

Sombrero: A PC-Tool To Calculate Shadows On Arbitrarily Oriented Surfaces

A. Niewianda, F.D. Heidt*

Dept. of Physics, University of Siegen, D-57068 Siegen, Germany

Abstract—Shaded areas of windows, solar thermal collectors or photovoltaic modules are of a major importance for the calculation of solar heating- and cooling-loads of buildings as well as for the determination of thermal or electrical output of corresponding solar equipment. SOMBRERO, a PC-program written in Turbo-Pascal, calculates the GSC (geometrical shading coefficient), the proportion of shaded area of an arbitrarily oriented surface surrounded by shading elements as a function of time and location. Shading elements are treated as polygons (not necessarily rectangles) located in a plane and can be combined to bodies such as buildings or trees. They may also represent overhangs and side-wings of the building under consideration. Elements, which are far away from the receiver area are treated as horizontal shading profiles. The reduction of (isotropic) diffuse radiation due to different kinds of obstacles is calculated by means of view-factors. Calculated results of the GSC are stored in ascii-format and can be used as an input for dynamic solar system simulation programs. This is demonstrated in two illustrative examples showing the coupling of SOMBRERO with SUNCODE and TRNSYS in order to calculate the effects of shading on passive solar heating and passive cooling, respectively.

1 Introduction

Shadows are indicators for light; in solar systems for instance they indicate the availability of sunshine. Shaded areas on the radiation collecting solar aperture, however, may be useful or detrimental for the system, depending on its type and objective. The knowledge of light and shadow finds very important applications within the solar architecture and urban planning. This is because already with the establishment of the local plan for development it is determined, how much sun a building can collect with its various facade elements [Goretzki (1989)]. For middle and northern latitudes, respectively in regions with relatively low solar irradiation, shaded parts on the southern front of a building, i.e. by other houses placed directly in front of it, must be avoided during the heating period. They would reduce solar gains and with this possible energy savings. In regions further south with a higher solar irradiation, however, passive cooling concepts are advantageous. Here, narrowly placed houses but also overhangs and side-wings can significantly reduce the cooling load. Shading may either lead to a minimum of energy consumption - or, at houses without air-conditioning, to an improvement of indoor thermal comfort.

A further important factor is daylighting. Severe shading of the solar aperture prevents sufficient illumination by daylight for the inner building. On the other hand, however, bright solar beam radiation will cause glarings, which can be avoided by certain shading measures, like overhangs. Shadings on solar thermal collectors, i.e. for domestic hot water, are always detrimental for a solar system. They reduce the radiation collecting area and beyond that, lead to inhomogeneous temperature conditions of the solar thermal plant. Same applies to photovoltaic generators or modules, where total or partial shading not only results in a reduction of energy absorbing area, but also in greater losses of bypass-diodes.

* Author to whom correspondence should be addressed; member of ISES.

Shadows are caused by obstacles between the source of light and the illuminated body, which may consist of several receiver planes (targets). The variety of such obstructions can be placed into three categories: - obstacles far away, described as horizontal profiles, - expanded objects of an average distance from the target, - areas belonging to the illuminated body, which cause a self-shading of certain targets. Furthermore, there are two types of light sources: solar beam radiation, originating from the time dependent position of the sun as well as diffuse solar radiation of variable intensity, originating from the whole hemispherical environment of the receiver plane. This means, that the shading depends in many ways on the orientation of the collecting surface, its surroundings, on the season and time of day. Suitable for a corresponding characterization of shading is the geometrical shading coefficient (GSC), the ratio between shaded and total receiver plane area [Yeziro and Shaviv (1994)]. Due to the complexity of geometry and its time dependency it is almost impossible to determine the GSC reliably just by an assessment.

Because of the before-mentioned significance for the performance of solar systems, precise GSC-data are essential for accurate simulations. For this reason, following described PC-tool SOMBRERO was developed. This tool allows the treatment of all three before-mentioned types of obstructions and the calculation of its GSC effects due to beam radiation and diffuse radiation. As a result for a chosen target plane, a monthly list of hourly GSC-values is obtained. These values are available in form of ascii data files which can be used as data inputs for other simulation programs.

The following will describe the methods and algorithms used to generate GSC-values for beam radiation and diffuse radiation. Also, further details are given in view of the PC- program, its user surface, input values and calculated results. Two illustrative examples are presented and discussed, which show the coupling of this tool SOMBRERO with other thermal simulation programs like SUNCODE [Palmiter et al. (1985)] and TRNSYS [Klein et al. (1990)]. Last but not least, they will demonstrate the importance of knowledge on shading for predicting energy consumption and temperature behaviour of such solar systems.

2 Method

2.1 Geometrical model

The idea of the algorithm goes back to Clarke (1985) and uses the technique of parallel point-projection onto a plane which contains the receiver area. If we consider one single receiver area, it is described by an array of up to N vertex-points $(x_1^R, y_1^R, z_1^R) \cdots (x_N^R, y_N^R, z_N^R)$ in a three-dimensional cartesian coordinate system. The index "R" is for "receiver." All points have to be located in a plane. A number of M obstacle-areas O_1 to O_M are described by their vertex-points $(x_1^{O_1}, y_1^{O_1}, z_1^{O_1}) \cdots (x_N^{O_1}, y_N^{O_1}, z_N^{O_1})$ to $(x_1^{O_M}, y_1^{O_M}, z_1^{O_M}) \cdots (x_N^{O_M}, y_N^{O_M}, z_N^{O_M})$.

They can be located in arbitrary positions. Each obstacle can consist of up to N points, too. Figure 1 shows the coordinate system, which subsequently will be called surface-of-earth-system (SOE-System) and represents a right-handed cartesian system.

