Technical Description of Color and Opacity Blending (COB) Debris Visualization Technique

David Massa and Timothy Carrico
Analytical Graphics, Inc.

ABSTRACT

Communicating the potential impact of orbital debris on satellites and payloads presents many unique analysis and visualization challenges. Both the numerical simulations and tracked datasets associated with these events can be large creating unique challenges for analysts who rely on computer generated imagery to enhance general cognition and rapidly draw conclusions.

This paper identifies a technique for displaying large space debris datasets using a unique combination of color and opacity blending through a common OpenGL graphics system and commercially available off-the-shelf software engine. This innovative approach to interactive 3D space debris visualization provides analysts with the enhanced ability to understand the relative densities of a numerical particle simulation that change over time.

BACKGROUND

NORAD is regularly tracking over 12,000 space objects. Of those objects regularly tracked, approximately 3,000 are considered payloads and approximately a third of the payloads are considered "active" assets important to global governments and economies. In order to better understand the effects of debris on these important assets, software systems have been developed to identify potential collisions [1].

These collision evaluations incorporate debris already in orbit, as well as debris from future orbital event simulations. In the case of existing orbital debris, the evaluation of collisions using track data only represents a subset of possible conjunctions as the data is limited to space objects greater than 10cm in diameter [2]. Consequently, in order to improve upon risk assessment of future debris events, numerical debris cloud simulations can be used to augment tracked datasets.

Typically, data of this nature is visualized through computer generated imagery (CGI) to facilitate general cognition. While some visualization methods rely on volume-rendering techniques that parcel 3D space into small volumes (voxels) [3], the computational overhead associated with this technique makes real-time visualization of time changing data impractical.

Other visualization techniques use 3D geometry or opaque markers to identify the location of each individual sample (Fig. 1).

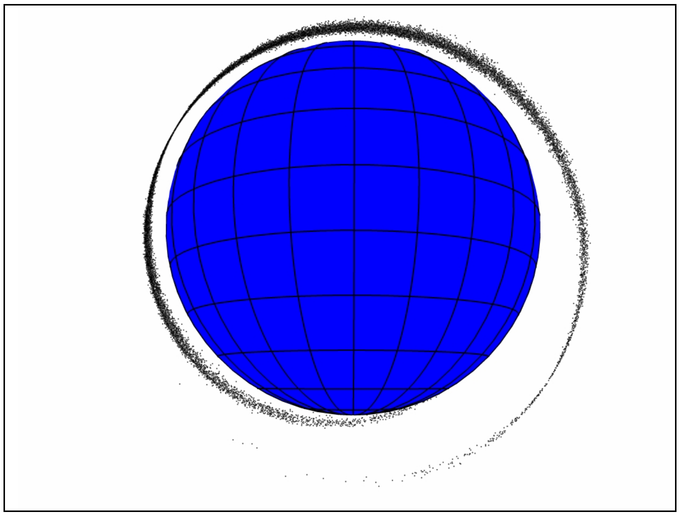

Figure 1: Location of Samples using Discreet Objects

While the usage of discreet objects depicted in Figure 1 can be an effective means of communicating sample location and quantity, it can inaccurately illustrate the relative degree of risk. For instance, a 2 pixel debris marker in a 200mi LEO orbit displayed on a standard 1024x768 pixel display using a typical field of view (i.e. 45 deg) equates to a 21.5 mile width (Fig. 2).

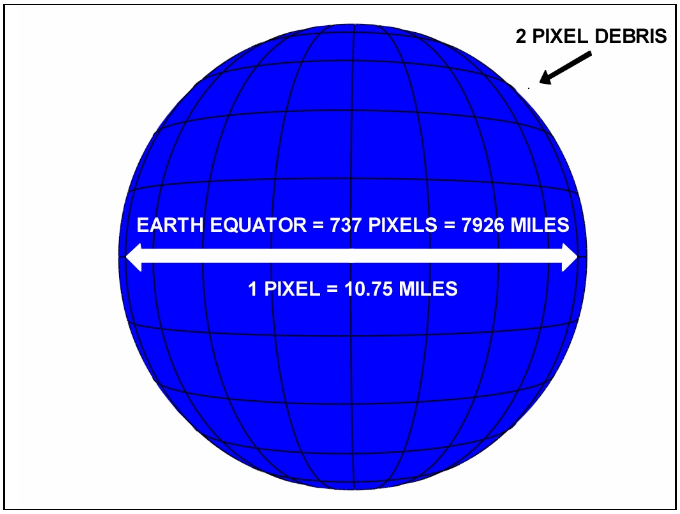

Figure 2: Location of Samples using Discreet Objects

If the 2 pixel marker in Figure 2 represented an actual 4cm piece of debris, the display would be over 865,000 times out of scale, thus exacerbating the degree of risk.

Other data visualization methods have sought to use distance-based volumetric rendering that relies on OpenGL graphics capabilities and pre-processing of the entire dataset [4]. While this volume-rendering technique can produce faster results than the voxel technique previously described, the need to pre-process data produces unnecessary computational overhead. Additionally, the volumetric blending technique used limits the range of color and opacity available.

In order to address the many visualization challenges associated with interactive display of large orbital debris datasets, a color and opacity blending technique was used in an OpenGL environment. This technique provides a means to visualize simulated debris clouds relative to payload trajectories which produces sufficient insight to the payloads requiring higher-fidelity analysis.

