Inter-building shading calculations based on CityGML geometric data

G.N. Lilis[1], G. Giannakis[1] D.V. Rovas[2]

[1]School of Production Engineering and Management, Technical University of Crete, Chania, Greece

[2]Institute for Environmental Design and Engineering, University College London, London, UK

Abstract

In order to account for the neighbor building shading effects into Building Energy Performance simulations, the building models should be enhanced with the geometric information of neighbor building envelope surfaces, taking into account the relative position of the neighbor buildings, with respect to the building of interest and the sun path, over a specific time interval. These surfaces, ranked by their total shading effect, are obtained by the introduced district neighbor shading function (DNS). DNS receives as input building envelope geometric representations from a CityGML file, the longitude and latitude of the district and a time interval of interest and returns the minimal set of shading buildings and surfaces, for the considered time interval. The algorithm is tested on CityGML data from a real district and it is assessed using EnergyPlus simulations.

Introduction

Recently, there is an increased interest for building energy performance simulations in a district environment following a bottom-up approach, where detailed simulation models are aggregated in order to form a larger simulation model for a whole district Reinhart and Davila (2016) Robinson (2012), justifying the transition from models of isolated buildings to models of buildings including the surrounding building information. Especially in dense city environments it has been shown that the inter-building effect becomes prevalent in energy as well as daylight simulations Li and Wong (2007), Pisello et al. (2012).

In order to capture such phenomenon and include the required data into the thermal simulation models, knowledge of the surrounding building geometries is required. Multiple data schemes for supporting such requirements have been developed, the most popular of which is the CityGML schema, which has been developed since 2008 by the Open Geospatial Consortium (OGC)Open Geospatial Consortium (2012). The shading effect of neighbor buildings have been studied before the introduction of the CityGML schema, using the shading mask concept on the sky dome Marsh (2005), and segmentations of the building's envelope surfaces using tessellations of voxel data models Hofierka and Zlocha (2012). Building shading computations on CityGML data have been attempted by several authors using ray casting on: grid points on building envelope surfaces Wieland et al. (2015) on sampled envelope surfaces using texture mapping and 3D voxel-grids for shading calculations Bremer et al. (2016), on triangulated building envelope surfaces for PV applications Alam et al. (2013). A thorough review of such methods can be found in Freitas et al. (2015). Most of these methods, are approximate as they depend on the resolution of the voxels, grid points, rays and triangles. In the present work an accurate algorithm for the calculation of building shading groups (the sets of surfaces which shade a particular building of interest at one or more time instances in a predefined time interval), for any CityGML building topology, is introduced. This algorithm is called District Neighbor Shading or DNS. DNS receives as input the geometric descriptions of the building envelopes contained in a CityGML file and returns the indexes of the surfaces of the neighbor buildings which shade a specific building at some time during a predefined time interval. The process is repeated for all the buildings in the district.

The proposed algorithm is intended to be used as a data enrichment process in the automatic simulation model generation process from IFC and CityGML data described in Lilis et al. (2016) and will be a component of a web-based platform developed for district retrofitting. During this process, a SimModel file O'Donnell (2013), will be formed from data of an existing IFC file of a building of interest (containing the buildings' second-level space boundary topology and external shading surfaces) and will be enriched with shading surfaces of neighbor buildings, obtained from the introduced DNS algorithm.

Certain initial definitions and geometric operations, used in the main algorithmic process, are described in the "preliminaries" section. The description of the DNS function follows in section "DNS algorithm". Demonstration examples, using as input CityGML data referring to a district in Santiago de Compostella in Spain, are presented after the description of the main process. Finally, limitations referring to the geometrical description of the district site and vegetation elements, are discussed in a separate section after the demonstration example. The work concludes with a discussion of future work plans referring to the different treatment of opaque and transparent building facade surfaces.

Preliminaries

A number of preliminary concepts are presented before the analysis of the main algorithmic process. These preliminary concepts are organized in a sequential manner, as described in the following sections. These concepts, have increasing complexity and each one is using the previous ones in its description. The mathematical notation presented in the fol-

lowing nomenclature section was adopted throughout the paper.

Nomenclature

Table 1: Nomenclature

Symbol	Description
P	Polygon P
$\hat{n}_P$	Normal vector of polygon P
a_P	Area of polygon P
$\mathcal{P}$	Polygon set P
j	Neighbor building index j
m	Neighbor polygon index m
$\mathbf{S} = \{(\mathcal{P}, j, m)\}$	Indexed set of polygon sets $\mathcal{P}$
$\mathbf{S}(i)$	The i^{th} triplet of set $\mathbf{S}$
$\mathbf{S}(i,j)$	The j^{th} entry of i^{th} triplet of $\mathbf{S}$
$\langle \cdot, \cdot \rangle$	Inner product notation
(c)	Polygon clipping notation
$\hat{n}_s$	Solar vector

Set operations

Three set operations are used in the present work: union $\cup$,intersection $\cap$ and difference $\setminus$. These operations are applied on regular sets, on coplanar polygons as well as on sets of coplanar polygons. In the case of coplanar and sets of coplanar polygons, these operations are implemented using the algorithm of Vatti (1992) and return polygon sets in general.

District geometric description

A district defined in a CityGML file contains a geometric description of a collection of finite buildings ($i = 1, ..., N$), the envelopes of which can be defined geometrically as sets of polygons:

$$\mathcal{P}_i = \{P_1^{(i)}, ..., P_{M_i}^{(i)}\}$$

where M_i the total number of envelope polygons of building i. Consequently, in the present work the geometric description of the district is a superset (set of polygon sets) $\mathbf{D}$ containing all the polygon sets $\mathcal{P}_i$:

$$\mathbf{D} = \{\mathcal{P}_1, ..., \mathcal{P}_N\}$$

Additionally, for every envelope polygon set $\mathcal{S}_i$ of building i, the set of envelope polygons of all the other buildings in the district except from building i is defined as the complement envelope polygon set of building i:

$$\mathcal{P}_i^c = \left\{P_m^{(j)}, j \neq i, j \in \{1, ..., N\}, m \in \{1, ..., M_j\}\right\}$$

Solar position function

The solar position function SolPos receives as input a simulation time instance t, the day duration in simulation instances D_d and the longitude (Lon)/latitude (Lat) of a district of interest and returns the two angles which determine the position of the sun disc on the sky dome: the solar azimuth angle $\hat{\phi}_s$ and the solar elevation angle $\hat{\theta}_s$, as described in the following expression:

$$[\phi_s, \theta_s] = \text{SolPos}(t, D_d, Lat, Lon)$$

Given a day duration D_d the possible simulation time instances are integers $t \in \{1, ..., 365 \times D_d\}$.

Solar vector

The solar vector $\hat{n}_s$, is defined as the unitary vector passing from the origin (0,0,0) and pointing to the sun disk on the sky dome. The solar vector is obtained from the solar azimuth and elevation angles as follows:

$$\hat{n}_s = [\cos(\phi_s)\cos(\theta_s) \quad \sin(\phi_s)\cos(\theta_s) \quad \sin(\theta_s)]$$

Solar plane

The solar plane is defined as the plane passing from the origin (0,0,0), which has normal vector equal to the solar vector $\hat{n}_s$, which is calculated using the solar position function defined previously.