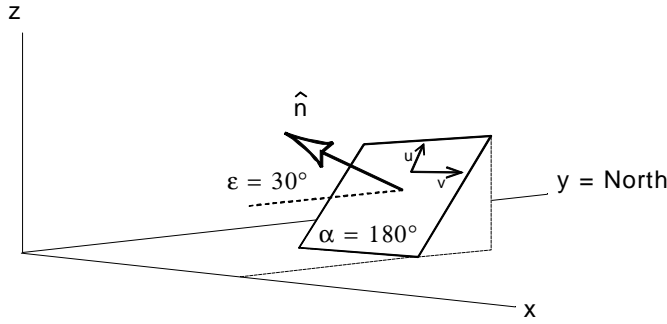


Fig. 1. The SOE-System with an area oriented to the south. The azimuth α is counted clockwise from the north, i.e. a south-oriented area has an azimuth of 180° . The elevation angle of the example area is 30° . The front-side is facing the sky, which is indicated by the normal vector \hat{n} . The normal-vector is obtained by following the polygon-vertices anti-clockwise with the fingers of the right hand. The direction of the thumb then indicates the direction of the normal vector (right-hand rule). The u-v-axes indicate the body-system.

To be able to calculate the projection of the obstacle areas on the receiver area, the position of the sun has to be known within the SOE-System. It is calculated according to Holland & Mayer (1988), where a FORTRAN-routine is described which gives the *sun's azimuth* \mathbf{a}_s and elevation \mathbf{e}_s as a function of location (longitude and latitude) and time t (time-of-the-year, described by day, hour and minute). This routine has been adapted, so that it counts the azimuth *clockwise from the north*. Positive elevation means that the sun is above the horizon, negative below. In order to run a shadow calculation properly, it is more convenient to describe the sun's position as a vector $\vec{S}(t)$. Therefore, it is necessary to transform the spherical coordinates $(\mathbf{a}_s, \mathbf{e}_s)$ into cartesian coordinates (x_s, y_s, z_s) , which present a point on the unit-sphere.

$$\vec{S}(t) = \begin{pmatrix} x_s \\ y_s \\ z_s \end{pmatrix} (t) = \begin{pmatrix} \cos \mathbf{e}_s(t) \cdot \sin \mathbf{a}_s(t) \\ \cos \mathbf{e}_s(t) \cdot \cos \mathbf{a}_s(t) \\ \sin \mathbf{e}_s(t) \end{pmatrix} \quad (1)$$

The program, written in Turbo-Pascal, deals with the objects as separate data sets, i.e. an area is described by an array of corner-points and the position of the sun as a vector on the unit-sphere. Matrix transformations, as described in the next paragraph, can be applied on vectors, regardless of whether they are corner-points of an area or represent a single point as the sun-vector.

Now we have a description of all necessary geometrical information. The next step is to transform the objects in a way to make them suitable for shadow-calculations. The corner-points of the obstacle-areas have to be projected onto the plane which contains the receiver area. The projection is carried out along the direction of the sun-vector. After that, we have to determine the shaded part of the receiver area.

2.2 The algorithm for beam radiation

To be able to perform the projection, it is necessary to know whether a vertex-point of an obstacle-area is located in front of the receiver area or behind it. Only points lying in front of it have to be projected. Therefore, some transformations are made. The meaning of the angles is shown in Figure 1.

The complete algorithm is as follows:

1. Rotation of all elements in a way that the receiver area is parallel to the $x - z$ plane of the SOE-System. For an arbitrarily oriented plane two rotations are required:

- A rotation around the $z -$ axis by the receiver-azimuth angle α_R .
- A rotation around the $x -$ axis by the receiver-elevation angle ϵ_R .

2. A displacement of all elements by the vector from $(0, 0, 0)$ to the lower left corner-point of the receiver area is carried out. The “lower left corner” is found by looking at the plane facing the normal vector from the front. Now the receiver area is part of the $x - z$ plane with the normal vector facing the north (i.e. it forms a parallel to the y -axes). The decision, whether a point $(x_i^{Oj}, y_i^{Oj}, z_i^{Oj})$ of an obstacle-area is lying in front of the receiver area is reached by looking at the y -coordinate of the point. If $y_i^{Oj} > 0$, the point has to be projected. By applying this test routine to all points of an obstacle-area, it is possible to eliminate those points which are behind the receiver plane. If only parts of the obstacle-polygon are lying behind the receiver plane, it can be replaced by a reduced polygon with all its points positioned in front of the receiver plane (see Figure 2).

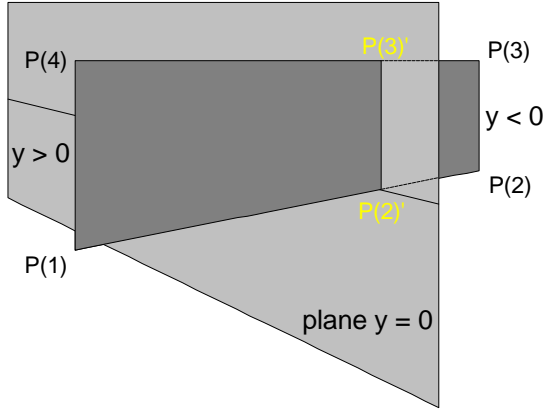


Fig. 2. A polygon that is located partially in front of and partially behind the plane $y = 0$ is replaced by a reduced polygon with all of its points lying in or in front of this plane.

3. Rotation of the sun: For every timestep, the sun-vector $\vec{S}(t)$ is calculated according to equation 1 and has to be transformed in the same manner. No displacement is necessary here. A projection has to be performed only for those timesteps where $y_s > 0$ and $e_s > 0$, i.e. the sun is standing in front of the $x - z$ plane and it is above the horizon.

