Discretization Errors in the Graphical Computation of the Physical Optics Surface Integral

Juan M. Rius, Daniel Burgos, Angel Cardama*

ABSTRACT. This paper studies the sources of discretization errors in the graphical computation of PO surface integrals. Three different PO models that associate to screen pixels patches of different shape and orientation are presented. The RCS versus frequency results obtained for a sphere show that when the resolution in the surface discretization is high enough for the working frequency, the best results are obtained with the triangle mesh PO model (Gordon formula) [2], but the tangent plane approximation of J.S. Asvestas [13] achieves the best trade-off between CPU time and accuracy. When the resolution in the surface discretization is not high enough for the working frequency, the best results are obtained in most cases with the heuristic approximation of J.M. Rius [11] [12], due to the use of interpolated unit normals in pixels inside the flat triangles of the rendering model.

I. INTRODUCTION

One of the most important applications of the physical optics (PO) approximation is the prediction of the radar cross section (RCS) of electrically large surfaces [1]. The computation of the PO surface integral for realistic radar targets requires adequate modeling of arbitrary surfaces. For that reason, computer-aided design (CAD) software packages are used for geometric modeling of the target. Arbitrary surfaces are usually modelled for RCS computation as:

- A collection of flat polygonal facets:
  - The PO surface integral can be computed for each facet with a simple formula [2]. This leads to simple computer code and fast computation [3] [4]. Since only facets illuminated by the incident field contribute to the scattered field, some additional processing is necessary to remove the contribution from eclipsed or shadowed facets.
  - However, the approximation of a curved surface with a reasonable number of facets introduces artificial edges, which leads to the so-called "facet noise" in the RCS computation.
- A collection of curved patches, mathematically represented as non-uniform rational B-splines (NURBS) parametric surfaces [5] [6]:
  - In contrast to facets, parametric surface patches can accurately approximate curved surfaces (all conics can be exactly represented as NURBS) and thus the modeling noise is usually negligible.
  - However, since NURBS are piecewise ratios of polynomials, direct computation of the PO surface integral for NURBS is not as computationally efficient as for flat facets [10] and asymptotic approximations are commonly used [7] [8] [9]. This involves the identification of illuminated stationary phase points [1] and the computation of radii of curvature, which leads to complex computer code and slow computation.

Recently a graphical processing procedure has been presented for simple and fast computation of the PO surface integral over parametric surfaces [11] [12]. Using a graphic workstation we can obtain a 3-D image of the target. If the viewpoint of the observer is located at the position of the monostatic radar, then the picture on the workstation screen contains only the illuminated surfaces and edges: the shadowed ones are not visible from the observer viewpoint because they have been removed by the 3-D visualization hardware.

Instead of processing the geometry database that mathematically defines the arbitrary surface, the 3-D image of the object is processed in order to obtain the relevant geometry information for illuminated surfaces, namely the (x, y, z) coordinates and the unit normal (n_x, n_y, n_z) associated with the image pixels. Computation of the PO surface integral is now simple, fast and independent of the geometric model (same code for facet

*Dept. of Signal Theory and Communications, Universitat Politecnica de Catalunya, edifici D-3, Jordi Girona 1-3, 08034 Barcelona, Spain.

or NURBS). Furthermore, since all the geometric transformations necessary to discretize parametric surfaces into pixels are performed by the hardwired graphics processor of the workstation, the CPU is freed from the computationally intensive geometric processing tasks.

However, the discretization of 3-D geometric models into pixels by the graphics processor produces some "modeling noise" in the RCS prediction. The aim of this paper is the study of discretization errors arising in the PO computation by graphical processing techniques, for both facet and parametric surface models.

II. Pixel discretization of a geometry model

For computation of the PO surface integral, an arbitrary surface is discretized into a collection of small patches associated to screen pixels. In this procedure, five different geometry models are sequentially processed:

1. **Original surface** (cylinder, sphere, etc.)

2. **Geometric model** created by the CAD software package and stored in a geometry database. As mentioned before, two kinds of geometric models are widely used:

   - Facets model: collection of flat polygonal facets.
   - Parametric surface model (NURBS): Each surface patch is represented as a piecewise function corresponding to a ratio of polynomials [5] [6].

3. **Rendering model.** The graphics processor is able to render only flat triangular facets, defined by the (x, y, z) coordinates and the unit normal $\hat{n} = (n_x, n_y, n_z)$ at triangle vertices. The CAD geometric model is thus discretized into a triangle mesh by the geometry engine of the graphics processor.

   - Facets model: Polygonal facets of more than three sides are subdivided into adjacent coplanar triangles. Unit normals (n_x, n_y, n_z) at the vertices of triangles are set equal to the unit normal to the facet.
   - Parametric surface model: NURBS are discretized into a triangular facets mesh by sampling the piecewise polynomials. This process is called tessellation. Unit normals at triangle vertices are set equal to the unit normals to parametric surface at the vertices location. The resolution in this triangular mesh tessellation is set as a user parameter.

