Solar Energy Collection Analysis Tool for Conceptual Aircraft Design

A Senior Project

presented to

the Faculty of the Aerospace Engineering Department

California Polytechnic State University, San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science

by

Grant Glazebrook

December, 2012

Solar Energy Collection Analysis Tool for Conceptual Aircraft Design

Grant M. Glazebrook[1]
California Polytechnic State University, San Luis Obispo, CA, 93407

As battery energy storage and solar cell technology improve, solar aircraft are increasingly being considered for High Altitude Long Endurance missions. Although solar vehicles may theoretically remain on-station indefinitely using the sun as a power source, their design and feasibility is sensitive to mission planning details as specific as the time history of the vehicle's deck orientation relative to the sun; the energy available for capture by the on-board solar array is governed by the solar incidence angle, and at certain orientations, the vehicle may cast shadows on itself and further reduce its energy capture capabilities. To quantify these losses, a batch mode program was developed that takes the vehicle geometry and sun orientation, integrates incidence and shadow losses, and outputs an equivalent effective solar array collection area for use in a vehicle and mission analysis environment. In this paper, the need for such a tool is identified, tool methodology is described, and example output and validation cases are presented.

Nomenclature

E	=	exposure
f	=	Fresnel correction factor
G	=	solar flux constant at Earth's surface, (W/m^2)
I	=	illumination, shadow fraction
N	=	number, e.g. number of triangles
n	=	index of refraction
P	=	packing efficiency factor
R	=	distance
r	=	radius
θ	=	solar incidence angle
φ	=	refracted beam angle
γ	=	azimuthal polar angle

Subscripts

1	=	air
2	=	solar cell anti-reflective material
c	=	cell
tri	=	triangle

I. Introduction

THERE are several advantages to an aircraft that can operate at a high altitude and remain on station for a long time. In warfare, these aircraft are exceptional strategic and tactical platforms. The Northrop Grumman RQ-4 Global Hawk's role as a strategic intelligence, surveillance, and reconnaissance (ISR) aircraft has proved invaluable in Operation Enduring Freedom by allowing the US Air Force to engage more targets per sortie than ever before[2]. With a service ceiling of 60,000 feet and an endurance of 28 hours, the Global Hawk is the highest flying, highest endurance unmanned aircraft in production as of 2012. Northrop Grumman classifies this aircraft as a high altitude long endurance (HALE) aircraft. But what if the aircraft could fly higher and remain on station longer? If such an

[1] Undergraduate Student, Aerospace Engineering, One Grand Avenue, AIAA Student Member.

aircraft existed and could perform a fundamentally different mission than the Global Hawk, it would be worthy of its own classification. In reality, the definition of a HALE aircraft is very loose and the term covers a broad spectrum of altitudes and endurances.

An aircraft that could operate as an "atmospheric satellite" and remain on station for a period of weeks, months, or years could provide fundamentally different services than those rendered by any "HALE" aircraft of 2012. It is proposed that the term "HALE" be reserved for such an aircraft. In terms of military utility, a true HALE aircraft could extend the current capabilities of the Global Hawk/Predator fleet by providing surveillance and security coverage without lapse or complicated logistics.

HALE aircraft may also perform scientific or communications missions, as in the AeroVironment Pathfinder, Helios, and Global Observer. Helios was designed as part of NASA's Environmental Research Aircraft and Sensor Technology program to examine the feasibility of solar powered aircraft as long endurance, sensor-carrying platforms. As part of its environment monitoring missions, aircraft like Helios could be used to detect forest nutrient status, forest regrowth, sediment/algal concentrations, and assess coral reef health. It was also considered as a monitor for hurricane development and forest fires. As a communications platform, the aircraft could provide emergency connectivity for relief workers over areas struck by natural disasters where communications infrastructure has been destroyed. In the same situation, an onboard surveillance package could relay live video of affected areas and assist aid organizations in prioritizing and planning aid delivery[7]. The ability to have such a platform on station uninterrupted is crucial to relief efforts.

Recognizing the evolving potential of HALE aircraft, in 2008 the Defense Advanced Research Projects Agency (DARPA) Vulture program requested proposals for a HALE vehicle capable of remaining on station at 100,000 feet of altitude for 5 years at a time with a 1,000 pound payload. In keeping with the strengths of this aircraft class, the vehicle would be used to tightly circle the same geographical location and act as an unblinking "eye-in-the-sky," providing real-time imaging to aid in tactical decisions and security. Vulture also requested proposals for a vehicle capable of performing a hurricane surveillance mission with similar performance criteria.

In 2007, NASA analyzed several HALE aircraft configurations using the DARPA Vulture requirements as reference missions. The study considered lighter-than-air and heavier-than-air designs utilizing both solar regenerative and non-regenerative fuel sources. In response to Ref. 8, the purpose of this tool is to improve upon the existing methods of analysis of HALE solar aircraft so that they may be designed and benchmarked with greater accuracy and speed.

II. Background

HALE aircraft designs can be broken into two general categories: fuel-burning and fuel-retaining. By definition, endurance is the key measure of merit in HALE vehicle designs. Fuel-burning aircraft have several drawbacks in terms of endurance. While the reduction in weight during cruise is a benefit to the range and endurance of a fuel-burning aircraft, the amount of fuel required for an aircraft to remain on station for days drives the design to large fuel fractions and hence large gross weights, which presents performance and structural problems. Conceptual design sizing studies given in Ref. 3 show that no fuel-burning aircraft can remain on station for more than a few days without returning to base for a refuel.

Fuel-retaining vehicles may be powered by an onboard nuclear reactor, beam-powered propulsion, or solar power. Nuclear aircraft were briefly experimented with in the 1950's when it was shown that a pair of GE turbofan engines could be powered by a nuclear reactor. And although the largest design challenge was adequately shielding the crew from radiation, which is not a concern in an unmanned aircraft, neither the designer nor the general public is likely to feel comfortable at the thought of nuclear reactors flying above.

Research into beaming power to an aircraft from the ground is ongoing, and the two most suitable technologies appear to be laser and microwave energy transmission. Each type of energy transmission has its pros and cons. The size of the onboard receiving antennas can be of great concern if they are so large as to severely impact the vehicles aerodynamics and weight. Laser-powered systems have smaller antennas than microwave-powered systems. Beaming microwave power may interfere with satellite communications systems and the filtering or frequency restrictions that may be required by global telecommunication regulators could be a barrier to the economic operation of such a system. Both laser and microwave power is attenuated by the Earth's atmosphere and weather, and given a laser's small wavelength, it is highly susceptible to power loss due to scattering. A last major concern is safety; because of the power flux density of a laser beam, which is much higher than that of a microwave beam, any intrusion into the beam by objects, people, or animals could be very serious for the health of the intruding object. Physical laws allow the power flux density of a microwave beam to be much lower, but the risk is still similar.

These physical constraints, not to mention geopolitical concerns or any technical immaturity, limit power beaming to an area of ongoing research[4].

Solar powered aircraft are unique systems. They are much more acceptable in terms of safety and cost, they operate from a highly predictable and reliable fuel source, and if the solar array can capture more energy than is required to fly for 24 hours under worst case conditions (winter solstice, high latitude, strong headwinds, end-of-life solar/battery system efficiency, etc.), then theoretically the aircraft can remain airborne indefinitely.

In the case of a multiple year mission, mission feasibility is a more true measure of merit as opposed to endurance. The mission is deemed feasible when the aircraft exhibits an energy balance under worst case conditions. Worst case conditions include minimum solar irradiance throughout the day, strong winds, and component failures. Onboard failures are a great concern on an aircraft on a mission exceeding several weeks, since, from and engineering standpoint, failures are essentially guaranteed. The aircraft must be designed with enough robustness and redundancy to carry out its mission in the presence of one or more engine failures, solar cell failures, or payload failures. Therefore, for HALE aircraft, mission feasibility is a complex measure of merit that incorporates many factors.

The typical variables that define mission feasibility are still the same as those that define endurance, and so the solar aircraft mission feasibility is very sensitive to typical aircraft endurance performance metrics such as lift-to-drag ratio, fuel fraction, and propulsive efficiency.

The propulsive efficiency of the aircraft's electric propulsion system is essentially fixed and known by design. However, for a solar aircraft, the power source is the sun, and so the available input power varies largely with the geometry of the aircraft and its deck orientation relative to the sun. For example, the amount of power available from the sun varies with the time of year, and the aircrafts altitude, longitude, and latitude. The available solar radiation at a given longitude varies sinusoidally through the period of one day on account of the Earth's spin, and the available solar radiation at a given latitude varies sinusoidally through the period of one year on account of the Earth's tilt and revolution around the sun.

Furthermore, at certain vehicle orientations, the aircraft may cast shadows on across the solar array, which can severely reduce the solar array's energy capture capabilities. Also, it is known that the solar power collected varies sinusoidally with the angle of incidence between the solar array and the sun. In all, several geometric features govern the availability of sun radiation to a solar aircraft's array, and input power becomes a very important variable in the calculation of solar aircraft endurance.

Figure 1 shows a plot of mission feasibility as it varies with two of the most important solar aircraft design parameters: battery energy density and solar cell efficiency. This plot, from Ref. 8, suggests that the mission feasibility for a Helios-like aircraft performing the DARPA Vulture hurricane science mission is only 48% with 2007 technology. Figure 1 also shows the sensitivity of mission feasibility to solar cell efficiency. If an aircraft were designed to operate on the 110% feasible contour, even a small reduction in cell energy capture capabilities may push the vehicle into the infeasible range. A reduction in solar cell efficiency is ultimately analogous and as mathematically significant to a reduction in input power or the solar energy incident on the solar array. Therefore, the ability to accurately quantify the actual exposure of the solar array to the sun's radiant energy is paramount to calculating mission feasibility.