2.2.1 The rotation angles

The rotation angles are determined by the following formulas applied to the *vertex-points of the receiver area*:

$$\left. \begin{aligned} X \text{ Sum} = i &= \sum_{j=1}^N (y_j z_j - z_j y_j) \\ Y \text{ Sum} = i &= \sum_{j=1}^N (z_j x_j - x_j z_j) \\ Z \text{ Sum} = i &= \sum_{j=1}^N (x_j y_j - y_j x_j) \end{aligned} \right\} \text{for } \begin{aligned} j &= i + 1 \\ j > N: j &= 1 \end{aligned} \quad (2)$$

Determination of the azimuth angle α_R .

$$\alpha_R = \begin{cases} \arctan\left(\frac{X \text{ Sum}}{Y \text{ Sum}}\right) & \text{for } Y \text{ Sum} > 0 \\ \arctan\left(\frac{X \text{ Sum}}{Y \text{ Sum}}\right) + \pi & \text{for } Y \text{ Sum} < 0 \end{cases} \quad (3)$$

For $Y \text{ Sum} = 0$:

$$\alpha_R = \begin{cases} -\pi/2 & \text{for } X \text{ Sum} < 0 \\ 0 & \text{for } X \text{ Sum} = 0 \\ \pi/2 & \text{for } X \text{ Sum} > 0 \end{cases}$$

Determination of the elevation angle ϵ_R :

For $X \text{ Sum}^2 + Y \text{ Sum}^2 \neq 0$:

$$\epsilon_R = \arctan\left(\frac{Z \text{ Sum}}{\sqrt{X \text{ Sum}^2 + Y \text{ Sum}^2}}\right), \quad (4)$$

and for $X \text{ Sum}^2 + Y \text{ Sum}^2 = 0$:

$$\epsilon_R = \begin{cases} -\pi/2 & \text{for } Z \text{ Sum} < 0 \\ 0 & \text{for } Z \text{ Sum} = 0 \\ \pi/2 & \text{for } Z \text{ Sum} > 0 \end{cases}$$

2.2.2 The first rotation (A)

First we rotate *every* vertex-point (x, y, z) anti-clockwise around the z-axis by the angle α_R :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos \alpha_R & -\sin \alpha_R & 0 \\ \sin \alpha_R & \cos \alpha_R & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (5)$$

After this rotation, the azimuth of the receiver plane is 0.

2.2.3 The second rotation (E)

Now we perform a clockwise rotation around the x-axis by the angle ϵ_R :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \epsilon_R & \sin \epsilon_R \\ 0 & -\sin \epsilon_R & \cos \epsilon_R \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (6)$$

The normal vector of the receiver plane now is a *parallel to the y-axis*. The receiver area is a parallel to the $x - z$ plane.

2.2.4 The displacement

Next, the displacement is carried out to place the receiver area into the $x - z$ plane:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \vec{d}, \quad (7)$$

where \vec{d} is the vector from $(0, 0, 0)$ to the lower left corner-point of the receiver area (This point can be calculated by applying a simple algorithm, which will not be further explained. Note that after the two rotations the y -coordinate of all receiver-vertices is the same).

2.2.5 The transformation of the sun-vector

For each timestep, the sun-vector $\vec{S}(t)$ has to be rotated in the same way as all the other vertex-points before. In short notation this is:

$$\vec{S}(t) = E \times (A \times \vec{S}(t)) \quad (8)$$

2.3 The projection-algorithm

In order to calculate the projected polygons, we have to project all obstacle-areas (indicated by the index "O") point-wise and for every time step t along the sun-vector $\vec{S}(t)$ onto the $x - z$ plane. This is done by solving the equation

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_O - \lambda(t) \cdot \vec{S}(t) = \begin{pmatrix} x^p \\ 0 \\ z^p \end{pmatrix}(t) \quad (9)$$

using the y - component (which has to be zero after projection) for the calculation of $I(t)$. If I is known, we are able to calculate the projected x^p - and z^p - components, which are located in the $x - z$ plane.

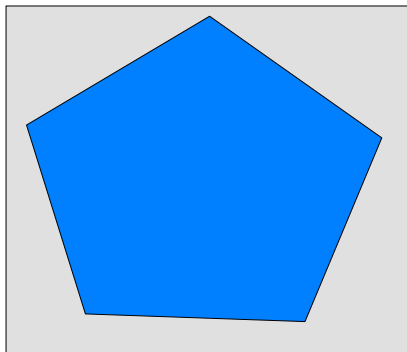


Fig. 3. Step 1: The graphic shows the shape of the receiver area before the projection of shadows due to the obstructive polygons. The points with the colour of the receiver area (blue) are counted.

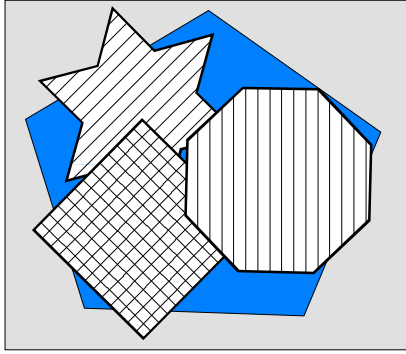


Fig. 4. Step 2: The shadows are drawn. In general they can overlap each other. By counting the blue points which are left over the shaded part of the receiver area can be determined.

2.4 The destination of the shaded part of the receiver area

The destination of the shaded part of the receiver area could in principle be performed analytically. However, complicated algorithms are required to solve this problem on a computer for all sorts of polygons and independently oriented obstacles, because we also have to consider shadows which overlap each other (see Figure 1 and 2). These calculations are time-consuming, so although an analytical solution appears to be the most elegant way, it is not very practical.

