A Comparison Among Fast Visibility Algorithms Applied to Computational Electromagnetics

[1] J. G. Meana, [1] F. Las-Heras, and [2] J. Á. Martínez-Lorenzo

[1] Área de Teoría de la Señal y Comunicaciones, Campus de Viesques, Gijón,
University of Oviedo, Spain
[2] The Gordon CenSSIS, Northeastern University, Boston, MA, USA
jmeana@tsc.uniovi.es

Abstract – This paper presents a set of fast algorithms developed for solving the visibility problem in large complex geometric scenarios. The implemented algorithms are: *Binary Space Partitioning (BSP)* –based on a binary tree structure- and three new ones: *Trimming Method* – facets which are partially occluded are trimmed –, *Cone Method* – a cone emerges from the point of view to discard shadowed facets – and *Pyramid Method* – a pyramid is used to eliminate non-lit patches – . All the proposed algorithms are tested on a pair of scenarios for determining the field of view of electromagnetic waves in order to calculate the corresponding induced currents on the surfaces. The scenarios consist of two electrically large spheres and two PEC plates respectively where the surfaces are meshed with a variable number of flat triangular patches. The first reflection contribution is calculated using Physical Optics (PO) for both cases. The results show that the computational time can be drastically reduced by assuming small percentage of error in the computed scattered fields.

I. INTRODUCTION

Fast interrogation techniques for distinguishing between shadowed and lit regions, relative to a source point, are important in many computational-engineering areas including: image processing, virtual reality, video games and electromagnetic problems. Radar applications and radio-electric coverage are examples of electromagnetic problems where this distinction is necessary. In this way, the existence of line of sight between observation and source points can be one of the most consuming activities to afford at complex scenarios.

Traditional software methods, such as Painter's or Z-buffer algorithms [1-3], were applied successfully for solving the visibility problem. These methods have been combined with some pre-processing techniques to accelerate the basic algorithms. One of those techniques consists of grouping many planar facets into parallelepipedic or conical macro-domains. This information is then used to perform a first interrogation at the macro-domain level and discard some of them based on the source and observation points. Finally, a refined interrogation at the facet level is performed and, as a result, a high amount of computational time is saved. *Space Volumetric Partitioning* (SVP) or *Angular Z-buffer* (AZB) algorithms are good examples of this pre-processing technique [4-7].

Currently, a wide variety of the visibility problems can be solved in a fast way by employing GPU's [8] instead of CPU's. Good examples of this concept can be found in the literature [9, 10]. In [9] a specialized graphic card was introduced into a personal computer in order to compute real-time RCS. Its main disadvantage is that it implies the building of machine dependent software. On the other hand, non real-time solutions, but suitable for many applications, can also be performed by means of low cost generic graphic cards. The interface for communicating with the generic graphic card is done by using a set of libraries like DirectX or OpenGL [11]. One of the flaws in the latter method is that these libraries do not allow extracting the required information directly without first depicting the geometry. This depicting step is of no interest in applications where the main goal of the problem is not a graphical representation of the data as it implies an increase in the computational cost. This is the case of some electromagnetic problems -like radio-electric coverage or RCS applications- where the required information consists in getting mathematical structures indicating the facets in the scenario a point source is able to see for a set of sweeping angular directions. At this point, a return to the traditional geometric algorithms can be thought as a suitable solution for the problem. As a response to this requirement, new efficient and faster than traditional algorithms are investigated and presented in this paper.

The main features of the proposed techniques in the resolution of the visibility problems are: hierarchical classification and storage in *BSP*, accurate determination of visible geometry in *Trimming*, fast discrimination in *Cone* and *Pyramid Methods*.

The paper structure is as follows: the next section presents and describes step by step how the four visibility algorithms *-Binary Space Partitioning*, *Trimming Method*, *Cone Method* and *Pyramid Method-* work; the third section presents the numerical examples where all

1054-4887 © 2009 ACES

the algorithms are tested and compared. The paper is closed with conclusion that summarizes the performance of each algorithm based on accuracy and computational cost.

II. ALGORITHMS DESCRIPTION

This section describes, step by step, all visibility algorithms. The mathematical notation and symbols are summarized in Table 1. The visibility algorithms satisfy the following conditions:

- The geometry is supposed to be modelled with flat triangular patches. As a consequence, the algorithms are optimized to these particular facets.
- The normal vector criterion is always taken in consideration. If the angle between the position vector and the outward normal vector of the patch is in the interval [-90º, 90º], the facet must be eliminated.

A feature to underline is that the source point $\mathbf{s}$ is located at the origin of the Cartesian coordinate system in order to simplify the equations. Any other emplacement can be considered by means of a translation transformation.

Table 1. Nomenclature for the visibility algorithms.

NOMENCLATURE		
Symbol	Description	Components
$\mathbf{s}$	Source point	$(0,0,0)$
$\hat{t}$	Unitary target direction	(t_1,t_2,t_3)
$\hat{n}^i$	Unitary normal outward vector to facet i	(n_1^i,n_2^i,n_3^i)
$\mathbf{P}_c^i$	Barycentre of facet i	$(P_{c1}^i,P_{c2}^i,P_{c3}^i)$
$\mathbf{V}_l^i$	Vertices of facet i ($l = 1,2,3$)	$(V_{l1}^i,V_{l2}^i,V_{l3}^i)$
$\mathbf{P}$	Generic point	(x,y,z)
F_i	Facet i	---
$'$	Prime coordinates denote projection onto a plane.	---

A. *BSP*

Binary Space Partitioning algorithm (*BSP*) [12, 14] envelops a hierarchical structure (binary tree) in which all the patches can be arranged easily. First, a facet is randomly taken as reference (this is the root of the binary tree). The plane in which this triangle is contained splits the space into two subspaces (front and back). Therefore, some other patches could be intersected by this plane. In such a situation, every intersected patch would generate two new flat polygons (usually a triangle and a quadrilateral –discretized in two triangles–) One belongs to the front-half-space and the other to the back-half-space. All the patches (new and original) located in the front-half-space will take place in the left subtree while the back-half-space ones will do it in the right. For every subspace a new reference is chosen. This is inserted in the BSP tree as a node and the whole process is repeated recursively. When one facet is the unique element in a half-space, the patch acts as a leaf in the tree and the recursion finishes.

The relationship among different nodes in the binary tree remains invariable even though the location of the source would change. Consequently the tree needs to be built only once and its computational cost t_{Build} is [15],

$$(N+1)\log_2(N+1) \le t_{Build} \le \frac{N^2}{2} \tag{1}$$

where N is the number of facets.

BSP algorithm is able to extract a list of patches which can be seen from a point of view when the direction of the target is specified. In order to extract those elements, the binary tree is walked and the decision of including the triangle in the list or not depends basically on the angle between the target direction and the outward normal vector of the patch (a parallel projection onto a plane perpendicular to the target vector is accomplished). This operation implies a linear cost t_{Walk} [15],

$$t_{Walk} = \Theta(N). \tag{2}$$

The list resulting from walking the tree satisfies that the nth element could be occluded by elements 1 to n-1 (it is not necessary to test all the pairs of patches in the geometry). Therefore an additional algorithm to know whether a facet obscures other patches is indispensable.

One of the disadvantages to underline is that the parallel projection restricts the scope to scenarios where the point of view is situated far from the observation points. Otherwise, the hierarchical structure can be useful in other applications apart from visibility.

Our implementation classifies the patches in front-half-space or back-half-space based on the coordinates of the barycentre. As a result, the process of clipping the intersected patches is avoided and kept back to the *Trimming Method*.