Direct solar irradiance function

The intensity of the sun rays and as a result the impact of the shading of neighbor building surfaces depend on the sun position on the sky dome and the time of the year. Consequently, in order to include the sun ray intensity into the introduced algorithm, the direct solar irradiance function, is defined. The direct solar irradiance function receives as input the solar elevation angle θ_s and the day of the year $d = \{1, ..., 365\}$ and returns the solar irradiance on the solar plane according to the following equation.

$$s_{ir} = \mathrm{D_r}(\theta_s, d)$$

Essentially the direct solar radiation function calculates the direct normal irradiance under clear sky conditions, based on a specific solar calculation method. In the present work the ASHRAE clear sky model ASHRAE (2007) is adopted, as the solar calculation method.

Polygon solar projection function

The polygon solar projection function is the function which receives a polygon in three dimensions P and the normal vector of a solar plane $\hat{n}_s$ and returns an output polygon P_p which is the projection of P onto the solar plane. Mathematically this function can be expressed as:

$$P_p = \mathrm{F_{pp}}(P, \hat{n}_s)$$

Envelope solar projection function

The envelope projection function essentially returns the union of the polygon solar projections $\mathcal{P}_{pi}$ of the envelope polygons of building i $\mathcal{P}_i$, onto the solar plane.

$$\mathcal{P}_{pi} = \mathrm{F_{ep}}(\mathcal{P}_i, \hat{n}_s), \qquad \mathcal{P}_{pi} = \bigcup_n \mathrm{F_{pp}}(P_n^{(i)}, \hat{n}_s)$$

Area function

The area function receives as input either a polygon P and returns its surface area A or a polygon set $\mathcal{S}$ and returns the sum of the areas of its polygons:

$$a_P = \mathrm{Area}(P), \qquad a_{\mathcal{P}} = \sum_p \mathrm{Area}(\mathcal{P}(p))$$

Polygon shading function

The purpose of the polygon shading function is to determine, given a sun position defined by $\hat{n}_s$, the indexes of polygons contained in a complement envelope polygon set $\mathcal{P}_i^c$, which shade a given polygon $P_n^{(i)}$ from the set $\mathcal{P}_i$ and the intersection of their solar projections with the solar projection of $P_n^{(i)}$. This function returns the intersections of the projected shading polygons $P_m^{(j)}$ with the projection of $P_n^{(i)}$ in an indexed set of polygon sets $\mathbf{S}_{out}$ containing the polygon intersection sets accompanied with the indexes of the shading polygons j, m. The operations of this function are described in Algorithm 1.

A prerequisite for an envelope surface $P_n^{(i)}$ to have shading surfaces, is to face towards the direction of the sun. This is captured by the positive inner product condition $\langle \hat{n}_P, \hat{n}_s \rangle > 0$ in Algorithm 1.

The polygon shading function is implemented into two phases: the solar projection and the intersection phase. During the solar projection phase the neighbor envelope surfaces (green polygons in Figure 1 A) are clipped by the envelope polygon $P_n^{(i)}$. During this clipping operation (c), the plane of $P_n^{(i)}$ dissects the neighbor envelope polygon $P_m^{(j)}$ into two parts: one part in the half space pointed by the normal vector of the clipping polygon ($\hat{n}_{P_n^{(i)}}$) and another part pointed by the vector which is opposite to the normal vector of the clipping polygon. The first part is returned by the clipping operation as a clipped polygon set $\mathcal{P}_{clp}$ (blue polygons in Figure 1 A). The clipping operation is performed in order to exclude neighbor building surfaces which are behind the considered envelope surface $P_n^{(i)}$ of building i. The solar projection phase resumes with the projection P_p of the clipping polygon $P_n^{(i)}$ and the projection P_{pc} of all the clipped envelope polygons of the set $\mathcal{P}_{clp}$ onto the solar plane using the solar projection function described earlier.

During the intersection phase the projection of the clipping polygon P_p (Figure 1 B) is intersected with the projections of the clipped envelope polygons of the set $\mathcal{P}_{pc}$ (Figure 1 B). Finally, the resulting polygon set $\mathcal{P}_{ip}$ (Figure 1 B) from these intersections, is stored into the output polygon set $\mathcal{S}_{out}$ along with the indexes of the neighbor building envelope surfaces (j, m) (Algorithm 1).

Algorithm 1 : Polygon shading function
$\mathbf{S}_{out} = \mathrm{F_{ps}}(P_n^{(i)}, \mathcal{P}_i^c, \hat{n}_s)$

$\mathbf{S}_{out} = \emptyset$ <Output indexed set of polygon sets>
if $\langle \hat{n}_P, \hat{n}_s \rangle > 0$ **then**
  $P_p = \mathrm{F_{pp}}(P_n^{(i)}, \hat{n}_s)$ <Projection of partition polygon>
  for $j = 1 : N, j \neq i$ **do**
    for $m = 1 : M_j$ **do**
      if $P_n^{(i)}, P_m^{(j)}$ are not coplanar **then**
        $\mathcal{P}_{clp} = P_n^{(i)}$ (c) $P_m^{(j)}$ <Clipped polygon set>
        $\mathcal{P}_{int} = \emptyset$ <$\mathcal{P}_{int}$ initialization>
        for $p_1 = 1 : |\mathcal{P}_{clp}|$ **do**
          $P_{pc} = \mathrm{F_{pp}}(\mathcal{P}_{clp}(p_1), \hat{n}_s)$ <Projected clipped>
          $\mathcal{P}_{ip} = P_p \cap P_{pc}$ <Intersection of projections>
          for $p_2 = 1 : |\mathcal{P}_{ip}|$ **do**
            $\mathcal{P}_{int} \leftarrow \mathcal{P}_{int} \cup \{\mathcal{P}_{ip}(p_2)\}$ <Update $\mathcal{P}_{int}$>
          end for
        end for
        if $\mathcal{P}_{int} \neq \emptyset$ **then**
          $\mathbf{S}_{out} \leftarrow \mathbf{S}_{out} \cup (\mathcal{P}_{int}, j, m)$
        end if
      end if
    end for
  end for
end if

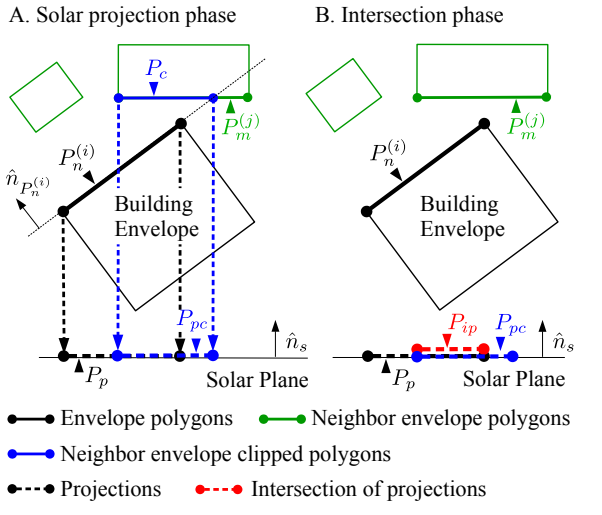

Figure 1: Illustration of operation of polygon shading function: Solar projection phase (A) and Intersection phase (B).