BREAK-UP MODEL DATA

In order to achieve the high number of debris pieces required for density visualization, a break-up modeling environment was set up in Satellite Tool Kit (STK) using the spacecraft maneuvering tools (Fig. 3).

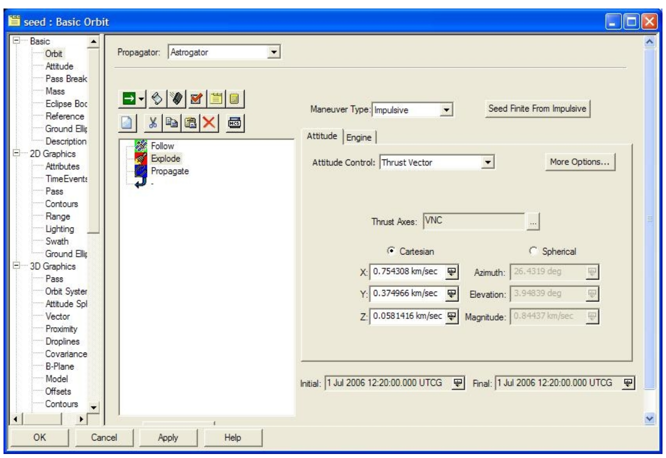

Figure 3: Debris Modeling Environment in STK

These tools allow the user to create and impart a force on an orbiting body, satellite or piece of debris and then propagate until re-entry or for the desired period of time. With this method, the break-up model was able to model thousands of debris trajectories in a numerically integrated full force model by automating the propagation of one debris trajectory at a time. The speed of modern computers and parallel techniques make this a suitable, expedient method.

The debris propagation inputs were developed from user selected distributions of mass, ΔV, material density, and area. Later, empirical data (TLE) analysis from a recent Chinese anti-satellite (ASAT) test against an aging weather satellite (Fenygun 1C) provided statistical information for the break up model, adding to the fidelity of the generated data. The output of the model was ephemeris data on each piece of debris which were subsequently loaded into objects in the STK scenario. Model runs from 20 to 50k pieces (Fig. 4) provided satisfactory density for the visualization technique.

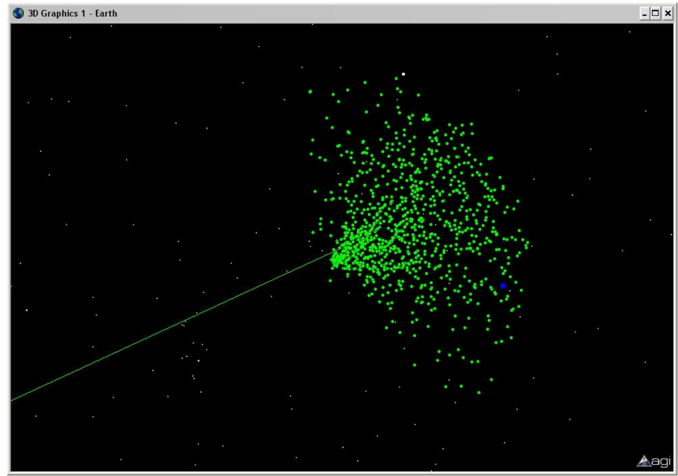

Figure 4: Generated Debris, Normal Distribution of ΔV

DEBRIS VISUALIZATION TECHNIQUE

In order to minimize the problems associated with displaying orbital debris as discreet objects, a Color and Opacity Blending (COB) visualization technique was developed. The COB technique is built around STK, an existing real-time analysis and visualization system, and utilizes OpenGL-based graphics to promote user interactivity while displaying large 3D datasets.

MULTI-TRACK OBJECT

In order to utilize the COB technique, the ephemeris data for each debris sample in the break-up model's data cloud is assigned as a data track within a Multi-Track Object (MTO). An MTO is a container object used by STK to display large time-varying datasets in an efficient manner (Fig. 5).

Begin Track
(Unique ID for Data Track)
(Interpolation order between data points)
(Removal of standard propagator)
Begin Points
Epoch Seconds and Central Body Fixed Coor.
End Points
End Track

Figure 5: MTO Data Track Structure

With the break-up model's ephemeredes loaded into an MTO, a texture map was assigned to each debris object's data track. While 3D polygonal geometry could have been used to represent the location of a debris sample, a texture map was used to take advantage of STK's processes for attitude and sizing of 2D markers within a 3D environment.

DEBRIS ATTITUDE DEFINITION

The break-up model used to produce the data cloud only reported ephemeris data. Without time-varying attitude data, the resulting debris orientation is fixed to a coordinate system. Though the default coordinate system for a “3D object” representation in STK is independent of the viewing angle, a “marker” representation type has a default coordinate system that is locked to the viewing vector (Fig. 6).

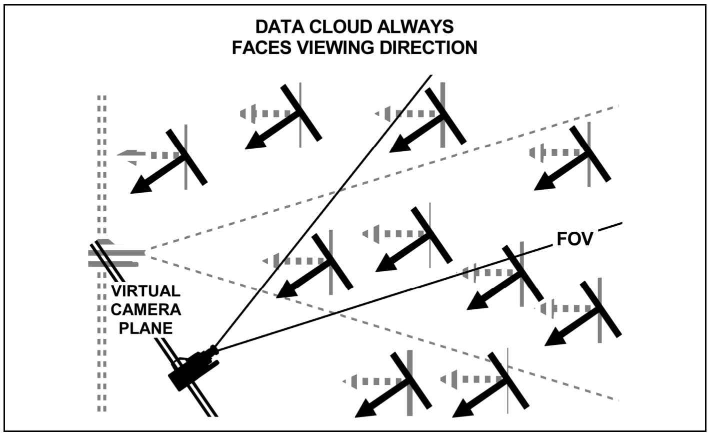

Figure 6: Viewer fixed coordinate system representation

Without a viewer fixed coordinate system, flat 2D debris representations would effectively disappear if the viewing angle became perpendicular to the object.