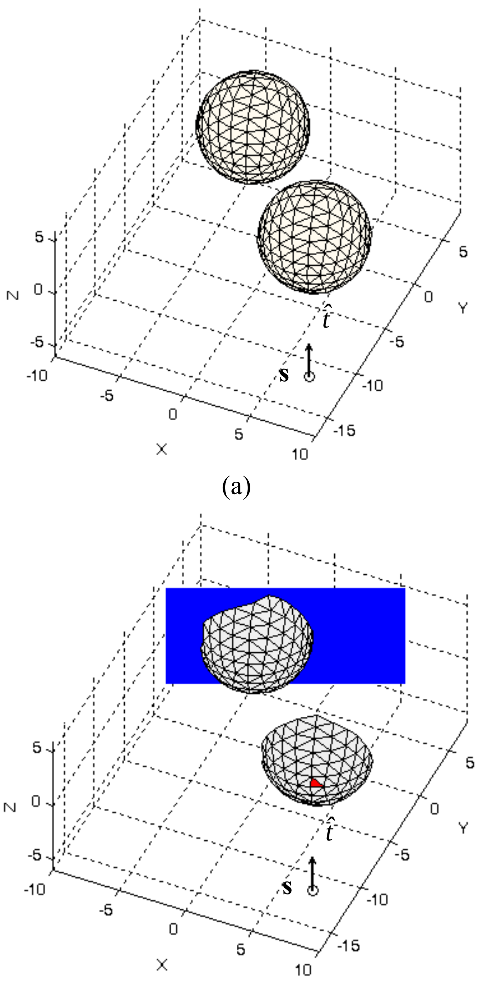
B. *Trimming Method*

The *Trimming Method* is the most accurate visibility algorithm due to the fact that the output geometry coincides exactly with the lit zones. Every partially lit facet is trimmed, so new completely lit and completely shadowed polygons appear. This is an exact solution to avoid fixing a threshold for deciding whether a partial occluded triangle is viewed or not.

First, the *N* facets (shown in Fig. 1 (a)) are sorted by their distance from the source point $\mathbf{s}$ with indexes $i = 1,2,\ldots,N$ (this is one of the basic actions to go into when a visibility problem must be solved). The nearest facet F_1 can always be seen (marked in Fig. 1 (b)). The projection plane is defined by the target direction $\hat{t}$ and the barycentre of the furthest facet $\mathbf{P}_c^N$ and its equation can be written as,

$$\hat{t}\cdot\left(\mathbf{P}-\mathbf{P}_c^N\right)=0 \tag{3}$$

(the projection plane has also been depicted in Fig. 1 (b)).



(a)

(b)

Fig. 1. (a) The original geometry, source point **s** and target direction $\hat{t}$. (b) Facets which verify the normal criterion. The nearest facet to the source point has been marked. The projection plane has been depicted too.

The three vertices $\mathbf{v}_d^i$ $(d=1,2,3)$ of each triangular patch $(i=1,2,\ldots,N)$ are projected onto the above mentioned plane by performing a perspective projection whose focus is the source point $\mathbf{s}$. The coordinates $\mathbf{P}'(x,y,z)$ of the projection of a generic point $\mathbf{P}(x,y,z)$ are,

$$\mathbf{P}'=\left(\hat{t}\cdot\mathbf{P}_c^N\right)\left[\left(\hat{t}\cdot\left(1,\frac{y}{x},\frac{z}{x}\right)\right)^{-1}\hat{x}+\left(\hat{t}\cdot\left(\frac{x}{y},1,\frac{z}{y}\right)\right)^{-1}\hat{y}+\left(\hat{t}\cdot\left(\frac{x}{z},\frac{y}{z},1\right)\right)^{-1}\hat{z}\right]. \tag{4}$$

Now let's take $\mathbf{P}_c^N=\mathbf{P}_c^{N}{}'$ the origin of a new rectangular coordinate system whose orthogonal vector basis is $\left[\hat{t},\hat{u},\hat{v}\right]$ where,

$$\hat{u}=\frac{\mathbf{V}_2^{N'}-\mathbf{V}_1^{N'}}{\left|\mathbf{V}_2^{N'}-\mathbf{V}_1^{N'}\right|} \tag{5}$$

and

$$\hat{v}=\frac{\hat{t}\times\hat{u}}{\left|\hat{t}\times\hat{u}\right|}. \tag{6}$$

It is clear that every point $\mathbf{P}'(t,u,v)$ in the projection plane can be expressed as,

$$\mathbf{P}'(t,u,v)=0\,\hat{t}+u\,\hat{u}+v\,\hat{v}=\mathbf{P}'(x,y,z)-\mathbf{P}_c^N \tag{7}$$

with

$$u=\left(\mathbf{P}'(x,y,z)-\mathbf{P}_c^N\right)\cdot\hat{u}\ , \tag{8}$$

$$v=\left(\mathbf{P}'(x,y,z)-\mathbf{P}_c^N\right)\cdot\hat{v}\ . \tag{9}$$

This transformation will be helpful to carry out union and intersection operations in two dimensions. Figure 2 shows the projection plane and all the projected triangular patches of the geometry in u-v coordinates.

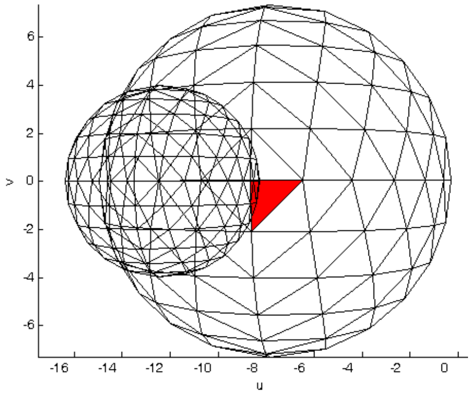


Fig. 2. Projection of the triangular patches onto the projection plane (u-v coordinates). The nearest facet is marked.

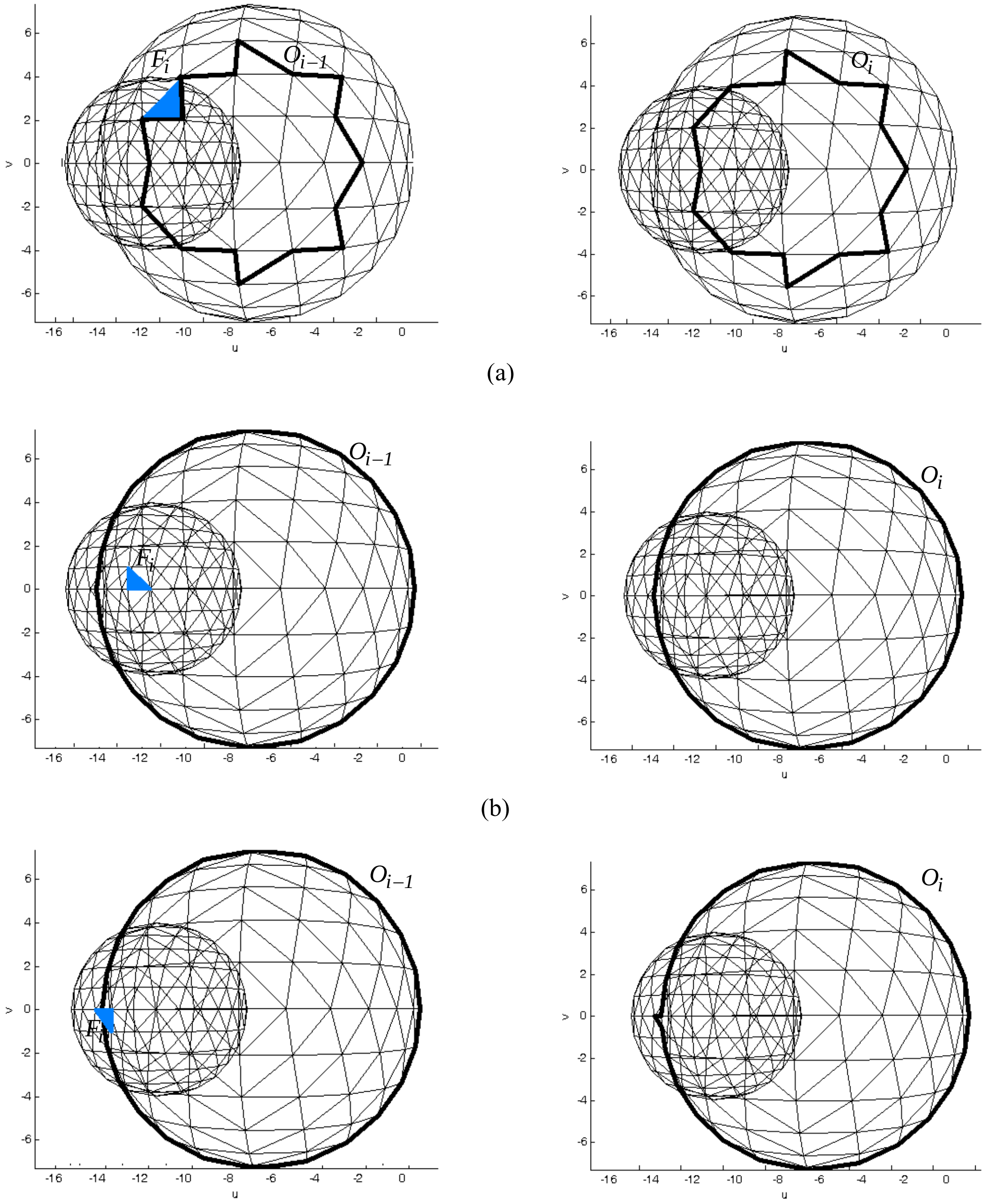
Afterward, the *test polygon O* is created with the aim of keeping updated the piece of surface associated with lit facets in the iteration *i* (*i=1,2,...,N*). That is, at the beginning,

