COMPUTATION OF CROSS SECTION OF COMPLEX BODIES IN ESA DRAMA TOOL

Noelia Sanchez-Ortiz[(1)], Raúl Dominguez-Gonzalez[1], Joaquim Correira de Oliveira[(1)], Johannes Gelhaus[(2)], Christopher Kebschull[(2)] Holger Krag[(3)]

[(1)] DEIMOS Space S.L.U., Ronda de Poniente 19, 2°2, Tres Cantos, Madrid, 28760 Spain,
Email:noelia.sanchez@deimos-space.com, joaquim.oliveira@deimos-space.com
[(2)] Technische Universität Braunschweig, Institute of Aerospace Systems, Hermann-Blenk Str. 23, 38108 Braunschweig, Germany, Email: j.gelhaus@tu-braunschweig.de
(3) ESA/ESOC (Space Debris Office), Robert-Bosch-Str.5, 64293 Darmstadt, Germany, Email: Holger.Krag@esa.int

ABSTRACT

In order to enable ESA space programs to assess their compliance with ESA's requirements on space debris mitigation, the DRAMA (Debris Risk Assessment and Mitigation Analysis) software is made available by ESA. The updated DRAMA is composed of six individual software applications, addressing different aspects of debris mitigation (see [1] for more details).

This paper focuses on the new CROC (CROss Section of Complex Bodies) module. It allows the computation of projected cross-sectional areas for user-defined bodies. The computation of the projected cross-section of the complex geometry needs to address two different issues: the cross section of every primitive and the shadowing effects among them, which does not allow to simple sum up every singular contribution.

1 INTRODUCTION TO CROC

The cross section of a satellite seen under a defined aspect angle defines the area of the satellite as seen from the mentioned direction. This area defines the surface perturbations acting on the satellite, like drag, solar radiation pressure (SRP), albedo, etc. (and also the relevant cross-section for collision risk analyses). Any of those perturbations are normally related to a different aspect angle with respect to the satellite (defined in body-fixed reference frame) and then, the projected area is different for the computation of the perturbation forces. For example, drag perturbation depends on the area projected into the along-track direction (velocity), whereas the albedo force is dependent on the cross section projected into the nadir direction. On the contrary, SRP depends on the area as seen from Sun.

For an orbiting satellite, these mentioned three directions may be associated to a very different cross-section. Only in case of a pure spherical body, the projection in these three directions will have the same value (πR^2, R being the radius of the sphere). For any other geometry, different cross-sections will be obtained in each of these three directions.

On CROC, the computation of cross sections from a point of view is made with the whole composition of elements (being those, spheres, spheres cap, boxes, cylinders and cones). This computation is made using projections determined using the given point of view direction and the Z-Buffer Algorithm for visible surface determination.

The Z-Buffer Algorithm is one of the simplest visible-surface algorithms to implement and use. No pre-sorting is necessary and no object-object comparisons are required.

Drawing the whole composition of elements using OpenGL, allowed us the use the above methods to project and determine visible surfaces. In fact, accessing directly the OpenGL Frame Buffer, it allowed us to determine the cross section at a given point of view.

With these premises, we have developed a Java Graphic Library for Java OpenGL that implemented drawing the elements: Sphere, Sphere Cap, Cylinder, Cone and Box on a 3D Coordinates Space; the use of a illumination for more realistic view, cross section calculation, an Intuitive UI (User Interface) for interacting with the model being drawn on DRAMA, and a standalone module for CROC usage outside DRAMA.

2 CROC FUNCTIONALITIES

Three different functionalities are available at CROC module:

- F1. User defined Aspect Angle
- F2. User defined Aspect Angle and Rotation Axis
- F3. Randomly Tumbling Satellite

Projecting the model with a given aspect angle - the point of view – and determining the visible surface are actions that can be made several times to cover the other

Proc. '6th European Conference on Space Debris'
Darmstadt, Germany, 22–25 April 2013 (ESA SP-723, August 2013)

two functionalities.

