



Real-Time Solar Exposure Simulation in Complex Cities

I. Muñoz-Pandiella^{1,2}, C. Bosch³, N. Mérillou¹, X. Pueyo² and S. Mérillou¹

¹XLIM Research Institute - CNRS, Université de Limoges, France
{imanol.munoz-pandiella, nicolas.merillou, stephane.merillou}@unilim.fr

²ViRVIG Research Center, Universitat de Girona, Spain
xavier.pueyo@udg.edu

³Eurecat, Centre Tecnològic de Catalunya, Spain
carles.bosch@eurecat.org

Abstract

In urban design, estimating solar exposure on complex city models is crucial but existing solutions typically focus on simplified building models and are too demanding in terms of memory and computational time. In this paper, we propose an interactive technique that estimates solar exposure on detailed urban scenes. Given a directional exposure map computed over a given time period, we estimate the sky visibility factor that serves to evaluate the final exposure at each visible point. This is done using a screen-space method based on a two-scale approach, which is geometry independent and has low storage costs. Our method performs at interactive rates and is designer-oriented. The proposed technique is relevant in architecture and sustainable building design as it provides tools to estimate the energy performance of buildings as well as weathering effects in urban environments.

Keywords: illumination rendering, rendering, real-time rendering, rendering, visibility determination, rendering

ACM CCS: I.3.3.b [Computer Graphics]: Picture/Image Generation—Bitmap and framebuffer operations, I.3.7.b [Computer Graphics]: Three-Dimensional Graphics and Realism—Colour, shading, shadowing and texture

1. Introduction

Estimating solar radiation on virtual models has implications in several areas of computer graphics, including global illumination techniques, modelling and simulating natural phenomena. Furthermore, solar radiation is an important value in various other fields such as renewable energy studies, climatology or urban planning. To the best of our knowledge, the existing techniques in such areas (summarized in the next section) do not allow for the complex geometries of buildings and are time-consuming when dealing with large cities. The related literature has clearly identified, the need for, and the lack of, methods that provide the scope to handle complex building structures [MM09]. To tackle this problem, we propose an efficient approach to compute the intervening factors; particularly solar radiation and its interaction with surfaces.

Solar radiation plays a central role in several research areas in computer graphics. This includes physically based rendering of outdoor scenes [PSS99, HW12], as well as indoor illumination with

daylight [DKN*94, FB12]. In other fields, solar radiation is required to calculate urban climate [Lin07], the electric power generated by photovoltaics [MGVMLL14] or even the simulation of vegetation growth [COMM94, MP96].

Sky visibility is a key mechanism in estimating solar exposure on surfaces. This parameter, which is only geometry-dependent, is commonly called the sky view factor (SVF). Introduced in a pioneering paper from Johnson and Watson [JW84], the SVF is defined as the fraction of radiant flux leaving a planar surface that is being intercepted by the sky. This factor is closely related to the well-known term ambient occlusion (AO) term [MFS09] used to estimate part of the rendering equation in computer graphics. The main difference is that while AO computes global accessibility, SVF only computes just sky accessibility.

In this paper, we propose a new technique for estimating detailed solar exposure on complex urban models. We represent solar radiation as a directional exposure map integrated over a desired time period. We then evaluate the sky view term using a

two-scale screen-space approach, and we use this information to compute the final exposure on surfaces. As a result, our method is geometry independent, have low storage requirements and offer interactive feedback. Our technique is amenable to dynamic modifications of the scene, useful in preliminary stages of applications where considerable accuracy is needed, particularly for preview purposes.

The proposed technique is also useful in the sustainable development of urban environments. In the coming decades, there will be a high demand for renewable energy use in order to improve the global energy efficiency and to protect the environment. Consequently, we will all have to adapt to this new paradigm and to address the all-important question of just how we will adjust. One of the multiple answers probably lies in better use of solar energy, both for heating and lighting. Not only do new buildings or future cities that provide access to and make the most of this renewable energy need to be designed, but existing buildings will also need to be upgraded and improved.

The remainder of this paper is organized as follows; in the next section, we present previous work, followed by an overview of our pipeline. The following sections concentrate on estimating of the sky directional exposure map and the screen-space visibility on two levels of accuracy: both global and local scales. Finally, we describe the solar exposure evaluation step and we present our results and discussion.

2. Previous Work

Urban solar radiation has been addressed in different applications, including global illumination techniques [AAP12], atmospheric rendering [DKY*00, REK*04], participating media [SRNN05] and lighting [WFA*05]. Moreover, solar radiation is also an important parameter in the ageing phenomena [MG08] as well as in modelling of plant growth [DGA04]. Other fields are also exhibiting a growing interest as a result of (among other factors) its importance as a tool to improving energetic efficiency, and the considerable progress being made today with computing technology. Estimating solar radiation at a specific location, and moreover in a complex urban scene, is a challenging task. Some tools and techniques such as Heliodon ([MAAB10]), the DIVA suite with ArchSim [DRM12] and interfaced with Radiance [War94] are already available, along with recent GPU-based studies (e.g. [LGLI15]). These are commonly based on ray-casting and texture-baking algorithms and although they are able to handle a large number of buildings, it is with simple geometry and at the cost of high memory usage and computational time. A compromise between accuracy and computation time [FCRB15] is still needed and our technique obtains interactive frame rates with low memory usage in complex and detailed scenes. Note, however, that for our purpose, we do not take into account specular bounces induced by glass-sheathed buildings for example (see NVIDIA case study, for example, [NV115]).

Solar radiation modelling is an important concept in many fields. In realistic rendering, it became popular due to the work of Preetham *et al.* [PSS99]. Preetham uses the Perez model [PSM93] to capture the non-uniform variations over the sky dome. To describe the luminance of clear skies, this method uses an analytic formula which

includes changes occurring at the circumsolar region and near the horizon. To replace the use of tabulated values in the Perez model, Preetham *et al.* [PSS99] compute reference simulations at different solar elevations and turbidity with the Nishita model [NDKY96], then they propose a set of linear functions that depend on one parameter: *turbidity*. They also introduced interpolated chromatic channels. In 2008, Bruneton *et al.* [BN08] proposed a model that could handle multiple scattering, was able to render sky from all viewpoints (ground to space) and needed a pre-computation step. Recently, Hosek and Wilkie [HW12] proposed extensions to the Preetham model [PSS99] to handle spectral data, to improve high turbidity situations and to account for ground albedo. This model has been shown to better characterize measured spectral data [KKN*14]. Besides analytical models, services such as Solargis [ŠCS11] or Soda-IS [SoD15] provide solar radiation data for specific locations and times, although they rarely provide directional information, nor do they offer enough freedom when tuning sky conditions. In this work, we make use of the Hosek and Wilkie technique [HW12, HW13] to model solar radiation in different atmospheric conditions which are controlled by adjusting the turbidity.

Screen-Space AO Due to the close proximity to SVF and AO (see Section 5), we propose handling the computation of SVF using an adapted version of screen-space AO. In their pioneering paper, Zhukov *et al.* [ZIK98] introduced ambient obscurance, replacing the visibility function with a distance-based falloff function. A large number of studies concerning both the approaches of obscurance and AO [MFS09, MOBH11] have since followed. After the initial screen-space approaches appeared [Mit07, SA07], a number of new techniques improving performance and quality [BSD08, RGS09, BS09] were developed. We build our SVF estimation chiefly on the horizon-based AO method presented by Bavoil *et al.* [BSD08].

Computing the SVF The SVF is typically estimated *in situ* or computed. In the first case, the estimation is performed using a fish-eye camera, positioned 1 m from the ground and looking vertically towards the sky [BGR01]. Image analysis techniques then compute the part of the visible sky and afterwards the SVF. On the other hand, the SVF can be derived from the digital data from the environment using raster models that represent data from GIS databases and Digital Elevation Maps [RR04, MM11]. However, these techniques are not designed to work with complex buildings or urban objects, and so do not present interactive computations. Our technique, however, overcomes these drawbacks through a screen-space method which handles arbitrary geometry at interactive rates.

3. Overview

The overview of our method is depicted in Figure 1. We start with pre-computing a directional exposure map by integrating daylight irradiance over the desired (typically a year) time period (see bottom-left box). Using Hosek and Wilkie technique [HW13], the map is computed for the full sky dome according to parameters such as the scene latitude or mean sky turbidity.