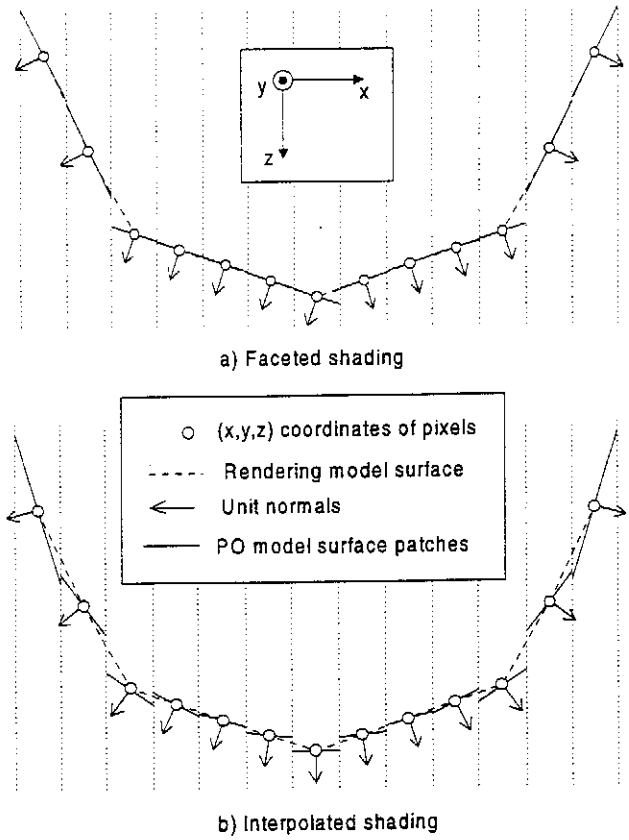

Figure 1 – Computation of the unit normal to pixels by different rendering techniques: a) flat or faceted shading sets all the normals equal to the normal to the facet, b) interpolated or smooth shading interpolates the normals at the triangle vertices. This figure has been obtained from the (x, y, z, n_x, n_y, n_z) information available at the *frame-buffer* and *z-buffer* of a Hewlett-Packard CRX-48Z graphics processor.

4. **Pixel model** is defined by the (x, y, z, n_x, n_y, n_z) information at each image pixel. The graphics processor provides the x, y coordinates from the pixel row and column, the z coordinate from the *z-buffer* [11] [6], and the (n_x, n_y, n_z) components of the unit normal from the color information available at the *frame-buffer* [11] [12]. The unit normal at each pixel depends on the shading technique used to render the image [6]:

   - Flat or faceted shading: The unit normal is set constant for all pixels inside a triangle (figure 1a).
   - Interpolated or smooth shading: The unit normal at pixels interior to triangles is obtained from bilinear interpolation of the unit normals at the vertices of triangles (figure 1b).

- For facets models, the image resulting from interpolated shading is the same as with flat shading.
- For parametric surface models, the triangle mesh with interpolated shading has a smooth, not faceted, appearance.

5. **Physical optics model**: From the pixel model information, (x, y, z, n_x, n_y, n_z), a surface patch must be associated to each pixel so that the PO surface integral is computed as the addition of contributions of all surface patches. There exist different PO models that associate to pixels surface patches of different shape and/or orientation.

III. Physical Optics Models

Three PO models have been programmed in the *GRECO* code and are compared in this paper. For simplicity, we will consider only the monostatic RCS case. We define the 3-D coordinate axis as follows: x, y are the 2-D coordinates of the workstation screen while z is perpendicular to the screen and points to the direction of observation. We assume that pixels are square in the xy plane, with side h and area h^2. The monostatic PO surface integral over the surface patch S associated to a pixel is

$$I = \int_S e^{j2k\hat{r}\cdot\vec{r}'} \cos\theta \; ds = \int_S e^{j2kz'} \cos\theta \; ds \qquad (1)$$

where k is the wavenumber, $\hat{r} = \hat{z}$ the direction of observation, $\vec{r}'$ the position vector to a point of S and θ the angle between the direction of observation and the unit normal, $\cos\theta = \hat{r} \cdot \hat{n} = n_z$. As mentioned before, different PO models associate to pixels surface patches S of different shape and/or orientation:

A. *Circular pixel approximation*

This model was developed in the first version of *GRECO* code [11] through an heuristic trial and error procedure, in order to reduce the modeling errors due to low-resolution triangular mesh tessellation of NURBS. This allowed a reduction of the number of triangular facets in the rendering model and, therefore, an improvement in the rendering time.

When the resolution in the triangular mesh is poor, a better approximation of the unit normals to the original surface by the unit normals of the pixel model can be achieved by rendering with the interpolated shading technique. The surface patch associated with each pixel is assumed as an ellipse centered at (x, y, z) with unit normal $\hat{n} = (n_x, n_y, n_z)$, that projects into the xy plane as a circle of area h^2 (figure 2). The PO integral for this surface patch is

