GRECO: 30 years of graphical processing techniques for RCS computation

Juan M. Rius$^{(1)}$, Eduard Ubeda$^{(1)}$, Alexander Heldring$^{(1)}$, etc.
rius@tsc.upc.edu, ubeda@tsc.upc.edu, heldring@tsc.upc.edu
$^{(1)}$CommSensLab, Universitat Politècnica de Catalunya - BarcelonaTech (Spain).

***Abstract*—This contribution to the special session in honor of Prof. Rafael Gómez-Martín will address the 30 year development of graphical processing techniques (GRECO) for fast computation of Radar Cross Section (RCS) of electrically large and complex targets. The development of GRECO started in 1988, in the frame of the "Applied research project for the development and validation of numerical methods for RCS prediction, analysis and optimization", in which I had the pleasure to know Rafael since our groups where participating together in the project.**

The development of GRECO never stopped, and recently it has been updated by replacing the graphical processing technique for computation of surface reflection and edge diffraction by a hybrid CPU-graphical processing approach. The resulting code has the same accuracy as conventional RCS computation techniques, but detection of shadowed surfaces and edges is one order of magnitude faster than the most efficient $\mathcal{O}(N \log N)$ implementations.

I. Introduction

Graphical Electromagnetic Computing *(GRECO)* technique was first introduced in 1993 [1] [2] for efficient high-frequency RCS prediction. The novel idea was to compute surface reflection with Physical Optics (PO) and edge diffraction with Physical Theory of Diffraction (PTD) [3] by processing the bitmap containing a 3D image of the target instead of the facet mesh or parametric surfaces of the geometry model. This new approach was called *Graphical Processing*. Although on one hand it allowed very fast computation with the CPU's available at that time, clocked only at tenths of MHz, on the other it introduced spurious errors in the results due to pixel discretization noise [4].

In order to take advantage of the power of modern CPU's, the GRECO technique was updated in 2014 by introducing a hybrid approach [5], in which bitmap graphical processing was used only to determine the visibility of facet vertices, and then PO was computed using the conventional Gordon's formula [6].

The comparison of computation speed and accuracy of both approaches (the original pixel-based and the new facet-based hybrid graphical processing) was presented in [5] for PO computation, so in this paper we will show only the comparison results for PTD.

Section II will outline the minimum necessary ideas to understand graphical processing. Then, section III will present the pixel-based algorithm for edge analysis in the target image. Finally, section IV will compare the PTD results of both graphical processing techniques.

II. Graphical Electromagnetic Computing

GRECO is based in processing the bitmap containing a 3-D image of the target. If the viewpoint of the observer is located at infinity with the same view direction as the monostatic radar, then the target image contains only the surfaces and edges illuminated by the incident plane wave: the shadowed ones, that do not contribute to high-frequency scattering, have been removed by the graphics card hardware because they are not visible from the observer viewpoint.

RCS is obtained by *GRECO* technique in two main steps: (i) the geometric and graphical processing part, which obtains the unit normal to the surface at each illuminated pixel of the target, and (ii) the electromagnetic part, which processes the unit normals using high-frequency approximations.

A. Computation of unit normal to surface

The surface unit normal at all pixels corresponding to visible surfaces is obtained by illuminating the scene with Phong's reflection model [7]. Using diffuse surface reflection, the brightness of all pixels is computed separately for each (R,G,B) color component proportionally to the projection $\hat{n}\cdot\hat{r}_i$ of the unit normal to the surface $\hat{n}$ into the incidence direction $\hat{r}_i$. If we set three light sources of purely red, green and blue colors, respectively located at infinity in the (x, y, z) coordinate directions, the three (R,G,B) color components of the image pixels are equal to the (n_x, n_y, n_z) components of the unit normal to the surface at these pixels. Therefore, the blending of the three (R,G,B) color components is equal to the (n_x, n_y, n_z) components of the unit normal. The ambiguity in the sign of (n_x, n_y, n_z) is resolved by displaying separately two different (R,G,B) images.