A graphic solution to the problem has been presented by Yezioro & Shaviv (1994) and has been implemented on a graphic-workstation. A CAD-program-display was used to determine the shaded area of building surfaces by simply drawing a picture of the scenery from the sun's point of view first with and then without shading elements. Comparisons of the pictures can be carried out using the Z-buffer of the graphic-workstation.

A different approach is presented in this paper: The projected shadows of the obstructive polygons are painted on the graphic display of the VGA-screen, which represents the plane of the receiver area (i.e. the $x - z$ plane of the SOE-system). The receiver area itself has a reserved colour (here: blue) and is painted *before* the projections are performed. The determination of the shaded part of the area is now very simple: Use the graphic-routines of Turbo-Pascal to count the points on the display of the screen which have the colour of the receiver area *before* the shadows are drawn, then paint them and look how many points of this colour are left over.

So for each time step under consideration, the GSC_B (for beam radiation) is calculated by the destination of the ratio:

$$GSC_B = 1 - \frac{\left(\begin{array}{l} \text{No. of blue points on screen} \\ \text{after projection of obstacles} \end{array} \right)}{\left(\begin{array}{l} \text{No. of blue points on screen} \\ \text{before projection of obstacles} \end{array} \right)} = 1 - \frac{\text{No.}(\text{blue})_a}{\text{No.}(\text{blue})_b} \quad (10)$$

This method has several advantages:

- No *hidden surface's removal analysis* has to be carried out for each time step.
- The algorithm does consider the diffuse radiation, which will be explained below.

- Making use of standard Turbo-Pascal graphic routines, this way of proceeding even allows us to deal with concave areas¹.

The display on the screen also serves as a visualization of the objects and can be used to control the input file for geometrical data.

2.5 The treatment of semi-transparent elements

Elements like trees may be partially transparent for light and also this transparency can vary with time, e.g. a deciduous tree loses its leaves in autumn. Therefore, shadows of trees have to be treated differently. Their opaque part of the area is determined by a monthly schedule. A schedule value SV of 0.8 means, that 20% of the area are transparent, the rest opaque.

As a consequence a receiver area totally covered by the shadow of a tree would be shaded by 80% only.

In order to make use of the graphic algorithm to determine the GSC, tree elements are projected *before* the other obstacle elements. They are painted in a reserved colour (green). If trees are placed between other elements and the receiver, and the sun projects these elements over them, (see Figure 5) their polygon-shaped shadows are (partially) painted over².

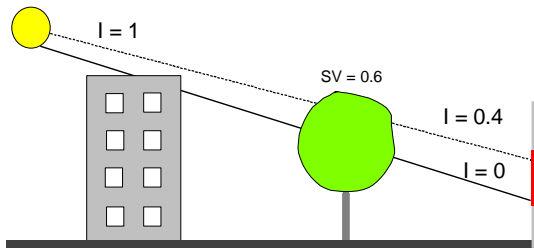


Fig. 5. A tree may be placed between a window and a building. If the sun ray only hits the tree, the shadow of the tree is counted according to its scheduled value SV . If it first hits the building, the tree's polygon will be covered by the polygons of the building and the beam-intensity I is zero.

So the formula for the GSC_B has to be changed to:

$$GSC_B = 1 - \frac{\text{No.}(\text{blue})_a + \text{No.}(\text{green})_a \cdot (1 - SV)}{\text{No.}(\text{blue})_b} \quad (11)$$

2.6 The algorithm for diffuse radiation

For a diffuse isotropic radiation model the determination of the incident diffuse radiation $H_d(\mathbf{e})$ on a tilted surface with elevation \mathbf{e} can be calculated according to Duffie & Beckman (1980) by using

¹ It is, in general, not a trivial problem to fill a space represented by pixels of the background colour with the colour of surrounding polygon-lines, if the polygon is concave.

² To avoid problems in the calculation of the GSC for receiver areas which do not cover the whole "target" area, it is necessary to store the points of the target area which are covered by the receiver polygon. The target area in this context means the reserved area on screen where the program is allowed to paint the receiver polygon.

$$H_d(\epsilon) = H_{dhor} \cdot (1 + \sin \epsilon)^2 + (H_{bhor} + H_{dhor}) \cdot \rho \cdot (1 - \sin \epsilon)^2 \quad (12)$$

$$\begin{aligned} &= H_{dhor} \cdot \nu_s + (H_{bhor} + H_{dhor}) \cdot \rho_G \cdot \nu_g \\ &= H_{dhor} \cdot \nu_s + (H_{bhor} + H_{dhor}) \cdot \nu_G \end{aligned} \quad (13)$$

where

- H_{dhor} : diffuse irradiation on a horizontal surface
- H_{bhor} : beam irradiation on a horizontal surface
- ρ_G : ground reflectance
- ν_s : view - factor of sky
- ν_g : view - factor of ground
- ν_G : ground factor

As equation 12 does not consider any shading elements the view-factors of ground and sky can be calculated analytically. For collectors on elevated roofs it may be acceptable to neglect such shading elements because their aperture is mainly directed to the sky and the remaining environment below them may be described by a constant ground reflectance. In buildings, however, windows that are tilted by 90° face their neighbourhood, which may consist of buildings, trees, sidewings and overhangs or horizontal shading profiles. In order to take these details of the environment into account, it is necessary to calculate the factors ν_s and ν_G in a better spatial resolution. Nevertheless, it is not the purpose of SOMBRERO to go into the details of the scattering of light and multiple reflection of light.