$$O_1 = F_1 \,. \tag{10}$$

The next facet to consider is F_2. Therefore an intersection operation between the *test polygon* and F_2 is performed. If the result is the empty set, the triangular patch F_2 will join the *test polygon* as an independent domain. If both the *test* and F_2 share a piece of surface the updated *test polygon* O_2 will result from the union operation between that polygon and the facet F_2. The algorithm iterates over all the facets.

$$O_i = O_{i-1} \cup \left(F_i - \left(O_{i-1} \cap F_i\right)\right) \;\; i = 2,3,\ldots,N \,. \tag{11}$$

Figure 3 summarizes different relative positions of the facet F_i and the *test polygon*.



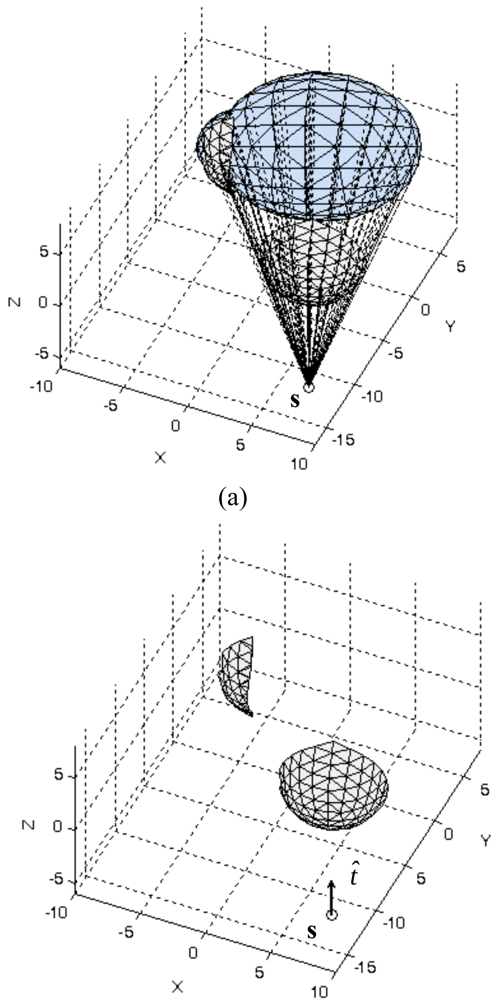
(a)

(b)

(c)

Fig. 3. Example of relative positions. Left, *test polygon* O_{i-1} and facet F_i. Right, *test polygon* O_i . (a) The facet is completely added to O_{i-1}. (b) The triangular patch is inside the test polygon. No modification of O_{i-1} occurs. (c) The facet is trimmed due to the intersection with O_{i-1}.

The following step consists of a reference system transformation and an inverse perspective projection to retrieve the 3D original coordinates (see Fig. 4). The trimmed geometry is made of the vertices in the *test polygon*. Nevertheless, a table which matches every projected point $\mathbf{P}'$ with its corresponding facet i has been previously filled (the projection of various points can coincide).



(a)

(b)

Fig. 4. (a) Inverse perspective projection of the trimmed facets. (b) Geometry after applying *Trimming method*.

An identical scheme can be maintained with two or more equally spaced projection planes (a division with dependence on the distance –in the plane XY– from the source point is now implemented). Let's denote B_1 the furthest projection plane from $\mathbf{s}$ and B_M, the nearest one (Figure 5 depicts three projection planes). A point $\mathbf{P}$ is located between two projection planes B_m and B_{m+1} when,

$$\left.\begin{array}{l}\hat{t}\cdot\mathbf{P}+D_m>0\\ \hat{t}\cdot\mathbf{P}+D_{m+1}<0\end{array}\right\}\qquad(12)$$

where m ranges from 1 to M-1 and D_m depends on the distance from B_m to the source point $\mathbf{s}$. Note that all the projection planes are parallel to B_l and, consequently, the normal vector is the same $\hat{t}$ for all of them.

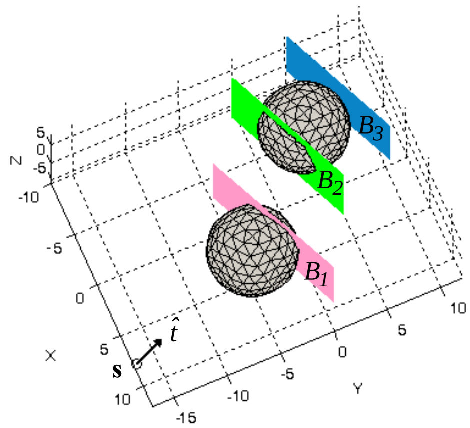


Fig. 5. Example of three projection planes ($M = 3$) in the *Trimming Method*.

The process starts with the facets in front of B_M. The resulting *test polygon* O_N is utilized as a filter to discard facets for B_{M-1}.This process is faster than making the comparison facet per facet. In spite of this, a previous trimming operation is carried out for the first *M-1* planes. As a consequence, the location of the source, $\mathbf{s}$, the target direction , $\hat{t}$, and the number of facets, N, influence the reduction (or increase) of the computational time.

The main advantage, accuracy, has already been discussed but the complexity of the algorithm because of union and intersection operations leads to a high computational cost in comparison with other methodologies (see Section III). Likewise, the trimming process implies the generation of new facets. Therefore, the mesh will contain more triangles and the deviation from their mean size would be, in general, greater than the original one.

C. Cone Method

This method employs an approximation from triangles to circles in order to accelerate the process of reckoning visibility.

A facet F_j is selected to compute shadow and lit regions. The source point $\mathbf{s}$ and each of the three vertices $\mathbf{V}_1^j$, $\mathbf{V}_2^j$, $\mathbf{V}_3^j$ define the segments $\overline{\mathbf{sV}_1^j}$, $\overline{\mathbf{sV}_2^j}$, $\overline{\mathbf{sV}_3^j}$. The angles between those segments and the target direction $\hat{t}$ are denoted by γ_1^j, γ_2^j, γ_3^j respectively. Then, the opening angle α^j can be easily calculated by weighting γ_1^j, γ_2^j and γ_3^j and stored for subsequent comparisons,

$$\alpha^j = \sum_{l=1}^{3} c_l \gamma_l^j \tag{13}$$

with c_l representing the weighting coefficients.