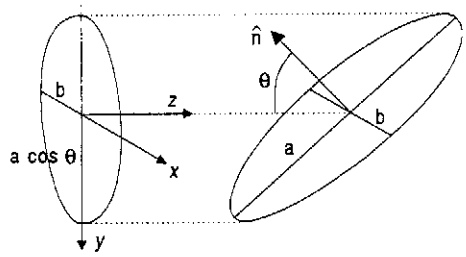

Figure 2 – Circular pixel model: the surface patch associated with a pixel is an ellipse of semiaxis $a = h/(\sqrt{\pi}\cos\theta)$ and $b = h/\sqrt{\pi}$, that projects into the xy plane as a circle of area h^2. The ellipse is centered at (x, y, z) and oriented with the normal $\hat{n} = (n_x, n_y, n_z)$.

$$I = 2h^2 \frac{J_1(\frac{2}{\sqrt{\pi}} kh \tan\theta)}{\frac{2}{\sqrt{\pi}} kh \tan\theta} \; e^{j2kz'} \qquad (2)$$

In order to speed up the computation, the $J_1(v)/v$ function in equation (2) can be tabulated against $\cos\theta = n_z$ [11]. However, the whole table must be recalculated for each radar operating frequency, and for that reason the first order Bessel function $2J_1\left(\frac{2}{\sqrt{\pi}}u\right) / \left(\frac{2}{\sqrt{\pi}}u\right)$ is approximated by $\eta_1(u) = \sin(u)/u$. These two functions have the main lobe almost equal, but the second one can be computed much faster using the floating point unit of the workstation. This results in the formula

$$I = h^2 \eta_1 (kh \tan\theta) \; e^{j2kz'} \qquad (3)$$

which is used in reference [11] to obtain equation (6).

In order to remove the spurious PO contribution from the shadow boundary [1] and to decrease the error from the region of fast phase variation, an optional correction factor $\cos^n\theta$ is included [11], where n is a parameter set by the user,

$$I = h^2 \eta_1 (kh \tan\theta) \cos^n\theta \; e^{j2kz'} \qquad (4)$$

which leads to equation (7) in [11].

B. *Tangent plane approximation*

In this approach, recently developed by J.S. Asvestas [13], the surface patch associated to each pixel is a flat plate with $\hat{n} = (n_x, n_y, n_z)$, limited by the square pixel boundaries in the (x, y) screen (figure 3). The PO contribution from each surface patch is:

$$I = h^2 \eta_1 \left(kh\frac{n_x}{n_z}\right) \eta_1 \left(kh\frac{n_y}{n_z}\right) e^{j2kz'} \qquad (5)$$

Defining $\hat{n} = \hat{x}\sin\theta\cos\phi + \hat{y}\sin\theta\sin\phi + \hat{z}\cos\theta$, (5) simplifies to

$$I = h^2 \eta_1 (kh \tan\theta\cos\phi) \, \eta_1 (kh \tan\theta \sin\phi) \, e^{j2kz'} \qquad (6)$$

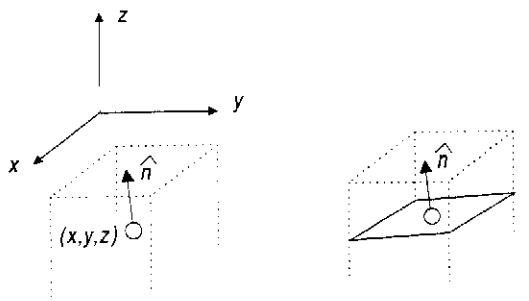

Figure 3 – Tangent plane approximation: the surface patch associated with a pixel is a quadrilateral with unit normal $\hat{n} = (n_x, n_y, n_z)$ limited by the square boundaries of the pixel centered at (x, y).

C. *Triangle mesh model*

The PO surface integral can be computed exactly over the rendering model triangle mesh. From the (x, y, z) coordinates associated with a group of four adjacent pixels arranged in a 2×2 pattern, the vertex coordinates of two 3-D triangles are obtained (figure 4). The PO contribution from each triangle is calculated by using Gordon's formula [2] [1]. In the implementation of this PO model, two questions arise:

1. When a surface is eclipsing part of another one, adjacent pixels that belong to different surfaces generate spurious triangles that are grazing to the incidence direction. The PO contribution from those triangles is small compared to that of triangles with broadside incidence, and thus no special care is required.

2. For each 2×2 pixel pattern, there are two possible triangle pairs. The choice between those two possibilities is arbitrary and of small significance in the final results.

The bistatic Gordon's formula for a polygonal facet with M sides of center points $\vec{\ell}_m$ and vector from origin to end vertices $\Delta\vec{\ell}_m$, with $\hat{h}_i$ polarization of the incident magnetic field and $\hat{e}_r$ polarization of the electric field receiver is [2] [1]:

$$I = \frac{-\hat{n}\cdot\left(\hat{e}_r \times \hat{h}_i\right)}{kT} \sum_{m=1}^{M} \hat{p}\cdot\Delta\vec{\ell}_m \; e^{-jk\vec{\ell}_m\cdot\vec{w}} \; \eta_1\left(\frac{k\Delta\vec{\ell}_m\cdot\vec{w}}{2}\right) \tag{7}$$

where in the monostatic case, with observation direction $\hat{r} = \hat{z}$, we have $\hat{e}_r \times \hat{h}_i = -\hat{z}$, $\vec{w} = \hat{i} - \hat{r} = -2\hat{z}$,

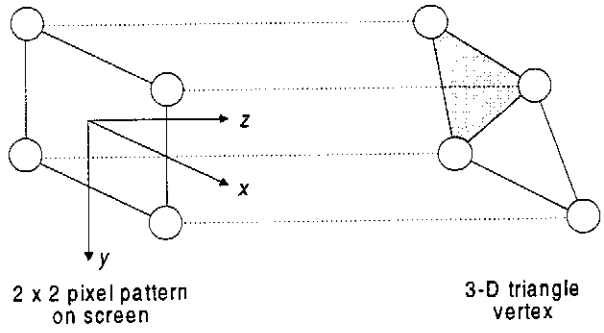

Figure 4 – Triangular mesh model. The surface associated with four adjacent pixels arranged in a 2×2 pattern in the screen is a set of two triangles with a common side.