Polygon set merging

The intersection polygon sets contained in the set $\mathbf{S}_{out}$ and obtained by applying the polygon shading function on two different envelope polygons of a

building of interest i, might involve the same neighbor building envelope polygon, as illustrated in Figure 2. In these cases and if the intersection polygons share common edges, they must be merged using the polygon union function. Such operations are performed by the polygon set merging function described in Algorithm 2.

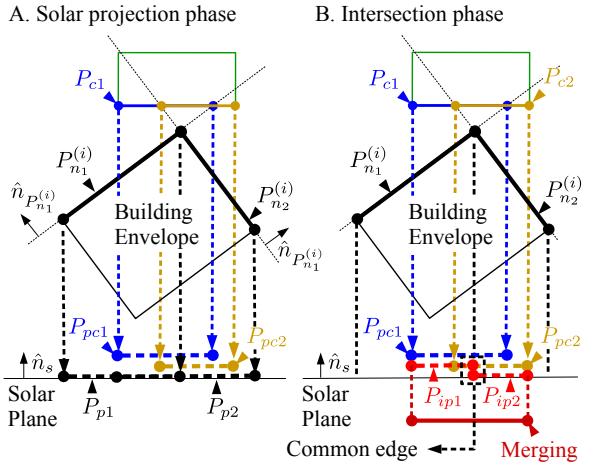

Figure 2: Illustration of merging of the intersections P_{ip1} and P_{ip2} of the solar projections of polygons $P_{n1}^{(1)}$ (P_{p1}) and $P_{n2}^{(1)}$ (P_{p2}), with the solar projections of the clipped polygons P_{c1} (p_{pc1}) and P_{c2} (P_{pc2}) of the same neighbor building surface $P_m^{(j)}$.

Algorithm 2 : Polygon merging function $\mathbf{S}'_{out} = \text{Mrg}(\mathbf{S}_{out})$

$\mathbf{S}'_{out} = \emptyset$ <Initialization of output set of polygon sets>
$s_1 = 1$
while $s_1 < |\mathbf{S}_{out}|$ **do**
  $\mathcal{P}_u = \mathbf{S}_{out}(s_1, 1)$
  $j = \mathbf{S}_{out}(s_1, 2), \quad m = \mathbf{S}_{out}(s_1, 3)$
  $s_2 = s_1 + 1$
  while $s_2 < |\mathbf{S}_{out}|$ **do**
    if $j = \mathbf{S}_{out}(s_2, 2)$ **and** $m = \mathbf{S}_{out}(s_2, 3)$ **then**
      $\mathcal{P}_u \leftarrow \mathcal{P}_u \cup \mathbf{S}_{out}(s_2, 1)$ <Merging operation>
      $\mathbf{S}_{out} \leftarrow \mathbf{S}_{out} \backslash \mathbf{S}_{out}(s_2)$ <Remove $\mathbf{S}_{out}(s_2)$ >
      $s_2 = s_2 - 1$
    end if
    $s_2 = s_2 + 1$
  end while
  $\mathbf{S}'_{out} \leftarrow \mathbf{S}'_{out} \cup (\mathcal{P}_u, j, m)$ <Update set of surface sets>
  $s_1 = s_1 + 1$
end while

The polygon sets $\mathbf{S}_{out}(s_1, 1)$ and $\mathbf{S}_{out}(s_2, 1)$ in the set $\mathbf{S}_{out}$ are merged if the respective building and envelope polygon indexes $j_1 = \mathbf{S}_{out}(s_1, 2)$, $m_1 = \mathbf{S}_{out}(s_1, 3)$ and $j_2 = \mathbf{S}_{out}(s_2, 2)$, $m_2 = \mathbf{S}_{out}(s_2, 3)$ are the same ($j_1 = j_2$ and $m_1 = m_2$) and the polygon sets intersect: $\mathbf{S}_{out}(s_1, 1) \cap \mathbf{S}_{out}(s_2, 1) \neq \emptyset$. If the polygon sets intersect they are merged in a union polygon set $\mathcal{P}_u$. The output of the merging operation is a set containing the merged sets (union polygon sets $\mathcal{P}_u$) $\mathbf{S}'_{out}$ along with the respective common indexes.

Maximum shading function

The surface of the sets of $\mathbf{S}'_{out}$ (output of the merging operation described previously), do not refer to the minimal set of envelope polygon surfaces which shade a particular building of interest at some moment in time, because overshadowing might occur, i.e the shadows of two or more shading polygons overlap as illustrated in figure 3. Consequently the surfaces of the sets $\mathbf{S}'_{out}$ with the maximum shading effect must be isolated first and any overshadowing surfaces in $\mathbf{S}'_{out}$ should be removed. The identification of the surface set of $\mathbf{S}'_{out}$ with the maximum shading effect is performed using the max shading function described by the following equation.

$$[m_s, \text{Rad}, \mathcal{R}] = \text{Msh}(\mathbf{S}'_{out}, \mathcal{R}, s_{ir})$$

Essentially the maximum shading function (Msh), described by algorithm 3, returns the index m_s of the set of polygons $\mathbf{S}'_{out}(m_s, 1)$ (first entry of the m_s element of the set $\mathbf{S}'_{out}$), which has the maximum shading effect over a surface set $\mathcal{R}$ formed by the projections of the envelope polygons of the building of interest onto the solar plane. This is achieved by populating the set $\mathbf{R}$ of remnant polygons sets and calculating the vector A formed by the total areas of the polygons contained in each of its elements ($A(s) = Area(\mathbf{R}(s, 1))$). The polygons of the s^{th} element of $\mathbf{R}$ (set $\mathbf{R}(s)$), are obtained by subtracting from the polygons of the set $\mathcal{R}$, the polygons of set $\mathbf{S}'_{out}(s, 1)$. As indicated in algorithm 3, the index m_s of the polygon sets of $\mathbf{S}$ with the maximum shading effect is equal to the index of the element of vector A with the minimum value because, the maximum shading effect is equivalent with the minimum not shaded (remnant) area.

Algorithm 3 : Maximum shading function $[m_s, \text{Rad}, \mathcal{R}] = \text{Msh}(\mathbf{S}'_{out}, \mathcal{R}, s_{ir})$

$\mathbf{R} = \emptyset$ <Indexed set of remnant polygon sets>
$A_R = Area(\mathcal{R})$
for $s = 1 : |\mathbf{S}'_{out}|$ **do**
  $\mathbf{R}(s, 1) = \mathcal{R} \backslash \mathbf{S}'_{out}(s, 1)$
  $\mathbf{R}(s, 2) = \mathbf{S}'_{out}(s, 2)$
  $\mathbf{R}(s, 3) = \mathbf{S}'_{out}(s, 3)$
  $A(s) = Area(\mathbf{R}(s, 1))$ <Area vector A population>
end for
$[A_m, m_s] = \min(A)$ <Min value of A and its index m_s >
$\text{Rad} = s_{ir}(A_R - A_m)$
$\mathcal{R} \leftarrow \mathbf{R}(m_s, 1)$ <Update set $\mathcal{R}$ >

As a final step, Msh function returns the amount of solar radiation blocked by the surfaces of $\mathbf{S}'_{out}(m_s, 1)$ and updates the surface set $\mathcal{R}$ with the remnant of $\mathcal{R}$ after the surfaces of $\mathbf{S}'_{out}(m_s, 1)$ are subtracted.