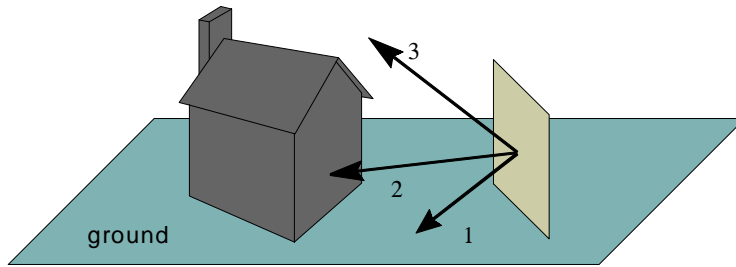


Fig. 6. A vector scanning the half-space in front of the receiver area may hit the ground (1), an obstacle element (2) or the sky (3). The treatment of these different characteristics is explained in the text.

To calculate ν_s and ν_G , we apply the algorithm which SOMBRERO uses for the destination of the GSC_B to scan the half-space in front of the receiver area (see Figures 6 and 7). Instead of performing a projection along the sun-vector, we now emit vectors along all directions with an elevation in the range of $\epsilon_R \pm 90^\circ$ and azimuth in the range of $\alpha_R \pm 90^\circ$. To make the model as easy as possible, we only distinguish the following cases (see Figure 6):

For vectors with an elevation $e < 0$:

1. The vector hits a point on the ground.
2. It hits an element which is then projected onto the receiver area.

For vectors with an elevation $e > 0$:

1. The vector strikes an element.
2. It sees the sky.

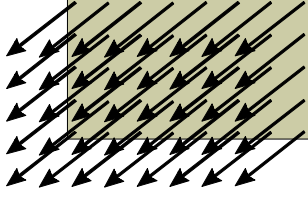


Fig. 7. To improve the resolution of the calculation, the evaluation of the view-factor for the whole receiver area is necessary. Therefore, a grid is placed on it with a step width of 7 pixels on the screen (default value). From all of these grid points a vector points in the same direction. The obstacle elements are projected along this vector onto the receiver-area. Afterwards, an averaged factor for the whole receiver area is calculated according to the colour of the grid-points after the projection and the elevation angle of the vector.

The calculation of the factors \mathbf{u}_S and \mathbf{u}_G is carried out according to the following equations:

$$\mathbf{n}_{\tilde{G}} = \text{vectors } (\mathbf{e} < 0) \sum_{\text{grid}} \sum \text{No.}(1) \cdot \mathbf{r}_G$$

$$\mathbf{n}_{\tilde{S}} = \text{vectors } (\mathbf{e} < 0, \mathbf{e} > 0) \sum_{\text{grid}} \sum [\text{No.}(2) \cdot \mathbf{r}_O + \text{No.}(3)]$$

$$\mathbf{n}_G = \frac{\mathbf{n}_{\tilde{G}}}{\text{all vectors } \sum_{i=1}^3 \sum \text{No.}(i)} \quad (14)$$

$$\mathbf{n}_S = \frac{\mathbf{n}_{\tilde{S}}}{\text{all vectors } \sum_{i=1}^3 \sum \text{No.}(i)} \quad (15)$$

where

- \mathbf{r}_O : Reflectance of obstacle surface
- $\text{No}(i)$: Number of grid-points with characteristic(i)

The reflectance of obstructing surfaces is attributed to the view-factor of the sky, because we assume that obstacles reflect only diffuse radiation. A default value for \mathbf{r}_O is 0.25. In this model the ground is the only part of the environment reflecting beam radiation. Scans are performed within steps of 5 degrees width³ for both azimuth and elevation angles of the vectors.

If the user has defined a horizontal shading profile (which is done by defining intervals of azimuth-angles with an elevation-angle above which the sun can be seen), the reflectance of the ground is taken also as the reflectance of the horizon.

3 Generation of the geometrical description

In order to supply SOMBRERO with a data base for the geometrical description of the scenery under investigation, the vertex-points of all areas have to be known within the SOE-system. In general, it is not an easy task to calculate these points for an arbitrarily oriented surface. Therefore, a module was developed that transforms easy-to-handle user-input data into the SOE-coordinates. The user describes an area by its azimuth, elevation and its vertex-points in the body-system (u-v-system), which is depicted in Figure 1. Because this system is 2-dimensional a description of the vertex-points is easy and can be achieved by introducing coordinates into a plan of the scenery. The origin of the u-v-system within the SOE-system can be used to position elements at their destination.

³ This is the default value; the user may change it according to his purposes.

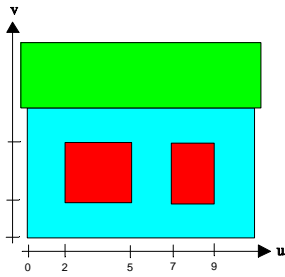


Fig. 8. View of a facade with two windows. They are described by their length, height and origin within the u-v-system. The u-v-system is regarded as the body-system of the facade. Units are in meters.

Figure 8 shows a view of a facade with two windows. If we consider them as receiver areas for a thermal zone, we have to describe their positions and their dimensions. Because they are rectangles, each of them can be described by a length (Δu) and a height (Δv) and an origin position in the u-v-system (u_0, v_0). After the positioning within the u-v-system, the relative orientation of the u-v-system and the SOE-system has to be known. This is achieved by describing the azimuth and elevation of the facade in the SOE-system. This facade, for instance, shall have an elevation of 0° and an azimuth of 135° , which is a vertical wall oriented to the south-east. The last step is to tell the module, where the origin of the u-v-system is located within the SOE-system. Again we can get this information from a plan which shows the scenery from above, e.g. a plan of a town/site. So we have a description of the scenery in its natural coordinates. For receiver areas it is also possible to create overhangs and side-wings by declaration of their relative elevation or azimuth according to the u-v-system.

For frequently occurring obstacles like houses or trees a simple generator has been written. A house, for instance, is described by its length, height, height of roof, its orientation and origin in the SOE-system. A maximum of 300 polygons with up to 12 points can be handled by the program.