2.1 F1. User Defined Aspect Angle

The projected view of the drawn model is achieved by using the User defined Aspect Angle. This parameter is actually the set of angles θ and φ that represents the direction from which the projection should be made or simply the point of view.

OpenGL uses this set of angles to determine the projection, calculating the parameters for the camera transformations. OpenGL is also in charge of maintaining the objects visibility according to the objects depth.

2.2 F2. User Defined Aspect Angle and Rotation Axis

This functionality is actually a set of calls made to the F1 functionality. These calls are made using a set of pre-computed view points achieved using the functionality parameters.

For each run, these parameters are computed using the User Defined Aspect Angle and Rotation Axis. The view directions were defined as evenly distributed directions around the rotation axis, with an angle equal to the User Defined Aspect Angle. Using an iterative set of calls to the F1 functionality allowed us to implement this functionality.

2.3 F3. Randomly Tumbling Satellite

This functionality is also a set of calls made to the F1 functionality. These calls are made using a set of pre-computed view points. The used set of points is defined in the basis of force equilibrium for points over a sphere.

These points are already pre-computed, as opposed to the F2 functionality where they are computed each run. These points are stored in the *F3.ViewPoints.inp* file. Although the file can be changed, it is suggested that the original file is used.

3 ALGORITHM IMPLEMENTATION

The basic algorithm implemented within CROC is based on the Z-buffer computation [2]. This is based on an array of $n \times n$ pixels that holds the visible pixels using the z value of the corresponding element.

Given a list a polygons, an array of intensity and an array of buffer (initialized with -∞), for each polygon P and for each pixel(x,y) that intersects P, the z-depth of P at pixel(x,y) is calculated. Z depth represents the distance to the Z plane of the element. The result of the application of this algorithm can be represented in the example of Figure 1.

The accuracy of the computed cross section depends on the window size it is fixed to 2560x1920 to ensure accurate results.

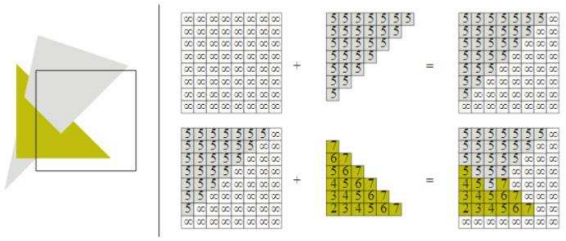


Figure 1. z-Buffer Algorithm representation

In the case of F1, a unique run of the former defined algorithm is executed in the basis of the defined aspect angles.

For F2, we can encounter two situations:

- R (rotation axis) and V (view direction) are not parallel, then the cross section can be computed as the average of the areas for a set of cases defined as steps on the rotation of the satellite around the rotation axis.
- R and V are parallel, then the cross section is computed in a unique step, since the projected area is the same for any rotation step of the satellite around the axis.

This first case can be based on a number of calls to the algorithm defined for the basic computation of cross section for an aspect angle.

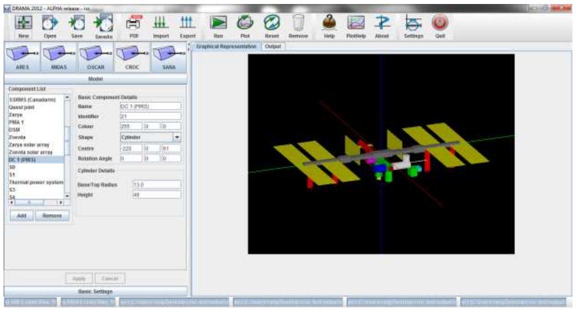
In the case of F3, the algorithm implemented for the computation of the average cross section of tumbling satellites is also based on the usage of a number of predefined points of view, as per the previous functionality, and then applying the computation defined in the basic case.