With the purpose of clarifying the procedure, Fig. 6 is presented: suppose that the three vertices $\mathbf{V}_l^j$ and the barycentre $\mathbf{P}_c^j$ are projected onto a plane perpendicular to the target direction $\hat{t}$, the same way *Trimming Method* does. Provided that $\mathbf{V}_l^{j'}$ and analogically $\mathbf{P}_c^{j'}$ are obtained from equation (4), the α^j value determines the radius R^j of the right cone whose axis is parallel to $\hat{t}$.

Any point in shadow will be situated behind the plane which contains the triangular patch F_j,

$$\hat{n}^j \cdot \left(\mathbf{P} - \mathbf{P}_c^j\right) < 0 \tag{14}$$

and inside the right cone previously defined. Therefore, in a local Cartesian coordinate system where the source point is in the origin, the angle between the position vector of that point and the target direction $\hat{t}$ must be smaller than the opening angle α^j,

$$\cos^{-1}\left(\frac{\mathbf{P} \cdot \hat{t}}{|\mathbf{P}|}\right) < \alpha^j . \tag{15}$$

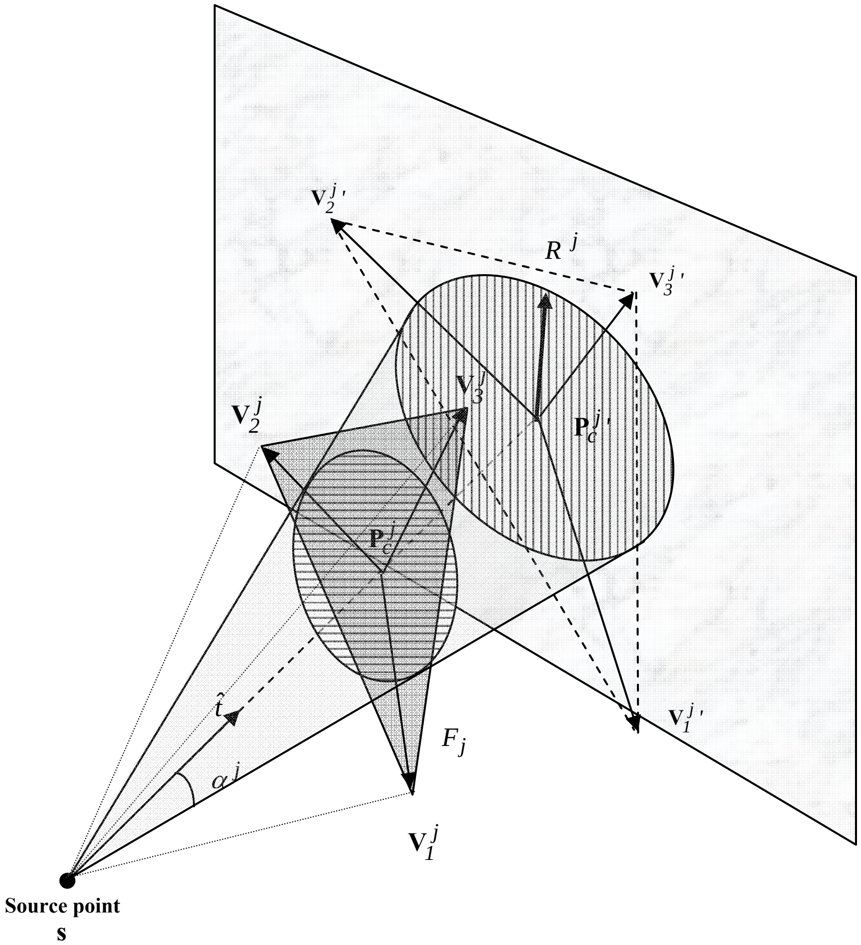


Fig. 6. *Cone Method*. The angle between the segments $\overline{\mathbf{sV}_1^j}$, $\overline{\mathbf{sV}_2^j}$, $\overline{\mathbf{sV}_3^j}$ and the target direction $\hat{t}$ is weighted so the opening angle α^j is obtained and consequently the radius of the cone R^j. Every facet whose barycentre $\mathbf{P}_c^i$ ($i \neq j$) is located inside the cone and behind the triangular patch F_j is assumed to be occluded.

The key of the accuracy in the *Cone Method*, apart from the number of points which characterizes a facet, is the choice of the factor which weights the angles γ_l^j : a great factor would not eliminate some contiguous triangular patches but a small one would keep some occluded facets. Simulations have proven that the best adjust to trimming results are achieved when the weighting factor corresponds with the arithmetic mean of the three angles.

Until this moment, the visibility problem in *Cone Method* has been reduced to points instead of facets. A facet is in shadow when there is, at least, one triangle which intersects the line of sight between the source point and its barycentre, although other characterizations can be performed for this and *Pyramid* methods –e.g. 3 vertices, inner points, etc– but an increase in the computational time must be expected. Finally, a sweep all over the facets in the sorted list is carried out.

D. Pyramid Method

The *Pyramid Method* is a special type of the *Cone Method* where the base is not circular. This algorithm determines if any point in the geometry is occluded by a specific facet.

Assume all the facets have been sorted by the distance from their barycentre to the source point $\mathbf{s}$. The closest triangular patch is always seen but could occlude other facets. The plane which contains the reference facet F_j is considered.

$$\hat{n}^j \cdot \left(\mathbf{P} - \mathbf{P}_c^j\right) = 0 \, . \tag{16}$$

Three additional planes are defined by taking into account every edge $\overline{\mathbf{V}_1^j\mathbf{V}_2^j}$, $\overline{\mathbf{V}_2^j\mathbf{V}_3^j}$, $\overline{\mathbf{V}_3^j\mathbf{V}_1^j}$ of F_j and $\mathbf{s}$. A generic point $\mathbf{P}$ in one of these planes must satisfy the corresponding plane equations in (17) to (19),

$$\frac{\mathbf{V}_2^j \times \mathbf{V}_1^j}{\left|\mathbf{V}_2^j \times \mathbf{V}_1^j\right|} \cdot \mathbf{P} = 0 \tag{17}$$

$$\frac{\mathbf{V}_3^j \times \mathbf{V}_2^j}{\left|\mathbf{V}_3^j \times \mathbf{V}_2^j\right|} \cdot \mathbf{P} = 0 \, , \tag{18}$$

$$\frac{\mathbf{V}_1^j \times \mathbf{V}_3^j}{\left|\mathbf{V}_1^j \times \mathbf{V}_3^j\right|} \cdot \mathbf{P} = 0 \, . \tag{19}$$

The intersection of the last three planes generates the faces of a pyramid whose base is F_j (see Fig. 7). In order to know if a point $\mathbf{P}$ is inside the pyramid, the left side in equation (20) is evaluated. When the result equals the number of vertices of the base, $\mathbf{P}$ is enclosed by the pyramid (the vertices of F_j are supposed to appear in clockwise or anti-clockwise order).

$$\sum_{i=1}^{3} sign\left|\frac{\mathbf{V}_{i+1}^j \times \mathbf{V}_i^j}{\left|\mathbf{V}_{i+1}^j \times \mathbf{V}_i^j\right|} \cdot \mathbf{P}\right| = 3 \tag{20}$$

where $\mathbf{V}_4^j = \mathbf{V}_1^j$. The other constraint is identical to equation (14) in *Cone Method*.

This method complements the *BSP* once the priority list has been obtained.

The mentioned idea about SVP and AZB in Section I is adapted to *Trimming*, *Cone* and *Pyramid* methods. Not only cubic or parallelepipedic domains are not created but also conical. Instead, the z-coordinate is neglected and equally-spaced angular sectors (corresponding with angle ϕ in a local cylindrical coordinate system) are delimited (Figure 8). Supposing L sectors, the equally-spaced limit angles are $\beta_1, \beta_2, \ldots, \beta_L$,

$$\beta_1 \le \phi \le \beta_{L+1} \, . \tag{21}$$

A facet F_j belongs to sector l if its barycentre $\mathbf{P}_c^l$ is inside its boundaries (except for *Trimming Method*, where the triangular patches are, of course, trimmed),

$$\begin{cases} -\mathbf{P}_{c1}^j \sin\beta_l + \mathbf{P}_{c2}^j \cos\beta_l > 0 \\ -\mathbf{P}_{c1}^j \sin\beta_{l+1} + \mathbf{P}_{c2}^j \cos\beta_{l+1} < 0 \end{cases} \tag{22}$$

with $l = 1, 2, \ldots, L-1$.