DEBRIS SIZING

While geometric information such as mass, density, and area formed the basis for the debris calculations, the resulting data cloud is simply a series of point-mass locations that change over time. As such, the debris objects had to be given a visual representation for size.

Within STK, the default characteristic for 3D geometry associated with a “3D object” is to change in size relative to the position of the virtual camera. As such, objects closer to the virtual camera for the 3D display will appear larger than those objects that are further away (Fig. 7).

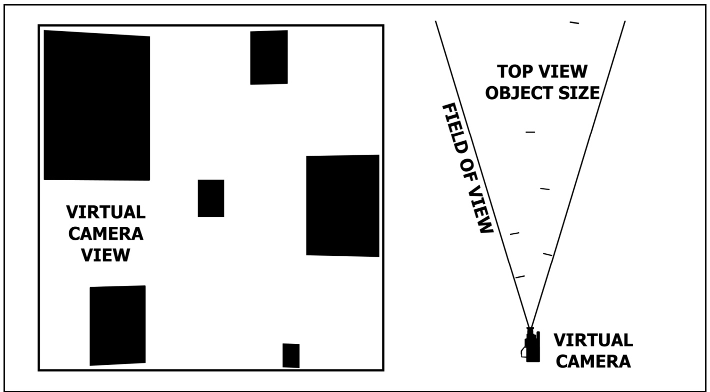

Figure 7: Default STK 3D object sizing

The default characteristic for sizing STK “markers” differs than “3D objects.” For markers, the visualization display size is fixed regardless of the distance from the object to the virtual camera (Fig. 8).

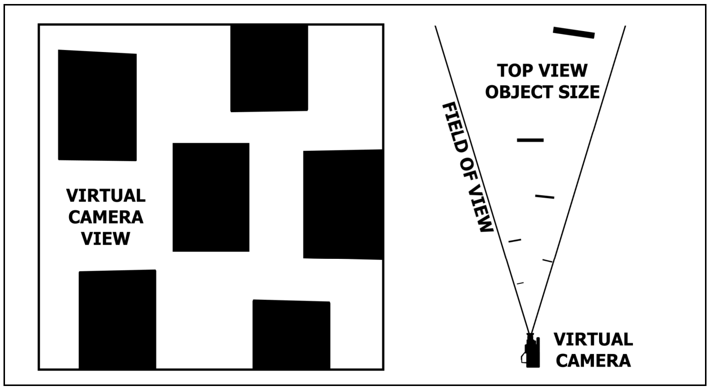

Figure 8: Default object sizing for STK markers

Beyond selecting a sizing technique, the dimensions of the debris representation must be defined with knowledge of the limitations of display devices. If we use the example of a 200mi circular Low Eart Orbit (LEO) at the equator with a standard 1024x768 pixel display, a single pixel represents 8.13mi.

$$\textit{Pixel dimension (P)} = \frac{\textit{Circular Orbit Display Width (D)}}{\textit{Resolution (R)}}$$

$$\textit{D = 2 x Semimajor Axis (a)}$$
$$\textit{a = satellite altitude + Earth equatorial radius}$$

$$a = 200mi + 3963mi = 4163mi$$
$$D = 2 \times 4163mi = 8326mi$$
$$P = 8326mi / 1024pixels = 8.13\ mi/pixel$$

Because most display devices can not produce graphics smaller than one pixel, a single piece of debris shown as part of the whole orbit previously described will be at least 8.13 mi/pixel. As such, in order to show a 4cm piece of debris to scale in the above example, you would need over 327,000 display devices stacked side-by-side.

$$P = 4cm/pixel = 0.00002485\ mi/pixel$$
$$R = D / P = 8326mi / 0.00002485\ mi/pixel$$
$$R = 335{,}050{,}302\ pixels$$

$$\textit{\# of displays (n) = R / individual display resolution (r)}$$
$$n = 335{,}050{,}302\ pixels / 1024\ pixels = 327{,}198$$

DEBRIS COLOR

With the location, attitude, and size of the samples in the debris cloud defined, the final step in characterizing the debris objects is to give them color (C) and opacity (α). Color in a typical computer graphics display is assigned with 24 bits of data that is equally divided into red, green, and blue 8-bit channels (C_R, C_G, C_B). Each 8-bit color channel has 256 available divisions designated as a color value between 0 and 255. Opacity in a typical computer graphics display is assigned 8 bits of data with a corresponding range of transparency between 0-255.

