 8 rectangles to solve the optimal openings, with different orientations and geographical locations.

### 6.4.1. Orientation

For orientation changes, it can be observed that the results for East, North and West orientations (Fig. 14) have better UDI values than the reference solution (Fig. 9). The highest UDI difference ( $\Delta\text{UDI}=939\text{ h}$ ) is presented when the window is oriented to the North.

Fig. 15 shows several results for East and West orientations. In Fig. 15(a)–(d) the optimal shape is shown without the symmetric restriction. As it is explained in Section 6.3, this case implies more variables (32) and more optimization steps (200,000) than the symmetric counterpart. In Fig. 15(c) and (d) we show the opening configuration obtained when the full year (24 h per day and 8760 h per year) is considered. Finally, in Fig. 15(e) and (f), we apply the symmetry again but for the full year.

The non-symmetric results found (Fig. 15(a)–(d)), slightly improve their symmetric counterparts (Figs. 14(a), (c), 15(e), and (f), respectively). An interesting question to discuss is if the symmetric configurations for East and West orientation should be the same when we optimize in a full year basis. The optimal configurations are quite similar. In Fig. 15(e) and (f), it can be appreciated that the East opening has only 12 patches more than the West opening. Also, they ought to be equal if we just consider a clear-sky model. On the other hand, the UDI hours have a difference of 86 h. So, perhaps the difference in the solutions is due to another factor more related to the optimization process. To evaluate the solutions, we exchanged East and West opening configurations looking for an improvement in the solutions, but the results worsen the optimal UDI values in 22 h and 13 h for East and West orientations, respectively. Thus, it appears that East and West solutions are different because of the climate-based data.

### 6.4.2. Geographical location

We compute optimal window shape for the other 4 geographical selected places: Quito-Ecuador, Kharga-Egypt, Barcelona-Spain, and Kiruna-Sweden (Fig. 16).

The locations cover different latitudes. All the opening configurations are quite similar, but the amount of UDI hours decreases significantly when the latitude increases. Regarding the solar path diagrams from the center of the office (Fig. 17), we can observe that both in the Equator line and the tropic (Fig. 17(a) and (b)) there is no direct Sun along the year. In Barcelona and the Arctic Circle (Fig. 17(c) and (d)), there are some hours of direct Sun coming through the upper part of the window, and in the Arctic Circle the light also comes through the bottom part.

Mean SVF due to the optimized windows are: 0.4% in the polar circle (i.e., a little more than in London), 0.7% in Quito-Ecuador 0.3% in the tropics and in Barcelona. This result is very interesting: on the Equator line, the Sun is always high for the south window, which can be more open. On the polar circle, the Sun is so weak that the window can be a little more opened. In other latitudes, the SVF is rather constant.

## 7. Conclusions and future work

A new method to calculate the illuminance of a set of sensors is presented. This method is based on the radiosity equation and the use of pinholes to model the flux of light passing through the opening. A set of matrix-based equations is defined, whose parameters are the luminance of the sky tiles ( $s$ ) and the geometry of the opening ( $W$ ). As examples of application of these equations,

the UDI metric and the daylight coefficients are formulated, both dependent on  $s$  and  $W$ . The achieved speed led us to implement an optimization method to find optimal openings based on the hourly sky conditions of a whole year. The work is presented here as a fast computational method for aiding design at an early stage of an opening shape analysis. Further steps should address real architectural construction, as well as the connection between the openings and other light sources ( $E \neq 0$ ), or the use of energy for heating and cooling. Another extension of our work is the inclusion of blinds or fenestration systems to expand the possibilities of lighting. Finally, the method could be applied to more complex scenes and to a higher number of sky tiles, taking advantage of the fact that computational cost of the optimization process does not depend on these parameters.

## Acknowledgments

The work was supported by project FSE.1.2014.1.102344 from Agencia Nacional de Investigación e Innovación (ANII, Uruguay) and TIN2014-52211-C2-2-R project from Ministerio de Economía y Competitividad, Spain.