$T = |\hat{n}\times\vec{w}| = 2\sqrt{n_x^2 + n_y^2}$, $\hat{p} = (\hat{n}\times\vec{w})/T = 2(n_x\hat{y} - n_y\hat{x})/T$. The monostatic version of (7) is, therefore:

$$I = \frac{n_z}{k}\sum_{m=1}^{M} \frac{n_x\Delta y_m - n_y\Delta x_m}{2\left(n_x^2 + n_y^2\right)} \; e^{j2kz_m} \; \eta_1\left(k\Delta z_m\right) \tag{8}$$

where $z_m = \hat{z}\cdot\vec{\ell}_m$ and $\Delta\vec{\ell}_m = \Delta x_m\hat{x} + \Delta y_m\hat{y} + \Delta z_m\hat{z}$.

For broadside incidence, $n_x = n_y = 0$, there is a removable singularity in (8), that can be replaced by the limit

$$\lim_{n_x^2+n_y^2\to 0} I = h^2 \; e^{2jkz'} \tag{9}$$

A very good technique for avoiding computational instabilities when computing the value of the integral I near the singularity is described in [14].

It must be noted that the unit normal information (n_x, n_y, n_z) available in the pixel model is not used. The unit normal in (8) is computed for each triangle from the vertices' coordinates as $\hat{n} = \left(\Delta\hat{\ell}_1 \times \Delta\hat{\ell}_2\right) / \left|\Delta\hat{\ell}_1 \times \Delta\hat{\ell}_2\right|$.

D. *Theoretical comparison of PO models*

A theoretical comparison of these three PO models leads to the following conclusions, which will be confirmed by numerical results: the tangent plane approximation and triangular mesh models are obviously more accurate than the circular pixel approximation. In fact, the later, equation (3), is equal to the tangent plane approximation (6) only if the observation direction is in the xz or the yz plane, $\phi = 0, \pi/2, \pi, 3\pi/2$. Large errors from a single pixel contribution are obtained for observation directions away from the xz and yz planes and $kh\tan\theta > 0.5$.

However, the main contribution to the PO surface integral over the whole object is obtained from the pixels

near the stationary phase point [1], which approximately has $\theta < 1/(2ka)$, where a is the radius of curvature of the surface at the stationary phase point and in the direction of evaluation of the integral; assuming that these pixels have the derivative of the phase with respect to the distance to the stationary phase point smaller than 1, $d(2kz')/dy < 1$. For these 'stationary phase pixels', the argument of the η_1 function in (6) is, thus, very small, $kh\tan\theta < h/(2a) \ll 1$, if the pixel size h is much smaller than the radius of curvature a. The value of the PO integral for stationary phase pixels is therefore $I \approx h^2\, e^{j2kz'}$ for both the circular pixel (3) and the tangent plane approximations (5) (6), independently of the electrical size of the pixel.

On the other hand, for electrically large objects the circular pixel approximation obtains a large error for pixels in the region of fast phase variation near the shadow boundary, where $kh\tan\theta > 0.5$. However, the contribution of these pixels is small, and can be further reduced with the $\cos^n\theta$ correction factor (4). This explains why results of the circular pixel approximation (3) (4) are acceptable for electrically large objects in references [11] [12].

The triangular mesh model gives the most accurate evaluation of the PO surface integral over the rendering model triangular mesh. Results should be slightly better than with the tangent plane approximation because the later uses the unit normal information at pixels. This leads to a PO model with a discontinuous surface at triangle edges if the faceted shading technique is used (figure 1a), or at pixel edges if the interpolated shading technique is used (figure 1b). On the other hand, the triangular mesh model does not use the unit normal information at pixels and therefore has a continuous surface corresponding exactly to the rendering model, with both faceted or interpolated shading techniques.

IV. Sources of discretization errors

Discretization errors arise from the transition between the five geometry models presented in section II.:

1. From the original surface to the CAD geometric model.
   - Significant discretization errors for facets models.
   - Negligible errors for parametric surface models.
2. From geometric model to rendering model (triangle mesh tessellation).
   - No error on facets models.
   - There is a discretization error for parametric surface models. However, since usually the resolution of the triangular mesh is much better than the resolution of a facets model, this error is often negligible compared to the discretization error in the facet models and needs to be accounted for only in low-resolution triangular meshes.