Begin Track
(Toggle display of track on or off)
(Define distance for marker, label, and point display)
(Disable use of 3D model attach points)
(Define default azimuth)
(Direct Z-axis coordinate towards Nadir)
(Remove function to follow previous default azimuth)
Begin Marker
(Toggle display on or off)
(Define the object as a polygon or a texture)
(Define marker shape – not used)
IsTransparent Soft *(Transparent marker)*
OneOneBlend Yes *(COB technique)*
(Toggle rotation adjustment on or off)
(Define angle of rotation adjustment)
(Define horizontal origin location)
(Define vertical origin location)
MinSize 3000 *(Debris display size in meters)*
MaxSize 3000 *(Debris max display size in meters)*
(Disable auto sizing relative to camera distance)
SizeInMeters Yes *(toggles size - pixels to meters)*
(Define minimum distance for marker display)
(Define maximum distance for marker display)
ImageFile DebrisMarker.tga *(texture for debris)*
(Disable rotation of marker relative to North)
End Marker
End Track

Figure 9: Fully characterized debris MTO file structure

COB VISUALIZATION

In order to visualize a fully characterized debris object (Fig. 9), the process of "rendering" must take place whereby a 3D environment is converted to pixels as part of 2D CGI for display. The resulting appearance of each pixel is dependant upon the characteristics of the virtual camera, the debris cloud, and other objects in the 3D environment (i.e. stars, Earth, and other satellites).

Virtual Camera Viewing Angle

The virtual camera is the unseen viewpoint for the viewing angle of a 3D display environment. Depending on the distance, orientation, and field of view of the virtual camera relative to the debris cloud, different debris objects will be in view (Fig. 10).

COB Pixel Pipeline

Once the virtual camera's location is defined, the rendering system must determine how each pixel with the virtual camera's field of view should be represented by evaluating the color and opacity of the debris objects residing in the pixel's space. The process of creating pixel color (PxlC) from debris objects in a 3D environment is referred to as the pixel pipeline.

The first step in determining PxlC is to start with the background color (i.e. black space) for the 3D environment (C_R, C_G, C_B) and store this color into the graphics system's frame buffer (FrmC). With FrmC as a background, source objects (SrcO) that fall within the pixel pipeline are examined for C and α. The source objects include stars, debris, satellites, Earth, etc. For illustrative purposes, the pixel pipeline is shown (Fig. 11) with objects that have a color (SrcC) of pure white (255, 255, 255) and an α of 25% (64 out of 256 available divisions).

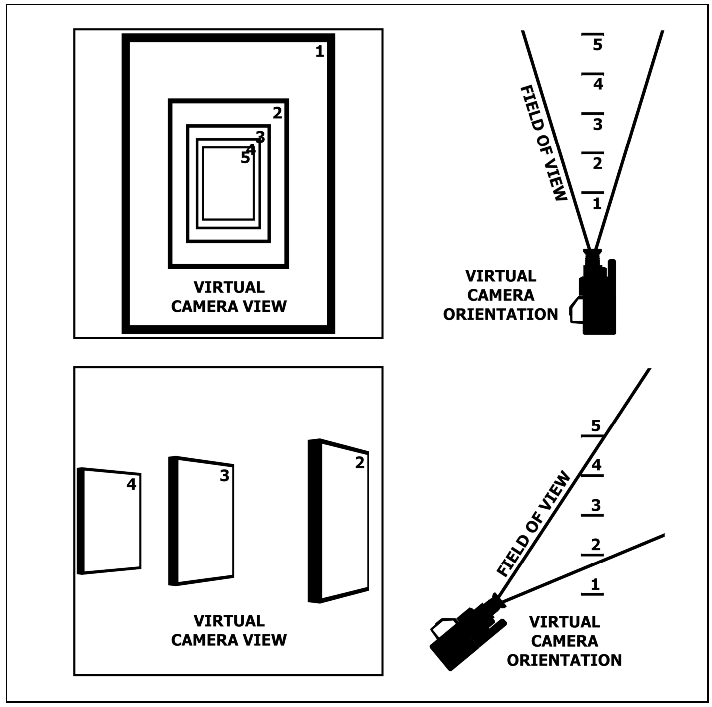

Figure 10: Virtual camera effect of viewing angle

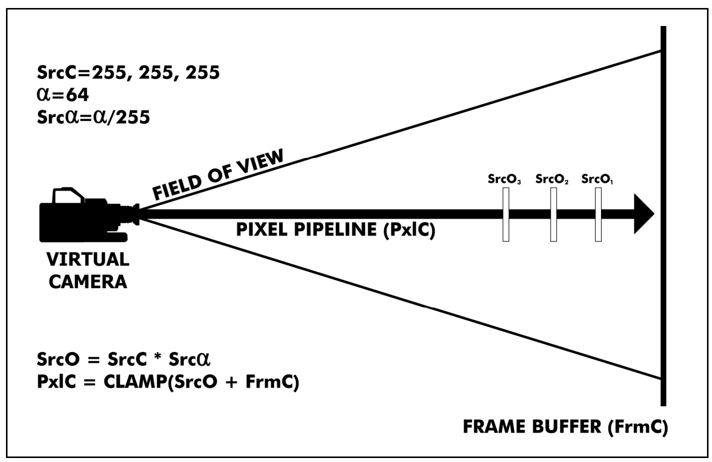

Figure 11: Pixel pipeline for debris environment

To calculate the effect of the α on each debris object's SrcC, the α 0 to 255 integer range must be normalized to a source opacity (Srcα) decimal range from 0 to 1. The resulting Srcα is multiplied by the SrcC to determine the resulting SrcO values.

$$SrcO = SrcC * Src\alpha$$

With the values for each SrcO determined, the resulting PxlC can be determined by clamping the sum of the