At runtime, we evaluate the visibility of the scene using a two-step screen-space approach. Using an adapted screen-space solution, large-scale visibility is computed from a global view of the

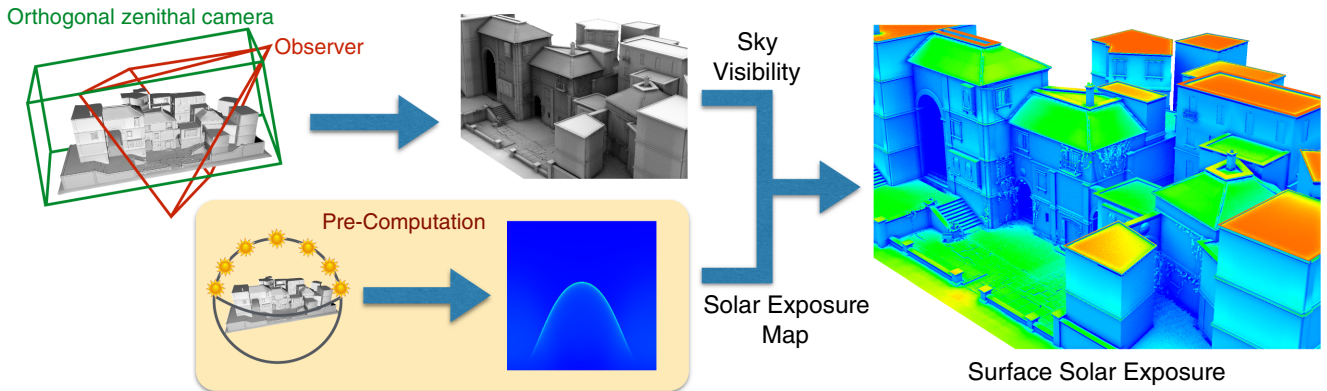


Figure 1: Overview of our system. A directional solar exposure map is pre-computed over the specified time period and under the given conditions (bottom row). At runtime, we evaluate visibility in screen space by combining a global and local view of the scene (top row). Surface solar exposure is finally computed by sampling the map within the sky regions visible from each point (right).

scene captured from the zenith (green frustum in left-top part). This visibility is then combined with a fine-scale term computed from the current viewpoint using common screen-space AO (red frustum).

Finally, we use the visibility information to sample the directional exposure map and compute the final solar exposure at each surface point (rightmost image). This sampling is performed taking advantage of mipmap pre-filtering, which reduces the number of required samples.

All these results are obtained in an interactive framework that provides the required accuracy from each view and only entails two geometry-independent textures being pre-computed (i.e. the exposure map and a global depth map). Our technique belongs to the category of interactive simulation approaches, which allow accuracy and building geometries to be changed on-the-fly. In addition, due to our two-scale process, we obtained a good match (compared to ground truth visibility and solar exposure) over detailed scenes, and achieved a high level of realism when compared to existing solutions.

4. Solar Radiation

The amount of solar radiation reaching a surface is usually split into the sum of three components [MAAB10]: direct or beam radiation, which is the radiation directly incident from the sun that is not being absorbed or scattered by the atmosphere, sky diffuse radiation, which originates from the solar beam but scatters towards the surface, and reflected radiation, which is reflected from the ground and neighbouring objects. These components are controlled on a global scale by the geometry of the earth, the relative location of the sun and the atmospheric transmittance, whereas on a local scale, they are controlled by surface orientation, elevation and albedo.

Of these components, the direct radiation is the most straightforward to compute, while the reflection from the ground is usually considered to have a lesser impact. On the other hand, the diffuse component is considered as the largest potential source of computational error [NMK08]. In this work, we rely on the Hosek skydome

model [HW13], as it significantly reduces the error related to the diffuse component when compared to the Preetham model [KKN*14]. In addition, we simplify our method by neglecting reflections from the ground or neighbouring surfaces.

4.1. Computing daylight radiance

Assuming no ground reflection, daylight effects due to the sun and the sky can be decomposed into two main components: the direct beam coming from the sun, L_{dir} , and the sky diffuse term due to light scattering on the atmosphere, L_{dif} . The incoming radiance L_i at a given point x is thus described by

$$L_i(x, \omega_i) = L_{dir}(\omega_i) + L_{dif}(\omega_i), \quad (1)$$

where ω_i is the incident direction defined over the hemispherical domain Ω . Note that both terms represent spectral quantities that change over time. Also, due to its relative distance from the scene, we can drop the dependence on the spatial position x .

The direct component is assumed to originate from the centre of the solar disk, and can thus be described by a Dirac delta function over the sun direction, ω_{sun} . In this direction, the incoming radiance can be computed as the extra-atmospheric radiance reaching the atmosphere, L_{ea} , weighted by the atmospheric transmittance τ . The latter oversees light attenuation resulting from absorption and scattering. Preetham provides measured spectral data for both factors [PSS99]:

$$L_{dir}(\omega_i) = \tau L_{ea}, \quad \text{if } \omega_i = \omega_{sun}. \quad (2)$$

Several models have been proposed to approximate the effect of the diffuse component, by relying on either an isotropic or an anisotropic description of the sky [NMK08]. Classical anisotropic luminance models, such as the CIE standard or the Perez model, assume that the sky is composed of three main terms: a circumsolar region, a horizon band and an isotropic back-scattering term. Each of these terms can be controlled by a set of parameters (a through

e) that can later be fit to measured data [PSM93] or simulations [PSS99, HW12, HW13]:

$$f(\theta_z, \gamma) = \left(1 + a \exp\left(\frac{b}{\cos \theta_z}\right)\right) \left(1 + c \exp\left(\frac{d}{\gamma}\right) + e \cos^2 \gamma\right), \quad (3)$$

where θ_z is the angle from the zenith and γ the angle from the sun. The function f models the relative luminance of the sky, usually with respect to the zenith luminance, $L_i(\omega_z)$. After normalizing and weighting by this luminance, the final diffuse term becomes:

$$L_{diff}(\omega_i) = L_i(\omega_z) \frac{f(\theta_{z,i}, \angle(\omega_i, \omega_{sun}))}{f(0, \theta_{z,sun})}. \quad (4)$$

Although these models were originally restricted to luminance, Preetham *et al.* use the same formulation to describe chrominance variations [PSS99], while Hosek and Wilkie rely on a similar expression for spectral bands [HW12, HW13].

4.2. Annual exposure

Many applications require the total solar energy reaching a surface over a specific period of time to be estimated. Conventional quantities relate to daily, weekly or monthly estimations, but annual estimations are especially relevant for long-term decisions such as building design or decay simulation. This term is often referred to as solar exposure, and expressed as MWh/m^2 .

We are interested in the distribution of energy over the sky hemispherical domain Ω , which will later be combined with visibility information to obtain the final exposure at each surface point. The distribution of energy over the sky depends on the sun path over the specific period (e.g. a year), the latitude our model occupies and the atmospheric conditions. The latter is represented by the turbidity parameter, which may vary over time. In our examples, we use a fixed turbidity value representing the average conditions at a specific location and for an explicit period of time, although a more accurate representation could be used with our technique in the case of yearly data being available for that location.

To estimate an annual map, we need to integrate Equation (1) over time for each hemispherical direction ω_i . We do this using numerical integration, i.e. by sampling the sky radiance distribution at uniform time steps within the daily hours of the period. The sun is also baked into the map to store its temporal behaviour over the same period. Due to the discrete nature of the sun, in this case, we rely on splatting its energy over the map at each time step in this case.

At each step, the radiance of the sun and the sky is obtained using [HW13], then integrated as previously explained. Our directional exposure map is finally obtained by weighting these values with the corresponding solid angle, i.e. the sky patch or the sun radius in each case. The cosine term needed for computing the final exposure will be incorporated at runtime, as it depends on surface orientation (see Section 6).