The facetization error from the original surface to the rendering model, arising either from the facets models or from the triangle mesh tessellation, sets an upper bound on the electrical size of the object. For example, a sphere of radius a discretized into M triangles has approximately $M/2$ non-planar quadrilaterals and $N = \sqrt{2\frac{M}{2}} = \sqrt{M}$ straight segments along its circumference (figure 5). The facetization error is therefore:

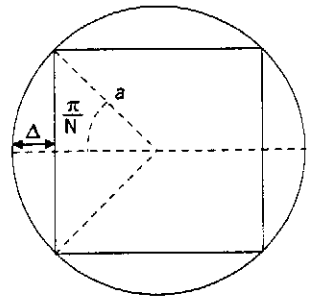

Figure 5 – A sphere discretized into $M = 16$ triangles, corresponding to $M/2 = 8$ non-planar quadrilaterals, has its circumference approximated by $N = \sqrt{2\frac{M}{2}} = 4$ straight segments.

$$\Delta \sim a\left(1-\cos\frac{\pi}{N}\right) \approx \frac{a}{2}\frac{\pi^2}{M} \qquad \text{if } \frac{\pi}{N} << 1 \quad (10)$$

The usual maximum allowed phase error of $\Delta < \lambda/20$ leads to a maximum frequency

$$f_{\max} = \frac{3}{100}\frac{M}{\pi^2 a} \simeq \frac{3M}{1000\,a} \qquad \text{GHz} \qquad (11)$$

3. From rendering model to pixel model. Errors at this step depend on the shading technique used for rendering the model:
   - Flat or faceted shading: no error.
   - Interpolated shading: due to bilinear interpolation, the unit normal at pixels interior to triangles is not perpendicular to the triangle mesh surface. The amount of noise introduced by this step depends on the PO model: The

triangular mesh PO model (8) does not use the unit normal information at pixels, and is, therefore, free from that source of error. On the other hand, some noise is introduced in the tangent plane (6) and circular pixel (2) approximations. In the later, this noise can cancel, in part, the facetization error due to parametric surface tessellation.

- Additional errors at this step are introduced by the hardwired graphics processor. The computation of z coordinate and unit normal $\hat{n} = (n_x, n_y, n_z)$ for all pixels and their quantization in, respectively, the *z-buffer* and *frame-buffer* may not be correct, specially near intersection curves of parametric surfaces. Figure 6 shows the errors in the z coordinate for the image of a right circular cylinder.

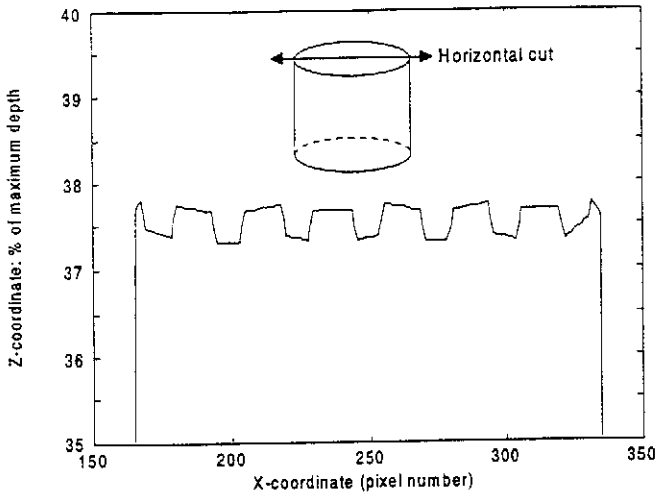

Figure 6 – Errors in z coordinate for the end cap of a right circular cylinder. The z coordinate in a horizontal screen row has been plotted versus the discrete x coordinate. z should be constant and equal to 37.5 % of the maximum depth, but an error of 0.25 % of maximum depth can be observed. The image has been obtained from the *z-buffer* of a Hewlett-Packard CRX-48Z graphics processor.

4. From pixel model to PO model:

   - Circular pixel approximation: PO surface integral is approximated by the summation of contributions from elliptical surface patches, which is not equal to the integral over the surface of the rendering model (triangular facets mesh). This introduces an error, which depends on the shading technique used and is small for pixels near the stationary phase point.
   - Tangent plane approximation:
     - Facets models with any shading and parametric surface models with faceted shading: Since unit normals at all pixels are perpendicular to the rendering model surface, the PO surface integral over the surface patches is almost equal to the integral over the rendering model. Small errors come from the surface discontinuity at triangle or facet edges (figure 1a).
     - Parametric surface models with interpolated shading: Since unit normals at pixels interior to triangular facets are not perpendicular to the triangle mesh surface, the surface patches associated to pixels are disconnected and not parallel to the rendering model (figure 1b). The PO surface integral obtained is thus not equal to the integral over the rendering model, but the error is in general smaller than the error due to the circular pixel approximation.
   - Triangular mesh model: Since the triangles defined by the (x, y, z) coordinates of pixels are tangent to the rendering model surface and the unit normal to the surface is obtained from the triangle vertices coordinates, this PO model obtains an exact computation of the PO integral over the rendering model.