The x, y coordinates are oriented in the horizontal and vertical directions of the image bitmap, while z is parallel to the observer view direction so that that the z coordinate of each pixel is equal to the distance from the observer to the associated surface element. The z-coordinate information, that is crucial for coherently adding the contribution of all surface elements to PO and PTD integrals, is retrieved from the graphics card *z-buffer*.

B. Pixel-based PO computation

The PO contribution is computed very efficiently from the (x, y, z) coordinates and the corresponding unit normal components (n_x, n_y, n_z) of each illuminated pixel using the Asvestas' tangent plane approximation [8]. Implementation details can be found in [4] and [5].

C. Facet-based hybrid PO and PTD computation

As thoroughly explained in [5], the latest GRECO software computes RCS using Gordon's formula [6] for all illuminated facets. Facet visibility is determined by projecting the 3-D coordinates of facet vertices on the 2-D target image, and comparing the resulting z coordinate of the vertex with that of the visible surface at the same x, y pixel position. If the image pixel is closer to the observer than the facet vertex (it has smaller z-coordinate), the vertex is located in a shadowed surface. Partially visible facets are subdivided.

Since in general the projected x, y coordinates of facet vertices do not exactly coincide with the center of image pixels, it is necessary to extrapolate the projected vertex z coordinate to the center of the pixel before the shadowing test [5].

The same procedure has been implemented for PTD computation using the conventional Ufimtsev's formulas [3]. The edge vertices are tested for visibility by comparing the projected-extrapolated z coordinate with that of the corresponding image pixel. Again, if the edge is partially visible it is subdivided.

Of course, in order to avoid testing the same vertex several times for all the facets and edges sharing this vertex, a vertex visibility look-up table is generated and only the table entries are checked when finding shadowed facets and edges.

III. Edge analysis with pixel-based graphical processing

The graphical processing approach for edge analysis presented in [1] [2] was based on detecting edges as discontinuities in the surface unit normal when the z-coordinate remained continuous. This approach was (i) prone to edge mis-detection and (ii) not able to know the unit normal of the hidden face of wedges having only one visible face.

Both issues where fixed later by developing a new graphical edge processing algorithm that has not been well-disseminated yet and that works surprisingly well for processing *only* the target image. This algorithm is as follows:

A. Unit normals to non-visible wedge faces

In order to obtain the unit normal to the hidden face of wedges with only one illuminated face, two more (R,G,B) color images are generated in which the unit normals of the model have been reversed (Figure 1). Since the graphics hardware removes the *back-facing* surfaces (the ones that have the unit normal pointing away from the observer) prior to rendering the *front-facing* ones (that have the unit normal pointing towards the observer half-space), the images with reversed surface normals contain the *back-facing* surfaces located immediately behind the surfaces visible in the conventional image (see Fig. 1). These *back-facing* surfaces that are now visible in the reversed-normals image include the hidden face of wedges, for which we need to compute the unit normal. The knowledge of these unit normals to hidden faces is necessary for computation of the edge diffraction coefficients and for detecting ray reflections in ray-tracing.

B. Edge detection

Now that we know the unit normal to surface both at the visible pixels and also at the hidden pixels located immediately behind the visible ones, we know the normal to both faces of wedges even if one of them is hidden: When only one face of the wedge is visible, then the other face is a *back-facing* surface and its unit normal can be obtained from the image with the unit normals reversed (Fig. 1). This allows us to compute the surface radii of curvature from the variation of the unit normal at contiguous pixels.

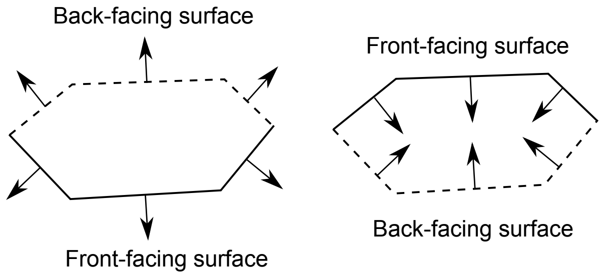

Fig. 1. Images with exterior normals to surfaces (left picture) and reversed normals (right picture). Only the surfaces whose normal points towards the observer half-space are rendered (*front-facing* surfaces, with solid lines in the picture), while the surfaces whose normals point away from the observed are removed (*back-facing* surfaces, with dashed lines in the picture). When all the unit normals have been reversed (right picture), the visible surfaces are now the previous *back-facing* ones located immediately behind the normally visible surfaces.