FrmC with the SrcOs. It is necessary to limit the resulting value to ensure that the color value does not exceed the 8 bits of available data (255, 255, 255 in the C_R, C_G, C_B) available to typical displays.

$$PxlC\ Limit(SrcO + FrmC)$$

This COB blending technique is unique in comparison to previously developed visualizations of space debris. Traditional methods of blending C and a produce gradients that only range from transparent to opaque for debris objects that possess a single color.

$$PxlC = SrcO + FrmC * (1.0 – Srca)$$

The advantage of the COB technique is to extend the range of the displayed representation for debris objects that are stacked within the camera's view. To illustrate this point, the pixel pipeline is shown (Fig. 12) with objects that have a color (SrcC) of blue (64, 64, 255 in the C_R, C_G, C_B) and an α of 25% (64 out of 256 available divisions). As different quantities of debris objects intersect the pixel pipeline, the PxlC changes from transparent to opaque blue to opaque white, giving the viewer a wider range of displayed density of debris particles.

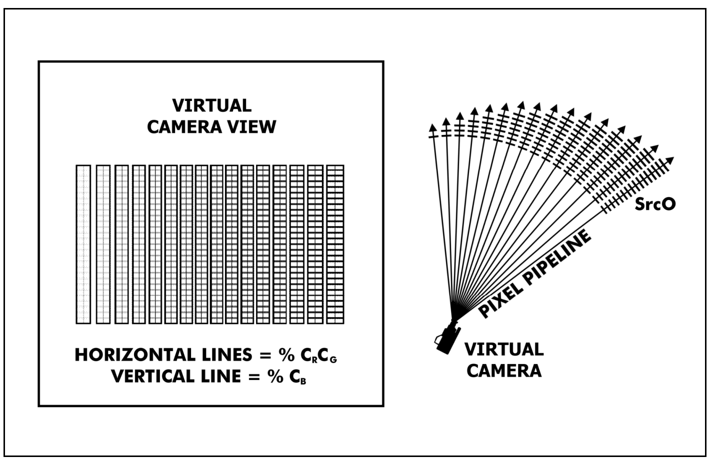

Figure 12: COB rendering technique gradients

Relative Debris Particle Densities

The result of the COB rendering method is a visualization of debris cloud densities from a specific vantage point. Fig. 13 shows a debris cloud that uses an opaque circular texture with a C_R, C_G, C_B values of 1, 3, 20 respectively. Because of the unique COB blending process, a single debris sample is nearly invisible even though no transparency values were assigned. As the debris samples increase in quantity within a viewing direction, their display representation transitions to dark blue, cyan, and onto white allowing the viewer to rapidly comprehend where the mathematical model is predicting the highest number of samples.

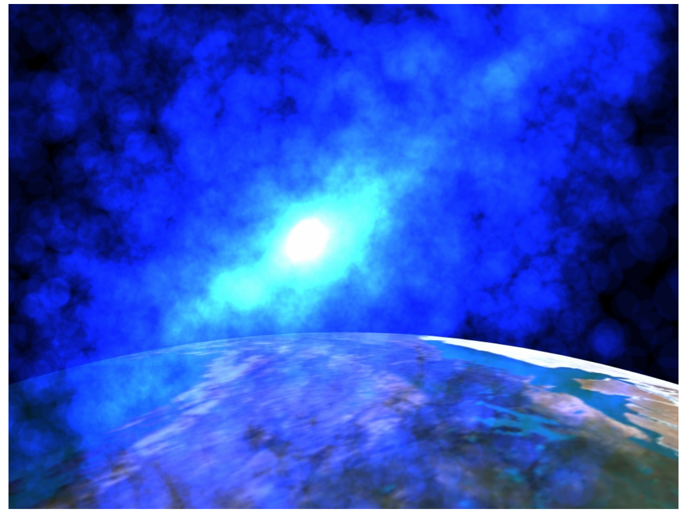

Figure 13: Result of COB cloud densities

Because the resulting visual demonstrates areas of high densities, the visual can serve as a rapid means of assessing the probability of a debris sample location versus an actual debris object location. For instance, in Fig. 14 we show an example of a break-up model that uses 10,000 samples as part of its simulation. Instead of predicting 10,000 debris pieces for this break-up, the model is predicting a Gaussian-based area in which debris particles could exist. If specific areas of the debris cloud contain more samples than others, then one could conclude that the characteristics of the breakup suggest an increased likelihood of particle existence within the areas of high number of samples. Conversely, if the break-up model produces a single debris sample on the periphery of the others, then one could conclude that a unique combination of characteristics had to align in order for a debris particle to exist in that location. Consequently, making that particle nearly invisible to the view is beneficial to understand where the highest probability of debris samples could be located.