5. *Frame-buffer* resolution: The *frame-buffer* is a portion of RAM memory at the graphics processor that contains the color information for all pixels at the workstation screen. This color information is associated with the (n_x, n_y, n_z) components of the unit normal [11], [12]. For obvious memory size restrictions, the *frame-buffer* size is usually equal to or less than 1280 x 1024 pixels. This produces a negligible quantization error for maximum size image windows (usually 1024 x 1024 pixels). However, since the CPU time for computing graphically the PO integral is proportional to the number of illuminated pixels in the object image, faster computation can be achieved by rendering in a smaller window, at the cost of increasing the discretization error.

   Even for small image windows, the error in the RCS computation is negligible for broadside incidence. However, surfaces with grazing incidence have a smaller projection on the workstation screen in the

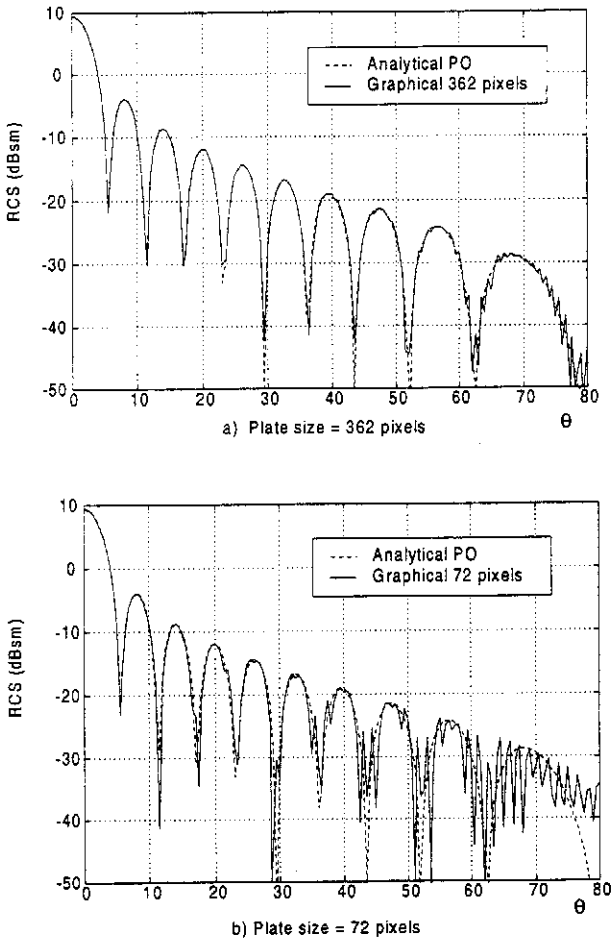

Figure 7 – Monostatic RCS of a $5\lambda \times 5\lambda$ square plate versus aspect angle at a frequency of 9.227 GHz. Analytical PO (dash line) is compared with the results of graphical computation (solid line) for a Hewlett-Packard CRX-24Z graphics processor. Plots a) and b) respectively correspond to 362×362 pixel and 72×72 pixel plates.

direction of fast phase variation, which produces a significant quantization error in the RCS results. For example, a square plate that with broadside incidence occupies 72 x 72 pixels at the workstation screen, with $\theta = 69$ deg incidence occupies 72 x 25.8 pixels. Since the plate size in the direction of grazing incidence must be approximated by either 25 or 26 pixels, an error up to 3.2 % is made in the plate size, which produces a 15 dB error in the computation of the PO integral. The result of the *frame-buffer* quantization error is a spurious oscillation in the monostatic RCS plot for large aspect angles. These oscillations are larger for poorer surface discretization in pixels. Figure 7 shows the RCS in dBsm versus aspect angle for a 5λ square plate at a frequency of 9.227 GHz, computed graphically with a Hewlett-Packard CRX-24Z graphics accelerator. PO integral has been computed with the circular pixel model (3). Figure 7 a) and b) correspond respectively to 362 x 362 and 72 x 72 pixel size of the flat plate.

In conclusion, the *frame-buffer* discretization errors are significant only for grazing incidence, when the PO contribution of the surface analyzed is usually masked by edge diffraction, the reflection from some specular point or the broadside PO contribution of another surface. We are encouraged by this fact to process small size images of the object in order to reduce computation time.

6. *Z-buffer* quantization: The *z-buffer* is a portion of RAM memory at the graphics processor that contains the z coordinates of all pixels at the workstation screen. If the z coordinate is discretized with b bits, there are 2^b depth levels between the point of the object closest and furthest to the observer. With a usual sampling of $\lambda/16$, the maximum size of the object is $2^b/16 = 2^{b-4}$ wavelengths. This is equal to 4000λ for a 16-bit *z-buffer*. For curved surfaces, the *z-buffer* quantization error is in general less restrictive than the facetization error from the original surface to the rendering model triangle mesh, equation (11). For flat surfaces, the *z-buffer* quantization error is less restrictive than the *frame-buffer* resolution error.

V. Results

In order to test numerically the accuracy and CPU times for the different PO models, a NURBS model of a sphere of radius equal to 1 meter has been analyzed. Figures 8, 9 and 10 show the monostatic RCS versus frequency normalized with the geometrical optics approximation πa^2 for a discretization into, respectively, $M = 6400$, 40000 and 160000 triangles. The maximum allowed frequency according to equation (11) is, respectively, 19 GHz, 120 GHz and 480 GHz. Results for the three different PO models and for the faceted and interpolated shading techniques are compared.