4 USER INTERFACE CAPABILITIES

CROC User Interface (UI) adds new UI capabilities to the existing DRAMA application for building and navigating through a drawn model. The building of models can be achieved using an ASCII files, allowing the user to use user-customized scripts to generate data from third-party sources and translate the model data to this application. All changes to the model can be observed in real time in the Graphical Representation tab of the CROC module.

The user interface allows defining all the characteristics of every primitive forming the defined satellite.

The representation of the resulting spacecraft is provided in a graphical output that can be zoomed, rotated and panned using mouse gestures.



Component List

SSRMS (Canadarm)
Quest joint
Zarya
PMA 1
DSM
Zvezda
Zarya solar array
Zvezda solar array
DC 1 (PIRS)
S0
S1
Thermal power system
S3
S4

Basic Component Details

Name	DC 1 (PIRS)		
Identifier	21		
Colour	255	0	0
Shape	Cylinder		
Centre	-220	0	61
Rotation Angle	0	0	0

Cylinder Details

Base/Top Radius	13.0
Height	49

Figure 2. Components Attributes definition with graphical representation of ISS

5 EXAMPLES OF USE

5.1 Simple Satellite

In this section we will show how a simple satellite can be built with a small set of elements. In particular, we have created a satellite, with the main body, solar panels, an antenna and a camera with eight primitives. The configuration of the satellite is defined as:

Table 1. Elements of a simple satellite

Element	Type Type	Dimensions (cm)	Position (XYZ, cm)	Rotation Angles (deg)
Main Body	Box	50x50x50	0,0,0	0,0,0
Solar Panel 1	Box	50x100x5	0,100,0	0,30,0
Solar Panel 2	Box	50x100x5	0,-100,0	0,30,0
Auxiliar Piece 1	Cylinder	5x25	0,25,0	-90,0,0
Auxiliar Piece 2	Cylinder	5x25	0,-25,0	-90,0,0
Antenna	Sphere Cap	40x50	50,20,80	0,-90,0
Auxiliar Piece 3	Cylinder	5x20	20,20,25	0,0,0
Camera	Cone	20x20	0,0,-30	0,0,0

Figure 4 provides the view of the satellite from different points of view. As it is easily observed from the plots, the projected area is very different. When observing the satellite from a view direction defined by aspect angles (45 deg, 15 deg), the projected area is 1.6234m^2. If we observe the satellite from X axis, aspect angles (0 deg, 0 deg), the area is 1.64311m^2. For the case of a top view (from Z axis), the observed area is much smaller (0.5387m^2).