## References

- [1] C.F. Reinhart, J. Mardaljevic, Z. Rogers, Dynamic daylight performance metrics for sustainable building design, *LEUKOS: J. Illum. Eng. Soc. N. Am.* 3 (1) (2006) 7–31.
- [2] A. Nabil, J. Mardaljevic, Useful daylight illuminance: a new paradigm for assessing daylight in buildings, *Light. Res. Technol.* 37 (January (1)) (2005) 41–59.
- [3] E. Fernández, G. Besuievsky, Inverse lighting design for interior buildings integrating natural and artificial sources, *Comput. Graph.* 36 (8) (2012) 1096–1108.
- [4] F. Cassol, P.S. Schneider, F.H.R. França, A.J. Silva Neto, Multi-objective optimization as a new approach to illumination design of interior spaces, *Build. Environ.* 46 (2) (2011) 331–338.
- [5] P.R. Tregenza, I.M. Waters, Daylight coefficients, *Light. Res. Technol.* 15 (2) (1983) 65–71.
- [6] E. Fernández, G. Besuievsky, Inverse opening design with anisotropic lighting incidence, *Comput. Graph.* 47 (2015) 113–122.
- [7] H. (Chair) Lisa, W. Van Den, Keven (Vice-Chair), M. Andersen, N. Digert, L. Fernandes, A. Keller, J. Loveland, H. McKay, R. Mistrick, B. Mosher, C. Reinhart, Z. Rogers, M. Tanteri, Approved Method: IES Spatial Daylight Autonomy (sDA) and Annual Sunlight Exposure (ASE), 2010, April.
- [8] B.J. Futrell, E.C. Ozelkan, D. Brentrup, Optimizing complex building design for annual daylighting performance and evaluation of optimization algorithms, *Energy Build.* 92 (2015) 234–245.
- [9] K. Hiyama, L. Wen, Rapid response surface creation method to optimize window geometry using dynamic daylighting simulation and energy simulation, *Energy Build.* 107 (2015) 417–423.
- [10] G. Ward, R. Mistrick, E.S. Lee, A. McNeil, J. Jonsson, Simulating the daylight performance of complex fenestration systems using bidirectional scattering distribution functions within radiance, *LEUKOS: J. Illum. Eng. Soc. N. Am.* 7 (4) (2011) 241–261.
- [11] A. McNeil, E.S. Lee, A validation of the radiance three-phase simulation method for modelling annual daylight performance of optically complex fenestration systems, *J. Build. Perform. Simul.* 6 (1) (2013) 24–37.
- [12] A. McNeil, The Five-Phase Method for Simulating Complex Fenestration with Radiance, Technical Report, BBLN, 2013.
- [13] G.J. Ward, The radiance lighting simulation and rendering system, in: *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '94*, ACM, New York, NY, USA, 1994, pp. 459–472.
- [14] V. Tourre, J.-Y. Martin, G. Hégron, An inverse daylighting model for CAAD, in: *SCCG '08: Proceedings of the 24th Inverse Conference on Computer Graphics*, Comenius University, Bratislava, 2008, pp. 95–102.
- [15] G. Besuievsky, V. Tourre, A daylight simulation method for inverse opening design in buildings, in: *Proceedings of the IV Iberoamerican Symposium in Computer Graphics*, Sociedad Venezolana de Computación Gráfica, DJ Editores, CA, 2009, June, pp. 29–46.
- [16] E. Fernández, G. Besuievsky, Efficient inverse lighting: a statistical approach, *Autom. Constr.* 37 (Complete) (2014) 48–57.
- [17] F. Goia, M. Haase, M. Perino, Optimizing the configuration of a façade module for office buildings by means of integrated thermal and lighting simulations in a total energy perspective, *Appl. Energy* 108 (2013) 515–527.
- [18] J. Wright, M. Moursheid, Geometric optimization of fenestration, in: *Proceedings of the Eleventh International IBPSA Conference (Building Simulation 2009)*, IBPSA, 2009, pp. 920–927.

- [19] M.F. Cohen, J. Wallace, P. Hanrahan, *Radiosity and Realistic Image Synthesis*, Academic Press Professional, Inc, San Diego, CA, USA, 1993.
- [20] M.F. Cohen, D.P. Greenberg, The hemi-cube: a radiosity solution for complex environments, *SIGGRAPH Comput. Graph.* 19 (3) (1985) 31–40.
- [21] B.W. Bader, T.G. Kolda, Algorithm 862: MATLAB tensor classes for fast algorithm prototyping, *ACM Trans. Math. Softw.* 32 (4) (2006, December) 635–653.
- [22] D.E. Knuth, *The Art of Computer Programming Seminumerical Algorithms*, vol. 2, 3rd ed., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.
- [23] E. Fernández, Low-rank radiosity, in: O. Rodríguez, F. Serón, R. Joan-Arinyo, E. Coto, J. Madeiras, J. Rodríguez (Eds.), *Proceedings of the IV Iberoamerican Symposium in Computer Graphics*. Sociedad Venezolana de Computación Gráfica, DJ Editores, CA, 2009, June, pp. 55–62.
- [24] P. Hansen, N. Mladenovic, Variable neighborhood search: principles and applications, *Eur. J. Oper. Res.* 130 (3) (2001) 449–467.
- [25] D. Luenberger, Y. Ye, *Linear and Nonlinear Programming*. International Series in Operations Research & Management Science, Springer, 2008.
- [26] MATLAB version 7.10, The MathWorks Inc., Natick, MA, 2010.
- [27] B.W. Bader, T.G. Kolda, et al., **MATLAB Tensor Toolbox Version 2.6, 2015, February** <http://www.sandia.gov/tgkolda/TensorToolbox/>.
- [28] **EERE, Weather Data, 2015, December** <http://energy.gov/eere/office-energy-efficiency-renewable-energy>.
- [29] B. Beckers, L. Masset, **Heliodon2: Software, References & Manuals, 2008** <http://www.heliodon.net/> (in French & Spanish).