The plots in figures 8, 9 and 10 lead to the following conclusions:

- The triangle mesh PO model gives exactly the same results for faceted and interpolated shading techniques, as expected.
- For $f < f_{\max}$ (11):
  - the best results are obtained with the triangle mesh PO model,

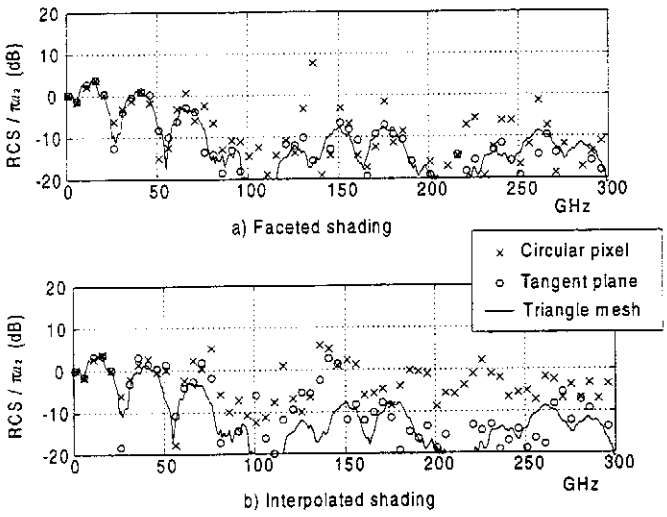

Figure 8 – Monostatic RCS versus frequency for a sphere with radius equal to 1 meter, normalized with the geometrical optics approximation πa^2. The parametric surface has been discretized into $M = 6400$ triangles; and, therefore, the maximum allowed frequency (11) is 19 GHz. Results for the three different PO models are compared.

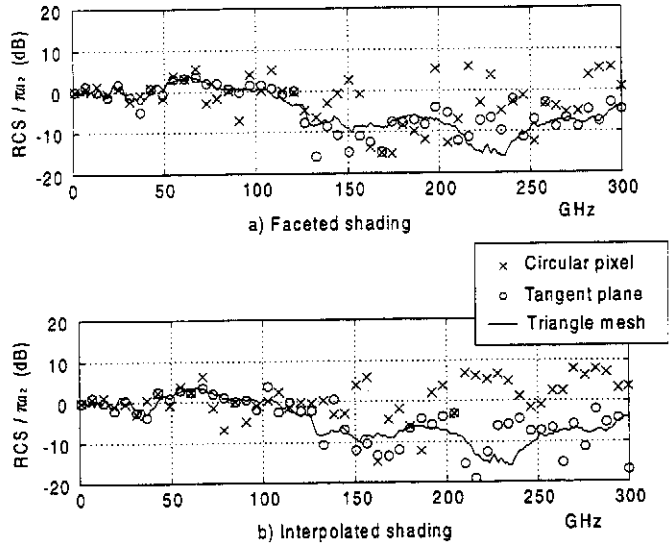

Figure 9 – Monostatic RCS versus frequency for a sphere with radius equal to 1 meter, normalized with the geometrical optics approximation πa^2. The parametric surface has been discretized into $M = 40000$ triangles; and, therefore, the maximum allowed frequency (11) is 120 GHz. Results for the three different PO models are compared.

– the tangent plane approximation is slightly worse than the triangle mesh model, probably due to the discontinuities at triangle edges (faceted shading) or pixel edges (interpolated shading),

– The circular pixel approximation is out (errors up to ±10 dB) for $f > f_{\max}/2$ in figure 9 and $f > f_{\max}/3$ in figure 10.

- For $f > f_{\max}$ (11):
  - all three PO models are out,
  - in the circular pixel and tangent plane approximations, the results obtained with faceted and interpolated shading techniques are completely different,
  - in most cases, the best results (average error about ±5 dB) are obtained with the circular pixel approximation and the interpolated shading technique.

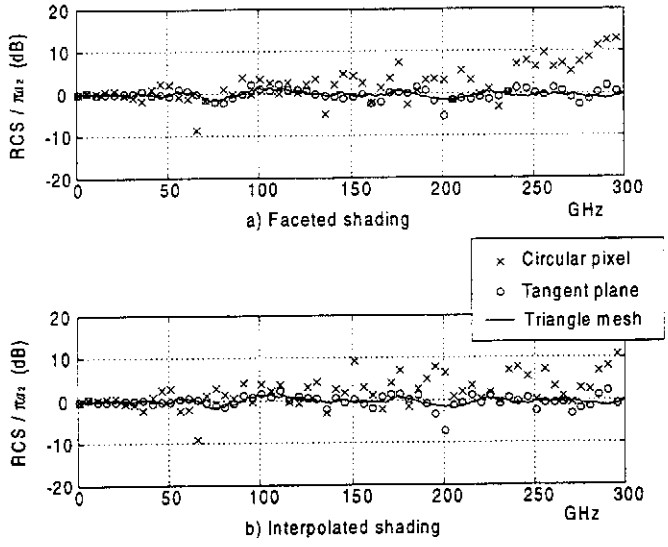

Figure 10 – Monostatic RCS versus frequency for a sphere with radius equal to 1 meter, normalized with the geometrical optics approximation πa^2. The parametric surface has been discretized into $M = 160000$ triangles; and, therefore, the maximum allowed frequency (11) is 480 GHz. Results for the three different PO models are compared.