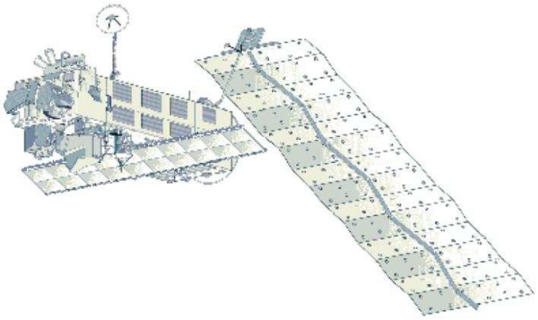


Figure 3. ENVISAT inflight configuration

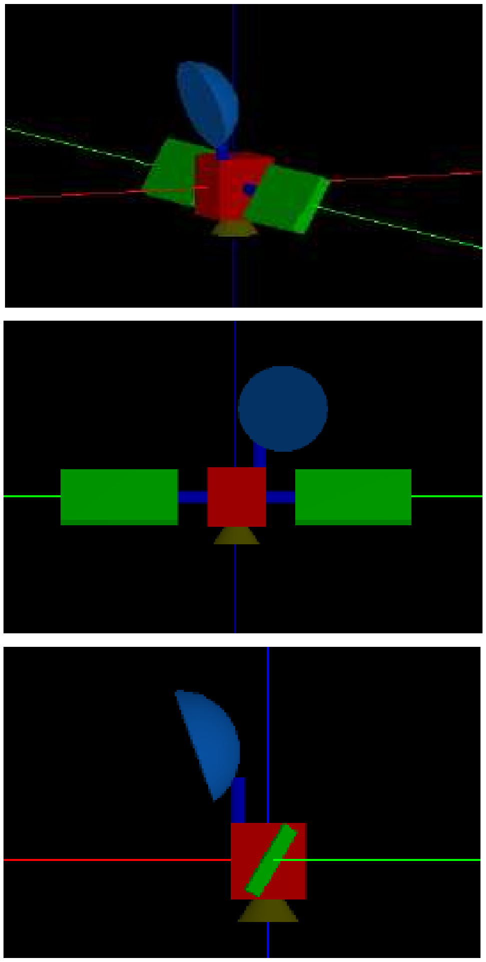


Figure 4. Simple Satellite from different Aspect Angles

5.2 ENVISAT case

In order to show the capabilities of the tool for assessing the cross section of a complex satellite, a model of ENVISAT spacecraft is created.

Envisat is an Earth-observation satellite which was operated by ESA. It was launched in 2002 with an estimated lifetime of 5 years. The mission was extended and the satellite was operative until 2012. In 2012, the ground segment lost contact with the satellite, and all attempts to recover contact failed. It was operating in a Sun synchronous polar orbit (768 km altitude, 98:4deg inclination).

The configuration of a satellite as ENVISAT is quite complex as it can be observed in Figure 3 defining the real configuration of the satellite. Figure 5 shows the model created with CROC.

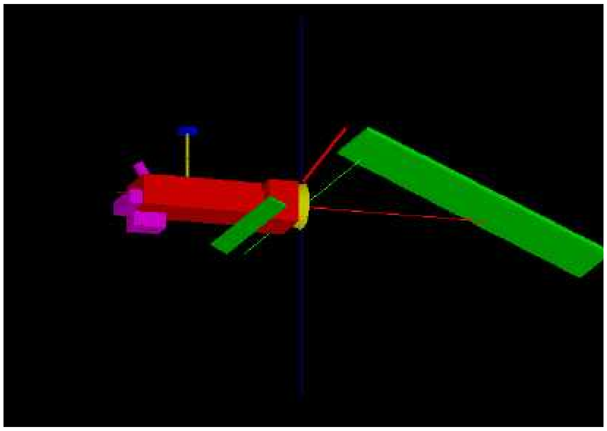


Figure 5. ENVISAT model created with CROC

A model comprising the most prominent components of the Envisat satellite is defined in table 2.

Figure 6 shows the results from functionality F2. Cross sections seen from a circle contained in the XY plane are shown. In addition to this, pictures from some representative viewpoints are shown.

Table 3 provides the results from random tumbling functionality, compared with the cross section values stored in the DISCOS database [4]. ESA's DISCOS Database (Database and Information System Characterising Objects in Space) enables registered users to retrieve information on a variety of data associated with space activities. The results are quite similar. However, it should be noticed that the cross sections from DISCOS were obtained from quite a simple model.

Table 2. Elements of ENVISAT model

Element	Type Type	Dimensions (cm)	Position (XYZ, cm)	Rotation Angles (deg)
Main Body	Box	635.60x278.30x159.80	-582, 0, -10	0,0,0
Body Joint	Box	35.0 x122.00x119.00	-243, 0, -10	0,0,0
Hollow box	Box	162.00x278.00x212.00	-144, 0, -11	0,0,0
Pole	Box	381.00x16.00x16.00	116, 0, 283	0,-53,0
Big panel	Box	1420.00x498.00x23.00	923, 0, 97	0,23,0
Antenna pole	Cylinder	7.00x238.00	-662, -132, 70	0,0,0
Dish	Cylinder	52.00x24.00	-662, -132, 300	0,0,0
Cone	Cone	119.00x396.00	0, 0, 0,	0,270,0
Front cylinder	Cylinder	29.00x118.00	-827, 0, 58	0,-32,0
ASAR antenna	Box	100.00x1000.00x10.00	-290, 0, -120	0,25, 0
Nose assembly	Box	70.00x278.00x73.00	-937, 0, -52	0,0,0
Nadir assembly	Box	174.00x139.00x82.00	-819, 70, -128	0,0,0