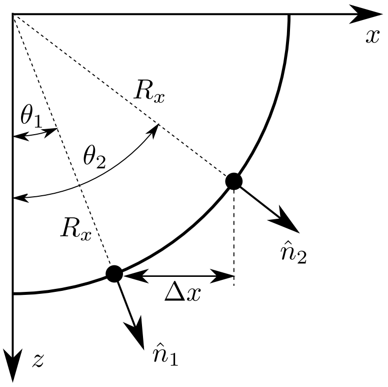

Fig. 2. Computation of the surface radius of curvature in the xz plane, $R_x = \Delta x / \Delta n_x$.

Figure 2 shows the procedure to compute the radius of curvature R_x in the xz plane:

$$\Delta x = R_x \sin\theta_2 - R_x \sin\theta_1 = R_x \left(n_{2x} - n_{1x}\right) = R_x \Delta n_x \tag{1}$$

The procedure for obtaining the radius of curvature in the yz plane, R_y, is analogous. The final expressions are:

$$R_x = \frac{\Delta x}{\Delta n_x}, \qquad R_y = \frac{\Delta y}{\Delta n_y} \tag{2}$$

These radius of curvature are not, in general, equal to the principal radius of curvature of the surface, and thus cannot be used to compute the geometrical optics contribution to the RCS.

Edges are detected when any of the radius of curvature (R_x in the xz plane or R_y in the yz plane) (2) defined by two contiguous pixels is smaller than a given threshold R_{min}, as shown in Fig. 3. The kind of discontinuities in the unit normal components to contiguous pixels that lead to edge detection can be also due to surface eclipses, but that situation is easily discriminated because in that case there is also a discontinuity in the z coordinate.

Once we have detected edges and know the unit normal to both faces, we can use the same formulas as in [1] [2] to obtain the angles that define the orientation of a wedge, which

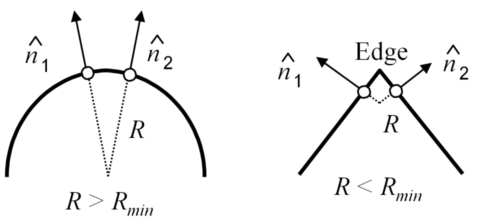

Fig. 3. Ege detection when radius of curvature R is smaller than R_{min}

are necessary to compute the PTD diffraction coefficients:

$$\begin{aligned}\alpha &= \arccos(-\hat{n}_1 \cdot \hat{n}_2) \\ \hat{t} &= \frac{\hat{n}_1 \times \hat{n}_2}{\sin\alpha} \\ \cos\beta_r &= \hat{t} \cdot \hat{r} = t_z \\ \sin\phi &= \frac{\hat{n}_1 \cdot \hat{r}}{\sin\beta_r} = \frac{n_{1z}}{\sin\beta_r}\end{aligned} \tag{3}$$

where $\hat{n}_1, \hat{n}_2$ are the unit normals to both faces of the wedge, $\hat{t}$ is the unit vector tangent to the edge, α is the wedge inner angle and β_r, ϕ are the spherical angles of the incident direction as defined in [3] and [2] (β_r is between the incident direction and $\hat{t}$ while ϕ is between the projection of the incident direction on the plane perpendicular to $\hat{t}$ and the first face of the wedge).

When the unit normals $\hat{n}_1$ and $\hat{n}_2$ are parallel, for example in the edges of a square plate, we cannot use (3) to compute the edge direction $\hat{t}$. In that case, a special processing of the edge is required in order to obtain the correct value of $\hat{t}$.