Acceleration is expected depending on the number of triangles the geometry is composed of.

III. SIMULATED RESULTS

Once the algorithms have been presented, a comparison in terms of accuracy and computational time is carried out among the CPU implementation of the *BSP Method* and the three new methods *Trimming*, *Cone* and *Pyramid*.

A. PEC Plates

This example consists of a first comparison among the already presented algorithms and a simple Z-buffer implementation where the facets are not trimmed.

The scenario is compound of two square parallel PEC plates. They are the same size, 10λ, and the distance between them is d_p=5 λ. The first plate is located in the XY plane. The position of the center of the second plate varies along the Y-axis from (0,-10λ,-d_p) to (0, -10λ, d_p) with step $\Delta y = \lambda$. Three different meshes have been considered to evaluate the computational time with respect to the number of facets: 200, 800 and 1800.

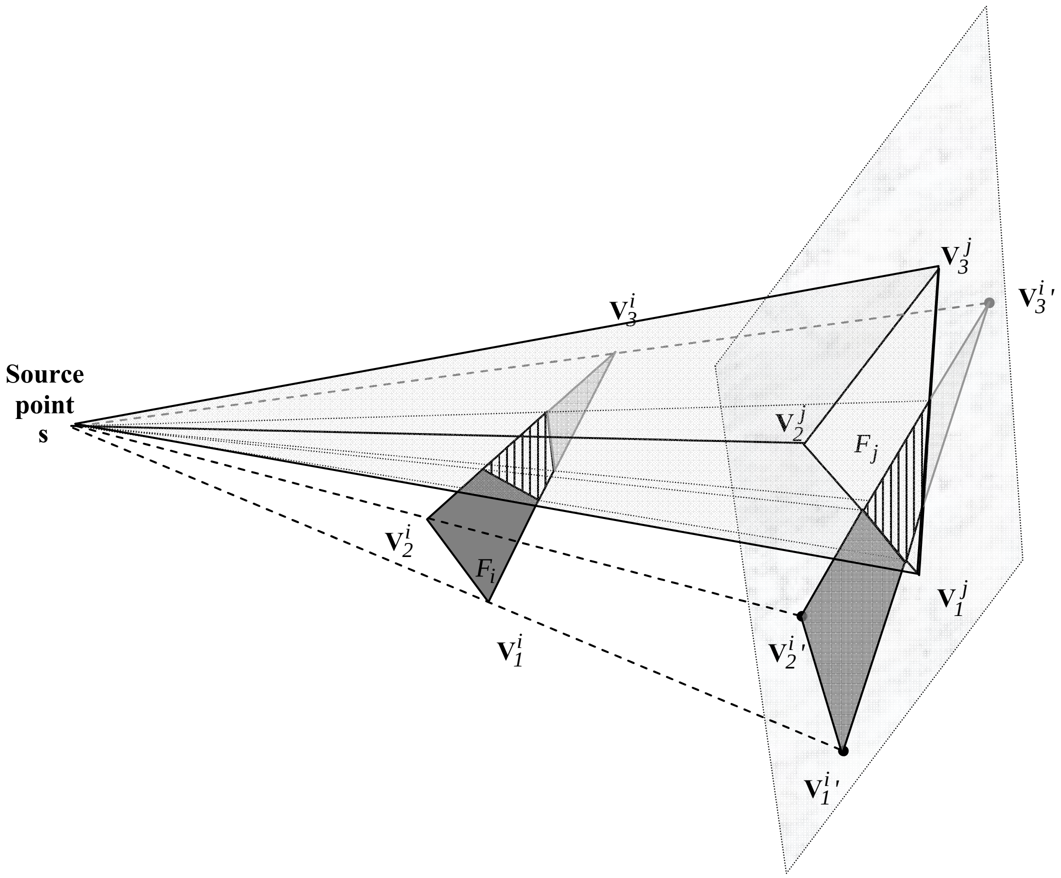


Fig. 7. *Pyramid Method*. The facet F_j acts as reference. Its edges and the source point $\mathbf{s}$ define three planes (pyramid). The facet F_i intersects some of them but is not in shadow because it is situated in front of F_j.

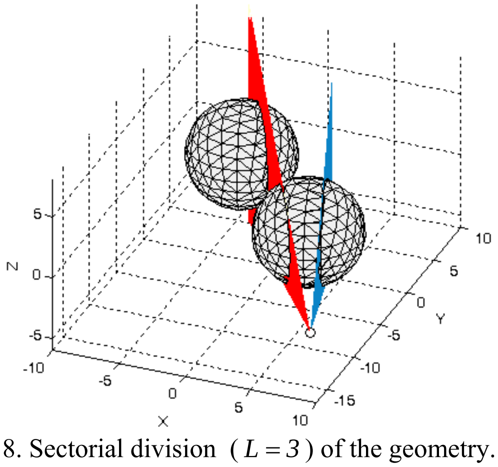


Fig. 8. Sectorial division $(L = 3)$ of the geometry.

An incident plane wave impinges the surface after determining which facets are illuminated. The Physical Optics approximation [16] establishes that the incident field induces an electric current density $\mathbf{J}$,

$$\mathbf{J} \approx 2\hat{n}\times \mathbf{H}^{inc} \tag{23}$$

with $\hat{n}$ the outward normal vector and $\mathbf{H}^{inc}$ the magnetic incident field in the patch.

The scattered electric field $\mathbf{E}^s$ due to a flat triangular patch is reckoned analytically in [17] and the first reflection contribution (denoted by σ) is calculated as,

$$\sigma = \lim_{r\to\infty} 4\pi \mathrm{r}^2 \frac{|\mathbf{E}_1^s|^2}{|\mathbf{E}^{inc}|^2} \tag{24}$$

where $\mathbf{E}^{inc}$ represents the incident electric field; $\mathbf{E}_1^s$, the total scattered electric field due to the contribution of the first reflection on the illuminated facets and r is the distance from the radar to the target.

The results in Fig. 9 show a perfect agreement where the facets do not need to be trimmed due to the selected step. Table 2 summarizes the computational time for each algorithm. Our simple Z-buffer implementation has associated higher computational time than *Pyramid* and *Trimming Method*s. Likewise, a generic scenario where the facets could be partially occluded would require a more complex Z-buffer implementation capable of trimming facets. Consequently its computational time is estimated to surpass even the *Trimming Method* values.

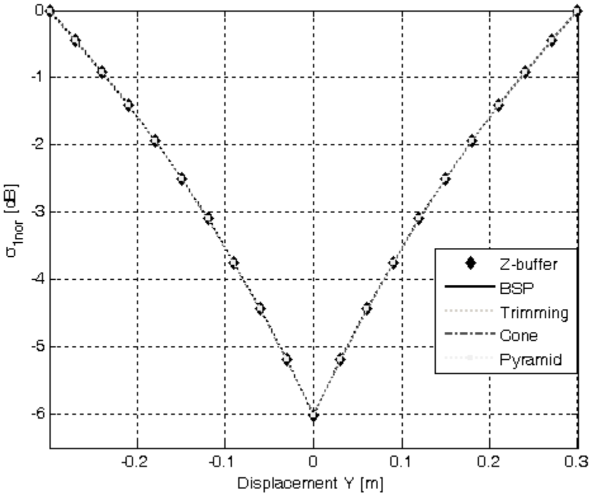


Fig. 9. Simulated results for two PEC plates. The first reflection σ contribution has been normalized.

Table 2. Plates example. Computational time comparison.