Table 3. ENVISAT cross sections in DISCOS database and obtained with CROC (F3)

Cross section	DISCOS	CROC
Minimum (m^2)	20.25	22.023
Maximum (m^2)	99.034	106.402
Average (m^2)	63.047	65.062

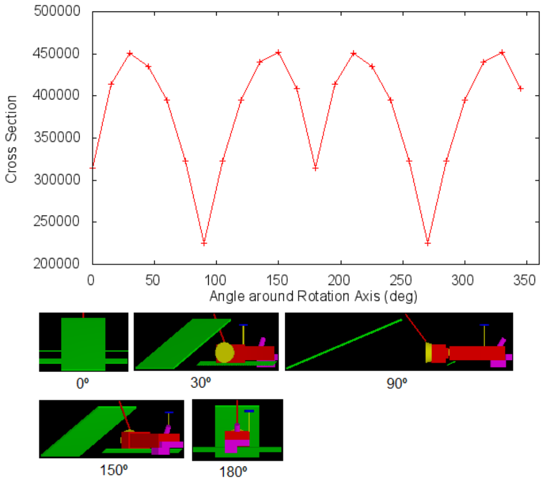


Figure 6: Envisat cross section (cm^2) from viewpoints around its Z-axis. Pictures below correspond to the graph extremes

Figure 7 provides the cross sections as a function of azimuth and elevation as obtained with the random tumbling functionality.

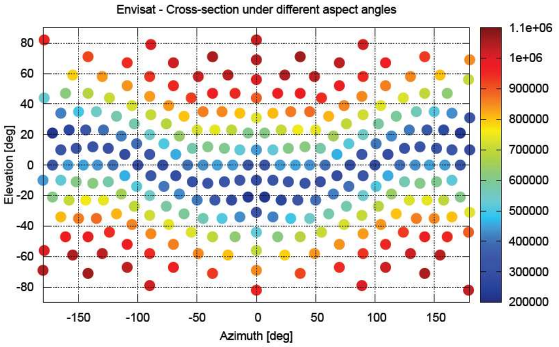


Figure 7. ENVISAT cross sections from different aspect angles (cm^2)

5.3 ISS case

The International Space Station is a particularly interesting test case. The space station comprises several elements, whose positions have changed over time. The most prominent contribution to the cross section is due to the solar arrays and radiators. As these elements are not in a fixed position, the ISS cross section will be largely dependent on their orientations.

For this document, the results of a simple ISS model are provided. The geometric specifications were obtained from [3] (Chapter 6, Table 6.1), upgrading the information regarding several components. Figure 8 shows the result (including the reference system used in the model). It should be noticed that solar arrays and radiators (yellow and red panels) might rotate.

[3] proposes a computation of the equivalent spherical cross section by finding a sphere in which the cuboid defined by the largest dimensions is inscribed. With that method, an equivalent radius of 68.5 m is obtained. Figure 9 shows the relative size of such cuboid and the ISS model.

With the particular configuration of solar panels and radiators used for this paper, CROC provides a 2481.324 m^2 cross section, which corresponds to a sphere of radius 28.1 m. Figure 9 provides a comparison of the relative sizes of the ISS model and a sphere of radius 28.1 m. In addition to this, the cuboid defined from the largest dimensions is also provided.