IV. Comparison of pixel-based with edge-based PTD computation

In order to compare the performance of the two graphical processing PTD algorithms, (i) the pixel-based one presented in sec. III and (ii) the hybrid approach of sec. II-C, we have selected two targets in which the presence of flat facets makes the PTD contribution significant compared with PO when the observation direction is not specular. One is a simple prism with very few edges and the other is a home-made model of an stealth aircraft made of hundreds of planar facets and no curved surfaces.

The RCS results obtained by GRECO for PO surface reflection combined with PTD edge scattering have been compared with the measurements and simulations for the benchmark targets published in [9]. One of the targets is a triangular prism as depicted in Fig. 4.

Fig. 5 shows the RCS measurements at 7GHz and FASCRO code simulations published in [9]. FASCRO code [10], is also based in PO and PTD high-frequency computations and is the seed of newFASANT commercial software [11]. It has been extensively validated by the authors and can be considered here as a reference benchmark.

Figure 6 shows the GRECO PO+PTD results for a stealth aircraft home-made model. The PTD pixel-based algorithm presented in in sec. III (red and black lines) is compared with the new hybrid facet-based approach (blue line) in which graphical processing is used only for detecting shadowed edges. The pixel based result for a small 506x123 pixels bitmap is contaminated with discretization noise. However, the result corresponding to a larger 1235x279 pixels bitmap agrees much better with the new facet-based hybrid approach, which is free from that kind of noise. The fact-based PO+PTD results are almost equal for the large and the small bitmaps, and only one of them has been shown here for clarity.

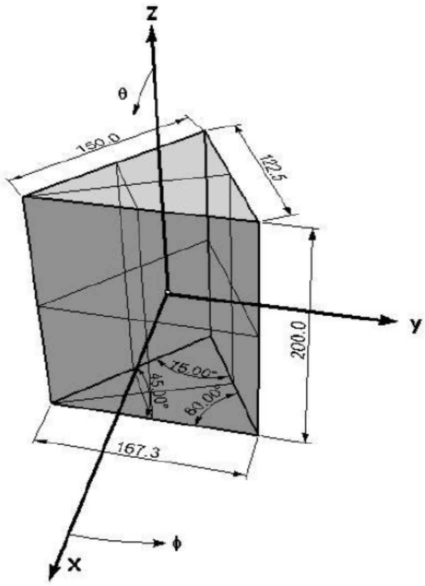

Fig. 4. Triangular prism defined in [9]. The dimensions are in mm and degrees. The height of the prism is 200mm and the largest face is normal to the x-axis. The sides of the triangular base are 167.3 mm (opposite angle 75), 122.5 mm (opposite angle 55) and 150 mm (opposite angle 60).

V. Conclusions

This paper has presented (i) the pure graphical processing algorithms for computation of surface reflection and edge scattering based on processing *only* a bitmap containing an image of the radar target and (ii) the new approach based on using the bitmap graphical processing algorithm to identify illuminated and shadowed facet mesh vertices [5], and then compute high frequency approximations for illuminated surfaces and edges in the conventional way.

The pixel-based approach is surprisingly accurate for simple targets, in spite of the crude discretization of the target surface into image pixels that produces a large amount of quantization noise for grazing incidence directions.

For complex targets in which specular reflection surfaces are avoided and the contribution of surfaces and edges with grazing incidence is not negligible, there is significant graphical processing noise in the pixel-based approach, specially for small bitmaps. When the bitmap size is increased, the amount of noise is reduced and the results converge to that of the facet-based hybrid approach.

Acknowledgements

This work was supported by FEDER and the “Spanish Plan Estatal de Investigación Científica y Técnica y de Innovación”, under projects TEC2016-78028-C3-1-P and TEC2017-84817-C2-2-R and the Unidad de Excelencia Maria de Maeztu MDM-2016-0600, which is financed by the Agencia Estatal de Investigación, Spain.