The use of the maximum shading process is illustrated in figure 4 where it is applied iteratively on a polygon set $\mathcal{R}$ and an indexed set of polygon sets (lying on the solar plane) $\mathbf{S}'_{out}$ containing three elements ($\mathbf{S}'_{out}(s_1), \mathbf{S}'_{out}(s_2)$ and $\mathbf{S}'_{out}(s_3)$). In each step the index s^* is returned by the maximum shading function and it is the index which minimizes the area of the subtraction polygon set $\mathcal{R}\backslash\mathbf{S}'_{out}(s^*, 1)$. Finally, the set $\mathcal{R}$ is updated by the difference: $\mathcal{R}\backslash\mathbf{S}'_{out}(s^*, 1)$.

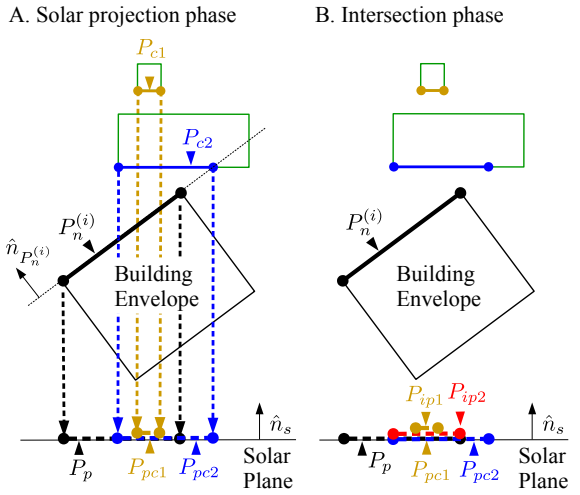

Figure 3: Illustration of overshadowing: Solar projection phase (A) and Intersection phase (B).

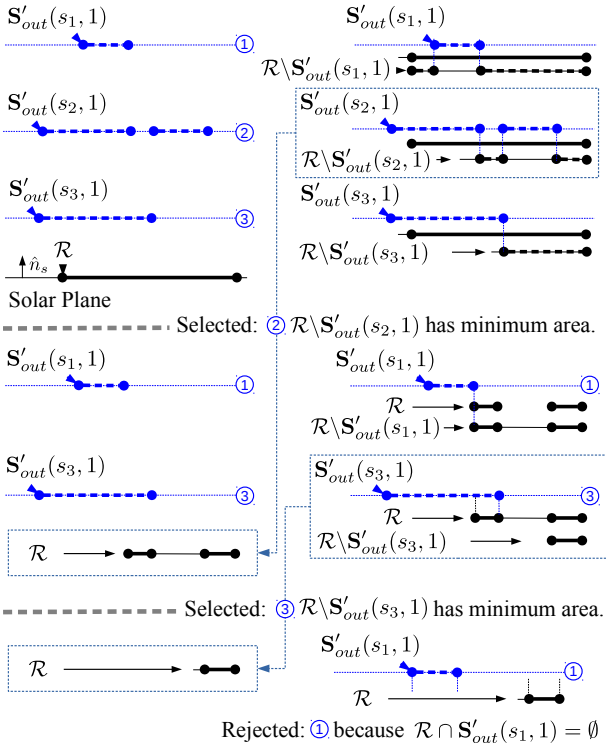

Figure 4: Illustration of maximum shading process. In each step the polygon set $\mathbf{S}'_{out}(s^*)$ which minimizes the area of $\mathcal{R}\backslash\mathbf{S}'_{out}(s^*, 1)$ and the set $\mathcal{R}$ is updated by the set $\mathcal{R}\backslash\mathbf{S}'_{out}(s^*, 1)$

In case the polygons in $\mathcal{R}$ and $\mathbf{S}'_{out}(s^*, 1)$ do not intersect the index s^* is not retained. The output of this iterative process is a sequence of indexes s which refer to polygons $\mathbf{S}'_{out}(s^*, 1)$ with a decreasing shading effect on the solar projection $\mathcal{R}$.

Envelope shading function

The envelope shading set of building i ($\mathcal{IR}_i$), is defined as the minimal set of triplets (j, m, Rad), where each triplet contains: the building (j) and envelope (m) indexes of the surfaces of $\mathcal{P}_i^c$, which block Rad amount of solar energy to the envelope polygons of building i (contained in $\mathcal{P}_i$), at a specific time instant, with the sun positioned at a point in the sky dome, indicated by the solar vector $\hat{n}_s$. The set $\mathcal{IR}_i$ is obtained by the envelope shading set function $\mathrm{F_{es}}$ as described next.

A neighbor envelope polygon from $\mathcal{P}_i^c$ shade a polygon from $\mathcal{P}_i$ if their respective projections on the solar plane intersect. The area of this intersection is proportional to the shading effect. In order calculate the minimal polygon shading subset of $\mathcal{S}_i^c$ and eliminate overshadowing effects, a remaining polygon set $\mathcal{R}$ is introduced and initialized as the union of the projections of the envelope polygons of building i on the solar plane. Then for every envelope polygon $P_n^{(i)}$ of $\mathcal{P}_i$ the polygon shading process is applied, in order to collect: (a) the indexes shading polygons of $\mathcal{P}_i^c$ and (b) the intersections of their solar projections with the solar projections of the envelope polygons $P_n^{(i)}$ in an indexed set of polygon sets $\mathbf{S}_{out}$. The set $\mathbf{S}_{out}$ obtained by collecting the outputs of the polygon shading function $\mathrm{F_{ps}}$ applied on all the polygons $P_n^{(i)}$ of $\mathcal{P}_i$) are merged (by applying merging function described earlier), in order to form a new set of polygon sets $\mathbf{S}'_{out}$.

After the formation of the merged indexed set of polygon sets $\mathbf{S}'_{out}$, the output minimal set of triplets $\mathcal{IR}_i$ is formed as a final step. $\mathcal{IR}_i$ contains triplets in the form (j, m, Rad) where: (j, m) are some index pairs of from the $\mathbf{S}'_{out}(:, 2)$ and $\mathbf{S}'_{out}(:, 3)$ entries, referring to envelope polygons of buildings neighbor to building i, and Rad is the amount of solar radiation these polygons block. These triplets are obtained by executing the maximum shading function (Msh) multiple times (one triplet every Msh execution), as follows. Every Msh execution returns the index m_s of the maximum shading polygon set of $\mathbf{S}'_{out}$. Using this index, the respective index pair (the m_s entries $\mathbf{S}'_{out}(m_s, 2)$ and $\mathbf{S}'_{out}(m_s, 3)$), referring to an envelope surface of a neighbor building ($\mathbf{S}'_{out}(m_s, 2)$-neighbor building, $\mathbf{S}'_{out}(m_s, 3)$-surface), is collected in the output set $\mathcal{IR}_i$ together with the amount of solar radiation this surface blocks, which is also returned by the Msh function. Finally the m_s element is removed from the set $\mathbf{S}'_{out}$ in order to be used for the next Msh execution.