Besides time integration, we also integrate quantities from the available spectrum. Although Hosek's approach is limited to a small

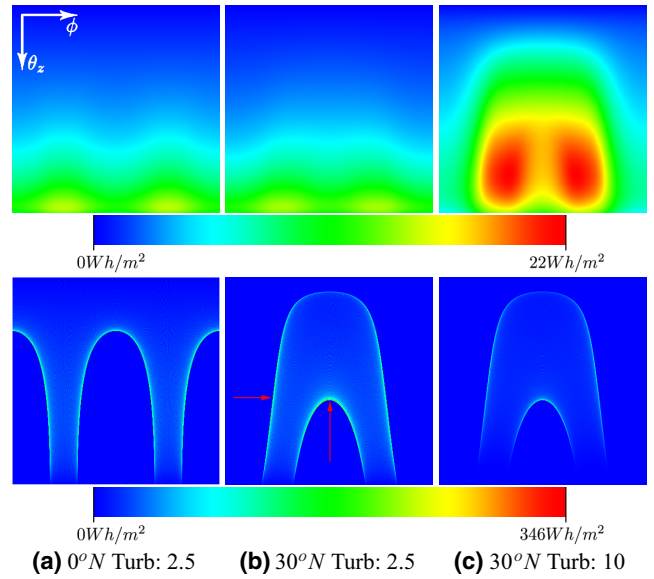


Figure 2: Annual exposure maps (diffuse on top and direct on bottom with their respective scales) obtained using different configurations: latitude $0^\circ N$ turbidity 2.5 (left), latitude $30^\circ N$ turbidity 2.5 (middle) and latitude $30^\circ N$ turbidity 10 (right). Red arrows mark the solstices sun band.

range of the light spectrum, one could focus on specific parts of the spectrum if necessary. Focusing on specific ranges such as visible, uv, or infrared light, might be of special interest for applications like weathering or plant growth simulation. The map obtained is ultimately represented as a floating-point texture using the classical Blinn–Newell parametrization of the hemisphere. We chose this parametrization because it is proper for the kind of sampling based on the directions that we use, as detailed in Section 6.

Figure 2 shows annual exposure maps for different latitudes and turbidities. Note that we use different scales between the direct and diffuse components to depict the maps. From centre to right, it is easy to observe how the contribution of the sky diffuse term increases for high turbidity values. From left to centre, the sun path, as expected, is greatly influenced by latitude changes. Also, notice how the energy attributable to the direct term concentrates at the declination limits of the sun band, which happens at the solstices (see red arrows in bottom-middle image of Figure 2).

5. Sky Visibility

After computing the exposure map, we need to determine the contribution this map makes by evaluating its visibility at each point. This means that, for every direction of the sky hemisphere, we need to know whether an object from the scene is blocking it. Since we want to be able to process large city models at interactive rates, we resort to screen-space techniques which are independent of scene complexity and can be easily evaluated with a simple depth map of the scene in question.

As we mentioned earlier, most techniques in the literature use SVF to estimate solar exposure. SVF can be defined as the amount of the sky hemisphere seen from a surface point. In some way, we can consider this factor complementary to the well-known (in computer graphics) AO. However, AO is defined as the amount of environmental occlusion from a surface point. For this reason, to make this complementary relationship true, AO needs to be adapted to take into account the sky hemisphere instead of the surrounding hemisphere.

Screen space AO (SSAO) is a well-known technique for computing visibility in screen space [MFS09], but it is commonly restricted to visibility from nearby surfaces that are directly observed from the camera's point-of-view. Here, we evaluate occlusion using a similar approach but resort to a two-step procedure. In a first step, we evaluate large-scale occlusion from the sky using a zenithal view of the city, which encompasses a larger portion of the scene. This information is then combined with standard SSAO from the camera's point-of-view to include fine-scale visibility.

5.1. Large-scale visibility

Screen-space methods typically consist of two stages: a rendering step that computes the depth map of the scene from the current camera position, followed by a second step that performs the computations using this depth map. Along with the depth, a normal map is often retrieved to determine surface orientation (see Figure 3).

Relying on information recovered from the current view is usually enough for evaluating local visibility at every point, but for global visibility, we need additional information. In our case, we include an additional view from the zenith using an orthogonal projection. This view is adjusted to encompass a large portion of the city model and its depth map is generated as a preprocess, independent of the current camera position. As it is view-independent, this map must only be generated again if the geometry of the scene changes. To compute global visibility, this depth map is evaluated on-the-fly, but only for the points visible from the main view. That is, we project each visible point from the main view to the zenithal view and evaluate visibility there (see input maps in Figure 3). In this way, computations are only evaluated for visible points, thus avoiding the need to pre-compute and store them in the model.

Once at the zenithal view, we sample the corresponding depth map in several directions (N_{dir}), similar to [BSD08]. For each direction θ in image space, we take multiple samples (S_{dir}) and we compute an elevation angle $h(\theta)$, also called the horizon angle, at each of them (see Figure 3). The highest angle represents the occluded portion of the environment for that particular direction. As in [BSD08], we apply a random rotation to all directions at each point to reduce aliasing artefacts.

Two common problems arise when using the previous approach. The first is due to sampling, which might lead to certain features being skipped during the evaluation. This is especially noticeable on corners, which tend to underestimate occlusion for directions opposing the surface normal. To solve this, we simply discard those directions based on their cosine angle.

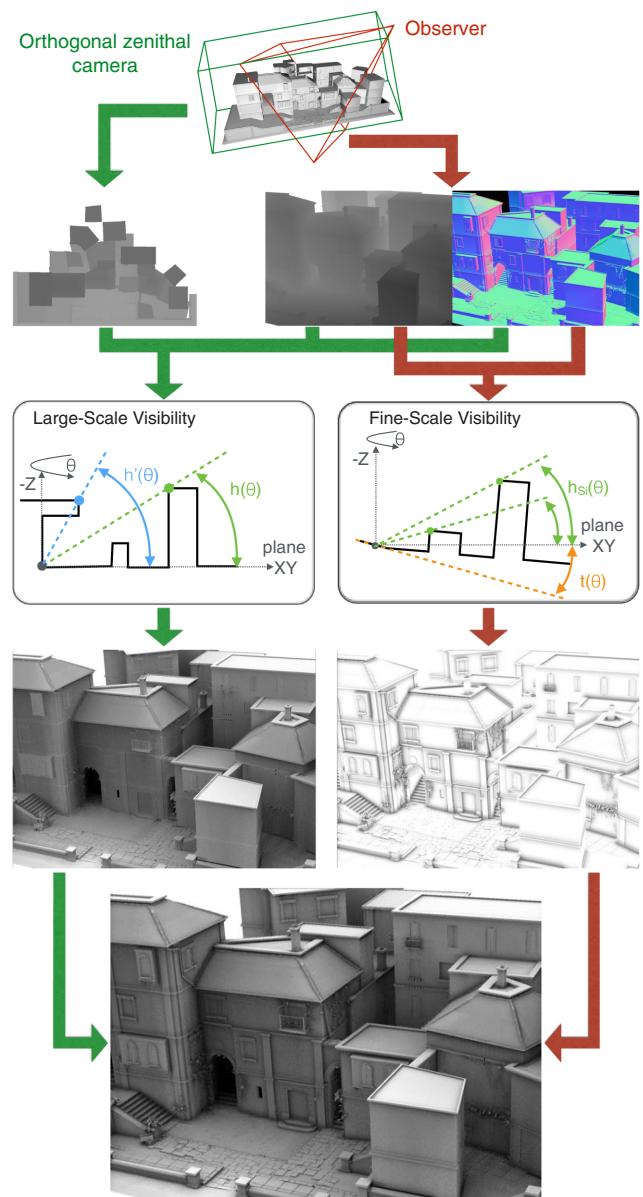


Figure 3: View-dependent sky visibility. Left: Large-scale visibility is computed in screen space using a zenithal orthographic view, accounting for non-heightfield features. Right: Fine-scale visibility is evaluated from the current camera position. Bottom: In the end, both results are combined.

The other issue is that relying on a single depth map inherently limits us to heightfield models as seen from the current (zenithal) view. This often results in facades or other elements being completely darkened due to salient roofs or balconies (see Figure 4). Ideally, one should resort to multiple views or depth peeling (multiple depths) to handle those situations [RGS09]. However, this would be computationally expensive. Here, we rely on a simpler approach based on computing a second elevation angle $h'(\theta)$ that we call the opening angle. For each sample direction, the opening angle is evaluated right before the first sample is found to be lower

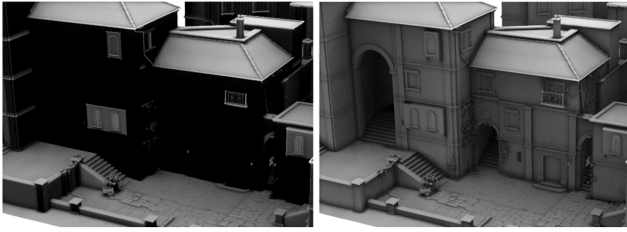


Figure 4: Visibility obtained using the opening angle $h'(\theta)$ (right) and without using it (left), computed using eight directions (N_{dir}) and eight samples per direction (S_{dir}).

than the current point (i.e. farther in depth). This term is combined with the subsequent elevation angle in order to obtain the final visible region for the current direction: $[h'(\theta), h(\theta)]$ (see Figure 3, left diagram).