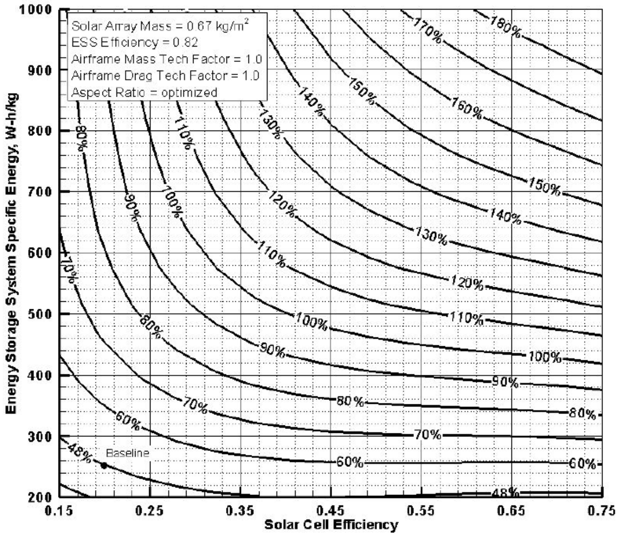


Figure 1. A solar aircraft hurricane science mission is only 48% feasible using 2007 technology[8].

On account of the importance of the time history of the aircraft-sun geometry, the contours of mission feasibility in Fig. 1 take into account energy losses due to aircraft self-shading and angle-of-incidence effects. This was done using a proprietary tool. As the analysis continued into 2010, Tom Ozoroski developed the Spiral 3 Solar Energy Analysis Tool, which succeeded the proprietary tool in use and function[12]. The Spiral 3 tool runs as a Microsoft Excel® macro and is programmed using VBA. It takes user-defined aircraft geometry in the form of rectangles, integrates solar energy losses, and outputs an equivalent solar collection area. The equivalent solar collection area is a quantity that mathematically represents the effective area that can collect solar radiation as if that area were described by a polygon in a plane normal to the incident radiation. It is essentially analogous to the reduction of the drag coefficient of an aircraft to a flat plate equivalent area.

Once the equivalent area is known, it can then be multiplied by the solar irradiance, G, (expressed in watts per unit area) at that altitude, longitude, latitude, and time of year to reveal the total input power available for collection by the solar array.

III. Solar Radiation and Computer Science Theory

In order to calculate the flat plate equivalent collection area, the incident solar energy on each cell of the solar array must be integrated. The incident solar energy is a function of shading, solar angle of incidence, and the solar cell packing structure.

A statement of the problem can be described by the double integral in Eq. 1,

$$A_{eq} = \iint_{solar\ array} E_c\, dA, \tag{1}$$

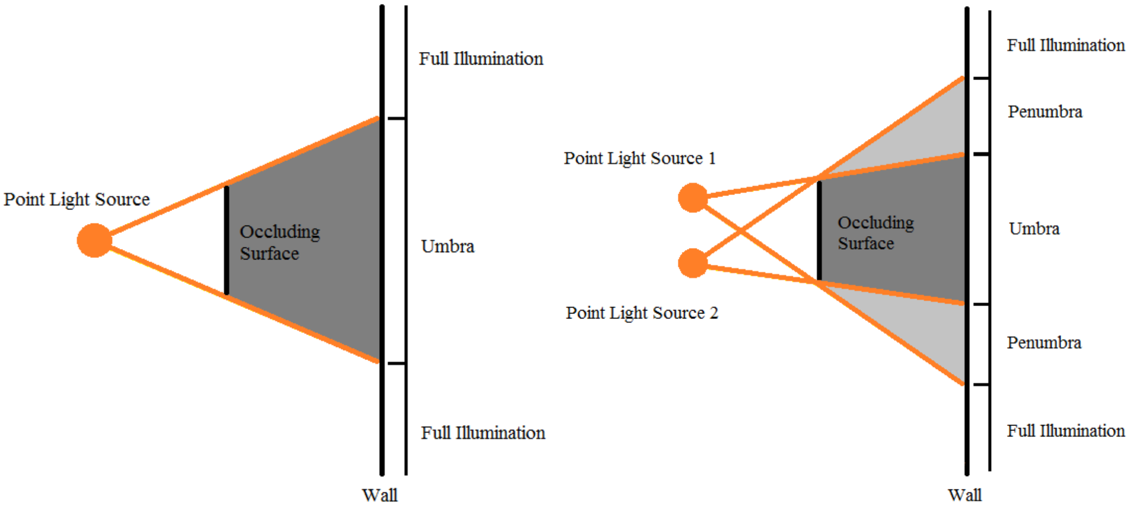
where E_c is the solar energy "exposure" at a given point on the vehicle geometry. The exposure is a non-dimensional correction factor that accounts for reductions in incident solar energy due to angle of incidence and self-shadowing.

Only a few simple equations are needed to describe angle-of-incidence effects, and packing efficiency is easily accounted for with a single correction factor, but accurately approximating the aircraft self-shadowing requires a powerful, numerical algorithm of which many exist.

A. Shadow Effects

1. Shadows

The loss of solar array sunlight exposure due to self-shadowing can be substantial at certain geometrical orientations, so accurately quantifying the effect is important. A point on a surface may be considered shadowed whenever the line-of-sight between that point and the light source is occluded by an object. This very simple definition is shown in Fig. 2(a). Here, the rays from a point light source that radiates equally in all directions are interrupted by an occluding surface before they hit the wall. The surface therefore casts a shadow based on the surface's geometry and the geometry of the sun-surface-wall system. In this case, a "hard" shadow is cast on the wall, i.e. the shadowed part of the wall has no illumination and is perfectly black. The perfectly black region is known as the umbra.



(a) Single point source.

(b) Multiple coplanar point sources

Figure 2. Multiple point sources create shadow penumbrae.

In the case of Fig. 2(a), there is a discontinuity in illumination at the wall between the umbra and the rest of the wall; the illumination is unity on one side of the boundary and zero on the other. This discontinuity represents no physical phenomenon and is merely an erroneous result of the simplicity of the model. In the real world, there is always a gradient between full illumination and zero illumination at the edge of a shadow, however minute. This gradient is generated by a light source with a finite surface area.

An area light source can be defined as the set of all points contained within a closed curve, while a volume light source can be defined as the set of all points contained within a closed surface. For the purposes of computation, an area or volume source only needs a sufficient number of points to accurately describe its radiant energy, e.g. an area light source may be represented by a finite number of coplanar point sources[9]. The set of coplanar point sources representing the area source generate the shadow gradient at the edge of a shadow, which allows for more realistic modeling. A simple version of this configuration is shown in Fig. 2(b). Here, a region of partial illumination appears between the full illumination and umbra sections of the wall due to the geometry of the system; this section is termed the penumbra. Shadows with penumbra elements are colloquially described as "soft" shadows.

Solar cells in the penumbra region may still collect energy from the sun, albeit at a fraction of their potential at full illumination. This fraction is exactly equal to the visible area of the light source divided by the total area of the light source. To maximize the accuracy of a shadowing algorithm and take advantage of regions of partial illumination, it is important to be able to quantify the effect of penumbra elements.

Considering only shadow effects, the solar cell energy exposure may be defined as

$$E_c = I \tag{2}$$

where E_c is the cell exposure, and I, the illumination, is the fraction of the area of the light source visible from the given cell or point on the array. I has a domain of [0,1].

2. Sun Model

Given that the sun is, on average, 93 million miles away from Earth, its entire volume can be represented as its circular cross section through its center along a plane perpendicular to the line of sight from the origin without a loss in accuracy. This reduces the sun's complexity from that of a volume source to an area source; fewer points are needed to accurately describe it. This sun may be modeled as having local coordinates (r,γ) with the center of the sun located at (0,0). The area source may be modeled as a number of point sources in a variety of ways; the most popular two ways are stochastically and deterministically.

In the stochastic model of the sun, a random number generator can be used to spread points over the region defined by ([0,r_s],[0,360]). Given a sufficient number of points, the disc can be modeled accurately. In the deterministic model, the points are spread out with more care. For reasons mentioned in Section IV, it is advantageous for each point on the sun to represent an equal amount of sun area. A method described in Ref. 14 does just this, and an example is shown in Fig. 3 with a preview of the subdivision shown in the first quadrant of the circle.

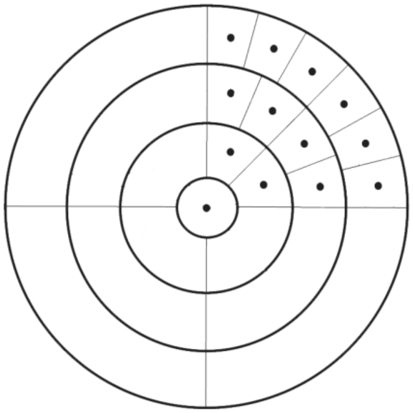


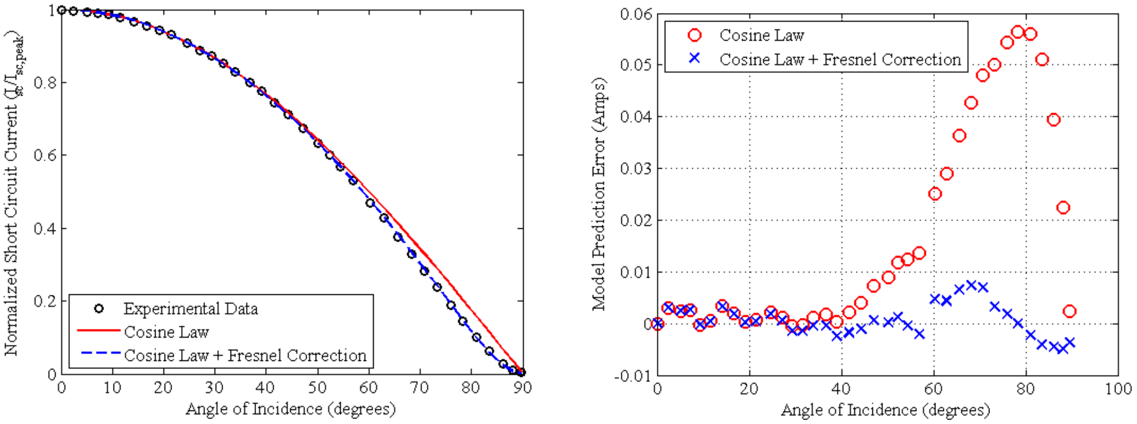
Figure 3. A deterministic area light model of the sun[14].

B. Angle-of-Incidence Effects

Photovoltaic cells only convert energy that arrives perpendicular to the cell's surface, and therefore the angle of incidence between the incomming irradiance and the cell's normal vector is important. Simple geometry reveals that irradiance is governed by Lambert's cosine law, which defines irradiance at a point on a surface as proportional to the cosine of the angle of incidence[13].