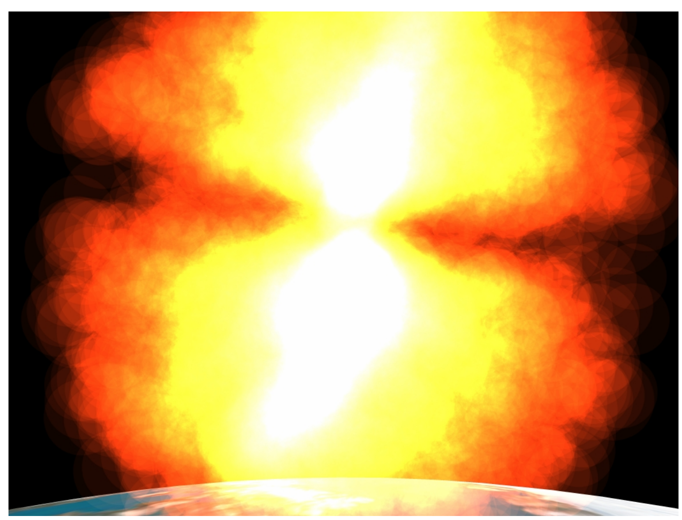

Figure 12: 10,000 sample debris cloud

Determining how gradient color areas map to a specific number of debris samples can by calculated using the C_R, C_G, C_B values used to define the visual representation for the debris sample. Using Fig. 14 as a visual reference, the debris samples were represented with C_R, C_G, C_B values of 16, 4, 1. With these C_R, C_G, C_B colors and knowledge that 255 is the maximum available value, the debris samples map to the values shown in Table 1 and the gradient shown in Fig. 15.

M = # of particles required to maximize a color channel
M = Integer (255 / C)

NOTE: If a C value = 0, then that color channel should be removed from the calculations listed below as a color channel with no value effectively does not use that color. Also, the calculations assume a black background for space. Other background colors should be added to the PxlC calculation as FrmC previously discussed.

$$\text{Red Channel } (M_R) = \text{Integer } (255/16) = 16$$
$$\text{Green Channel } (M_G) = \text{Integer } (255/4) = 64$$
$$\text{Blue Channel } (M_B) = \text{Integer } (255/1) = 255$$

$$S_0 = 1 \text{ particle characterized as } C_R, C_G, C_B$$
$$S_1 = \text{Smallest } M \text{ value}$$
$$S_2 = \text{Next largest or equal } M \text{ value}$$
$$S_3 = \text{Largest } M \text{ value}$$

NDS = Number of debris samples
Lim = Limit of value to 255

$$PxlC_{NDS1} = Lim(C_R *S1), Lim(C_G *S1), Lim(C_B *S1)$$
$$PxlC_{NDS2} = Lim(C_R *S2), Lim(C_G *S2), Lim(C_B *S2)$$
$$PxlC_{NDS3} = Lim(C_R *S3), Lim(C_G *S3), Lim(C_B *S3)$$

$$PxlC_{NDS1} = Lim(16*16), Lim(4*16), Lim(1*16)$$
$$PxlC_{NDS2} = Lim(16*64), Lim(4*64), Lim(1*64)$$
$$PxlC_{NDS3} = Lim(16*255), Lim(4*255), Lim(1*255)$$

$$PxlC_{NDS1} = 255, 64, 16$$
$$PxlC_{NDS2} = 255, 255, 64$$
$$PxlC_{NDS3} = 255, 255, 255$$

NDS	S_0	S_1	S_2	S_3
	1	16	64	255+
PxlC	16,4,1	255,64,16	255,255,64	255,255,255
Color	Black	Red	Yellow	White

Table 1: Debris samples quantities by color

Figure 15: Number of debris samples by color legend

POTENTIAL SOURCES OF ERROR

While the COB visualization technique provides a user with the unique ability to study the density of debris cloud simulation areas, it can also produce results that are misinterpreted. The potential sources for error are from the selection of a viewing angle or the visual representation of the debris cloud.

SELECTION OF VIEWING ANGLE

The viewing angle produced by the position, orientation, and field of view of the virtual camera determines how the COB technique renders the debris cloud (Fig. 10 - 12). An object that is rendered as part of the debris cloud (i.e. the Earth) due to the selection of the viewing angle can complicate the comprehension of the density, as does the selection of virtual camera characteristics.

Extraneous objects in the COB pixel pipeline

Because the COB visualization technique combines the view of all of the 3D objects that exists within the viewpoint of the virtual camera, objects that are not part of the debris cloud can affect the visual representation of the debris cloud. As depicted in Fig. 16, the debris cloud appears to be denser in the left image due to the Earth's influence on the image.

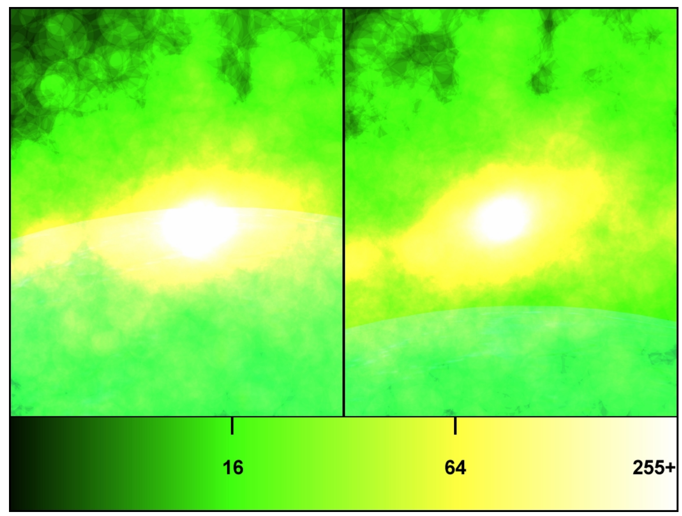

Figure 16: View Comparison with Earth Interference

Effect of virtual camera characteristics

By simply changing the orientation of the virtual camera, the debris cloud density can change in appearance. While this can be a source of error, it can also be a means of studying a specific characteristic of the debris cloud. The radial density dispersion of the debris cloud in Fig. 17 appears significantly less than the in-track density for the same debris area and times (Fig. 18).