Since we cannot detect intercepting features causing complete occlusion, this approach might lead to underestimated occlusion. However, in practice, it works well in most common situations, as shown in Figure 5 in the comparison with ground truth generated in a 7 h average using ray casting. Additionally, a user-provided threshold can be employed to discard openings far from the current point.

Note that the SVF for a specific direction can be obtained using:

$$SVF(\theta) = \sin h'(\theta) - \sin h(\theta). \quad (5)$$

The average of this term over all θ directions would then give us the final SVF term. For isotropic sky models, this term can simply be used as a weighting factor of the uniform diffuse exposure. For anisotropic models, we rather need to use the specific visible regions to explicitly sample and weight the exposure map, as detailed in Section 6.

5.2. Fine-scale visibility

The occlusion estimated using the method above lacks enough detail for an appropriate estimation. This is mainly due to the resolution used for the depth map and the differences in viewing angle. To incorporate fine-scale visibility, we resort to common SSAO computed from the camera view, which perfectly suits our needs in this case.

Similar to our large-scale approach, we sample the depth map from the current view in different directions, each time computing the horizon angle with respect to the viewing plane (XY in Figure 3). For each direction, the samples here are combined using a distance function $W(s_i)$, similar to [BSD08]. The tangent plane at the current point defines a second angle $t(\theta)$ from which the actual horizon angle should be defined. The local visibility term for a specific direction is then computed as

$$LVF(\theta) = \sum_{i=1}^{S_{dir}} ((\sin h_{s_i}(\theta) - \sin t(\theta)) \cdot W(s_i)). \quad (6)$$

As before, the final visibility term is obtained by averaging $LVF(\theta)$ over all θ directions.

The combination of the fine-scale occlusion term with the large-scale term should, ideally, be done by merging their visible regions, which would require handling their differences in sampled directions. A much simpler approach consists of weighting the obtained exposure with the fine-scale term which, in practice, proved to be sufficient (see Figure 5).

6. Surface Solar Exposure

After evaluating the visibility, we are ready to compute the final solar exposure at each surface point. The large-scale visibility step gives us the visible regions of the exposure map that will contribute to this final term. Inside these regions, we take a set of samples to recover the corresponding values, and weight their contribution according to the cosine term based on the current normal (see Figure 6). To avoid using a lot of samples, we resort to pre-filtering the map by means of mipmapping, as we detail next.

Note that the cosine term cannot be baked into this map as it changes for each surface point and direction. Because of its smooth condition, however, we consider it as constant within each mipmap sample. Using other pre-integration approaches, such as summed-area tables, would still require an appropriate cosine weighting within subregions.

Recall, from Section 5, that a visible region over a specific direction is defined by a pair of angles $[h'(\theta), h(\theta)]$. Within this range, we take uniform samples along the current direction, where the number of samples is adapted according to

$$n_s = \max\left(\frac{2 * (h'(\theta) - h(\theta)) * t_{height} * N_{dir}}{\pi * t_{width}}, 1\right). \quad (7)$$

Here, t_{height} and t_{width} are the dimensions of the exposure map and N_{dir} is the number of visibility directions (see vertical slices in Figure 7). According to this, n_s is proportional to N_{dir} and it depends on both the visible region and the aspect ratio of the exposure map.

Our sampling strategy consists of uniformly sampling the visible region corresponding to each direction and appropriately fitting the samples within this region (see Figure 7). For the last sample, we then adjust its position and size according to the remaining portion. Compared to a classical overlapping of samples, this approach does not overestimate exposure. Additional samples could always be taken into the remaining portion if necessary.

In order to select the appropriate level b within the mipmap, we take into account the limits of the current sample in elevation and the height of the texture:

$$p_{height} = \frac{2 * (h_{sup}(\theta) - h_{inf}(\theta)) * t_{height}}{\pi}, \quad (8)$$

$$b = \max(0, \log_2(p_{height})), \quad (9)$$

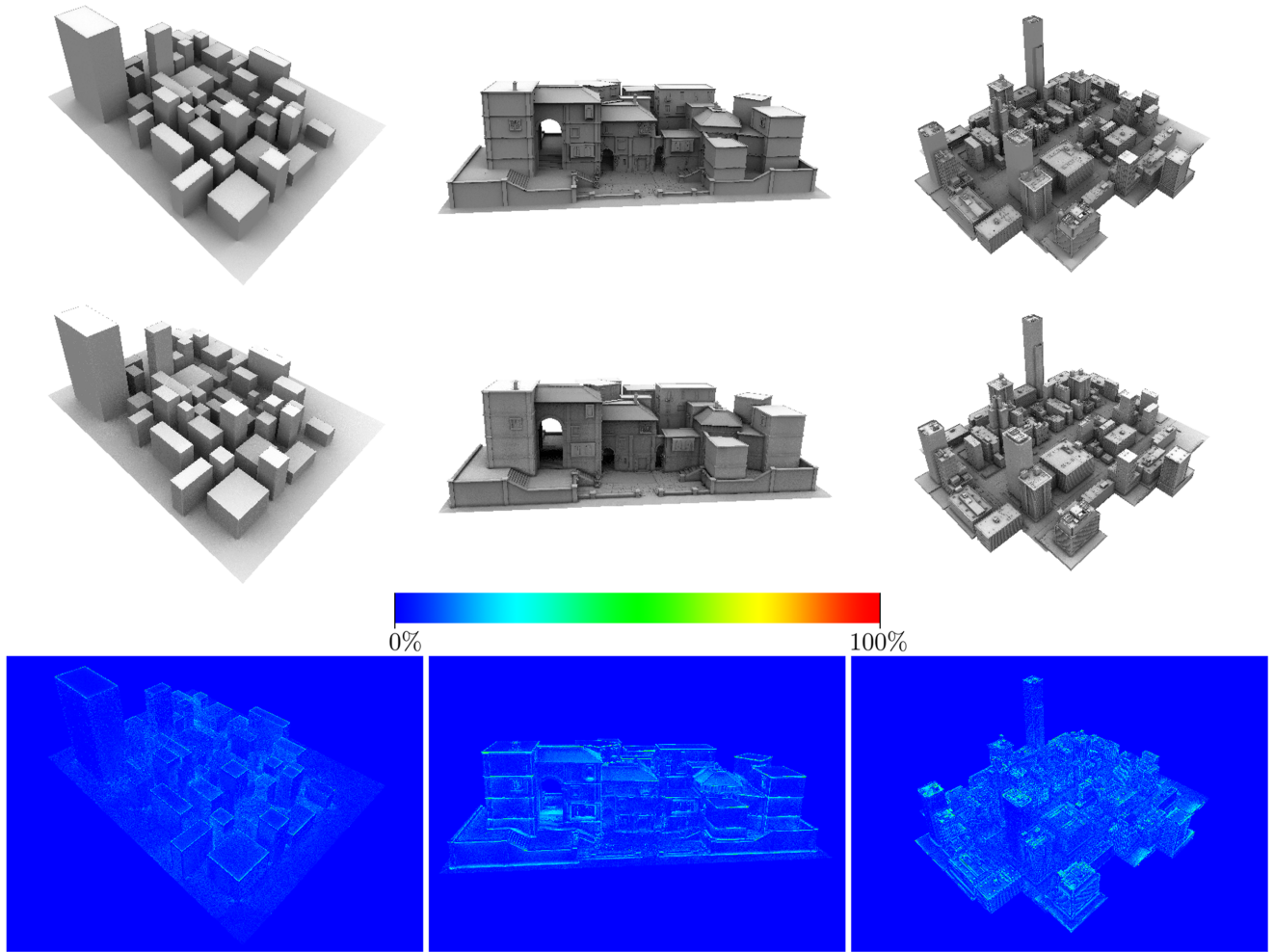


Figure 5: Sky visibility comparison between ground truth (top) and our method (centre), including difference images (bottom). Ground truth is based on ray casting using 128 samples per point. Our method uses 16 directions (N_{dir}) and 16 samples per direction (S_{dir}). Difference images show the difference of visibility percentage. The heightfield model uses our large-scale visibility term (centre left), while the Mediterranean and Urban Sprawl models use our combined visibility (centre middle and right).

where $h_{sup}(\theta)$ and $h_{inf}(\theta)$ define the limits of the current sample in elevation (see Figure 7), and p_{height} represents the height of the sample in pixels.

After retrieving the values from the mipmapped texture, they represent averaged values. Hence, to obtain the total exposure, we subsequently weight them according to the number of pixels (n_p) subtended by the mipmap patch:

$$n_p = p_{height}^2. \quad (10)$$

The value obtained from each sample is finally weighted by the cosine term and accumulated to obtain the final exposure of the surface point.