A comparison of solar energy collected by a solar cell to the cosine law prediction is shown in Fig. 3(a). Here a disparity can be seen between Lambertian theory and experimental data at incidence angles above forty degrees. This difference is largely due to the optical effects of the solar array cover glass.

The principle physical phenomenon responsible for the difference at higher angles is light reflection off of the solar cell cover glass or anti-reflective material. The effect is termed Fresnel reflectivity. The Fresnel equations account for the reflection of a portion of the incident light at the interface between two optical media having different indices of refraction.



(a) Absolute Comparison[2]

(b) Error comparison

Figure 4. The Fresnel equations allow for greater accuracy at higher angles of incidence.

With the cosine law and Fresnel equations, the solar exposure of the solar cell becomes

$$E_c = I * \left(1 - \mathrm{f}(\theta)\right) * \cos(\theta), \tag{3}$$

$$f(\theta) = \frac{1}{2}\left[\frac{tan^2(\theta-\varphi)}{tan^2(\theta+\varphi)} + \frac{sin^2(\theta-\varphi)}{sin^2(\theta+\varphi)}\right], \tag{4}$$

where f is the fresnel reflectivity factor, θ is the angle of incidence, and φ is the refracted beam angle, which is defined as

$$\varphi = \arcsin\left(\frac{n_1}{n_2}\sin(\theta)\right), \tag{5}$$

where n_1 is the index of refraction of air (taken to be unity for air and vacuum) and n_2 is the index of refraction of the solar array cover glass[11]. The latter index of refraction depends largely on the anti-reflective material used to coat the cover glass and the glass media itself. Anti-reflective materials are used to prevent solar energy from reflecting off of the cover glass and help the system achieve efficiencies closer to that of the ideal cosine law case.

The cosine law with the Fresnel correction model as it compares to experimental data is shown in Fig. 4(a). The Fresnel equations more appropriately capture the underlying optical physics at high incidence angles and, as a consequence, model end-behavior more realistically. This is confirmed in Fig. 4(b), which compares each model's predictive capabilities and shows that the Fresnel corrections minimize error at high solar incidence angles as compared to the pure cosine law model. The experimental data in Fig. 4 was normalized to by the value at peak output.

The error in the Fresnel correction data set represented by the x's may be due to experimental error, especially at large incidence angles; the cosine value changes very rapidly at large incidence angles so small measurement errors are magnified[3].

From the perspective of Eq. (3), it is difficult to see why an anti-reflective coating is beneficial to system performance; a non-zero Fresnel reflectivity factor, r, reduces the exposure of the solar cell. The Fresnel reflectivity factor has a range of values from zero to unity corresponding to indices of refraction from unity to infinity, respectively, and so it seems that it would be most advantageous to choose an index of refraction of unity, i.e., have no anti-reflective material at all. The advantage of anti-reflective coatings presents itself at the photovoltaic cell level, which the mathematical modeling does not consider.

Eliminating the anti-reflective coating would not increase cell exposure since the protective glass underneath it has its own index of refraction that is greater than one. In this case, the Fresnel reflectivity correction factor is greater than zero anyways. However, coating the glass with a material can have the effect of reducing the overall losses due to reflectivity as light passes from air to anti-reflective material to glass. For a given dielectric anti-reflective coating sandwiched between two dielectric media (air and glass), the index of refraction of the anti-reflective coating can be chosen based on the index of refraction of the air and the glass such that reflectivity is reduced for all wavelengths of light. The thickness and index of refraction of the coating are often chosen to place the minimum reflectance near 0.6 microns of wavelength, which is where peak solar irradiance in the solar spectrum occurs[11]. A comparison of reflectivity for bare glass to coated glass is shown in Fig. 5. Based on the chosen thickness and index of refraction, the reflection can be nullified for a given wavelength and angle of incidence. Creating multiple layers of coatings can produce multiple minima as opposed to the single minimum shown in Fig. 5.

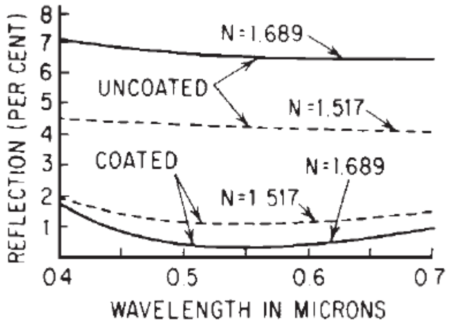


Figure 5. The reflectivity of a single-layer MgF_2 coating, compared with the reflectivity of uncoated glass[10].

While equations can be derived for the total reflectivity of a coated glass system, considering only the reflectance of the coating produces results accurate to within 1%, as seen in Fig. 4(b). For this reason, the model need only consider the index of refraction of the anti-reflective material.

C. Solar Cell Packing Efficiency

A solar panel is comprised of individual solar cells held together by a structure. The face of the solar panel is not completely covered in cell material capable of converting sun energy in to electricity; there is some structure between the individual solar cells. The fraction of actual collection area to the total area of the array is herein termed the “packing efficiency.” For a given amount of solar array area, the packing efficiency further reduces the array exposure in accordance with,

$$E_c = P * I * \left(1 - \mathrm{f}(\theta)\right) * \cos(\theta), \tag{6}$$

where P is the packing efficiency that has a domain of [0,1].

D. Computational Methods

Given a geometry and a set of coplanar point light sources arranged in a concentric circle pattern, shadow and solar incidence angle losses may be calculated and stored for each geometric orientation relative to the sun. The bulk of this computation is spent determining shadow location and intensity.

1. Angle of Incidence

Angle of incidence can be calculated using the dot product of two vectors: the geometric primitive’s surface normal and the vector from a point on the geometric primitive to the center of the sun. In the case of the latter vector, the point could be the centroid of the primitive. Angle-of-incidence effects can be computed with minimal computational effort.

2. Methods of Shadow Calculation

With regards to computer graphics, the definition of a shadow is very loose. In most circumstances, a shadow *is* due to an opaque body that prevents light rays from permeating regions in space. However the body may not be entirely opaque: it could be transparent or translucent; the lighting may be indirect due to reflection, scattering, and diffraction; and changes in atmospheric density can produce volumetric shadows.

In response to the many shadow phenomena, computer shaders and their algorithms – the programs that calculate the appropriate levels of light and darkness within a scene – have evolved with ever greater complexity. In general, the goal of the shadow algorithm is the same: to find, for all surfaces, the amount of light received from a particular light source. Figure 6 shows a general taxonomy of shadow generation algorithms[15].

Figure 6 presents shadow algorithms as falling under three general classes: object-based, image-based, and hybrid. Object-based algorithms perform their work on the objects before the image frame buffer is populated. These techniques often iterate over all of the objects in the scene. The accuracy of the geometry of the objects is more important than the resolution of the frame buffer to the correct display of an image. Image-based algorithms perform their work as objects are being converted into pixels in the frame buffer. In these algorithms, the resolution of the frame is very important to the accuracy of the image since work is performed on a pixel-by-pixel basis. Hybrid algorithms mesh features of both types of algorithms in a wide variety of ways. Figure 6 does not represent a complete taxonomy of all currently existing shadow algorithms; it only includes the most popular ones since state-of-the-art research is often focused on blending the aforementioned methods or accelerating their computation. A survey of the shadow algorithms presented in Figure 6 is given in Ref. 12.

For simplicity, the definition of shadows has been limited to point or area light sources and opaque objects, from which penumbra and/or umbra shadow regions can be generated. These assumptions greatly reduce the complexity required of the shadow algorithm. For brevity, the only method discussed here will be ray tracing, which ultimately was chosen as the shadow calculation algorithm.

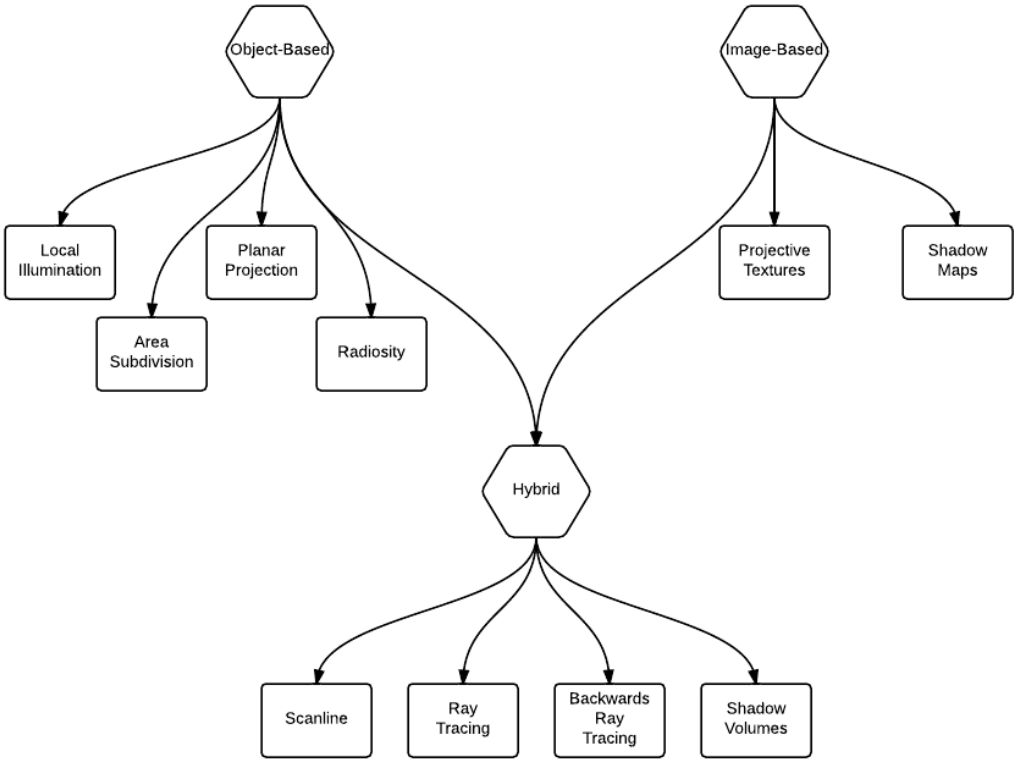


Figure 6. Taxonomy of popular shadow algorithms[15].