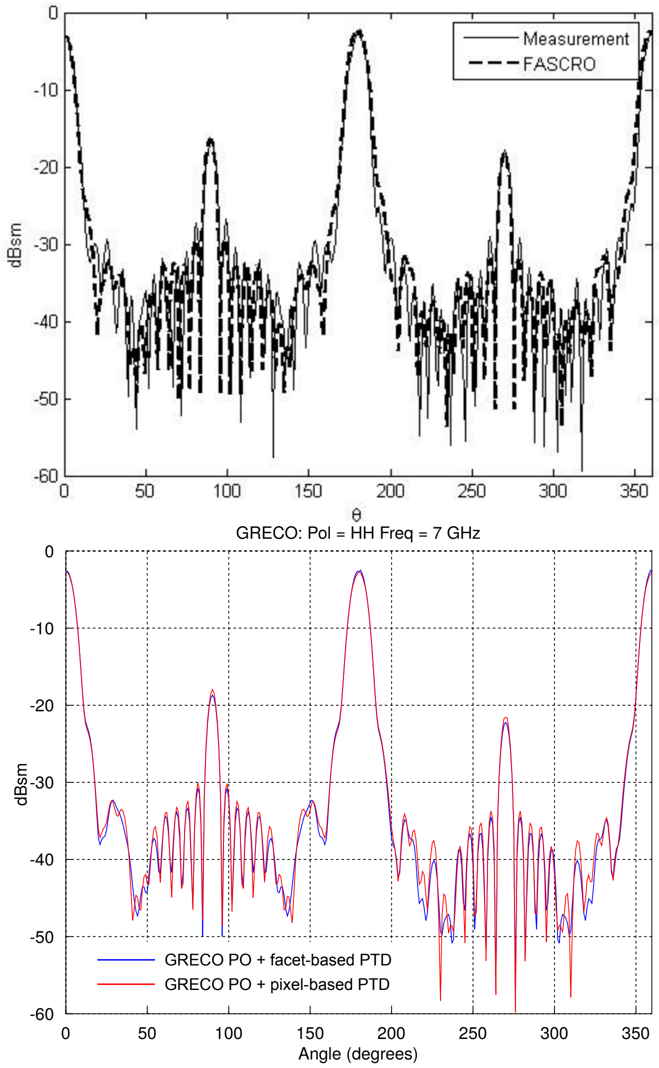

Fig. 5. Monostatic RCS results for the prism defined in Fig. 4 for HH polarization, $\phi = 90$ and θ ranging from 0 to 360 with 1 step at 7 GHz.
Upper plot: Published in [9]. Measurements obtained by the Spanish "Instituto Nacional de Técnica Aeroespacial" (INTA) compared with simulations of FASCRO high-frequency (PO+PTD) code [10].
Lower plot: GRECO computation using PO+PTD. Edge diffraction is computed either with the pixel-based graphical processing approach presented in sec. III (red line) or with the hybrid facet-based approach of sec. II-C (blue line).