Figure 7 shows an example of the sampling performed over three visible regions obtained from a surface point. Note how the number of samples and their size is adapted to each region.

7. Results

We have tested our approach on three different models: one simple heightfield model and two more complex city models (*Mediterranean* and *UrbanSprawl*). Table 1 shows the performance we obtained for an un-optimized code on a “MacBook Pro 15” with an Intel Core i7 2.6GHz, 16GB memory and an NVIDIA GeForce GT 750M with 2GB memory. It is noticeable that even when using a considerable number of samples, our model still exhibits interactive performance. Also, due to our screen-space approach, the performance does not change considerably with the scene complexity. The timings for the pre-computation step, generating a 1-year integration exposure map with an un-optimized code, are 5 s for the direct beam part and over 3700 s for the diffuse part. The extra memory costs of our method, with respect to traditional SSAO methods, are a float texture of 512×512 pixels for the exposure map, which takes up to 1 MBytes, and a 32-bit texture of 800×600 pixels for the zenithal depth map with a cost of 1.92 MBytes.

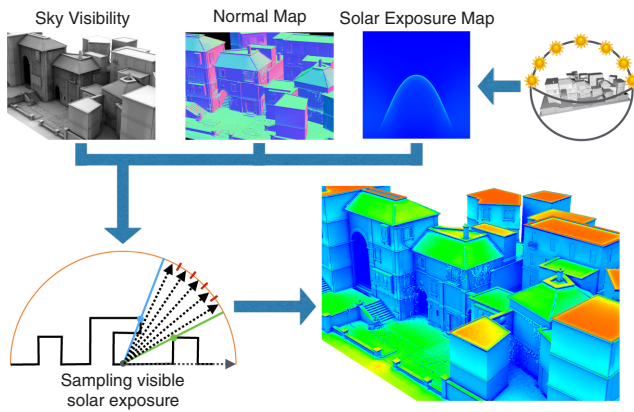


Figure 6: Computing surface solar exposure. The input exposure map is sampled within the visible regions obtained at each point. The normal map provides the surface orientation for the final weighting with the cosine term.

In order to display our results, the visibility represents here the SVF. Moreover, all the exposure maps used in the results presented here account for a 1-year integration period.

Figure 8 shows the final exposure computed on the *Urban Sprawl* and *Mediterranean* models for different latitudes and turbidities. At latitude $0^\circ N$ (top images), the variations on turbidity affect the model less. Only some small differences can be observed in the regions with less direct beam impact (mainly facades less oriented towards the sun path). At latitudes $30^\circ N$ and $60^\circ N$ (bottom images), how the exposure associated to the orientation changes, with regard to the sun path, is more noticeable (again on facades and pitched roofs). One can see how the regions less affected by the direct beam increase their exposure when turbidity is increased (see facades north oriented at $30^\circ N$ images) due to an increase in the diffuse component. Similarly, the exposure is reduced in regions affected by the direct beam (see roofs at $60^\circ N$ images).

While the variations on turbidity are intuitive (decreasing on direct and increasing on diffuse components), the influence of the latitude (for a same polar orientation) is clearly noticeable and proves

Table 1: Performance of our technique for different models. The Samples column specifies the number of visibility directions [N_{dir}] \times (samples per direction [S_{dir}] + maximum solar exposure samples [n_s]). Note that for a square solar exposure texture, as used in the examples, the latter equals to N_{dir} in the worst case (full visibility). Timings in fps were obtained with 60 Hz vsync enabled on an 800×600 viewport. The range of values means performance obtained for far and close views, respectively.

Model	Size	Samples (worst case)	Large-scale visibility (fps)	Large+fine scale visibility (fps)	Solar exposure (fps)
City Blocks	4K Faces	$8 \times (8+8)$	60–39	60–25	45–12
City Blocks	4K Faces	$16 \times (16+16)$	42–15	33–10	25–4
Mediterranean	604K Faces	$8 \times (8+8)$	60–34	60–21	37–11
Mediterranean	604K Faces	$16 \times (16+16)$	34–12	30–9	15–5
Urban Sprawl	777K Faces	$8 \times (8+8)$	57–27	50–18	36–10
Urban Sprawl	777K Faces	$16 \times (16+16)$	27–10	24–8	16–4

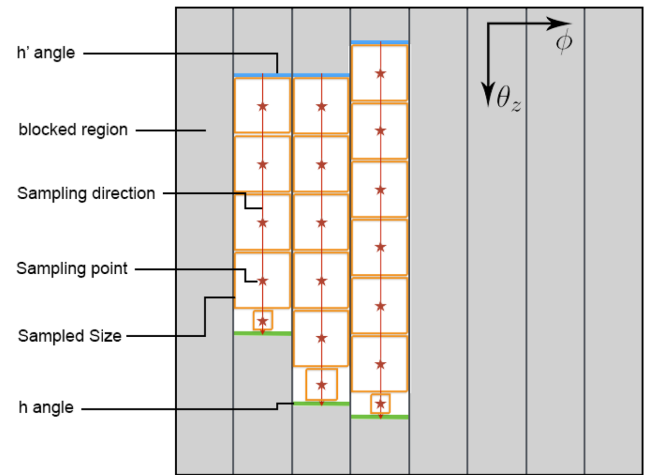


Figure 7: Sampling of the solar exposure map. We take samples along each visibility direction, adapting the number of samples to the visible region (red stars). We resort to mipmap pre-filtering to retrieve the exposure of each sample, whose size is fitted to the visible region (orange boxes).

to be very important. The combination of both a high latitude and a high turbidity leads to a low exposure (bottom-right results of *Mediterranean* model in Figure 8, for example). Also, the behaviour of common urban features, such as porches, is well handled.

Figure 9 shows different views of the visibility and solar exposure computed on the models *Mediterranean* and *Urban Sprawl* for a fixed latitude and turbidity. We can see that visibility and exposure tend to be correlated in some situations (like for some orientations in latitudes near to 0), but not for the rest: in particular, on some planar horizontal surfaces or on vertical walls oriented differently. Moreover, the same Figure 9 shows the influence of the sampling configuration used for computing sky visibility and solar exposure. One can notice that even with a low configuration, our technique is still capable of accounting for small geometric features such as vegetation that grows on the facade (see close-up in the bottom part of Figure 9).