		Computational time (sec)		
		200 facets	800 facets	1800 facets
Methods	Trimming	0.6689	2.6486	6.0955
	Cone	0.0499	0.4123	1.4716
	Pyramid	0.0841	0.4432	1.6278
	BSP	0.1890	1.3012	5.2432
	Z-buffer	0.1711	0.6806	1.7561

B. *PEC Spheres*

A geometry consisting of two perfect electric conductor (PEC) spheres (radius r=10λ) whose centres are separated $d = 3r = 30\lambda$ is defined (Figure 10).

In order to analyze the influence of the discretization in the results, three meshes have been selected: 392, 1352 and 2888 triangles per sphere (the increase in the number of patches improves the correct modelling of the spherical surface). Therefore, the difference between two configurations lies in the fact of considering lit or hidden a facet which is occluded partially (when *Trimming* is not active).

A directional sweep is performed from $\phi = -45^o$ to $\phi = 135^o$ with step of $\Delta\phi = 1^o$. The simulated results are calculated analogically to the previous section. Tables 3 to 5 show the average computational time for each geometry discretization and aspect angle as a function of the number of sectors (except for *BSP Method* where the time due to the creation of the BSP tree has been uniformly distributed among the *Q=181* aspect angles). The least values (best time) have been marked and their corresponding first reflection contributions have been depicted in Figs. 11 to 13. The proposed methods present a good coincidence in the central area of the graphics but not in the limits where one sphere starts to partially occlude the other. In this context, the dimensions of the patches and the lit/shadowed classification based on their barycentre have relevance.

Table 3. Spheres example. Computational time and relative error comparisons. 784 facets.

784 facets		Computational time (sec) | Relative Error δ(%)		
		1 sector	3 sectors	6 sectors
Methods	Trimming	**1.54**	1.96	2.48
	Cone	0.045 | **4.58**	**0.035** | 4.92	0.036 | 5.24
	Pyramid	0.080 | **8.12**	**0.075** | 9.51	0.079 | 10.48
	BSP	**0.49 | 13.04**		

Table 4. Spheres example. Computational time and relative error comparisons. 2704 facets.

2704 facets		Computational time (sec) | Relative Error δ(%)		
		1 sector	3 sectors	6 sectors
Methods	Trimming	**5.02**	5.78	6.74
	Cone	0.29 | **6.42**	0.16 | 6.48	**0.13** | 6.54
	Pyramid	0.42 | **10.42**	0.30 | 11.11	**0.26** | 13.52
	BSP	**3.47 | 10.89**		

Table 5. Spheres example. Computational time and relative error comparisons. 5776 facets.

5776 facets		Computational time (sec) | Relative Error δ(%)		
		1 sector	3 sectors	6 sectors
Methods	Trimming	**11.52**	12.25	13.17
	Cone	1.15 | **4.67**	0.51 | 5.79	**0.32** | 5.94
	Pyramid	1.37 | **5.81**	0.78 | 6.29	**0.60** | 6.46
	BSP	**13.19 | 6.42**		

In order to compare the differences between the *Trimming Method* and the other techniques, the absolute error Δ has been calculated in equation (25),

$$\Delta = \left|\sigma_i - \sigma_i^{ref}\right| \tag{25}$$

where σ_i and σ_i^{ref} are the first reflection value and first reflection reference value (*Trimming*) in the ith

aspect angle respectively. Figures 14 to 16 represent these absolute errors for the best fit solution. The graphics confirms that the error increases near $\phi = -45^o$ and $\phi = 135^o$ as mentioned. Equation (26) introduces a relative error $\delta(\%)$.

$$\delta(\%) = \frac{1}{Q}\sum_{i=1}^{Q}\left|\frac{\sigma_i - \sigma_i^{ref}}{\sigma_i^{ref}}\right| \cdot 100 \,. \qquad (26)$$

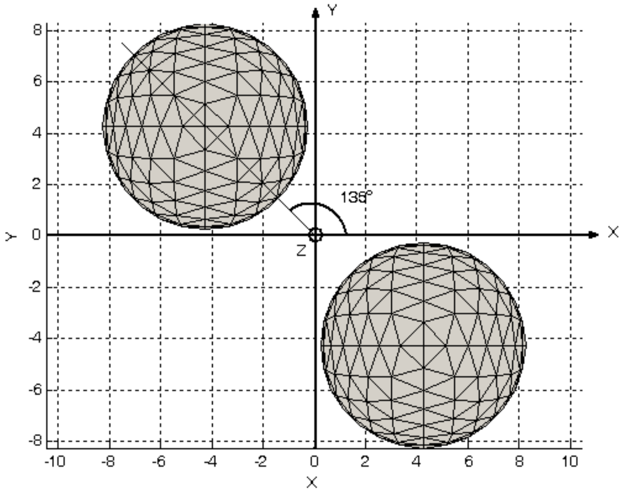


Fig. 10. Spheres example. Geometry description. Spherical coordinates for the centre of the spheres are: $C_1(\rho,\phi,z) = (1.5r, -\pi/4, 0)$ and $C_2(\rho,\phi,z) = (1.5r, 3\pi/4, 0)$. The radius r equals to 10λ and the separation between centres is $d = 3r = 30\lambda$.

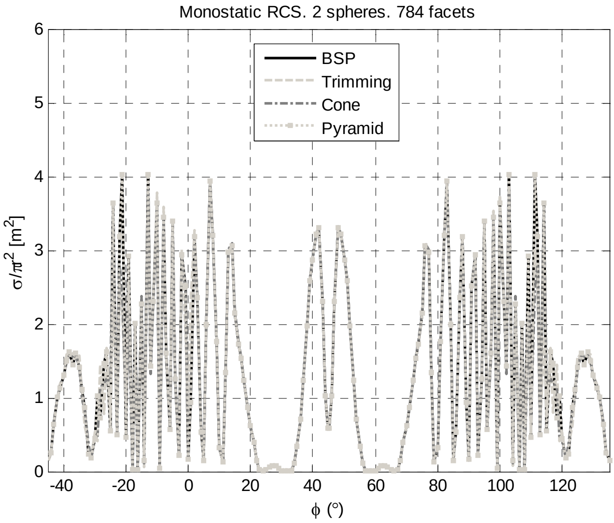


Fig. 11. Spheres example. First reflection contribution. Best time comparison, 784 facets.

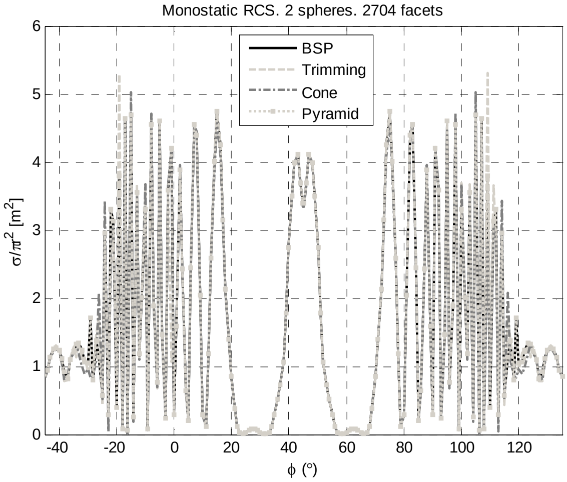


Fig. 12. Spheres example. First reflection contribution. Best time comparison, 2704 facets.

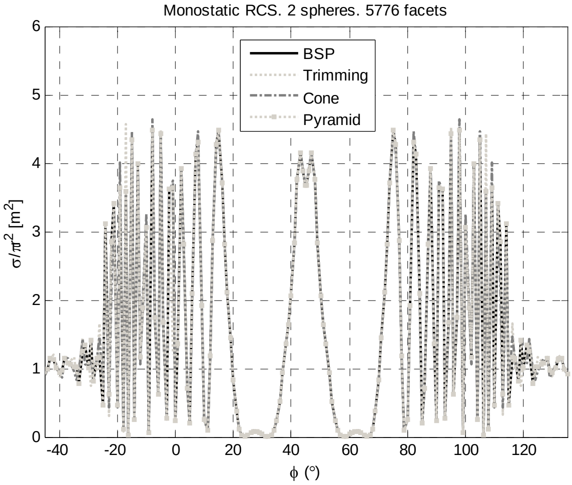


Fig. 13. Spheres example. First reflection contribution. Best time comparison, 5776 facets.