The multiple executions of Msh reduces the remaining polygon set $\mathcal{R}$ by polygon set subtraction. The

executions of Msh terminate when either the remaining set of polygons $\mathcal{R}$ becomes empty or the polygons sets of $\mathbf{S}'_{out}$ are exhausted. These operations are described in final while loop of algorithm 4.

Algorithm 4 : Envelope shading function
$\mathcal{IR}_i = \mathrm{F_{es}}(\mathcal{P}_i, \mathcal{P}_i^c, s_{ir}, \hat{n}_s)$

$\mathcal{IR}_i = \emptyset$ <Initialize index shading set for building i>
$\mathcal{R} = \mathrm{F}_{ep}(\mathcal{P}_i, \hat{n}_s)$ <Remaining polygon set initialization>
$\mathbf{S}_{out} = \emptyset$ <Initialize intersection projection set>
for $p = 1 : |\mathcal{P}_i|$ **do**
  $\mathbf{S}_{out} \leftarrow \mathbf{S}_{out} \cup [\mathrm{F_{ps}}(P_i(p), \mathcal{P}_i^c, \hat{n}_s)]$ <Shading of $\mathcal{P}_i(p)$>
end for
$\mathbf{S}'_{out} = \mathrm{Mrg}(\mathbf{S}_{out})$ <Merging>
while $(\mathcal{R} \neq \emptyset) \wedge (\mathbf{S}'_{out} \neq \emptyset)$ **do**
  $[m_s, \mathrm{Rad}, \mathcal{R}] = \mathrm{Msh}(\mathbf{S}'_{out}, \mathcal{R}, s_{ir})$
  $\mathcal{IR}_i \leftarrow \mathcal{IR}_i \cup (\mathbf{S}'_{out}(m_s, 2), \mathbf{S}'_{out}(m_s, 3), \mathrm{Rad})$
  $\mathbf{S}'_{out} \leftarrow \mathbf{S}'_{out} \backslash \mathbf{S}'_{out}(m_s)$
end while

DNS algorithm

After the definition of the preliminary functions the district neighbor shading (DNS) function can be introduced. This function receives as input: the geometric definition of a district **D** queried from a CityGML file, an index i of a building of interest, the day duration in simulation instances D_d, a simulation time window measured in days T_w (during the simulation window - which has a default value of 10 days - the solar path does not change significantly), and the longitude (Lon) / latitude (Lat) of the district.
The DNS function returns: the minimal set $\mathcal{I}_i$ of the triplets (j, m,Rad), containing pairs of indexes of neighbor building envelope polygons j, m which block Rad solar energy amount from building i, during certain time periods in the time interval defined by the day duration D_d and the simulation time window T_w. As an additional out put the DNS algorithm returns the minimal set $\mathcal{E}_i$ of pairs (j, Rad), where j are building envelope indexes which shade the particular building of interest i and Rad is the amount of solar energy these buildings block, in the considered time interval. The entries of the output sets $\mathcal{I}_i$ and $\mathcal{E}_i$ are sorted with respect to their shading effect.
Initially DNS function evaluates the number of time windows n_w during a whole year from:

$$n_w = \lfloor \frac{365}{T_w} \rceil + 1$$

Then, the set of simulation instances $\mathcal{T}$ is calculated as a union of sets of time instances which belong to separate time intervals (one for each set of instances), according to the following expression:

$$\mathcal{T} = \bigcup_{w=1:n_w} \{1 + D_d(T_w(w-1)), ..., D_d(1 + T_w(w-1))\}$$

According to figure 5, the set $\mathcal{T}$ contains n_w time intervals which have duration of one day and they are T_w days apart. Each of the time interval contains D_d time instances which are $\frac{24}{D_d}$ hours apart.

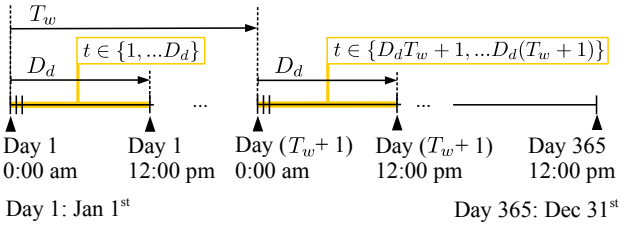

Figure 5: Illustration of the simulation times set $\mathcal{T}$ on a time line.

After the generation of $\mathcal{T}$, DNS populates the output set of index pairs $\mathcal{I}_i$, by calling $\forall t \in \mathcal{T}$ the shading set function $\mathrm{F_{es}}$ and unifying its respective output sets of triplets (j, m,Rad), as described in Algorithm 5.
The formed set of triplets $\mathcal{I}_i$ is then merged using the following rule. If the set $\mathcal{I}_i$ contain multiple triplets (obtained from different simulation instances), which have the same building and surface indexes (the first two j, m entries) and different radiation energy values (third entries Rad), then these triplets are replaced by a single triplet containing the common index j, m values (first two entries) and as radiation value (third entry), the sum of the radiation values of the initial triplets.
Finally the triplets of the merged set $\mathcal{I}_i$ are sorted with respect to the third entry, using the process $\mathrm{Srt_I}$ which is similar to the sorting operation applied on the triplets of the indexes of the shading surfaces ($\mathcal{I}'_{out}$) described earlier, with the area value (a) replaced by the radiation value (Rad).
When the sorting operation of $\mathcal{I}_i$ is completed the entries of $\mathcal{I}_i$ referring to the same building envelope (with the same first entry) are collected and their energy blocking quantities are summed together.
The resulted common building envelope indexes and the respective energy summations are collected as pairs (j, Rad) in a new output set $\mathcal{E}_i$. Finally, the set $\mathcal{E}_i$ is sorted using the $\mathrm{Srt_E}$ function, with respect to the second entry of its pairs (Rad) and is returned as the second output of the DNS algorithm.
The second output set $\mathcal{E}_i$ is generated in order to rank the shading effect of whole building envelopes to the building of interest i, since there are cases that although the shading effect of a single isolated building envelope polygon might be smaller than the total shading effect of the whole envelope of another building.
Although, the accuracy of the shading estimation on the solar radiation calculations depend on the level of detail of the CityGML data Biljecki et al. (2016), the introduced algorithm is can be extended to higher than 2 levels of detail of CityGML data, by collecting all the polygons of the building envelopes into separate polygon sets $\mathcal{P}_i$.