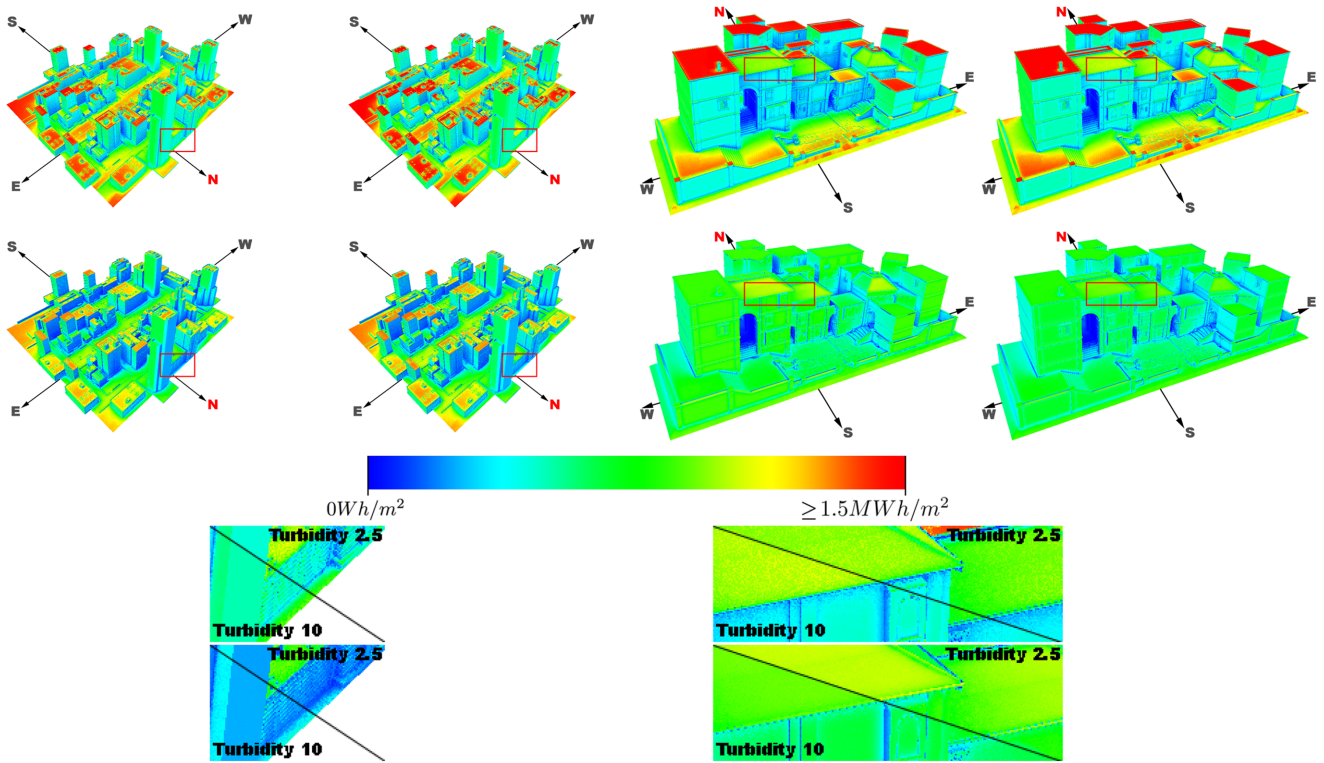


Figure 8: Solar exposure computed in the UrbanSprawl (left) and Mediterranean (right) models using 16 directions (N_{dir}), 16 samples per direction (S_{dir}) and a maximum of 16 samples for the exposure map (n_s). Latitudes on each row: $0^\circ N$ (up) and $30^\circ N$ (down) for UrbanSprawl and $0^\circ N$ (up) and $60^\circ N$ (down) for the Mediterranean. Turbidities on each column: 2.5 (left) and 10 (right) for each model. Bottom row shows close-ups for the red boxes.

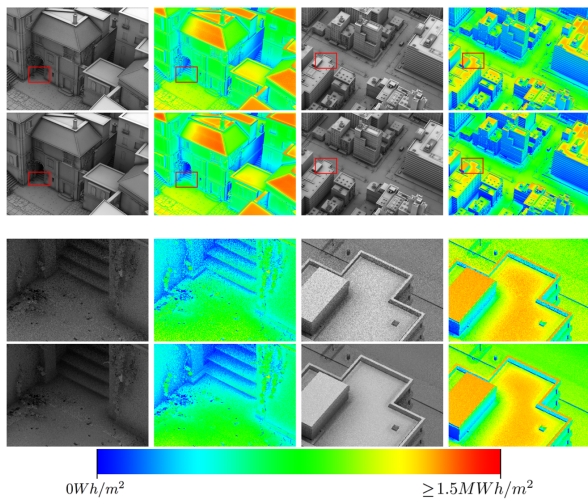


Figure 9: Visibility and solar exposure ($30^\circ N$ latitude and a 2.5 turbidity) for the Mediterranean (left) and Urban Sprawl (right) models, with their respective close-ups below, using different parameters: top, 8 directions (N_{dir}), 8 samples per direction (S_{dir}) and a maximum of 8 samples in the exposure map (n_s); bottom, 16 directions (N_{dir}), 16 samples per direction (S_{dir}) and a maximum of 16 samples in the exposure map (n_s).

In Figures 5 and 10, we compare our technique against a ground truth solution based on ray casting. Figure 5 compares the visibility term, while Figure 10 compares the final solar exposure. In the later, the ray casting approach directly samples the sky model without resorting to our pre-computed map. Main differences between the two methods are mostly due to slight differences in visibility angles, which cause the exposure to slightly shift over the surfaces (see Figure 10 left), or slightly differ on facades with salient elements such as roofs, balconies or porches (see Figure 10 middle and right). These differences are due to the resolution of the depth map, our sampling strategy, as well as our approximation of the opening angle. Our method, however, provides a good match compared to ground truth visibility and exposure overall, achieving such results in real time.

Finally, Figure 11 shows a comparison between existing tools and our method, specifically comparing our solution against *Heliodon2* and *Autodesk Revit 2017* with *Insight 360* plug-in. We have used a simple model, similar to *City Blocks*, due to the limitation to load complex geometry into *Heliodon2*. Both solutions compute the solar exposure in several points of the geometry, they store it in memory and, finally, interpolate between these sampling points. The main difference between them is the computational model. On the one hand, *Heliodon* uses its own algorithm to compute the direct beam based on ray-tracing, while the diffuse component is approximated using an isotropic atmospheric transmittance value weighted by the

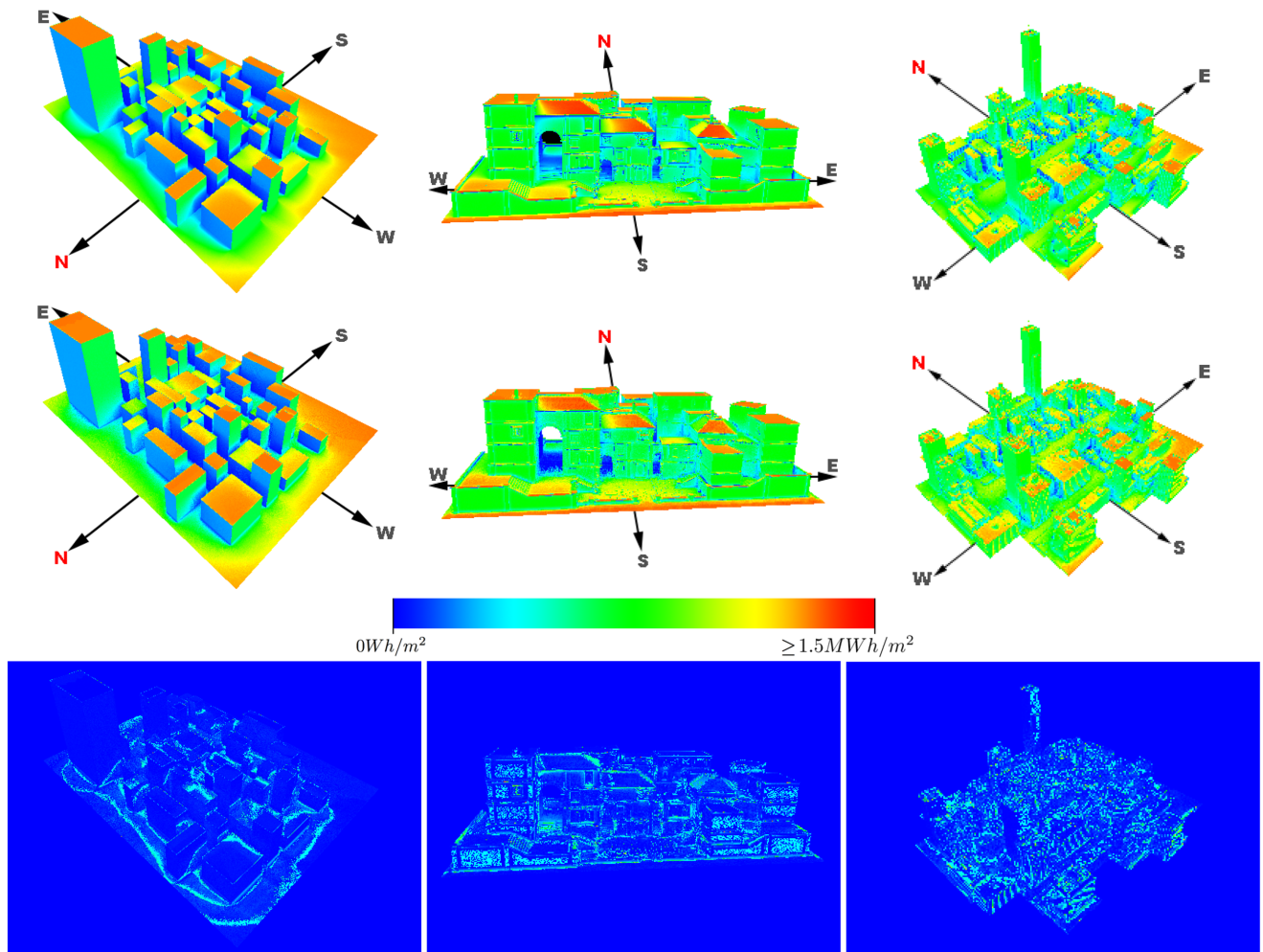


Figure 10: Solar exposure comparison between ground truth (top) and our method (centre), including difference image (bottom). Ground truth is based on ray casting, which samples the sky each hour using one ray for direct irradiance and 128 for indirect irradiance per pixel. Our method uses the pre-computed map and samples visibility using 16 directions (N_{dir}) and 16 samples per direction (S_{dir}).