CPU run times corresponding to the three PO models for a Hewlett Packard 735 CRX 48-Z workstation are shown in table 1.

VI. Conclusions

The sources of discretization errors in graphical computation of the PO surface integral have been stud-

PO model	Indexed tables	PO integral
Circular pixel	16 ms	40 ms
Tangent plane	91 ms	138 ms
Triangular mesh	163 ms	577 ms

Table 1. CPU run times in a Hewlett Packard 735 CRX 48-Z workkstation for the sphere model described in figures 9, 10 and 11. The second column shows the indexed tables computation time. Those indexed tables, that contain the values of the complex exponential and η_1 functions to speed up computation, must be recomputed for each frequency. The third column shows the PO surface integral computation time, for a single frequency and view angle.

ied. Three different PO models have been presented. The RCS versus frequency plots for a sphere show that when the resolution in the surface discretization is high enough for the working frequency, $f < f_{\max}$ (11), the best results are obtained with the triangle mesh PO model (Gordon formula) [2]. Tangent plane approximation of J.S. Asvestas [13] gives slightly worse results, but the computation is 4 times faster. The circular pixel approximation of J.M. Rius [11] [12] is the fastest, but errors up to ± 10 dB for the highest frequency ranges make the results unacceptable for most applications. Accordingly, for high resolution surface discretization the best trade-off between CPU time and accuracy is achieved with the tangent plane approximation.

When the resolution in the surface discretization is not high enough for the working frequency, $f > f_{\max}$ (11), all three PO models are inaccurate. In most cases, the best results (average error about ± 5 dB) are obtained with the circular pixel approximation and the interpolated shading technique. This is due to the use of interpolated unit normals in pixels inside the flat triangles of the rendering model.

VII. Acknowledgements

This work has been supported by the Spanish "Comisión Interministerial de Ciencia y Tecnología" (CYCIT) under the project TIC95-0983.

References

[1] E.F. Knott, J.F. Shaeffer, J.F. Tuley, *Radar cross section*, Artech House, 2nd Ed., (1993).

[2] W.B. Gordon, "Far Field Approximation of the Kirchhoff-Helmholtz Representation of Scattered Fields", *IEEE Trans. on Antennas and Propag.*, **23**, no.5, pp. 864-876, July 1975.

[3] N.N. Youssef, "Radar cross section of complex targets", *Proc. IEEE*, **77**, no.5, pp. 722-734, May 1989.

[4] M. Domingo, R.P. Torres, M.F. Cátedra, "Calculation of the RCS from the interaction of edges and facets ", *IEEE Trans. on Antennas and Propag.*, *42*, no.6, pp. 885-887, June 1994.

[5] G. Farin, *Curves and surfaces for computer aided geometric design: a practical guide*, Academic Press, (1988).

[6] J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes, *Computer Graphics: Principles and Prcatice*, Addison-Wesley, (1992).

[7] J. Pérez, M.F. Cátedra, "RCS of electrically large targets modelled with NURBS surfaces", *Electronics Letters*, **28**, no. 12, pp. 1119-1121, June 1992.

[8] J. Pérez, M.F. Cátedra, "Application of physical optics to the RCS computation of bodies modeled with NURBS surfaces", *IEEE Trans. on Antennas and Propag.*, **42**, no. 10, pp. 1404-1411, October 1994.

[9] J. Pérez, J.A. Saiz, O.M. Conde, R.P. Torres and M.F. Cátedra, "Analysis of Antennas on Board Arbitrary Structures Modelled by Nurbs Surfaces", *IEEE Transactions on Antennas and Propag.*, vol. 45, no. 6, pp. 1045-1053, June 1997.

[10] S.W.Lee, "XPATCH: A High-Frequency Electromagnetic Scattering Prediction Code and Environment for Complex Three-Dimensional Objects", *IEEE Antennas and Propag. Magazine*, **36**, no. 1, pp. 65-69, February 1994.

[11] J.M. Rius, M. Ferrando, Ll. Jofre, "RCS of complex radar targets in real time", *IEEE Trans. on Antennas and Propag.*, **41**, no.9, pp. 1308-1319, September 1993.

[12] J.M. Rius, M. Vall-llossera, A. Cardama, "GRECO: Graphical Processing Methods for High-Frequency RCS Prediction ", special issue on RCS of complex objects of the *Annals of Telecommunication*, **50**, no. 5-6, May-June 1995.

[13] J.S. Asvestas, "The Physical Optics Integral and Computer Graphics", *IEEE Trans. on Antennas and Propag.*, **43**, no. 12, pp. 1459-1460., December 1995.

[14] J.S. Asvestas, "A class of functions with removable singularities and their application to the physical theory of diffraction", *Electromagnetics*, **15**, pp. 143-155, 1995.