3. Ray Tracing

Like any other method, ray tracing is used in computer graphics to generate a two dimensional image from a three dimensional scene. A ray tracing algorithm begins at a virtual “eye” (representing the eye of the viewer) and emits a ray from the eye through a point (pixel) in a closed planar surface representing the computer screen. The ray then goes on to intersect objects in the environment and information about each intersection (texture type, shadowing, color, diffuse or reflective light, etc.) is stored along the way. A simplified visual representation of this process is shown in Fig. 7. Because ray tracing algorithms closely mimic optical physics, the result is a very realistic image.

Ray tracing is most often implemented using triangles as geometric primitives to describe scene geometry. The overwhelming majority of the computation involved in ray tracing is, therefore, calculating the intersections of rays and triangles; in order to properly map the scene to the image plane or “screen,” the algorithm must be able to properly predict which triangles are visible and which aren’t.

On account of the relatively large amount of time spent on ray-triangle intersections and the general requirement that *multiple* rays per pixel must be analyzed for an accurate, smooth image, a number of ray tracing acceleration techniques have been developed to increase the speed of the algorithm.

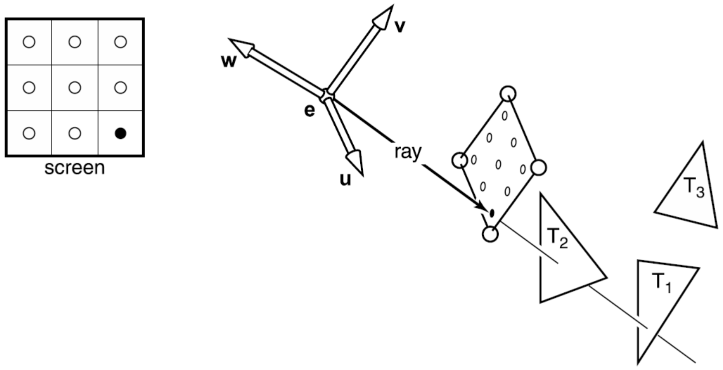


Figure 7. A general ray tracing setup. Camera coordinates (u,v,w) with the "eye" at the local origin are used to generate the screen. Rays are cast from the eye through sample point on the screen to map the geometry to the screen. In this example, the geometry is represented by triangles, T1 – T3[9].

4. Ray Tracing Acceleration Techniques

There are several means of decreasing the execution time of a ray tracer. In a naïve approach, the time required to execute one ray-triangle intersection scales linearly with the number of triangles in the scene. Acceleration techniques attempt to allow the algorithm to scale sublinearly with respect to the number of triangles. A general taxonomy of ray tracing acceleration techniques is given in Fig. 8, and a brief survey of each type is given in Ref. 6.

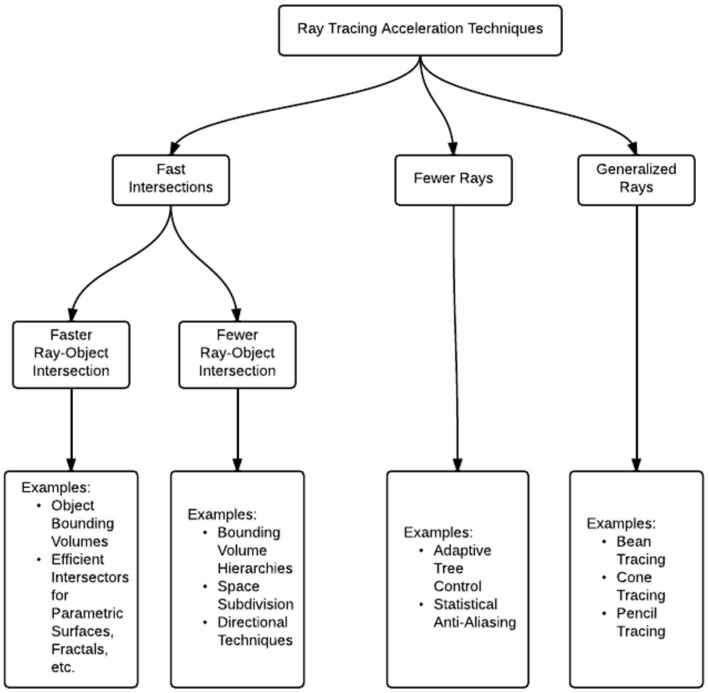

Ray Tracing Acceleration Techniques
Fast Intersections
Fewer Rays
Generalized Rays
Faster Ray-Object Intersection
Fewer Ray-Object Intersection
Examples:
• Object Bounding Volumes
• Efficient Intersectors for Parametric Surfaces, Fractals, etc.
Examples:
• Bounding Volume Hierarchies
• Space Subdivision
• Directional Techniques
Examples:
• Adaptive Tree Control
• Statistical Anti-Aliasing
Examples:
• Bean Tracing
• Cone Tracing
• Pencil Tracing

Figure 8. Taxonomy of ray tracing acceleration techniques[6].

Bounding volume hierarchies are popular and efficient means of reducing the execution time of a ray tracer. A bounding volume hierarchy (BVH) takes the form of a tree structure. Each geometric object in the scene is wrapped in its own bounding volume that forms a leaf node on the tree. Groups of bounding volumes can be bounded together by a larger bounding volume that then forms a node on the tree that is a parent in relation to the geometric objects it contains. This process can continue until a bounding volume is defined such that it encompasses the entire scene. An example of such a tree structure is shown in Fig. 9.

A popular BVH class is the axis-aligned bounding box (AABB). A AABB structure takes the triangular mesh and completely encloses it in a bounding volume that is then subdivided into smaller boxes in a recursive fashion until each bounding box contains a user-defined amount of triangles. This concept is illustrated in Fig. 10.

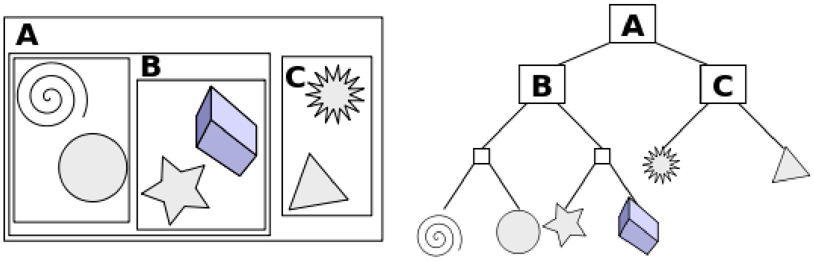
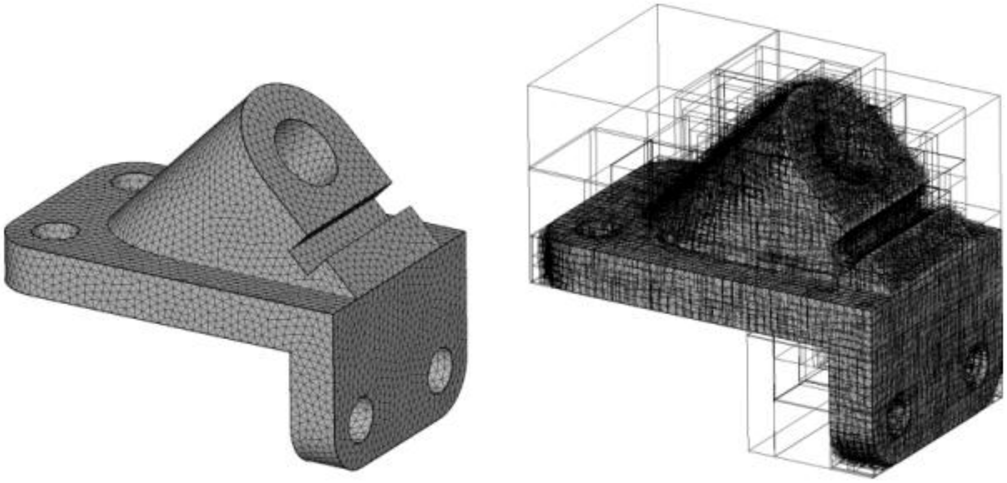


Figure 9. An example of a bounding volume hierarchy.

The AABB tree structure accelerates line-triangle intersection queries by reducing the amount of triangles that need to be tested for intersection. Because the tree structure incorporates nodes and parent-child relationships, if a line does not intersect a box (node) within the bounding volume, its child elements do not need to be tested. The bifurcation of the tree structure reduces the execution time of a single ray-triangle intersection (described in big O notation) from O(N_{tri}) to O(log(N_{tri})), where N_{tri} is the number of triangles in the three dimensional environment.



(a) Original triangular mesh .

(b) Mesh with AABB tree structure visible

Figure 10. The AABB tree subdivides the bounding box containing a triangular mesh[1].

IV. The Solar Energy Collection Analysis Tool (SECAT)

SECAT takes an input geometry in the form of a triangular mesh with solar cells marked and calculates the equivalent solar collection area for a given sun azimuth and orientation.

SECAT was designed with the philosophy that the tool should be fast, accurate, open source, and able to handle general geometries. The design philosophy was used to guide the development of the tool to one that improved upon the features of the Spiral 3 Solar Energy Analysis tool.

To accommodate the ability to handle general geometries, SECAT uses triangles as geometric primitives and therefore supports a wide range of CAD programs that can represent geometry as triangular meshes. On account of the alignment of its underlying algorithm with optical physics and ease of implementation, ray tracing was deemed perfect for application in SECAT. To ensure a fast implementation, SECAT was programmed in C++ to minimize computational overhead and improve cross-platform accessibility.

Since SECAT is programmed in C++, existing libraries of data structures and algorithms were used to improve the quality of the program. Ultimately, the Computational Geometry Algorithms Library (CGAL) satisfied all elements of the design philosophy; it is open source, employs C++ for speed, and includes triangle-based data structures and ray tracing-friendly algorithms for accuracy.

A. Interface and Inputs

SECAT is run by calling the executable from the command line. The executable itself has no inputs and instead reads input information from an organized text file. This text file includes the name of the vehicle geometry file, the range of sun azimuths and elevations to be analyzed, the fidelity of the sun model, solar cell packing efficiencies, and solar component identifiers. An example text file is shown in Appendix B. SECAT is a command line executable that uses input files to make it easy for other programs to call SECAT in a computerized multidisciplinary design environment.