4 Applications

4.1 Coupling of SOMBRERO with SUNCODE

Ground plan

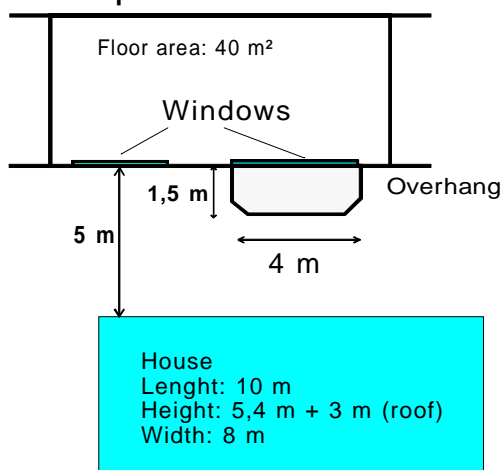


Fig. 9. The scenery for the simulation with SUNCODE is a flat with two windows facing the south. One of them is placed underneath a balcony. In a distance of 5 m a house is positioned in front of the south-facade. The U-value of the outer walls is $0.23 \text{ W/m}^2\text{K}$. The windows consist of three glass layers. Their U-value is $1.8 \text{ W/m}^2\text{K}$. Both windows belong to the same thermal zone so that an overall shading factor of the window-area is sufficient.

View from south

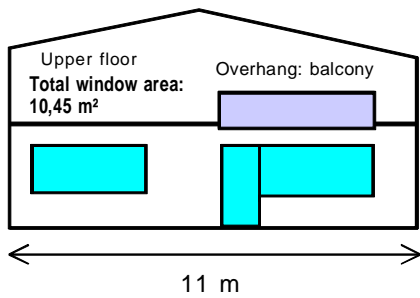


Fig. 10. View of the south facade of the building. The ceiling to the upper floor is considered as adiabatic.

The thermal simulation program SUNCODE allows to simulate the heating and cooling demand of a building taking into account solar gains through windows. SUNCODE does consider shadows by overhang-, siding- and horizontal obstructions. The factor SUNCODE uses for the reduction of the solar gains by shadows of overhangs was replaced by values for the GSC_B calculated with SOMBRERO. To do this, the source code had to be changed. Instead of calculating the factor the program reads the monthly averaged hourly GSC_B -values from a data file. For the following illustrative example only the shading of the beam radiation was taken into account.

For a description of the scenery under consideration refer to Figures 1 and 2. Three variants were simulated:

1. Simulation of the solar gains and energy consumption without shading elements.
2. Simulation with shading by the balcony only (overhang).
3. Simulation with shading by balcony and house.

The interesting time-period is winter (climatic data of Essen, Germany), when the shadow of the adjacent house influences the heating demand of the flat because of the reduced solar gains due to low elevations of the sun.

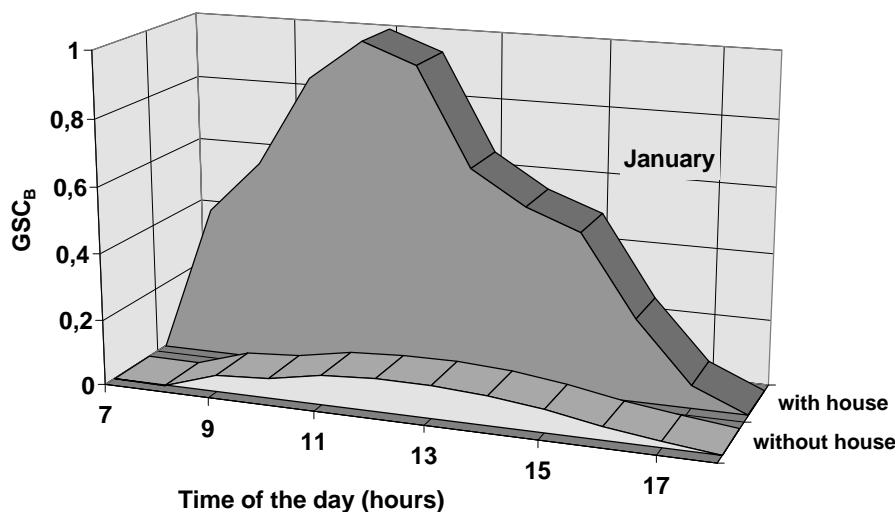


Fig. 11. Monthly averaged values of the hourly GSC_B . The lower curve shows the results of SOMBRERO for the overhang only, the upper one for both, overhang and house.

During the heating period the balcony on its own would not influence the solar gains very much (Figure 11). The comparison of the results for direct solar gains and the heating demand is shown in Figure 12.

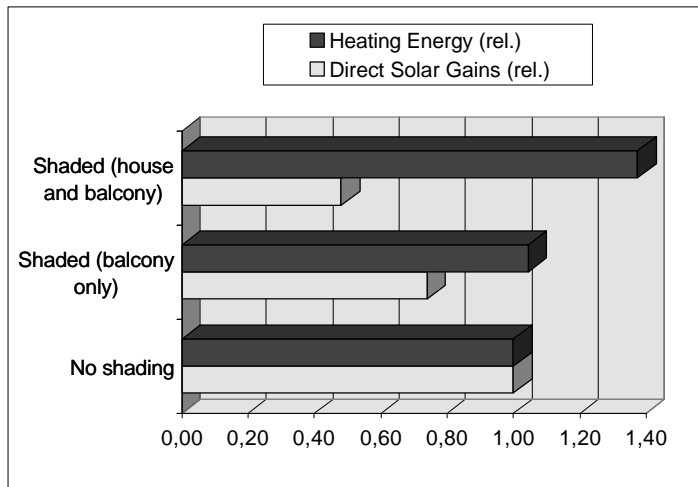


Fig. 12. Relative energy consumption and solar gains of the flat under different shading conditions.