From that figure, the following should be noticed:

- The sphere defined by the average cross section does not cover the ISS model. Particularly, its largest elements are left outside the sphere.
- The sphere matches the average cross section seen from any direction. In the case of the ISS, however, the cross section seen from the flight direction is more relevant, as most close encounters happen from that direction. In addition to this, the cross section from that direction is smaller than the total average. In the worst case, only one of the four truss solar panels contributes to the total cross section (see Figure 10)

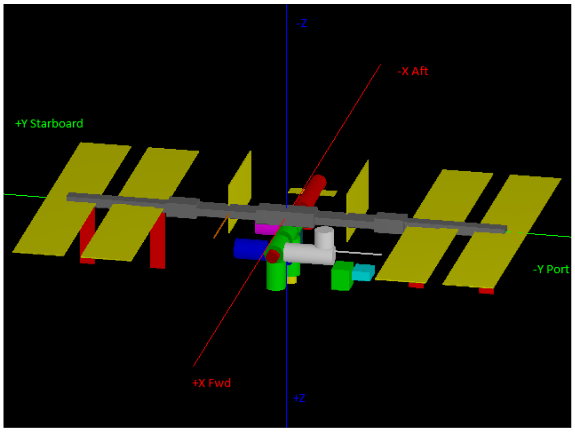


Figure 8: ISS model created with CROC

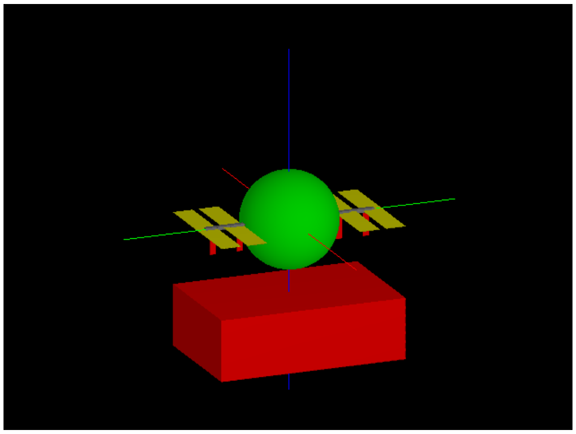


Figure 9: ISS model with the average cross section sphere computed by CROC. The cuboid below is provided for comparison

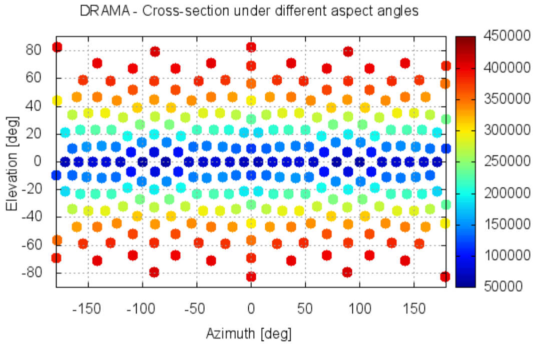


Figure 10: ISS cross sections from different aspect angles (dm^2)

6 CONCLUSIONS AND ACKNOWLEDGEMENT

CROC has been developed to allow the computation of projected cross-sectional areas for satellites defined by the user. This cross section computation can support the evaluation of satellite re-entry time, by means of DRAMA OSCAR tool, or collision risk computation and mean number of avoidance manoeuvres by means of the DRAMA ARES tool.

The examples provided in this paper show the flexibility of the tool and the capacity of simulating complex satellites with good computation accuracy.

The work presented in this paper is done within ESA/ESOC Contract No. 4000104977/11/D/SR. The ESA project is carried out by Institute of Aerospace Systems (ILR) /Technische Universit at Braunschweig (TUBS) and DEIMOS Space S.L.U.

7 REFERENCES

1. Gelhaus, J. et al. Upgrade of ESA's Space Debris Mitigation Analysis Tool Suite, Final Report.
2. Foley, van Dam, Feiner, Hughes, Phillips, Introduction to Computer Graphics, Addison-Wesley Publishing Company Inc, MA, US, 1996, ISBN 0-201-60921-5
3. F. Kenneth Chan. Spacecraft collision probability. The Aerospace Press. American Institute of Aeronautics and Astronautics, Inc. 2008, ISBN 978-1-8844989-18-6
4. DISCOS database; http://mas3.esoc.esa.de:9000