SECAT relies on a triangular mesh representing the vehicle geometry as an input. One way to obtain such a mesh is by using Open Vehicle Sketch Pad (OpenVSP), an open source parametric geometry tool used for rapid aircraft geometry generation. Open VSP can output aircraft geometry in the form of a .tri file from which data can be easily parsed within SECAT. Any program that can export .tri files may be used in place of OpenVSP. Input geometry, from an aircraft point of view, is defined as: +x from nose to tail, +y out of the right wing, +z upward. Sun azimuth and elevation is defined in the input geometry's body coordinates. Sun azimuth ranges from 0 to 360 degrees counterclockwise around the +z axis, and sun elevation ranges from -90 to 90 degrees where 0 degrees places the sun coplanar with the plane defined by the x and y axes and 90 degrees aligns the sun with the +z axis.

Given an input file and a triangular mesh, SECAT then uses CGAL's triangle data structures to store the mesh in an AABB tree, which accelerates sunray-triangle intersection queries. Then the tree may be passed to one of two equivalent area calculation methods. Each method is essentially an approximation of Eq. (1) where E_c is modeled as shown in Eq. (6). A comparison of the two methods is given in Accuracy and Performance.

In both methods, the sun is generated as discussed in Section III; it is important for points to be distributed such that each point represents the same amount of sun "area" so that each sun ray represents the same amount of power.

B. Shadow Ray Method

1. *Overview*

SECAT's shadow ray method is closely akin to an object-based shadow algorithm. A general setup is shown in Fig. 11. The shadow ray method is termed so because of the way it calculates shadows. The method generates line segments representing sunrays from the geometry surface point of interest to each point light source. The shadow fraction, I, is then the number of unoccluded line segments divided by the total number of line segments cast. In this method, each line segment must be checked against each triangle in the vehicle's 3D triangular mesh for intersection. However, once *any* intersection is detected, the computation may stop as that segment must be occluded.

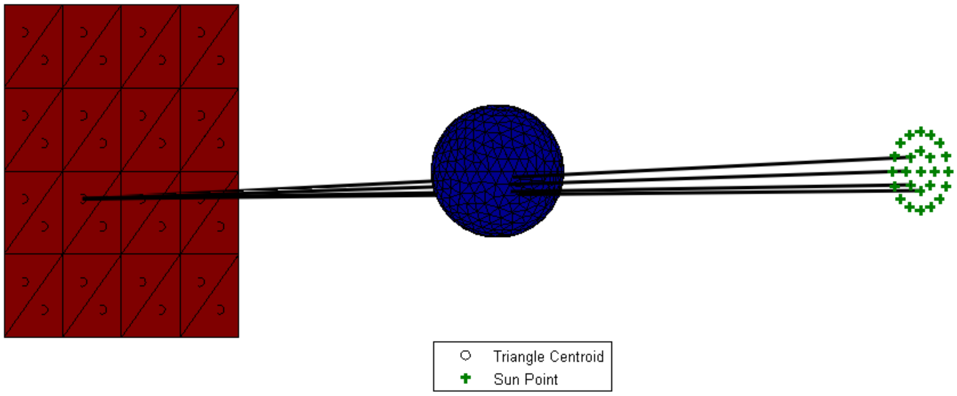


Figure 11. A visual representation of the shadow ray method. Rays originate from the centroid of each solar triangle and point to each sun point.

As cells are represented as triangles, the equivalent solar collection area of a given triangle is the triangle's area corrected for Fresnel reflectivity and shadow effects. The total equivalent solar collection area for the entire input geometry is then the sum of the equivalent solar collection areas of all solar triangles in the model. A pseudocode description of the shadow ray method is detailed in Fig. 12 below. The pseudocode may be run for a set of sun azimuths and elevations, and the resulting equivalent areas are written to a delimited text file in matrix form.

```
generate sun
for each solar triangle
    for each sun point
        define a segment between the solar triangle centroid and the sun point
        find the angle of incidence between the solar triangle normal and segment
        if the triangle normal faces away from the sun
            break (to next solar triangle)
        end if
        if the segment intersects the vehicle geometry (segment is dark)
            continue (to next sun point)
        end if
        calculate cosine and Fresnel reflectivity (CFR) correction factor
        update fresnel sum counter with CFR factor
    end for
    calculate triangle area
    lookup packing efficiency (PE) for solar triangle
    update area sum counter with product of triangle area, CFR factor, and PE factor
end for
return area counter
```

Figure 12. A pseudocode description of SECAT's shadow ray method. The above pseudocode is for one sun-to-deck orientation.

2. *Time Scaling*

The execution time of the shadow ray method scales with the number of points used to represent the sun, the number of solar triangles in the model, and the number of total triangles in the model. This relationship, in Big O notation, is shown in Eq. 7. Note that the number of solar triangles will generally scale linearly with the total number of triangles, so N_{tri} may be used in place of the exact number of solar triangles.

$$t_{execution} = O(N_{tri} * N_{sun} * \log(N_{tri})) \tag{7}$$

In Eq. (7), N_{sun} is the number of points used to describe the sun using the method of Ref. 14.

3. *Accuracy*

The accuracy of the solution can be increased by describing the geometry with more triangles, N_{tri}, which presumably would allow for a more accurate representation of geometric curvature. This would all for shadows to be modeled more closely to those that would appear on the actual geometry and would also fine tune angle-of-incidence effects. Increasing N_{sun}, serves only to increase the accuracy of the shadow effects; more points on the sun means penumbrae can be modeled better.

4. *Advantages and Disadvantages*

The main advantage to the shadow ray method is its ease of implementation on account of its simplicity. No point of intersection needs to be generated and stored since segment intersection queries are forcibly defined from the triangle centroid.

However, the shadow ray method has some disadvantages. In order to increase the accuracy of the solution, the total number of triangles in the model must be increased. This increases the number of intersection tests run and also increases the time it takes to perform an intersection test, albeit the execution time of a single intersection test scales only logarithmically with the number of triangles in the model. In a multidisciplinary design environment, it would be undesirable and inconvenient to require the complete regeneration of a model.

C. Viewport Method

1. *Overview*

SECAT's viewport method is more closely akin to a traditional ray tracing method and is an image-based solution. In general terms, it works by projecting the image of a model as seen by the sun onto a plane normal to the sunrays. Thus, the area of the image on this plane is the equivalent solar collection area once adjusted for Fresnel reflectivity. A general setup of the method is shown in Fig. 13.

The viewport is defined by projecting image of the model onto a plane normal to the line connecting the sun with the origin. The viewport is placed such that it is between the model and the sun. By projecting the image of the model onto the viewport plane, the viewport can be sized intelligently so that most of the rays cast through it will intersect the geometry.

From each sun point, rays are cast through the viewport and are intersected with the vehicle geometry. If the first intersected triangle is a solar cell, the flat plate equivalent area is incremented in such a way as to account for Fresnel reflectivity and shadow effects. If the first intersected triangle is not a solar cell, then there is no effect on the equivalent solar collection area. This process is repeated for all combinations of sun points and viewport points.

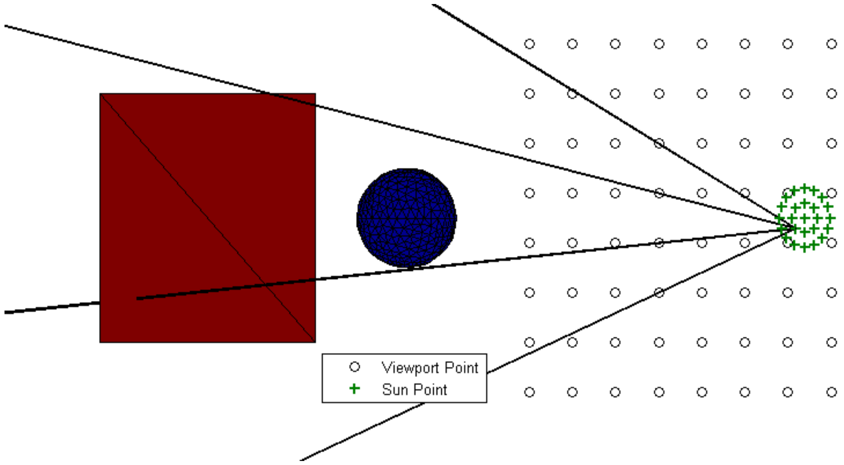


Figure 13. A visual representation of the viewport method. Rays originate from the sun points and point to each viewport point.

The mathematical model by which shadows are accounted for is as follows: when an intersection with a solar triangle is detected, the equivalent area is incremented by the viewport pixel area divided by the number of points on the sun. Thus, the shadow fraction is accounted for by determined by dividing the number of sun points that project the image of a given solar triangle onto the viewport plane with the total number of sun points.

The distinguishing feature of the method is the use of the viewport, which is identified in Fig. 13 and is analogous to the "screen" in a typical raytracer. Ultimately, the viewport is used to capture the projection of the solar cells onto the viewport plane. The projection area, corrected for Fresnel reflectivity, is the equivalent solar collection area. A pseudocode description of the viewport method is detailed in Fig. 14 below. The pseudocode may be run for a set of sun azimuths and elevations, and the resulting equivalent areas are written to a delimited text file in matrix form.

```
generate viewport, sun
calculate viewport pixel area, pixel area scaling factor
for each sun point
    for each viewport point
        cast a ray from the sun point through the viewport point
        if first intersected triangle is a solar triangle
            find sunray angle of incidence with solar triangle
            calculate Fresnel reflectivity (FR) correction factor
            lookup packing efficiency (PE) for solar triangle
            update sum counter with product of FR and PE factors
        end if
    end for
end for
return product of sum counter, viewport pixel area, pixel area scaling factor, and reciprocal of
    number of sun points
```

Figure 14. A pseudocode description of SECAT's viewport method. The above pseudocode is for one sun-to-deck orientation.

2. Time Scaling

The viewport method's execution time scales with the number of viewport points, the number of points used to represent the sun, and the total number of triangles in the model. This relationship, in Big O notation, is shown in Eq. 8.

$$t_{execution} = O(N_{viewport} * N_{sun} * \log(N_{tri})) \tag{8}$$

This relation is comparable to the time scaling of the shadow ray method.

3. Accuracy

As with the shadow ray method, increasing the number of triangles or points on the sun increases the accuracy of the solution. Further solution refinement may be obtained (up to the limit of the accuracy of the geometric model) by increasing the number of viewport points, $N_{viewport}$. By increasing $N_{viewport}$, the solar cells can be mapped to the viewport plane with a higher resolution.