4.2 Coupling of SOMBRERO with TRNSYS

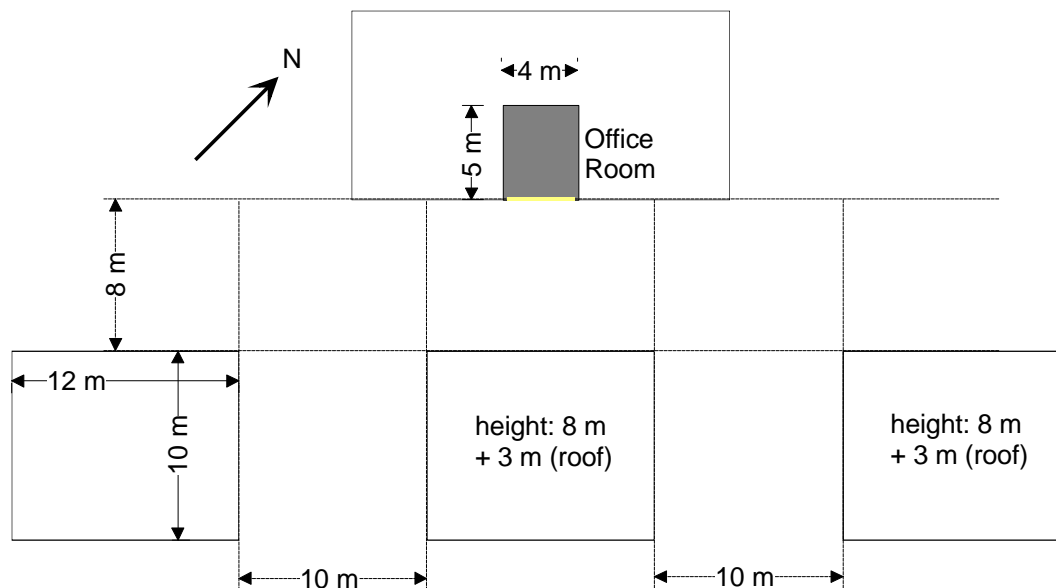


Fig. 13. The plan shows an office room in a town surrounded by buildings. The orientation of the window is south-east. The window-size is 3.5 m x 2 m, the height above the ground is 3 m.

The energy consumption of buildings, especially office buildings, can be divided into energy for heating, cooling, lighting and equipment [Santamouris M. et al. (1994)]. The following example deals with the influence of shading by external buildings on room temperatures and, therefore, indirectly also on cooling loads.

The transient simulation program TRNSYS is able to read pre-processed hourly values for the shading coefficients of SOMBRERO from a data file. With some modifications in the TRNSYS deck-file even the calculation of the shaded diffuse radiation on a tilted surface $H_d(\theta)$ is possible by using the view-factors for sky v_s and ground v_G . Usually the radiation processor of TRNSYS calculates $H_d(\theta)$ for isotropic radiation, taking into account the ground reflectance r_G . But as it also provides us with the

quantities of H_{bhor} , H_{dhor} and $H_b(\mathbf{a}, \mathbf{e})$ it is possible to calculate $H_d(\mathbf{e})$ for every time step using equation 13. EQUATION-statements which carry out this calculation can be introduced into the deck-file. This file contains the description of the simulation parameters and the description of the coupling between the components. The output of these equations is used as an input for a TYPE 56 multi-zone building model. This input provides a wall or a window of a predefined orientation with the time-dependent radiation. The shading of the beam radiation can be calculated by the introduction of additional equations.

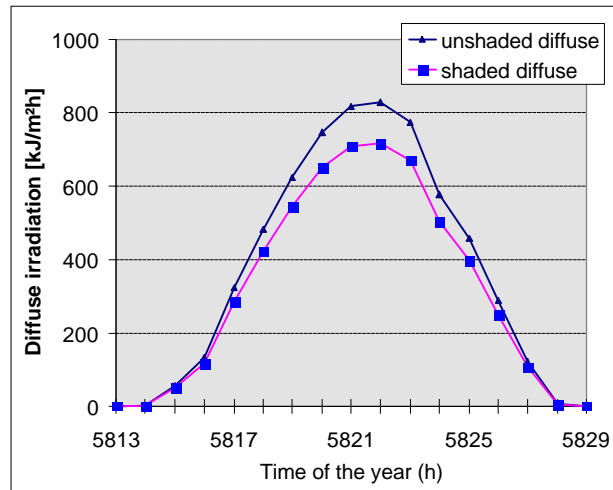


Fig. 14. The graphic shows the difference in the diffuse irradiation on a south-east oriented vertical surface surrounded by several buildings. The upper curve is the result of the radiation processor of TRNSYS, the lower was calculated with the view-factors from SOMBRERO. Ground reflectance was assumed as 0.3.

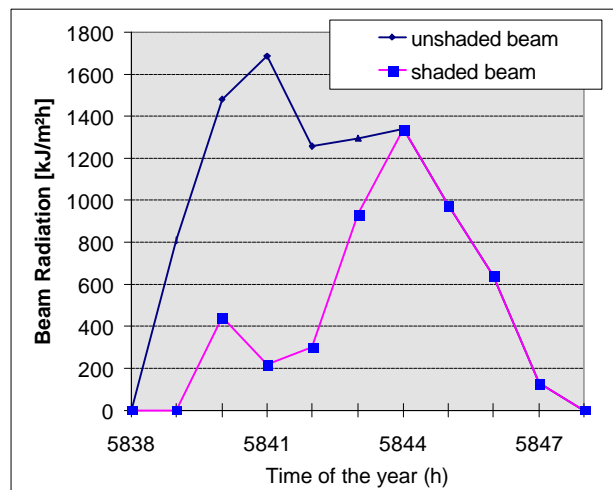


Fig. 15. Beam radiation on the office window for a sunny day as a function of time. Due to the surrounding buildings the sun hits the window with its full intensity only in the afternoon.