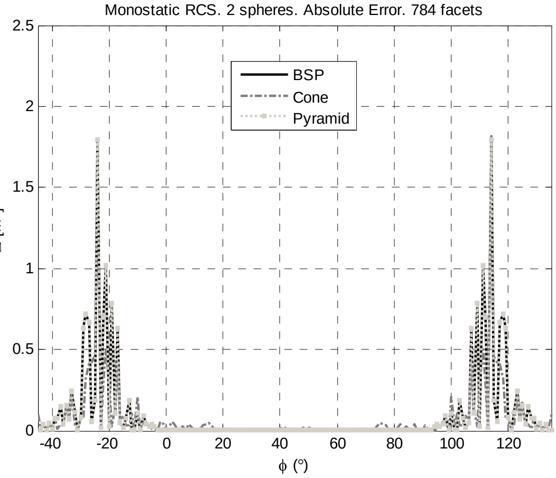


Fig. 14. Spheres example. Absolute error of the first reflection contribution with respect to *Trimming Method*, 784 facets.

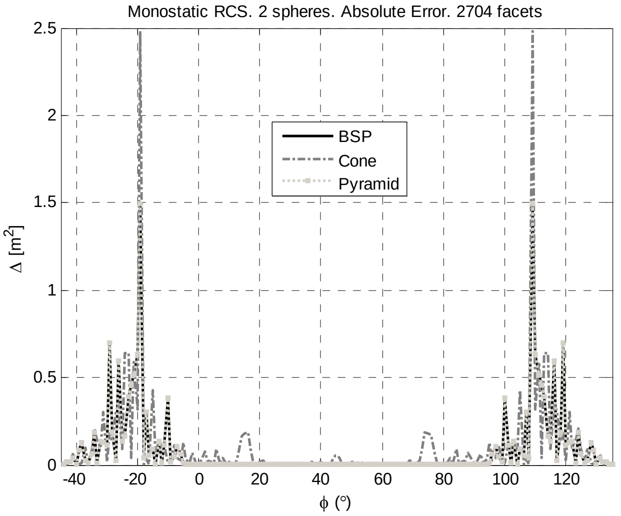


Fig. 15. Spheres example. Absolute error of the first reflection contribution with respect to *Trimming Method*, 2704 facets.

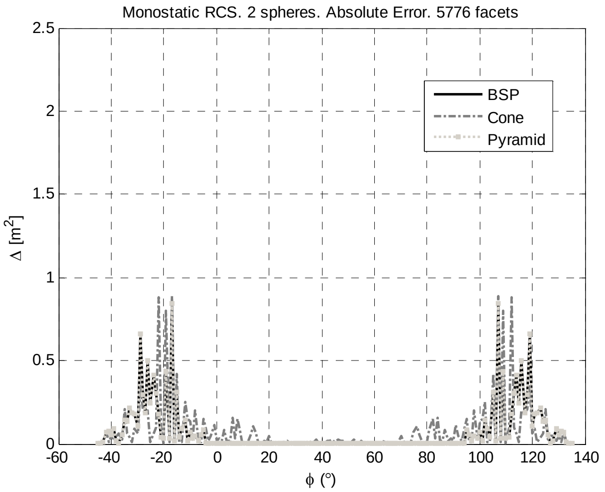
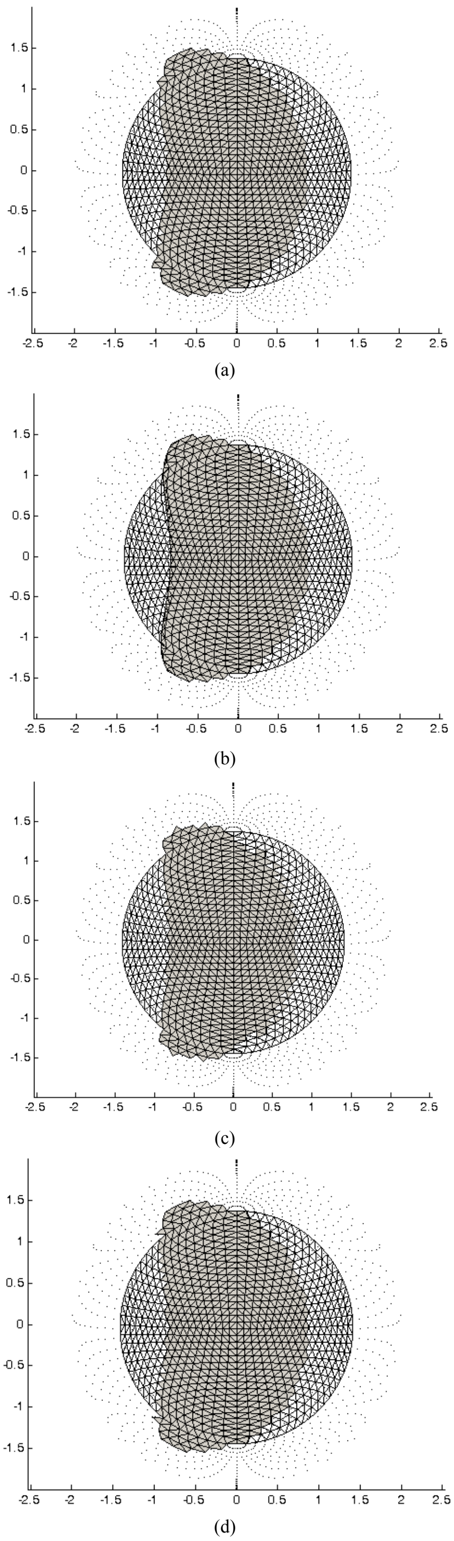


Fig. 16. Spheres example. Absolute error of the first reflection contribution with respect to *Trimming Method*, 5776 facets.

Tables 3 to 5 summarize these errors. The underlined cells are associated with the least errors for each of the techniques and discretizations. Best time and best fit results do not usually come from the same configuration. Therefore a midway solution must be adopted and an important computational time reduction is usually preferred, even though it would imply an error increase. *BSP* errors exceed *Trimming* and *Cone* results and all errors tend to increase with the number of sectors (because of the rough classification of the facets). Finally, an example of the lit triangular patches is presented in Fig. 17 in a Lambert azimuthal equal-area projection for the sphere in the second quadrant. Figure 17(b) corresponds to the *Trimming Method* and acts as reference for the graphics *(a)*, *(c)* and *(d)*.



(a)

(b)

(c)

(d)

Fig. 17. Lambert azimuthal equal-area projection of the sphere in the second quadrant. Light grey facets correspond to a semi-sphere projection (grid). Dark grey facets represent the lit triangles due to (a) *BSP*, (b) *Trimming Method,* (c) *Cone Method,* (d) *Pyramid Method.* (Aspect angle=-21°).

IV. CONCLUSION

This paper has presented four different methods to solve the visibility problem and a comparison of the first reflection contribution in terms of computational time and error has been performed based on CPU implementations.

Binary Space Partitioning uses a hierarchical structure to sort the patches in the space. The implementation of this well known algorithm has been divided into three phases: building and walking the binary tree and discarding the occluded triangles. It is suitable for scenarios where the point of view is situated outside the geometry because of the parallel projection which is carried out. Nonetheless, the computational cost is greater than the other techniques which provide similar or better accuracy (only a huge amount of angular directions could justify the creation of the BSP tree in terms of computational cost).