4. Advantages and Disadvantages

One advantage of the viewport method is its ability to intersect the same triangle multiple times by means of manipulating the number of viewport points. In contrast to the shadow ray method, which only analyzes one point per triangle, a sufficient number of viewport points can mean the same triangle is intersected multiple times. This gives rise to greater accuracy without regenerating the model.

Like the shadow ray method, the viewport method only tests solar triangles that face the sun; but since rays are cast from the sun, there is no need to waste any processing effort eliminating back-facing triangles as is necessary in the shadow ray method.

This method has the disadvantage that some of the rays cast will be "wasted," i.e. the geometry of the sun and viewport almost guarantees that some rays cast will not intersect the model at all. However, the use of an axis-aligned bounding box to accelerate intersection queries means that rays that do not intersect the model at all are generally easy to process, since pass through the empty space close to the model that is described by relatively large sub-boxes. Again, the larger the relative size of a sub-box, the faster the tree can be traversed to see if it contains any triangles.

D. Spiral 3 Solar Energy Analysis Tool Method

1. Overview

The method employed by the Spiral 3 tool is worth discussing since it was the inspiration for the development of SECAT. A user of Spiral 3 first defines aircraft geometry in the form of hand-inputted points that define rectangles. Each rectangle is considered to be a "part." The user then defines the number of "segments" to divide the rectangle into and then the number of "bins" to divide each "segment" into. Each part is given a packing efficiency for each side of the part face. The code accounts for Fresnel reflectivity, so an index of refraction for the solar cell cover glass material is required.

Given these inputs, the program may then be executed per the pseudocode in Fig. 15. Note that the pseudocode presented is very generalized, and there are additional computational elements in the actual code that help to speed up the execution, but for clarity these are not presented here. Also, a "shadow-casting" part is defined as any part that is not the current "shadow-target" part, which is the part currently being evaluated for shadowing and angle-of-incidence effects.

```
determine the relative orientation and spatial arrangement of every part w.r.t. every other part, vehicle axes, and sun axes
calculate cosine and Fresnel effects on each strip and bin on each part (CFR)
identify which parts face the sun and which face away from the sun (sun-facing panels are grouped as “shadow-target” parts)
for each “shadow-casting” part (all parts)
    trace the sunlight ray vectors from the sun through each corner of the shadow-casting part
    for each shadow-target part
        project the sunlight ray vectors to the plane containing the shadow-target part (create shadow vectors)
        determine the shadow cast distance, calculate darkness variations due to atmospheric scattering (ASC)
        for each projected shadow
            determine the intersection points of the shadow-target vector and the shadow vectors
            create shadowing function $S(w,l)$
            for each segment
                for each bin
                    determine shadowing correction factor (SCF)
                    update sum counter with product of SCF, ASC, CFR factors
                end for
            end for
        end for
    end for
end for
return sum counter
```

Figure 15. A pseudocode description of the Spiral 3 method. The above pseudocode is for one sun-to-deck orientation[12].

2. Time Scaling

The Spiral 3 method’s execution time scales with the square of the number of parts, the number of segments per part, and the number of bins per segment. This relationship, in Big O notation, is shown in Eq. 9.

$$t_{execution} = O(N_{bin} * N_{segment} * {N_{part}}^2) \quad (9)$$

Upon inspection and comparison, this relation does not scale well relative to SECAT’s shadow ray method or viewport method. Time scaling increases with the square of the number of primitives defining the geometry; the sub-linear scaling of the viewport and shadow ray methods is far more preferable than the super-linear scaling of Spiral 3. For simple geometries that can be easily described with rectangles and have very few parts, this behavior is not particularly debilitating. But if curvature needs to be accurately represented (e.g. the top of a curved aircraft wing or fuselage is coated with thin film solar cells) by a large number of geometric primitives, then this method will scale poorly as compared to the viewport or shadow ray method.

3. Accuracy

As in SECAT’s methods, Spiral 3 accuracy is dependent on the resolution of the geometry and the number of rays traced. The more geometric primitives (N_{parts}) used to describe the geometry, the higher the accuracy. Each part can then be subdivided further by increasing N_{bins} and $N_{segments}$ to give better shading accuracy across the part.

4. Advantages and Disadvantages

The main advantage of the Spiral 3 method is its user-friendly interface. The user may use Microsoft Excel®, a very popular spreadsheet program, to input data and view the results.

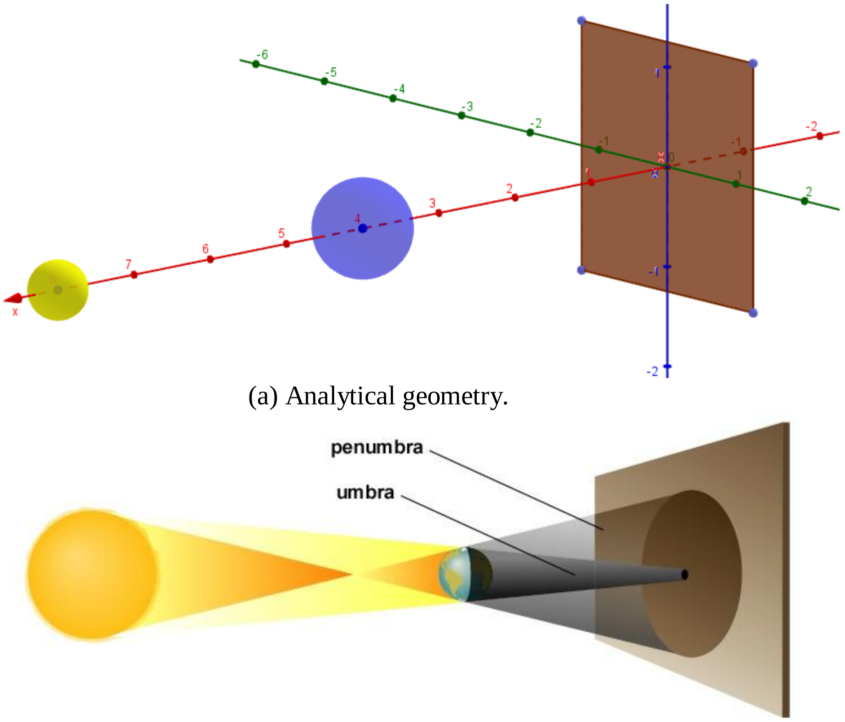
However, this advantage comes at the cost of open-source accessibility, speed, and generality. The Spiral 3 tool can only accept rectangles as geometric primitives. While this does not require the program to make highly inaccurate approximations, it is very cumbersome to input and removes the ability to easily handle curvature and more complex geometries.

Also, by using user-defined rectangles as inputs and forcing each part to be divided up into the same number of segments and bins, the program ignores a great advantage of curvature-based meshing used in programs like OpenVSP. Tight curvature must necessarily described by a large number of triangles, but in planar regions of the geometry, OpenVSP allows the triangle size to grow, reducing the number of primitives required to define the model.

V. Measured Accuracy and Performance

A. Accuracy

To ensure the accuracy of SECAT, its results were compared to the analytical solution of the simple sun-sphere-plate setup shown in Fig. 16(a). For reference, this setup is the same as is shown in Figs. 11 and 13. The analytical solution was derived by integrating the shadow and angle-of-incidence effects over the plate; this derivation is shown in Appendix A. By using a spherical occluding object and a spherical light source, the shadows cast on the plate are circular as shown in the general example in Fig. 16(b). The geometry of the sun-sphere-plate system has been chosen to maximize shadowing of the plate so that the accuracy of the shadow calculations can be rigorously tested; the plate is sized so that circular penumbra shadow region on the plate is tightly circumscribed by the plate, thus minimizing regions of full exposure.



(a) Analytical geometry.

(b) Example of sun-sphere-plate system shadow geometry[5].

Figure 16. A sun-sphere-plate system used to verify the accuracy of SECAT.

Figure 17 shows that both viewport and shadow ray SECAT methods converge to within 1% of the analytical solution with a sufficient number of rays traced. The error present is introduced by modeling the sun as an area light source as opposed to a volumetric light source. This specific type of error is inversely proportional to the distance between the light source and the solar cell. In the release version of SECAT, this distance is on the order of 10^7 units, so the error has no appreciable effect in SECAT's normal analysis mode.

Figure 17 shows that the viewport method exhibits a noisier convergence than the shadow ray method. This is transient behavior and it dies out with a significant number of rays traced. Because the viewport and the sun are modeled as deterministic patterns, the rays-triangle intersection points are also patterned across the plate, with some areas having a high concentration of points and others having a low concentration of points. As intersection points are not distributed evenly across the plate, increasing the number of viewport points tends simultaneously distributes points and concentrates them further and thereby simultaneously increases and decreases accuracy at the same time. Ultimately, increasing the number of viewport points serves to increase the accuracy, albeit it takes much longer for the transient error to die out as compared to the shadow ray method. As would be expected, this type of error transient does not manifest itself when the sun is modeled as a single point; the method more smoothly converges on the solution.

Note that for the purposes of theoretical analysis, a special definition for the number of rays traced was created for the viewport method. The viewport method will cast rays that do not intersect the geometry; these "wasted" rays were not included in the total count of rays. This was done because it is possible that the viewport may be defined so intelligently that all rays cast through it end up intersecting only solar cells on the model. This would be the limiting case for the theoretical performance of the viewport method.

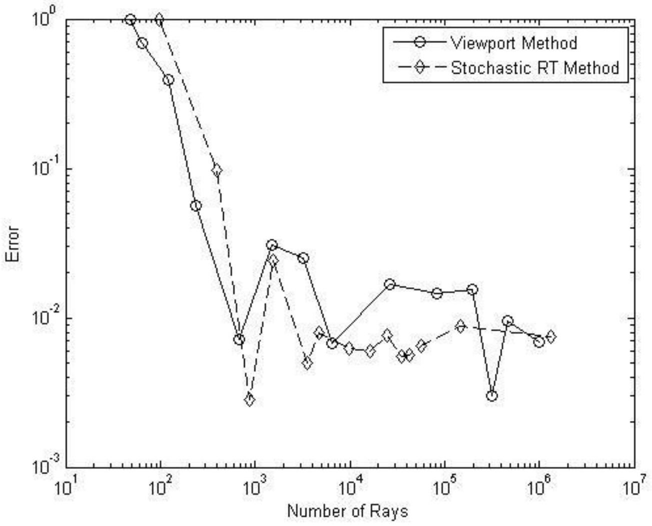


Figure 17. Both SECAT methods converge to within 1% of the solution to the sun-sphere-plate system.