To show the influence of shading through external buildings on the overheating risk in summer, we simulated the temperature evolution in an office room during a typical summer situation of five days length. In our example we considered a single zone only. The office room is part of a big building. Its window is facing the south-east (see Figure 13). The external wall is massive with 8 cm insulation material on the outside surface. The U-value of the windows is 3.0 W/m²K. The airchange rate was set to 2/hour. The simulation period covers some hot, sunny days in august (climatic data of Essen,

Germany). Opposite the window several higher buildings are located in a distance of 8 m across the road.

The evolution of the temperature in the room was simulated once with and once without the influence of the shading buildings. Figure 14 shows the effects of shading by the surrounding buildings on diffuse radiation, whereas Figure 15 shows the same on beam radiation, each for a typical day. The results of the development of room temperatures are presented in Figure 16.

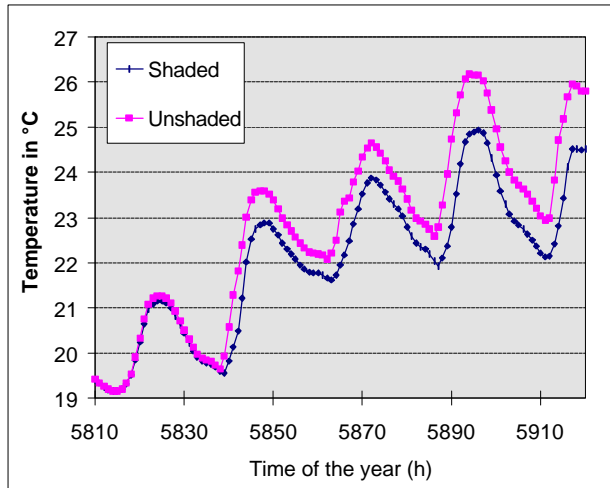


Fig. 16. Development of the office room temperature during a five day hot and sunny weather period. The upper curve refers to the unshaded case.

The two curves show that during the whole period the temperature of the partially shaded room rises slower than that of the room without shading and never exceeds 25 °C. This difference in behaviour affects the thermal comfort as well as the cooling load of the building. Such results are important for the development of a strategy to keep the energy consumption for cooling at minimum values.

5 Outlook

These examples have shown the influence of shadows on passive solar systems. In these systems shadows are always present. During the heating period they are disadvantageous, whereas during the summer they are welcome to keep houses cool. Of course, the influence of shadows on active components is not less important. The behaviour of active components like thermal collectors or transparently insulated walls could be simulated with the SOMBRERO / TRNSYS coupling, too.

For a coupling with a photovoltaic simulation program an overall shading coefficient for a receiver area is not sufficient. Shaded cells influence the energy output of the system depending on their location in a panel. The necessary information for a better resolution of the GSC is already present in the actual version of SOMBRERO. However, the results are no output variables, they would lead to an immense data volume. So one of the next steps is to find a coupling to a photovoltaic simulation program after reducing the output of the graphic algorithm.

6 Conclusion

The PC-program SOMBRERO calculates the quantities and time dependency of the shadow coefficients GSC_B , v_S and v_G . Shadow simulation of complex sceneries can be carried out on a PC with

a minimum expenditure in reasonable time and without the use of further programs. It was shown that the influence of shadows on thermal properties of buildings and solar energy equipment is substantial to the performance of these systems.

Detailed output is available in formats which allow the coupling of SOMBRERO with other simulation programs. This was demonstrated in two examples for thermal simulation with SUNCODE and TRNSYS.

7 Acknowledgements

This work was supported by the Ministry of Science and Technology of North-Rhine-Westfalia, Germany, under Contract No. 253-011-91 (AG-Solar NRW). The authors would like to thank Mr. A. Eicker for his great help and excellent ideas in hacking the source code, Dipl.-Phys. T. Braeske for preparing the algorithm to calculate the position of the sun, and Dipl.-Phys. A. Kock for the preparation of the building description files.

References

- [1] Clarke J.A. (1985) Energy simulation in building design. Adam Hilger Ltd., Bristol, pp.146-158.
- [2] Duffie J.A., Beckman W.A. (1980) Solar engineering of thermal processes. John Wiley & Sons, Inc., New York, p. 86.
- [3] Goretzky P. (1989) The computer program "GOSOL", an instrument to increase passive solar energy use in urban planning. Proc. of 2nd European Conference on Architecture. In: Steemers T.C. & Palz W. (eds.), Science and technology at the service of architecture. Kluwer Academic Publishers, Paris, pp. 45-47 (1989).
- [4] Holland P.G., Mayer I. (1988) On calculating the position of the sun. International Journal of Ambient Energy, Vol.9, No.1.
- [5] Klein S., Beckman W.A., Duffie J.A. (1990) TRNSYS - A Transient System Simulation Program. Program Manual, Version 13.1. Solar Energy Laboratory, Madison, University of Wisconsin.
- [6] Palmiter L., Wheeling T. and Ecotope Inc. (1985) SUNCODE - PC, A Program User's Manual, Seattle, Washington.
- [7] Santamouris M. et al. (1994) Energy Characteristics and Savings Potential in Office Buildings. Solar Energy, Vol.52, Nr.1, pp. 59-66.
- [8] Yezioro A., Shaviv E. (1994) A design tool for analyzing mutual shading between buildings. Solar Energy, Vol.52, No.1, pp. 27-37.