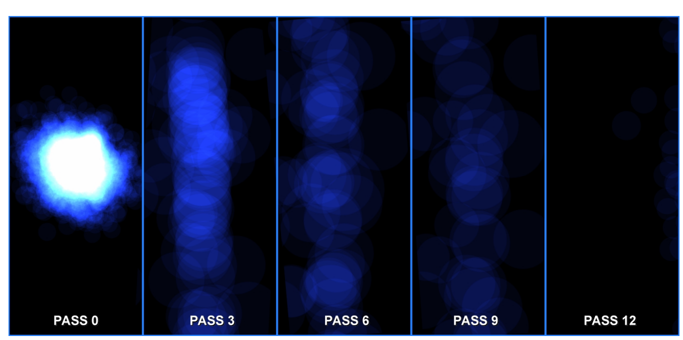

Figure 17: ECI based radial debris dispersion

Figure 18: ECI based in-track debris dispersion

VISUAL REPRESENTATION OF DEBRIS CLOUD

The technique used to visualize debris can introduce error into the analysis. The perception of black space, available color gradient, displayed number of particles, and size of the representation can all produce error.

Perception of black space

In order to enhance the available range for the particles, small C_R, C_G, C_B values are often used. Consequently, one debris sample can appear as the same color as the background. The absence of color in a COB visualization could demonstrate both a lack of density or an absence of samples. Figure 19 uses discreet objects to demonstrate the location of debris samples used for the COB rendering in Figure 17.

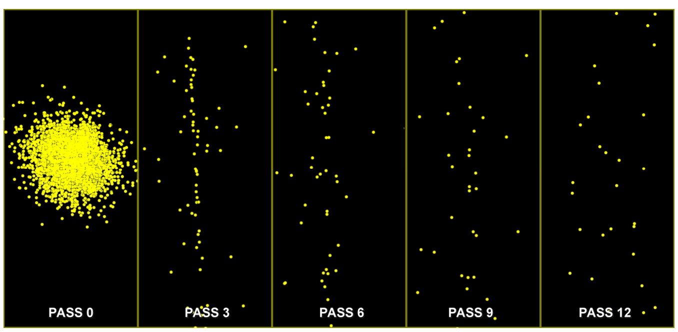

Figure 19: ECI based Radial debris dispersion with discreet markers used to represent the debris samples

Available color gradient

The C_R, C_G, C_B values used to represent an object can make comprehending the number of samples difficult to gage. Figure 18 demonstrates the appearance of a data cloud using C_R, C_G, C_B values of 48,8,16. Figure 21 uses the same debris cloud, but uses C_R, C_G, C_B values of 3,1,2. In Fig. 20, the maximum value for all color channels is reached with a display of 32 particles where the Fig. 21 gradient uses all 256 available gradient values.

Displayed number of particles

The COB visualization technique is one designed to study the density of debris samples versus the discreet location of any one particle. Consequently, smaller quantities of debris samples exported from a break-up model could incorrectly portray a lower density of points. Fig. 22 shows a comparison between two debris clouds that have identical break-up model characteristics, but one (top graphics) has 75% less samples as part of the calculation.

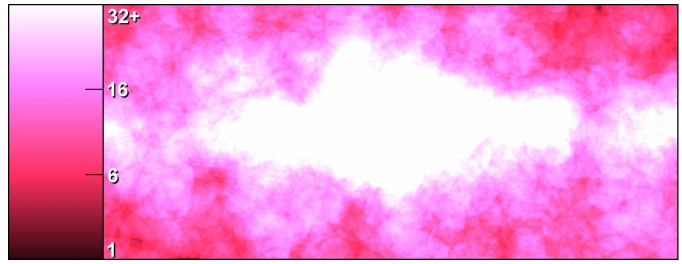

Figure 20: Data cloud with C_R, C_G, C_B values of 48,8,16

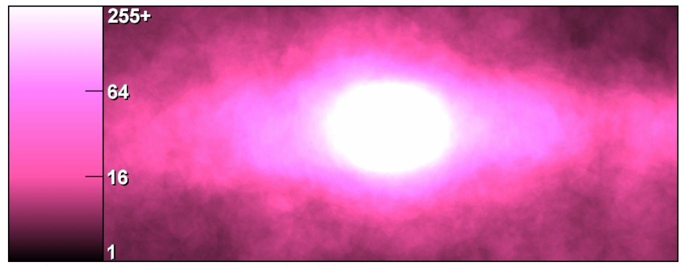

Figure 21: Data cloud with C_R, C_G, C_B values of 3,1,2

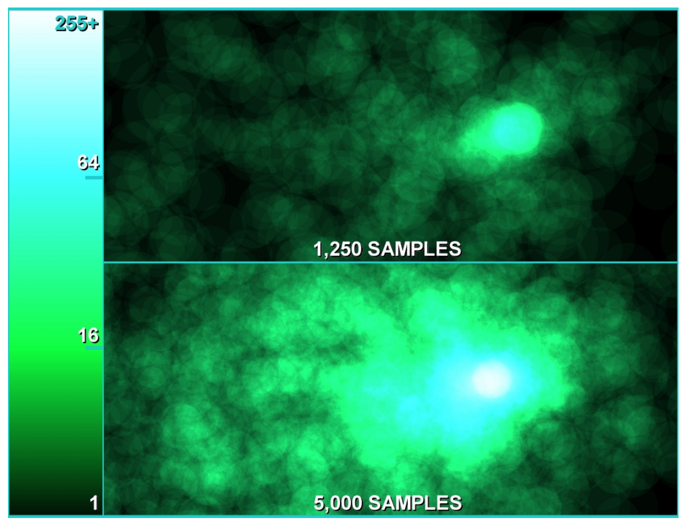

Figure 22: Debris cloud comparison of sample size

Size of debris sample representation

The debris samples generated from a numerical simulation are often "point masses" that are represented with 3D location and time, but not size. Depending on the threshold for density predictions, different representations may be required. Similarly, debris densities can misinterpreted if not evaluated along with its represented size. In Fig. 23, the same debris cloud is represented with a debris sample that is 2km in size and 10km in size.