B. Performance

Figure 17 also shows that the viewport and shadow ray methods require a comparable amount rays to achieve a given level of accuracy. Therefore, comparing the *speed* of convergence requires an examination of the time scaling of each method. As shown in Eqs. (7) and (8), the number of rays traced per second varies directly with N_{tri} and N_{view} in the respective methods. And while tracing a given number of rays does yield similar accuracy, it does not ensure similar time scaling, since increasing N_{tri} also increases the number of triangles in the AABB tree in the shadow ray method. All else being equal in implementation, the viewport method should arrive at results of a given accuracy faster than the shadow ray method.

SECAT also receives a substantial performance boost from parallel processing. Parallelization is done at the outermost loop over the range of input elevations, i.e. there is only a performance boost in the case of multiple input elevations. This is done to maximize the use of the processor, since if a single orientation (inner loop) was parallelized, the thread team would have to communicate via counters, which would lead to thread collisions and wasted processor time.

All data that is shared by the thread team is read-only thread safe, so there is no opportunity for one thread to interrupt another thread's access. Since some elevations can be more computer-intensive to calculate than others, thread scheduling is done dynamically during runtime as opposed to statically at the start. This ensures that no one thread gets stuck for too long on an exceptionally difficult chunk of analysis while the other threads wait for it to finish. Such a scenario usually takes place at the last few moments of execution. On account of these features, SECAT execution time decreases in direct proportion to the number of active cores.

VI. Example Results

As an example of its use on solar aircraft, SECAT was used to evaluate the solar collection capabilities of the solar array onboard Icaré 2, a solar-powered sailplane designed and built by the University of Stuttgart. The aircraft and the location of its solar cells can be seen in Fig. 18. A Spiral 3 model of the Icaré 2 is shown in Fig. 19(a), and an OpenVSP model of the aircraft is shown in Fig. 19(b). Both models were generated from known specifications and three-view drawings.

Per the global coordinate definitions defined above, the Icaré 2 exhibits symmetry about the x-axis, so azimuth was evaluated from 0 to 180 degrees. Furthermore, there are no solar cells facing the –z direction, so sun elevation was run through the range 0 to 90 degrees. SECAT's shadow ray method was used to analyze the equivalent solar collection area over the aforementioned range of solar azimuths and elevations, and the results are presented in Fig. 20. Furthermore, the Spiral 3 tool was also used to analyze the equivalent solar collection area over the same range of sun azimuths and elevations, and its output is also presented in Fig. 20 for comparison. Note that on account of the way OpenVSP defines aircraft geometry, zero degrees azimuth provides the viewer with a tail-on view of the aircraft while 180 degrees azimuth gives a head-on view of the aircraft.

Figure 20 shows good agreement between the two tools at sun elevations above 15 degrees. The general trend suggests that at a given elevation, more energy can be captured if the sun is facing the tail as opposed to the nose. This is because the solar cells on the Icaré 2 are mostly located aft of the quarter chord of the wing, which means the most of the cells' normal vectors are canted slightly towards the rear of the aircraft.

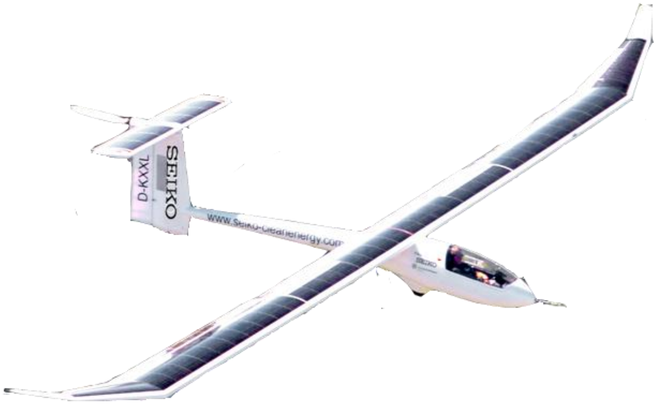
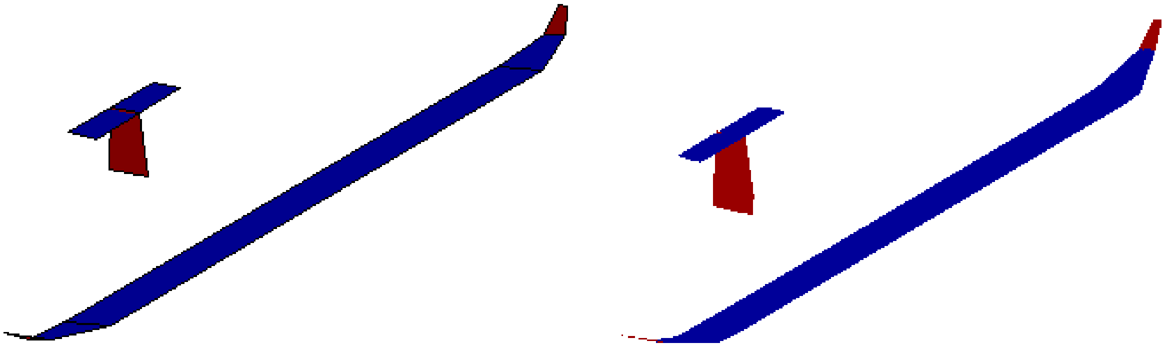


Figure 18. The Icaré 2 solar-powered sailplane. Dark areas on top of the wing and horizontal stabilizer are solar cells.

Below 15 degrees, SECAT suggests that there is generally more solar collection area than Spiral 3 predicts. At high angles of incidence, geometry becomes very important to accuracy since small absolute errors can mean large relative errors on account of the rapid rate of change of the cosine law and Fresnel corrections near 90 degrees. Modeling the solar array as a series of flat plates is likely introducing most of the error. Furthermore, it is known that the Spiral 3 tool's methodology has difficultly quantifying shadow effects when the target plane is nearly parallel to the vectors cast through the corners of shadow planes, as is the case at low elevations.

Given the data produced by SECAT in Fig. 20 and the time history (or an approximation of it) of the aircraft's deck orientation relative to the sun, the equivalent effective solar array collection area could then be multiplied by the solar flux constant G and the solar cell energy conversion efficiency factor to give the power generated by the solar array. Given a known value for propulsive power, the design of the rest of the aircraft and its mission may be further iterated upon.



(a) Spiral 3 model comprised of 7 quadrilaterals.

(b) SECAT model comprised of 3857 triangles

Figure 19. The Icaré 2 solar energy collection analysis tool input models. Blue components have solar cells and red components do not.

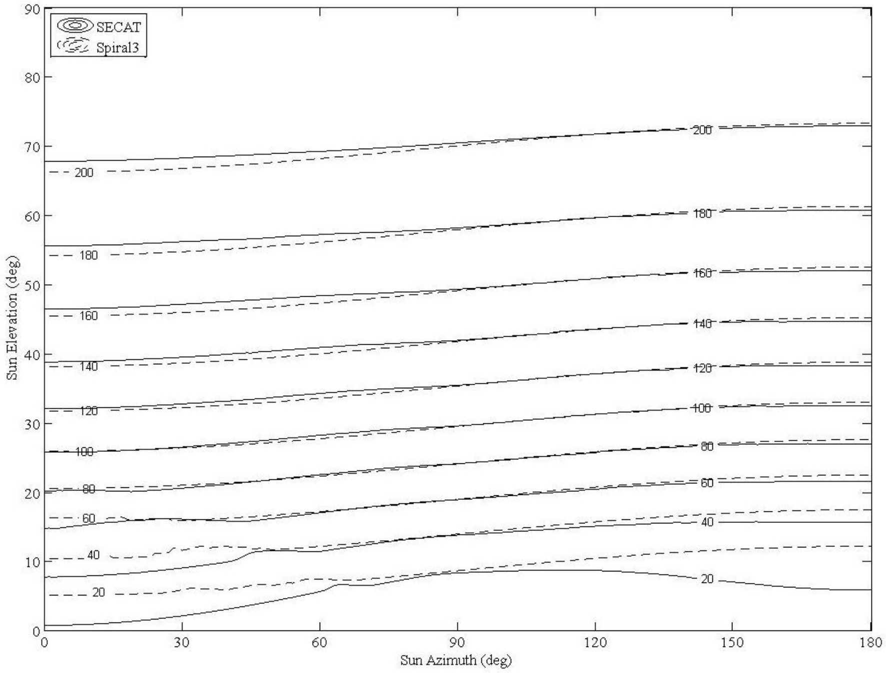


Figure 20. Contour plot of solar array equivalent collection area (square feet) for Icaré 2 as analyzed by SECAT and Spiral 3 tools.

VII. Conclusion

Using optical physics and computer graphics techniques, the Solar Energy Collection Analysis Tool evaluates angle of incidence and soft shadow losses for a given vehicle geometry to give an equivalent effective solar array area. SECAT allows solar aircraft to be sized with greater accuracy and confidence with respect to specific vehicle missions; the feasibility of ISR and science missions using HALE solar vehicles has been shown to be sensitive to the aircraft's deck orientation with respect to the sun. With SECAT, solar vehicle configurations may be analyzed and traded with greater autonomy, accuracy, and speed than previously possible.

It is well within the methodology behind SECAT to improve its utility even further. Solar panel efficiency is governed by the temperature of the cells, so it is possible to integrate a model for predicting the temperature of the cells given relevant conditions and further correct the equivalent solar collection area for temperature effects.

Furthermore, given a set of most probable sun azimuths and elevations throughout a 24 hour cycle, SECAT could also be programmed to make recommendations on where to place solar cells so that they "yield" the most energy collection throughout the day.

Even in its current implementation, SECAT is not perfect. CGAL is used to perform the bulk of the computation, and it does not currently include a function for finding the first intersected triangle as is required by the viewport method. Instead, all intersections must be calculated and the distance to each intersected triangle is calculated and compared to find the "first" intersected triangle. Clearly this method is not efficient and finding a way to incorporate the intersection test into the AABB tree searching algorithm more directly is desired. While CGAL is currently the best available choice, it does have some deficiencies. Also, modeling the sun deterministically caused the viewport method to contain more transient error for a given number of rays traced as compared to the shadow ray method. The error was determined to be caused by the patterns, which could be avoided by employing a randomized model of the sun as is done in more typical ray tracers.