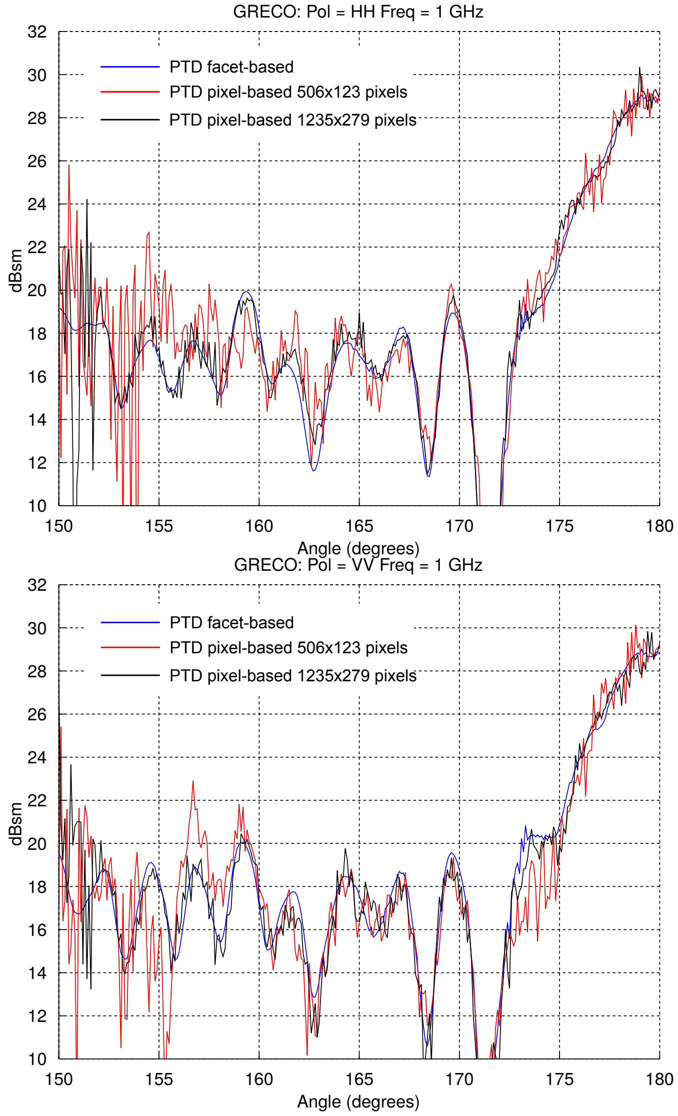

Fig. 6. GRECO PO+PTD results for a stealth aircraft home-made model. PTD pixel-based algorithm presented in in sec. III (red and black lines) is compared with the new hybrid facet-based approach (blue line) in which graphical processing is used only for detecting shadowed edges. The pixel based result for a small 506x123 pixels bitmap is contaminated with discretization noise. However, the result corresponding to a larger 1235x279 pixels bitmap agrees much better with the new hybrid facet-based approach.

References

[1] J. Rius, M. Ferrandc, and L. Jofre, "Greco: Graphical electromagnetic computing for rcs prediction in real time," *IEEE Antennas and Propagation Magazine*, vol. 35, no. 2, pp. 7–17, 1993, cited By 102.

[2] J. Rius, M. Ferrando, and L. Jofre, "High-frequency rcs of complex radar targets in real-time," *Antennas and Propagation, IEEE Transactions on*, vol. 41, no. 9, pp. 1308–1319, 1993.

[3] E. Knott, J. Shaeffer, and M. Tuley, *Radar Cross Section*. Artech House, 1993.

[4] J. Rius, D. Burgos, and A. Cardama, "Discretization errors in the graphical computation of the physical optics surface integral," *Applied Computational Electromagnetics Society (ACES) Journal*, vol. 13, no. 3, pp. 255–263, 1998.

[5] J. Rius, A. Carbo, J. Bjerkemo, E. Ubeda, A. Heldring, J. Mallorqui, and A. Broquetas, "New graphical processing technique for fast shadowing computation in po surface integral," *Antennas and Propagation, IEEE Transactions on*, vol. 62, no. 5, pp. 2587 – 2595, 2014.

[6] W. Gordon, "Far-field approximations to the kirchoff-helmholtz representations of scattered fields," *Antennas and Propagation, IEEE Transactions on*, vol. 23, no. 4, pp. 590–592, 1975.

[7] B. T. Phong, "Illumination for computer generated pictures," *Commun. ACM*, vol. 18, no. 6, pp. 311–317, Jun. 1975.

[8] J. Asvestas, "The physical-optics integral and computer graphics," *Antennas and Propagation, IEEE Transactions on*, vol. 43, no. 12, pp. 1459–1460, 1995.

[9] D. Escot-Bocanegra, D. Poyatos-Martnez, R. Fernandez-Recio, A. Jurado-Lucena, and I. Montiel-Sanchez, "New benchmark radar targets for scattering analysis and electromagnetic software validation," *Progress In Electromagnetics Research*, vol. 88, pp. 39–52, 2008.

[10] M. Domingo, F. Rivas, J. Perez, R. Torres, and M. Catedra, "Computation of the rcs of complex bodies modeled using nurbs surfaces," *Antennas and Propagation Magazine, IEEE*, vol. 37, no. 6, pp. 36–47, Dec 1995.

[11] "Newfasant electromagnetic simulation software." [Online]. Available: www.fasant.com