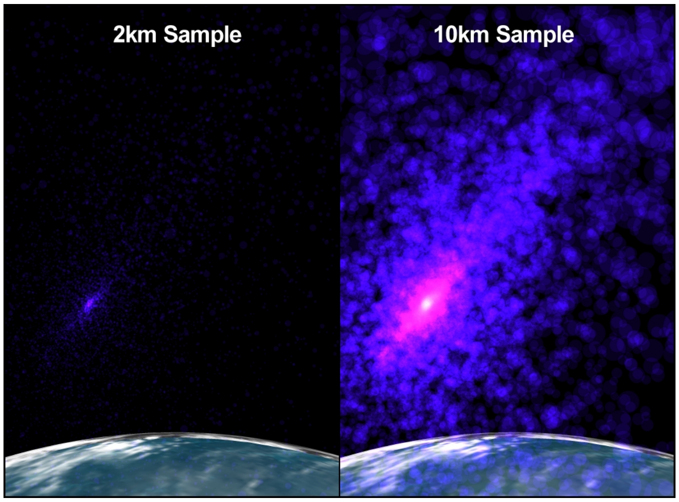

Figure 23: Debris cloud representation size comparison

CONCLUSION

Predicting the effects of debris events on active space payloads is a complicated activity surrounded by many unique challenges. The ability to utilize color and opacity blending techniques in an OpenGL environment is an effective means of addressing some of these challenges including computational overhead, inaccurate display representations, and the need for rapid comprehension of initial assessments.

Large, time-varying space debris datasets produced from break-up models can be an effective means of augmenting tracked datasets through its ability to represent probabilistic (versus deterministic) densities to assess debris effects. By visualizing these simulated debris clouds relative to payload trajectories, analysts can rapidly identify high-risk assets that can be targeted for higher-fidelity analysis.

VIDEO EXAMPLE OF TECHNOLOGY

An example of the color and opacity blending visualization technique described in this paper can be downloaded from **www.agi.com/ss**.

REFERENCES

1. Kelso, T.S. and S. Alfano "Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space (SOCRATES)", 15th AAS/AIAA Space Flight Mechanics Conference, AAS 05-124, January 23-27, 2005.
2. "Technical Report on Space Debris," UNITED NATIONS PUBLICATION, Sales No. E.99.I.17, ISBN 92-1-100813-1, pg. 5.
3. Malzbender, Thomas and Michael Goss, U.S. Patent 6,278,459
4. Sodenberg, Brian and Mark Kenworthy, U.S. Patent 5,339,386

CONTACT

David Massa (dmassa@agi.com)
Product Manager, Analytical Graphics, Inc.
David Massa is an aerospace engineer who serves Analytical Graphics, Inc. (AGI) through its Product Management team. For over a decade, Mr. Massa has applied his education and experience to a wide range of physics-based data visualizations. In addition to promoting techniques for addressing technical challenges for the aerospace and defense communities through all of AGI's communication channels, Mr. Massa's current responsibilities includes providing organizational leadership, counsel, and tactical direction to driving AGI's visualization technology.

Timothy Carrico (tcarrico@agi.com)
Principal Software Eng., Analytical Graphics, Inc.

DEFINITIONS, ACRONYMS, ABBREVIATIONS

- **Break-Up Model:** A computer program that attempts to simulate debris dispersion.
- **CGI (Computer-Generated Imagery):** A term to describe the rendering of a 2D image from 3D data.
- **COB (Color and Opacity Blending):** An OpenGL-based technique for visualizing large datasets by blending color and opacity.
- **Data Cloud:** A large number of processed data points that can be instantiated within a 3D volume.
- **Data Track:** Data with time-varying attributes.
- **Ephemeris:** A tabulation of computed positions and velocities of an orbiting body at specific times.
- **Ephemeredes (or Ephemerides):** The plural of ephemeris.
- **Full Force Model:** A computational model that accounts for the accelerations that an object experiences beyond basic equations of motion to include factors such as solar radiation pressure.
- **MTO (Multi-Track Object):** A container object used by STK to display large time-varying datasets.
- **Numerical Integrator:** An algorithm to calculate the results of a mathematical expression using approximate numerical methods.
- **OpenGL (Open Graphics Library):** A standard specification for writing applications that produce graphics from 3D data.
- **Render:** The process of creating 2D computer-generated imagery by evaluating the data properties of objects in a 3D environment.
- **STK (Satellite Tool Kit):** A commercial software application used by the aerospace and defense communities for analysis and data visualization.
- **Texture Map:** A means of adding color to an object in a 3D display using a 2D digital image.
- **Virtual Camera:** The unseen viewpoint for the viewing angle of a 3D display environment.