SVF, the latter being computed using a projection of the neighbour buildings. On the other hand, *Autodesk Revit 2017* uses a ray-casting solution with the Preetham model, combined with measured information of the weathering conditions in near locations. Notice how the obtained results differ between the three tools as they rely on different sky models, but the patterns of solar exposure distribution fit well in all cases. In comparison, our technique requires much less computing time.

8. Discussion

Our screen-space approach for estimating visibility provides interactive results with a good match compared to ray-casted visibility, as presented in Figure 5. The large-scale visibility step is able to handle non-heightfield configurations, although it does have limitations with estimated opening angle accuracy. Features not captured by the zenithal depth map will not be detected by this stage, although the

fine-scale visibility step based on the observer's depth map tends to compensate for this. When those features are not captured by either view, then one should provide additional views or resort to depth peeling [RGS09].

The combination of the visibility information from the two scales based on a final weighting is fast and provides reasonable results in most situations. Ideally, the visible regions from the two scales should be intersected on a common frame and used during the computation of the exposure. Despite the required effort, such a strategy should compensate for over-occluded places.

As for surface exposure, our pre-filtering strategy requires a few sets of samples to be taken for each direction, which are then automatically fitted according to the visible regions. Our technique provides interactive results with a good comparison with respect to a ground truth generated in a 40 h average using ray-casting, as shown in Figure 10. Moreover, Figure 11 shows how our technique obtains good results in much shorter computation time when

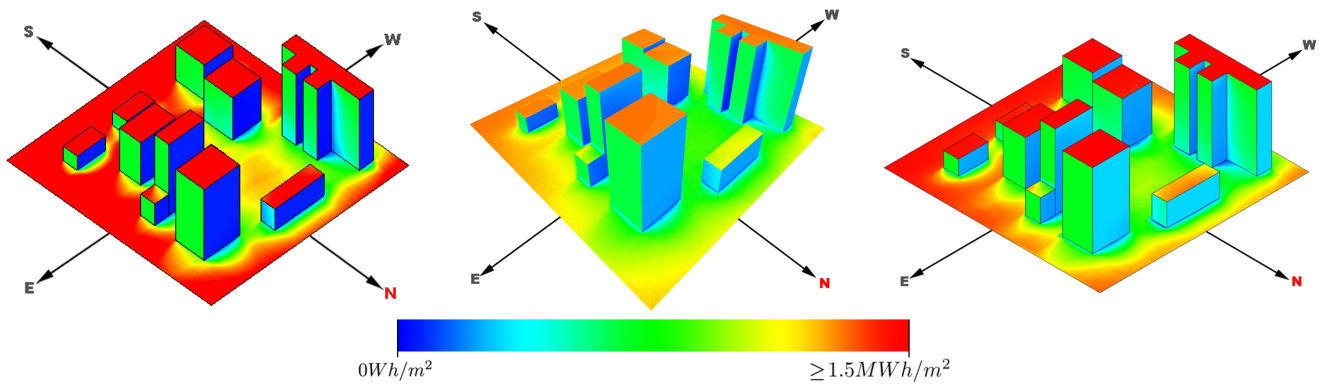


Figure 11: Comparison between existing techniques and our method at $30^\circ N$ latitude and a 2.5 turbidity. Left: Heliodon image generated in 345 s, sampling every 15 min with 1 m resolution; right: image generated with Autodesk Revit 2017 and Autodesk Insight 360 tool in 15 s, sampling every hour with 0.45 m resolution; centre: our solution generated in 0.07 s using per-pixel resolution (average in this view: 0.2 m) using 16 directions (N_{dir}), 16 samples per direction (S_{dir}) and a maximum of 16 samples in the exposure map (n_s).

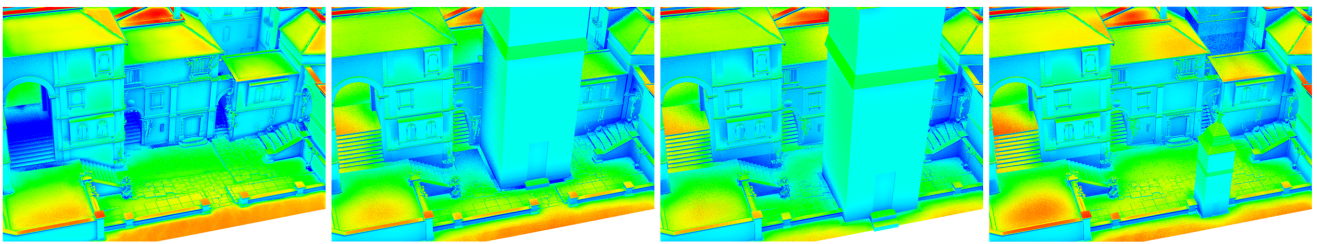


Figure 12: Frames extracted from an animation demonstrating how our technique handles dynamic scenes with interactive solar exposure.

compared to existing solutions: we achieve two orders of magnitude speed-up with respect to *Autodesk Revit* and three orders of magnitude speed-up with respect to *Heliodon*. In addition, we can handle dynamic scenes as shown in Figure 12. This option is very useful in urban planning and solar engineering in order to evaluate specific configurations of buildings or elements. An example of the relevance of the proposed technique is in the design and location of solar panels or other solar devices either for capturing and transforming solar energy or to light indoor environments with daylight. Another strategy, which could improve the results and reduce the computation time even further, would be to perform an adapted version of importance sampling, along the lines of Colbert's work [CK07], by considering not only visibility, but also the cosine and the directional exposure terms in order to guide the sampling.

Our pre-computed exposure map relies on the Hosek model to capture different sky conditions which, similar to Preetham, is restricted to continuous conditions of clear or, at most, overcast skies. Yearly conditions can be adjusted through the turbidity parameter to approximate varying conditions of cloudiness over time. For partially cloudy skies, however, one should preferably rely on measured data.

Last but not least, if changes in sky conditions or scene latitude are needed, one might pre-compute several maps to be interchanged in runtime. A better solution would be to develop specific analytical

models able to capture the integrated temporal behaviour or, at least, to approximate the diffuse component given its low contribution, in order to easily evaluate them interactively.

9. Conclusions

In this paper, we have presented an interactive method, based on sky maps integrated over arbitrary periods of time and under different conditions, to compute and evaluate solar exposure on complex urban scenes. Our system provides interactive feedback and is thus amenable to dynamic modifications of the scene as one can see in the video included. We believe that our technique can be useful in various scientific areas such as architecture and energy efficiency, in particular for interactive preview purposes prior to more accurate and costly simulations for final planning decisions. Future work is not only oriented towards solving the drawbacks discussed in the previous section, but also towards additional aspects such as indirect reflections caused by the ground and neighbouring objects; neither of which have not been considered here. Also, our two-scale screen-space techniques could be extended to handle natural phenomena other than the sun, for instance, precipitations (rain, snow, etc.), wind fields in urban scenes or weathering from environmental sources. Last but not least, we plan to improve the sky model's accuracy by introducing climatological data during the computation of the map.

Acknowledgement

The city models have been used with permission. This work was partially funded by TIN2014-52211-C2-2R project from Ministerio de Economía y Competitividad (Spain) and by the ANR under ref. ANR-16-CE33-0001 (France).