The new *Trimming Method* solves the visibility problem in an exact manner by means of a perspective projection and the definition of a *test polygon*. As a result, the original facets are partially occluded are trimmed. Therefore, only the lit surface is returned. The number of patches increases and the facet mean size tends to be smaller than the initial one. The re-definition of the triangular patches is not convenient for computing reflections. On the other hand, the high number of operations lead to a high computational cost.

The *Cone Method* replaces the original facets by circles and creates a fictitious right cone whose vertex is the source point. The piece of surface behind the plane which contains a facet and inside its cone is in shadow. This technique is the fastest one and Figs. 11 to 13 and Tables 3 to 5 have shown that the corresponding relative error is small.

The *Pyramid Method* tests whether a point satisfies equation (20), where each summand is related to one of the faces of the pyramid in Fig. 7. It employs the plane equation to quickly solve the problem; but it still takes twice the computational time taken by *Cone Method*. In addition, the technique can be extrapolated to n-edged polygons, where substituting polygons by circles would be inadequate.

A simple CPU implementation of the Z-buffer algorithm, where the geometry is not trimmed, has been also evaluated. The results have proven Z-buffer algorithm to be less time-efficient for our purpose than *Pyramid* and *Cone Methods.*

As the number of patches in the geometry ranges from 784 to 5776, the first reflection contribution varies because of the improvement on the modelling of the geometry. In this context, a smoother surface leads to more realistic results.

ACKNOWLEDGEMENT

This work has been supported by Ministerio de Ciencia e Innovación of Spain /FEDER under projects TEC2008-01638/TEC (INVEMTA) and CONSOLIDER-INGENIO CSD2008-00068 (TERASENSE); by Unión Europea-Fondo Europeo de Desarrollo Regional under project EQP06-015; by Gobierno del Principado de Asturias- Plan de Ciencia y Tecnología (PCTI)/FEDER-FSE under grants BP08-082, BP06-101, BP06-155 and projects FC-08-EQUIP-06 and PEST08-02 and by Cátedra Telefónica- Universidad de Oviedo and CenSSIS, the Gordon Center for Subsurface Sensing and Imaging Systems under the ERC Program of the NSF (Award number EEC-9986821).

The authors would like to thank Mss. Sarah Brown for her kindness in correcting the English in the text.

REFERENCES

[1] B. R. Dewey, *Computer graphics for engineers*. USA: Harpercollins College Div., 1998.

[2] J. D. Foley *et al.*, *Computer Graphics*. USA: Addison-Wesley, 1992.

[3] J. Bittner and P. Wonka , "Visibility in computer graphics," *Environment and Planning B: Planning and Design* vol. 30, no. 5, pp. 729 – 755, 2003.

[4] M. F. Cátedra and J. P. Arriaga, *Cell planning for wireless communications*. Boston: Artech House, 1999.

[5] M. F. Cátedra, J. Pérez, F. S. de Adana, and O. Gutiérrez, "Efficient ray-tracing techniques for three-dimensional analyses of propagation in mobile communications. Application to picocell and microcell scenarios," *IEEE Antennas and Propagation Magazine*, vol. 40, no. 2, pp. 15-25, 1998.

[6] A. Woo and J. Amanatides, "Voxel occlusion testing: a shadow determination accelerator for ray tracing," *Proceedings of Graphics interface '90*, pp. 213-220, June 1990.

[7] Y. Zhou and H. Ling, "On the multiplaten Z-buffer algorithm for ray tracing in high frequency electromagnetic scattering computations," *Microwave and Optical Technology Letters*, vol. 43, pp. 298-301, 2004.

[8] M. Reyer, T. Rick, and R. Mathar, "Graphics hardware accelerated field strength prediction for rural and urban environments," *Proceedings of the Second European Conference on Antennas and Propagation –EuCAP 2007*, November 2007.

[9] J. M. Rius, M. Ferrando, and L. Jofre, "GRECO graphical electromagnetic computing for RCS prediction in real time," *IEEE Antennas and Propagation Magazine*, vol. 35, no. 2, 1993.

[10] M. J. Inman and A. Z. Elsherbeni, “Programming video cards for computational electromagnetics applications,” *IEEE Antennas and Propagation Magazine*, vol. 47, no. 6, 2005.

[11] D. Shreiner, *OpenGL reference manual: the official reference document to OpenGL, Version 1.4.* London: Addison-Wesley, 2004.

[12] H. Fuch, Z. M. Kedem, and B. F. Naylor, “On visible surface generation by priori tree structures,” *ACM Siggraph Computer Graphics*, vol. 14, no. 3, pp. 124-133, 1980.

[13] D. Gordon and S. Chen, “Front-to-back display of BSP trees,” *IEEE Computer Graphics and Applications*, vol. 11, pp. 79-85, 1991.

[14] C. W. Fu, T. T. Wong, *et al.,* “Binary-space-partitioned images for resolving image-based visibility,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 1, 2004.

[15] F. Las-Heras, J. L. Jambrina, and E. Iguacel “Different approximations for the double reflection contribution to the RCS in a Physical Optics algorithm,” *Proceedings of the 7th European Electromagnetic Structure Conference*, pp. 11-14, September, 1993.

[16] C. A. Balanis, *Engineering electromagnetics*. New York, USA: John Wiley and Sons, 1989.

[17] A. M. Arias, J. O. Rubiños, I. Cuiñas, and A. G. Pino, “Electromagnetic scattering of reflector antennas by fast physical optics algorithms,” *Recent Res. Devel. Magnetics*, no. 1, pp. 43-63, 2000.

J. G. Meana was born in Gijón, Spain, in 1982. He received his M.S. degree in Telecommunications Engineering in 2005 from the University of Oviedo (Spain). He joined the R&D department of CTIC Foundation in 2005 and since 2006 he is a Research Assistant with the Area of Theory of Signal and Communications (University of Oviedo). He was awarded a Ph.D. Scholarship of the Principado de Asturias and is currently pursuing the Ph.D. degree in electrical engineering. He received the “Novel Engineer Award. XI La nit de les Telecomunicacions Awards” in 2005. His interests and research studies are focus on the evaluation of electromagnetic coverage in rural/urban scenarios by means of high frequency techniques.

Fernando Las-Heras was born in Zaragoza, Spain. He received the M.S. degree in 1987 and the Ph.D. degree in 1990, both in telecommunication engineering, from the Universidad Politécnica de Madrid (UPM), Madrid, Spain. From 1988 to 1990, he was a National Graduate Research Fellow. From 1991 to 2000, he held a position of Associate Professor at the Department of Signals, Systems and Radiocommunications of the UPM. From 2001 to 2003, he held a position of Associate Professor at the Department of Electrical Engineering of the University of Oviedo, pioneering the Area of Theory of Signal and Communications at that University. Since December 2003, he has held a Full Professor position at the University of Oviedo where he is presently Vice-Dean for Telecommunication Engineering degree at the Polytechnic School of Engineering at Gijón, Spain. His main research interests include the analysis and design of antennas, electromagnetic interference (EMI) and the inverse electromagnetic problem with application to diagnostic, measurement and synthesis of antennas.

José Ángel Martínez-Lorenzo was born in Madrid, Spain, in 1979. He received the M.S. and Ph.D. degrees both in telecommunications engineering from the University of Vigo, Vigo, Spain, in 2002 and 2005, respectively. From 2002 until 2004, he worked as a teaching and Research Assistant at the University of Vigo. In 2004, he joined the faculty of the Department of Signal Theory and Communications, University of Oviedo, Gijon, Spain, where he was an Assistant Professor until 2006. During spring and summer 2006, he was a Visiting Researcher at the Bernard Gordon Center for Subsurface Sensing and Imaging Systems (Gordon-CenSSIS) Engineering Research Center, Northeastern University, Boston, MA. Since October 2006, he has been with Gordon-CenSSIS as a Senior Research Scientist and a Part-time Lecturer in the Electrical and Computer Engineering Department at Northeastern University. He has authored over 45 technical journal and conference papers in the areas of microwave antenna design, electromagnetic wave propagation and computational electromagnetics.