Algorithm 5 : District Neighbor Shading function $[\mathcal{I}_i, \mathcal{E}_i] = \mathrm{DNS}(\mathbf{D}, i, D_d, T_w, Lat, Lon)$

$\mathcal{I}_i = \emptyset$ <Initialize index set>
$n_w = \lfloor \frac{365}{T_w} \rfloor + 1$ <Number of simulation windows>
$\mathcal{T} = \bigcup_{w=1:n_w} \{1 + D_d(T_w(w-1)), ..., D_d(1 + T_w(w-1))\}$
– –
< Population of $\mathcal{I}_i$ >
for all $t \in \mathcal{T}$ **do**
  $[\phi_s, \theta_s] = \mathrm{SolPos}(t, D_d, Lat, Lon)$ <Solar angles>
  $d = \lfloor \frac{t}{D_d} \rfloor + 1$ <Day index>
  $s_{ir} = \mathrm{D_r}(\theta_s, d)$ <Solar irradiance>
  $\hat{n}_s = [cos(\phi_s)cos(\theta_s) \;\; sin(\phi_s)cos(\theta_s) \;\; sin(\theta_s)]$
  $\mathcal{IR}_i = \mathrm{F_{es}}(\mathcal{P}_i, \mathcal{P}_i^c, s_{ir}, \hat{n}_s)$ <Envelope shading set>
  $\mathcal{I}_i \leftarrow \mathcal{I}_i \cup \mathcal{IR}_i$ <Update $\mathcal{I}_i$ set>
end for
– –
<Merging of $\mathcal{I}_i$ >
$\mathcal{I}_t \leftarrow \mathcal{I}_i$ <Temporary storage of $\mathcal{I}_i$>
$\mathcal{I}_i \leftarrow \emptyset$ <Initialize new $\mathcal{I}_i$>
while $\mathcal{I}_t \neq \emptyset$ **do**
  $j = \mathcal{I}_t(1,1),\; m = \mathcal{I}_t(1,2),\; \mathrm{Rad} = \mathcal{I}_t(1,3)$
  $k = 0$
  while $k \leq |\mathcal{I}_t|$ **do**
    if $(\mathcal{I}_t(k,1) = j) \wedge (\mathcal{I}_t(k,2) = m)$ **then**
      $\mathrm{Rad} = \mathrm{Rad} + \mathcal{I}_t(k,3)$ <Update Rad>
      $\mathcal{I}_t \leftarrow \mathcal{I}_t / \mathcal{I}_t(k)$
      $k = k - 1$
    end if
    $k = k + 1$
  end while
  $\mathcal{I}_i \leftarrow \mathcal{I}_i \cup (j, m, \mathrm{Rad})$ <Update $\mathcal{I}_i$>
  $\mathcal{I}_t \leftarrow \mathcal{I}_t / \mathcal{I}_t(1)$
end while
– –
<Sorting of $\mathcal{I}_i$>
$\mathcal{I}_i \leftarrow \mathrm{Str_I}(\mathcal{I}_i)$
– –
<Population of $\mathcal{E}_i$>
$\mathcal{E}_i = \emptyset$
$\mathcal{I}_t \leftarrow \mathcal{I}_i$ <Temporary storage of $\mathcal{I}_i$>
while $\mathcal{I}_t \neq \emptyset$ **do**
  $j = \mathcal{I}_t(1,1),\; \mathrm{Rad} = \mathcal{I}_t(1,3)$
  $k = 0$
  while $k \leq |\mathcal{I}_t|$ **do**
    if $\mathcal{I}_t(k,1) = j$ **then**
      $\mathrm{Rad} = \mathrm{Rad} + \mathcal{I}_t(k,3)$ <Update Rad>
      $\mathcal{I}_t \leftarrow \mathcal{I}_t / \mathcal{I}_t(k)$
      $k = k - 1$
    end if
    $k = k + 1$
  end while
  $\mathcal{E}_i \leftarrow \mathcal{E}_i \cup (j, \mathrm{Rad})$ <Update $\mathcal{E}_i$>
  $\mathcal{I}_t \leftarrow \mathcal{I}_t / \mathcal{I}_t(1)$
end while
– –
<Sorting of $\mathcal{E}_i$>
$\mathcal{E}_i \leftarrow \mathrm{Str_E}(\mathcal{E}_i)$

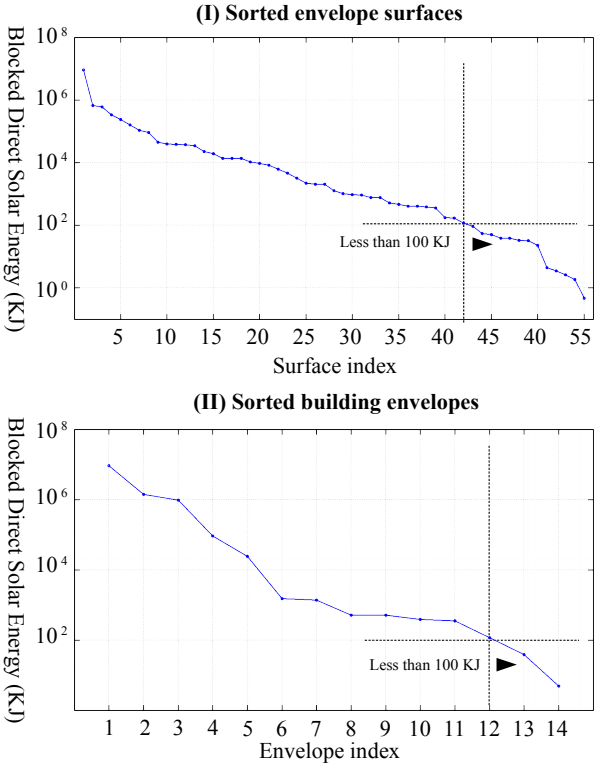

Figure 6: Sorted envelope surfaces (I) and building envelopes (II) by the amount of the blocked direct solar energy, which shade the building of interest.

Example

The algorithm is applied on the geometric data contained in a CityGML file referring to a district in the city of Santiago de Compostella in Spain (Figure 7). These data define 67 building envelopes in Level of detail (LoD) 2 ranging from residential buildings to public buildings. Out of these buildings, a central one is selected for demonstration purposes, as a building of interest (indicated with cyan color in Figure 7) and is replaced by a rectangular shape three story building with multiple openings placed in each of the four facades, in order to probe the solar radiation received from all possible azimuth angles.

The main algorithm is then applied on the selected building with the following input parameter values: day duration $D_d = 24$ (hour simulation time step), time window $T_w = 10$, latitude $Lat = [42^o\ 53'\ 14.8128'']$ and longitude $Lon = (-1) * [8^o\ 32'\ 44.614'']$. After applying the proposed algorithm on the 561 neighbor envelope surfaces of the district 55 surfaces are obtained as shading surfaces which block the amounts of direct solar energy presented in a decreasing order in plot (I) of figure 6. These amounts are extracted from the output set $\mathcal{I}_i$. As it is also reported in the output set $\mathcal{E}_i$, out of the 67 total building envelopes only 14 have a shading effect on the building of interest. These 14 building envelopes are also indicated as envelopes with at least one black surface (shading surface) in Case I of figure 7. These envelopes block the amounts of direct solar

energy presented in a decreasing order in the plot II of figure 6.

For simplicity, only 8 scenarios were simulated with EnergyPlus using the automatic simulation model generation process described in Lilis et al. (2016): A scenario where all 561 envelope surfaces are considered, a scenario with the 55 shading surfaces of the building of interest obtained from the algorithm and 10 scenarios where 5 surfaces with the smallest shading effect are removed consecutively from the set of 55 surfaces, resulting to scenarios with 50,45,40,35,30,25,20,15,10 and 5 surfaces.