References

- [AAP12] ARGUDO O., ANDÚJAR C., PATOW G. A.: Interactive rendering of urban models with global illumination. In *Proceedings of Computer Graphics International* (2012).
- [BGR01] BROWN M., GRIMMOND S., RATTI C.: Comparison of methodologies for computing sky view factor in urban environments. In *Proceedings of the International Symposium on Environmental Hydraulics* (2001).
- [BN08] BRUNETON E., NEYRET F.: Precomputed atmospheric scattering. *Computer Graphics Forum* 27, 4 (2008), 1079–1086.
- [BS09] BAVOIL L., SAINZ M.: Multi-layer dual-resolution screen-space ambient occlusion. In *SIGGRAPH '09: SIGGRAPH 2009: Talks* (New York, NY, USA, 2009), ACM, pp. 45:1–45:1.
- [BSD08] BAVOIL L., SAINZ M., DIMITROV R.: Image-space horizon-based ambient occlusion. In *SIGGRAPH '08: ACM SIGGRAPH 2008 Talks* (New York, NY, USA, 2008), ACM, pp. 22:1–22:1.
- [CK07] COLBERT M., KŘIVÁNEK J.: GPU-based importance sampling. In *GPU Gems 3*. H. Nguyen (Ed.). Addison-Wesley (Boston, MA, USA, 2007), pp. 459–479.
- [COMM94] CHIBA N., OHKAWA S., MURAOKA K., MIURA M.: Visual simulation of botanical trees based on virtual heliotropism and dormancy break. *The Journal of Visualization and Computer Animation* 5, 1 (1994), 3–15.
- [DGA04] DESBENOIT B., GALIN E., AKKOCHE S.: Simulating and modeling lichen growth. *Computer Graphics Forum* 23, 3 (2004), 341–350.
- [DKN*94] DOBASHI Y., KANEDA K., NAKASHIMA T., YAMASHITA H., NISHITA T., TADAMURA K.: Skylight for interior lighting design. *Computer Graphics Forum* 13, 3 (1994), 85–96.
- [DKY*00] DOBASHI Y., KANEDA K., YAMASHITA H., OKITA T., NISHITA T.: A simple, efficient method for realistic animation of clouds. In *SIGGRAPH '00: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2000), ACM Press/Addison-Wesley Publishing Co., pp. 19–28.
- [DRM12] DOGAN T., REINHART C., MICHALATOS P.: Urban daylight simulation: Calculating the daylight area of urban designs. In *Proceedings of SimBuild* (2012), pp. 613–620.
- [FB12] FERNANDEZ E., BESUIEVSKY G.: Inverse lighting design for interior buildings integrating natural and artificial sources. *Computers & Graphics* 36, 8 (2012), 1096–1108. Graphics Interaction Virtual Environments and Applications 2012.
- [FCRB15] FREITAS S., CATTI C., REDWEIK P., BRITO M.: Modelling solar potential in the urban environment: State-of-the-art review. *Renewable and Sustainable Energy Reviews* 41 (2015), 915–931.
- [HW12] HOSEK L., WILKIE A.: An analytic model for full spectral sky-dome radiance. *ACM Transactions on Graphics* 31, 4 (July 2012), 95:1–95:9.
- [HW13] HOSEK L., WILKIE A.: Adding a solar-radiance function to the hošek-wilkie skylight model. *IEEE Computer Graphics and Applications* 33, 3 (2013), 44–52.
- [JW84] JOHNSON G. T., WATSON I. D.: The determination of view-factors in urban canyons. *Journal of Climate and Applied Meteorology* 23, 2 (1984), 329–335.
- [KKN*14] KIDER JR. J. T., KNOWLTON D., NEWLIN J., LI Y. K., GREENBERG D. P.: A framework for the experimental comparison of solar and skydome illumination. *ACM Transactions on Graphics* 33, 6 (November 2014), 180:1–180:12.
- [LGLI15] LIANG J., GONG J., LI W., IBRAHIM A. N.: A visualization-oriented 3d method for efficient computation of urban solar radiation based on 3d–2d surface mapping. *International Journal of Geographical Information Science* 28, 4 (2015), 780–798.
- [Lin07] LINDBERG F.: Modelling the urban climate using a local governmental geo-database. *Meteorological Applications* 14, 3 (2007), 263–273.
- [MAAB10] MERINO L., ANTALUCA E., AKINOGLU B., BECKERS B.: Solar energy inputs estimation for urban scales applications. In *Proceedings of the 8th International Conference on System Simulation in Buildings* (Liege, December 13–15, 2010).
- [MFS09] MÉNDEZ-FELIÚ À., SBERT M.: From obscurances to ambient occlusion: A survey. *The Visual Computer* 25, 2 (2009), 181–196.
- [MG08] MÉRILLOU S., GHAZANFARPOUR D.: A survey of aging and weathering phenomena in computer graphics. *Computers and Graphics (Pergamon)* 32, 2 (2008), 159–174.
- [MGVMLL14] MARQUEZ-GARCIA A., VARO-MARTINEZ M., LOPEZ-LUQUE R.: New model for the estimation of solar radiation on façades in urban environments. *Renewable Energy and Power Quality Journal* 12 (April 2014), pp. 729–734.
- [Mit07] MITTRING M.: Finding next gen: Cryengine 2. In *SIGGRAPH '07: Proceedings of the ACM SIGGRAPH 2007 Courses* (New York, NY, USA, 2007), ACM, pp. 97–121.
- [MM09] MATZARAKIS A., MATUSCHEK O.: Estimation of sky view factor in urban environments. In *Mettools VII: A Conference on Environmental Meteorology of the German Meteorological Society* (Hamburg, Germany, September 1–3, 2009), Borntraeger.

- [MM11] MATZARAKIS A., MATUSCHEK O.: Sky view factor as a parameter in applied climatology – Rapid estimation by the skyhelios model. *Meteorologische Zeitschrift* 20, 1 (February 2011), 39–45.
- [MOBH11] MCGUIRE M., OSMAN B., BUKOWSKI M., HENNESSY P.: The alchemy screen-space ambient obscurance algorithm. In *HPG '11: Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics* (New York, NY, USA, 2011), ACM, pp. 25–32.
- [MP96] MĚCH R., PRUSINKIEWICZ P.: Visual models of plants interacting with their environment. In *SIGGRAPH '96: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1996), ACM, pp. 397–410.
- [NDKY96] NISHITA T., DOBASHI Y., KANEDA K., YAMASHITA H.: Display method of the sky color taking into account multiple scattering. In *Pacific Graphics* (1996), vol. 96, pp. 117–132.
- [NMK08] NOORIAN A. M., MORADI I., KAMALI G.: Evaluation of 12 models to estimate hourly diffuse irradiation on inclined surfaces. *Renewable Energy* 33 (2008), 1406–1412.
- [NV115] What's cooking in pro graphics: How real-time ray tracing can avert a real-life “death ray” (2015). <https://blogs.nvidia.com/blog/2015/03/19/ray-tracing-death-ray>. Accessed on September 2016.
- [PSM93] PEREZ R., SEALS R., MICHALSKY J.: All-weather model for sky luminance distribution—preliminary configuration and validation. *Solar Energy* 50, 3 (1993), 235–245.
- [PSS99] PREETHAM A. J., SHIRLEY P., SMITS B.: A practical analytic model for daylight. In *SIGGRAPH '99: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 91–100.
- [REK*04] RILEY K., EBERT D. S., KRAUS M., TESSENDORF J., HANSEN C.: Efficient rendering of atmospheric phenomena. In *EGSR'04: Proceedings of the 15th Eurographics Conference on Rendering Techniques* (Aire-la-Ville, Switzerland, Switzerland, 2004), Eurographics Association, pp. 375–386.
- [RGS09] RITSCHEL T., GROSCH T., SEIDEL H.-P.: Approximating dynamic global illumination in image space. In *I3D '09: Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2009), ACM, pp. 75–82.
- [RR04] RATTI C., RICHENS P.: Raster analysis of urban form. *Environment and Planning B: Planning and Design* 31, 2 (2004), 297–309.
- [SA07] SHANMUGAM P., ARIKAN O.: Hardware accelerated ambient occlusion techniques on gpus. In *I3D '07: Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2007), ACM, pp. 73–80.
- [ŠCS11] ŠŮRI M., CEBECAUER T., SKOCZEK A.: Solargis: Solar data and online applications for pv planning and performance assessment. In *Proceedings of the 26th European Photovoltaics Solar Energy Conference* (2011).
- [SoD15] SoDa: Solar radiation Data: Solar energy services for professionals, 2015. URL: <http://www.soda-is.com/eng/index.html>. Accessed on September 2016.
- [SRNN05] SUN B., RAMAMOORTHI R., NARASIMHAN S. G., NAYAR S. K.: A practical analytic single scattering model for real time rendering. In *SIGGRAPH '05: Proceedings of the ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), ACM, pp. 1040–1049.
- [War94] WARD G. J.: The radiance lighting simulation and rendering system. In *Proceedings of the 21st Annual Conference on Computer graphics and Interactive Techniques* (1994), ACM.
- [WFA*05] WALTER B., FERNANDEZ S., ARBREE A., BALA K., DONIKIAN M., GREENBERG D. P.: Lightcuts: A scalable approach to illumination. In *SIGGRAPH '05: Proceedings of the ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), ACM, pp. 1098–1107.
- [ZIK98] ZHUKOV S., IONES A., KRONIN G.: An ambient light illumination model. In *Rendering Techniques*. G. Drettakis and N. L. Max (Eds.). Springer (1998), pp. 45–56.

Supporting Information

Additional Supporting Information may be found in the online version of this article at the publisher's web site:

Video S1

Video S2