Furthermore, a ray-tracing-inspired shadow calculation algorithm is not necessarily the best approach to shadow calculations; it was chosen simply because it was easy to implement and provided promise of increased analysis capability and speeds over the Spiral 3 tool. A trade study on shadow algorithms is warranted before SECAT is developed any further.

Although SECAT does not incorporate all physical phenomena that govern the collection capability of a solar array, its accuracy, speed, and ability to accept general geometries with ease gives it the capability to aid in the design of a vast array of systems that rely on solar energy collection.

Appendix A

The derivation of the flat plate equivalent area for the sun-sphere-plate system begins with Eq. (1):

$$A_{eq} = \iint_{solar\ array} E_c\, dA \tag{1}$$

From Fig. 16(b), it is known that the umbra and penumbra regions are circular and can best be represented in polar coordinates. The plate, which is square, is best represented in Cartesian coordinates. In either case, the origin is at the center of the square plate. Furthermore, umbra regions have no exposure and contribute nothing to the flat plate equivalent area. Given these observations, Eq. (1) can be represented as

$$A_{eq} = \iint\limits_{no\ shadows} E_c\, dA + \iint\limits_{penumbra} E_c\, dA \tag{11}$$

$$A_{eq} = 4\int_0^{r_p} \int_{\sqrt{r_p{}^2 - x^2}}^{c} E_{c,plate} dydx \; + \; 4\int_{r_p}^{c}\int_0^{c} E_{c,plate} dydx \; + \; 2\pi \int_{r_u}^{r_p} E_{c,penumbra}\, rdr \tag{12}$$

where r_p is the penumbra radius, r_u is the umbra radius, and c is the plate edge half-length. For integration simplicity, the plate edge half-length should be chosen so that it is greater than or equal to the penumbra radius. In the case where $c = r_p$, the second term in Eq. (12) is zero and may be removed. The penumbra and umbra radii are functions of the geometry of the sun-sphere-plate system; considering a view of the system with the plate edge-on, the umbra radius may be found by extending a line tangent to the top of the sun and tangent to the top of the sphere to the plane of the plate. If this line crosses the centerline defined by the origin and the center of the sun, then there is no umbra and there is an anti-umbral region. Such cases are not handled by the mathematics discussed here. Otherwise, the umbra has a finite radius.

The penumbra radius, considering the same geometric perspective discussed above, is found by extending a line tangent to both the top of the sun and the bottom of the sphere to the plate. The distance from the origin to the point of intersection on the plate is then the penumbra radius.

The umbra and penumbra radii are calculated as follows

$$r_u = \left(\frac{-R_{pb}}{R_{ps} - R_{pb}}\right)(r_s - r_b) + r_b, \tag{13}$$

$$r_p = \frac{R_{pb}(r_s/R_{is})}{\sqrt{1-\left(\frac{r_s}{R_{is}}\right)^2}}. \tag{14}$$

Where R_{pb} is the distance from the plate to the center of the sphere (**b**all), R_{ps} is the distance from the plate to the center of the sun, r_s is the radius of the sun, r_b is the radius of the sphere, and R_{is} is an intermediary value defined as

$$R_{is} = R_{ps} - R_{pb} - \frac{r_s\left(R_{ps} - R_{pb}\right)}{r_s + r_b}\,. \tag{15}$$

Now that the limits of integration are fully defined, the exposure for the plate and the penumbra must be derived. Points on the plate outside of the penumbra are not shadowed and therefore only angle of incidence and packing efficiency effects need to be considered. The exposure of the plate is defined as

$$E_{c,plate} = P * \left(1 - f(\theta)\right) * \cos(\theta), \qquad (16)$$

$$\theta = \tan^{-1}\left(\frac{r}{R_{ps}}\right) = \tan^{-1}\left(\frac{\sqrt{x^2 + y^2}}{R_{ps}}\right).$$

(17)

The exposure inside of the penumbra is similar to Eq. (14) except that shadow effects must be taken into account.

$$E_{c,penumbra} = P * I * (1 - f(\theta)) * \cos(\theta) \qquad (6)$$

The shadow term, I, is defined as the area of the eclipsed sun divided by the total area of the sun as viewed from a given point on the plate. Equation (18) describes ratio of the area of overlap of two circles to the area of one of the circles; it is being used to describe how the sphere overlaps the sun based on the perspective of a viewer at a given radial station on the plate. In order to do this, the sphere's image as seen from the viewer must be projected onto the plane containing the center of the sphere and that is parallel to the plate. This "apparent" radius of the sphere is described by Eq. (20). This operation accounts for how a golf ball, when held up to the sun, can appear to fully eclipse the sun as seen by the viewer on account of the geometry of the system. By projecting the sphere at the sun, the golf ball and the sun still appear the same size as seen by the viewer's eye, but now an equivalent radius is known for use in the eclipse calculation.

$$I(r) = \frac{1}{\pi {r_s}^2}\left[{r_{b,app}}^2 \cos^{-1}\left[\frac{d(r)^2 + {r_{b,app}}^2 - {r_s}^2}{2 * d(r) * r_{b,app}}\right] + {r_s}^2 \cos^{-1}\left[\frac{d(r)^2 + {r_s}^2 - {r_{b,app}}^2}{2 * d(r) * r_s}\right] - \frac{1}{2}\sqrt{\left(-d(r) + r_{b,app} + r_s\right)\left(d(r) + r_{b,app} - r_s\right)\left(d(r) - r_{b,app} + r_s\right)\left(d(r) + r_{b,app} + r_s\right)}\right],$$

(18)

$$d(r) = \frac{\left(R_{ps} - R_{pb}\right)\left(r/R_{pb}\right)}{\sqrt{\left(\frac{r}{R_{pb}}\right)^2 + 1}},$$

(19)

$$r_{b,app} = R_{ps}\left(\frac{r_b}{R_{pb}}\right).$$

(20)

In the equations above, $r_{b,app}$ is the apparent (equivalent) radius of the sphere, and d is the distance between the center of the circles representing the projected images of the sphere and the sun. Note that I is only a function of r and does not depend on the angular station since, for a given radius, the area of overlap is the same for all angles. This is the reason the double integral representing the penumbral region in Eq. (11) can be reduced to the single integral in Eq. (12).

Given Eqs. (11) – (20) and some inputs, the flat plate equivalent area of the plate may be calculated. For the setup corresponding to Fig. 16(a), the values in Table 1 were used.

Using these values, the flat plate equivalent area comes to 3.30 square units.

Variable	Value
P	1
R_{ps}	8
R_{pb}	4
r_s	0.3
r_b	0.5

Table 1. Inputs to the calculation of the flat plate equivalent area analytical solution for the sun-sphere-plate system.

Appendix B

The shadow ray method SECAT input file used to generate the data in Fig. 17 is shown below in Table 2 with a description of the inputs. The input file for the viewport method differs only in that it has an extra input for the user to define the number of viewport ports per unit length of viewport.

Description	File Contents
TRI filename (with file extension)	Icare2.tri
LowerElevationLimit UpperElevationLimit LowerAzimuthLimit Upper Azimuth Limit (degrees, space separated)	0 90 0 180
Number of components	5
Number of components with solar cells	2
Solar cell component ID numbers (space separated)	1 2
Packing efficiencies corresponding to all component numbers (space separated)	0.7 0.85 0 0 0
Number of sun layers (0 (point source) to inf)	3
Solar cell coverglass index of refraction	1.33
Analysis flag (all or top)	top

Table 2: Example Icaré 2 SECAT input file.

Acknowledgments

The autho would like to thank Tom Ozoroski for providing his work on the Spiral 3 solar energy analysis tool, and Dr. McDonald of Cal Poly for his ongoing support.

References

[1] Alliez, P., Tayeb, S., and Wormser, C., "3D Fast Intersection and Distance Computation," *CGAL User and Reference Manual* [Web], 2012.

[2] Blom, J. D., "Unmanned Aerial Systems: A Historical Perspective," U.S. Army Combined Arms Center, Occasional Paper 37, Fort Leavenworth, KS, 2010.

[3] Burger, D. R., and Mueller, R. L., "Angle of Incidence Correction For GaAs/Ge Solar Cells with Low Absorptance Coverglass," 25th PVSC, IEEE, Washington, D.C., 1996, pp. 243-246.

[4] Dickinson, R. M., and Grey, J., "Lasers for Wireless Power Transmission," TR 2014/16855, NASA and AIAA, 1999.

[5] Fernando, R., et al., *GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics*, Addison Wesley Professional, Boston, MA, 2004.

[6] Glassner, A., et al., *An Introduction to Ray Tracing*, Academic Press Limited, San Diego, CA, 1989.

[7] Hunley, J. D., and Kellogg, Y., "ERAST: Scientific Applications and Technology Commercialization," *NASA ERAST Proceedings*, CP-2000-209031, NASA, 1999, pp. 7-8.

[8] Nickol, C. L., Guynn, M. D., Kohout, L. L., and Ozoroski, T. A., "High Altitude Long Endurance Air Vehicle Analysis of Alternatives and Technology Requirements Development," AIAA 2007-1050, 2007.

[9] Shirley, P., et al., *Fundamentals of Computer Graphics*, A K Peters, Wellesley, MA, 2005, Chap. 10.

[10] Smith, W. J., "Modern Optical Engineering," McGraw-Hill, New York, 2000, Chap. 7.

[11] Solar Cell Array Design Handbook, Vol. 1, CR-149364, NASA, 1976, pp. 4.2-1 & -2.

[12] Spiral 3 Energy Analysis Tool, Software Package, Ver. 1.0, Swales Aerospace, Hampton, VA, 2012.

[13] Suffern, K., *Ray Tracing from the Ground Up*, A K Peters, Wellesley, MA, 2007, Chaps. 13, 18.

[14] Winckelmans, G. S., Topics in Vortex Methods for Computation of Three- and Two-Dimensional Incompressible Unsteady Flows, Doctor of Philosophy, California Institute of Technology, Pasadena, California, February 1989.

[15] Woo, A., Poulin, P., Fournier, A., "A Survey of Shadow Algorithms," *IEEE Computer Graphics & Applications*, Vol. 10, No. 6, Nov. 1990, pp. 13-32.