The performance results of previous twelve scenarios are presented in table 2. Two quantities were extracted from the EnergyPlus simulations' output: the total beam incident radiation on an external building surface facing east at the ground floor of the building (S) in Watts and the total transmitted solar radiation energy through the windows to the 6 zones of the test building (Z) in Joules. These quantities were estimated every one hour for a whole year resulting to a total of 365x24=8760 values. For these values the mean absolute percentage error (MAPE) was used as a performance measure and the scenario with all 561 envelope surfaces present, as a reference. For the above quantities two performance metrics $\mathrm{MAPE_s}$ and $\mathrm{MAPE_z}$ were defined and calculated according to the following expression:

$$\mathrm{MAPE_x} = \frac{100}{|I|} \sum_{t \in I} \left| \frac{X_t^{(N)} - X_t^{(561)}}{X_t^{(561)}} \right| \%$$

where X is referring to the measured quantity (S for the incident solar radiation on east facing external surface and Z for the total transmitted solar energy through the openings to the building zones), the superscript N is referring to the number of surfaces of the simulated scenario: $N \in \{5, 10, ..., 55, 561\}$, I is the set of indexes $t \in \{1, ..., 8760\}$ where the quantity $X_t^{(561)}$ remains positive and $|I|$ is its cardinality.

Apart from $\mathrm{MAPE_s}$ and $\mathrm{MAPE_z}$ two additional performance measures were used for each scenario (two last columns of table 2): the simulation run time $\mathrm{R_T}$ and the total blocked direct solar energy $\mathrm{E_b}$ calculated from the algorithm's output set $\mathcal{I}_i$.

It can deduced from the results of table 2 that significant improvement in the performance metrics $\mathrm{MAPE_s}$ and $\mathrm{MAPE_z}$ (around 1%) is achieved for the scenarios after $N = 15$ ($\mathrm{MAPE_s}$ is falling form 7.35 to 1.24 % and $\mathrm{MAPE_z}$ is falling from 2.7 to 1.67%). Furthermore, for the scenarios after N=40 (N=45,50,55) the amount of blocked direct solar radiation is increased at the order of 100KJ.

The last observation is evidenced from the plots of figure 6: if an 100KJ energy threshold is imposed, the scenarios with N> 42 are omitted and at the same time 12 out of the 14 shading building envelopes are retained.

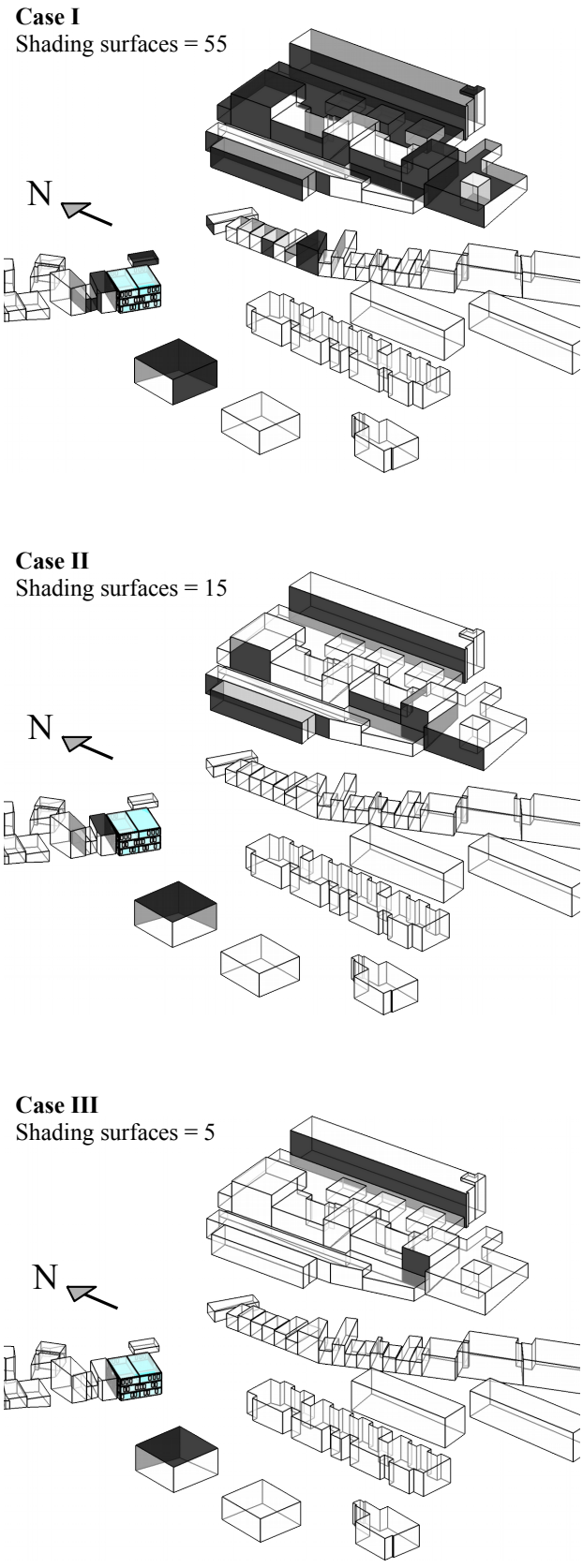

Figure 7: Results of the algorithm applied on the demonstration site, with N=55 (Case I), 15 (Case II) and 5 (Case III) surfaces.

It is important to mention that for the evaluation of MAPE_x metrics, only the time instances t with $X_t^{(561)} > 0$ were considered. There are no instances satisfying $Z_t^{(N)} > 0$ and $Z_t^{(561)} = 0$ and the number of time instances (t) satisfying $S_t^{(N)} > 0$ and $S_t^{(561)} = 0$ are falling from 90 out of 8760 for scenario N=5 to 12 out of 8760 for scenario N=55. The values $S_t^{(N)}$ at those instances are insignificant with respect to the maximum values of $S_t^{(561)}$ obtained during the respective days. Additionally, the MAPE_x performance metrics diverge from the expected zero values because secondary reflections from the shading surfaces (not included in the N=55 scenario) affect the EnergyPlus results of the N=561 scenario.

Table 2: Results
N: *Number of shading surfaces,* $\mathbf{MAPE_S}$*: Mean Absolute Percentage Error for surface incidence radiation,* $\mathbf{MAPE_Z}$*: Mean Absolute Percentage Error for zone transmitted radiation,* $\mathbf{R_T}$*: Simulation run time,* $\mathbf{E_b}$*: Blocked direct solar energy.*

N	$\mathbf{MAPE_s}$	$\mathbf{MAPE_z}$	$\mathbf{R_T}$(sec)	$\mathbf{E_b}$(GJ)
5	11.6304	5.2029	66.76	10.9892
10	7.3502	2.7068	68.18	11.4333
15	**1.2445**	**1.6747**	**70.88**	**11.5859**
20	1.0149	1.6027	72.49	11.6472
25	0.9639	1.5061	76.68	11.6719
30	0.9560	1.4064	80.78	11.6792
35	0.7638	1.2247	81.82	11.6826
40	**0.5303**	**0.9716**	**82.29**	**11.6844**
45	0.4675	0.8822	84.58	11.6848
50	0.4097	0.9258	88.27	11.6850
55	0.4431	0.7064	91.79	11.6850
561	**0.0000**	**0.0000**	**1128.53**	11.6850

Finally, figure 7, contains visual representations of the results of the algorithm for the cases where only 55 (Case I), 15 (Case II) and 5 (Case III) surfaces are considered as shading surfaces with the biggest shading effect. For the case of N=5 only the surfaces in the East, West and Northwest directions are retained.

Limitations

District site

There are special cases of districts with big variation in the slope of their site, in which the shading surfaces obtained by the proposed algorithm are incorrect. Such cases are illustrated in the figure 8, where an incorrect surface (surface 1) is obtained by the proposed algorithm when the site geometrical description is absent.
As figure 8 demonstrates, if the site geometrical description is provided as a single building envelope, then the algorithm will return the correct neighbor shading results, which in the case of the example of figure 8, include the site surfaces 2.

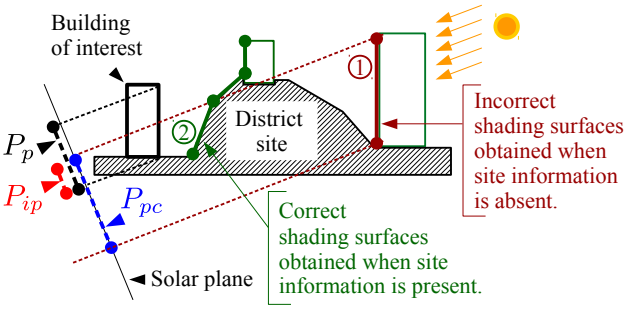

Figure 8: Example of an incorrect neighbor shading surface obtained by the proposed algorithm, when site information is absent.

District vegetation

Additionally, the algorithm does not take into account city vegetation elements. These elements depending on their density allow a fraction of the sun light to pass Al-Sallal and Al-Rais (2013) and as a result, they cannot be considered pure shading objects. Such cases require additional modeling elements to be introduced into the proposed algorithm.

Diffuse radiation

Finally, the diffuse component of the solar radiation, which is affected by the cloud cover, the sky view factor and the albedo of the external surfaces, is not taken into account in the calculations of the proposed algorithm, since it requires a different approach than the one presented here and is a topic of further research and future work.

Conclusions

A geometric algorithm which generates the minimal set of neighbor building surfaces and building envelopes, which shade a specific building of interest in a district environment, at some time during a considered time interval, is introduced. The algorithm receives as input: the geometric representations of all the building envelopes in the district contained in a CityGML file, the index of the building of interest, the duration in simulation instances of a single day and a time window in days. The introduced process returns two outputs: (a) a minimal set of triplets (j,m,Rad) $\mathcal{I}_i$, where each triplet refers to a shading polygon and contains its building envelope and polygon indexes j, m and the amount of direct solar energy (Rad) this polygon blocks (by shading) to building i during the considered time interval defined by the day duration and time window values and (b) a minimal set of pairs (j,Rad) $\mathcal{E}_i$ where each pair refers to a building envelope and contains its index j and the amount of direct solar radiation (Rad) the envelope blocks during the considered time interval.
The introduced algorithm is applied on the CityGML LoD2 data which describe a district in Sandiago de

Compostella in Spain, where a single building envelope is chosen as a building of interest and is replaced by the detailed IFC description of a single imaginary building. The performance of the algorithm is tested using EnergyPlus simulations with simulation models generated using the automatic simulation model generation process which combines IFC and CityGML data described in Lilis et al. (2016). Eight simulation scenarios are considered, where the total direct solar radiation incident on all the external surfaces of the building of interest and the total transmitted solar radiation in its zones are the calculated quantities for each scenario. The scenario with all the envelope surfaces present as shading surfaces is treated as a reference. As the performance metric the mean absolute percentage error between the calculated values of every scenario and the values of the reference scenario. The evaluation results verified that significant reduction in the overall simulation execution time can be achieved if the shading surfaces obtained by the proposed algorithm are adopted for the simulation model generation without sacrificing accuracy. Finally, certain limitations of the proposed algorithm referring to cases of terrain slope variations and city vegetation are also discussed.

Additionally, the shading effect of neighbor buildings on a surface of a building of interest, depend on the type of the surface. If the surface of the building of interest is a transparent surface, then the shading effects from neighbor buildings will impact the thermal simulation more than in the case of an opaque surface. Consequently, if CityGML data of Level of Detail 3 and higher are available, the proposed algorithm should be applied twice: one time for the transparent surfaces and one time for the opaque surfaces of the building facades, with priority given to the results referring to the transparent facade surfaces. Such investigation is a topic of further research.

Acknowledgements

This work is supported by the E.U. project: Innovative design tools for refurbishing of buildings at district level H2020-EeB-05-2015 #680676 (OptEEmAL).

References

Al-Sallal, K. A. and L. Al-Rais (2013). A novel method to model trees for building daylighting simulation using hemispherical photography. *Journal of Building Performance Simulation 6*(1), 38–52.

Alam, N., V. Coors, and S. Zlatanova (2013). Detecting shadow for direct radiation using CityGML models for photovoltaic potentiality analysis. *Urban and Regional Data Management; Ellul, C., Zlatanova, S., Rumor, M., Laurini, R., Eds*, 191–196.

ASHRAE (2007). Solar energy use. *ASHRAE Handbook - HVAC Applications (SI) 33*, 5–6.

Biljecki, F., H. Ledoux, and J. Stoter (2016). Does a finer level of detail of a 3d city model bring an improvement for estimating shadows. *Advances in 3D Geoinformation; Springer International Publishing: Cham, Switzerland*.

Bremer, M., A. Mayr, V. Wichmann, K. Schmidtner, and M. Rutzinger (2016). A new multi-scale 3d-gis-approach for the assessment and dissemination of solar income of digital city models. *Computers, Environment and Urban Systems 57*, 144–154.

Freitas, S., C. Catita, P. Redweik, and M. Brito (2015). Modelling solar potential in the urban environment: State-of-the-art review. *Renewable and Sustainable Energy Reviews 41*, 915–931.

Hofierka, J. and M. Zlocha (2012). A new 3-d solar radiation model for 3-d city models. *Transactions in GIS 16*(5), 681–690.

Li, D. and S. Wong (2007). Daylighting and energy implications due to shading effects from nearby buildings. *Applied Energy 84*, 1199–1209.

Lilis, G. N., G. Giannakis, and D. Rovas (2016). Simulation model generation combining IFC and CityGML data. In *European Conf. on Product and Process Modelling (ECPPM)*.

Marsh, A. (2005). The application of shading masks in building simulation. In *Proceedings of the Building Simulation 2005 Conference*.

O'Donnell, J. (2013). SimModel: A domain data model for whole building energy simulation. In *proc. of SimBuild*, Sydney, Australia, pp. 382–389.

Open Geospatial Consortium (2012). OGC City Geography Markup Language (CITYGML) Encoding Standard. <`https://portal.opengeospatial.org/`>.

Pisello, A., J. Taylor, X. Xu, and F. Cotana (2012). Inter-building effect: Simulating the impact of a network of buildings on the accuracy of building energy performance predictions. *Building and Environment 8*, 37–45.

Reinhart, C. and C. Davila (2016). Urban building energy modeling - a review of a nascent field. *Building and Environment 97*, 196–202.

Robinson, D. (2012). *Computer modelling for sustainable urban design: Physical principles, methods and applications*. Routledge.

Vatti, B. (1992). A generic solution to polygon clipping. *Communications of the ACM 35*(7), 56–63.

Wieland, M., A. Nichersu, S. M. Murshed, and J. Wendel (2015). Computing solar radiation on CityGML building data. In *18th AGILE international conference on geographic informaton science*.