



FULL SCENE ANTIALIASING ON QUANTUM3D PCIGs

Antialiasing Methods,
Benefits and Drawbacks

Author: Brian Dickens

Date: 7/24/2002 Copyright 2002 Quantum3D, Inc.

PAGE 1

What is aliasing?

When generating computer graphics images a continuous signal representing the displayed objects projected onto a view plane is sampled at discrete points defined by the array of pixels in the frame buffer. Most forms of spatial aliasing are the result of sampling the objects using a rectangular grid of points that represent the location of pixels in the frame buffer. Since the projected objects are not constrained to lie on integer grid points, certain artifacts are introduced into the image. These are due to the fact that the object information is of too high a frequency for the sampling grid to capture. Typical aliasing artifacts present on objects in the image are called stair-steps, jaggies, and pixel flicker/jitter. Other aliasing of texture on the objects results in scintillating images and moiré patterns. When maneuvering your eyepoint through a 3D model, aliasing becomes even worse since these jaggies move or crawl in the image.

Figure 1 below shows the major cause for aliasing in a system without antialiasing. The dots indicate the pixel sample points. **Figure 2** shows the results of aliasing in a graphics scene.

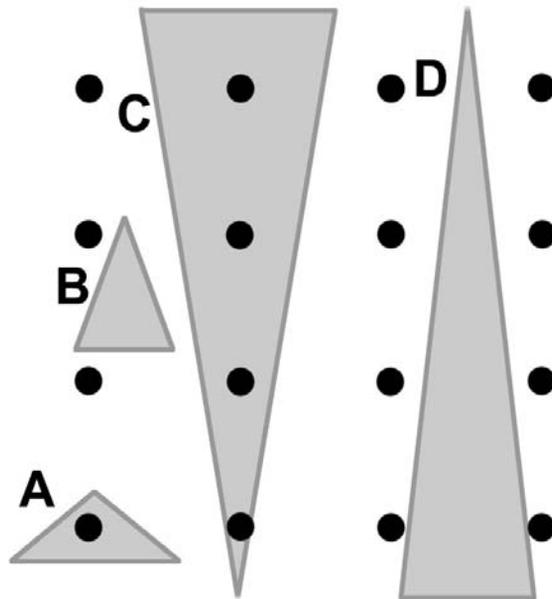


Figure 1: Why aliasing occurs in point sampled graphics (no antialiasing). Objects A and C are represented by the samples, B and D are not.



Figure 2: Jaggies and Subpixel flicker. Note that the telephone poles should be complete. When moving through the database, the poles flicker on and off as they cross the center of the sampled pixel. In this picture parts or complete poles are missing.



Figure 3: Moiré patterns or high frequency noise

What is antialiasing?

Antialiasing is a method of reconstructing the continuous signal of projected objects to ensure that the discrete pixel array in the frame buffer represents the objects as closely as possible. This increases the image quality and realism of the resulting image on the screen.

What is full-scene antialiasing (FSAA)?

Contrary to some antialiasing techniques, FSAA performs antialiasing on each and every pixel that represents the scene. FSAA is independent of drawing order, edges, object boundaries, textures or other object attributes. Implemented properly, FSAA is completely independent of the application and the objects being drawn. The most common method of implementing FSAA is called supersampling or multisampling. This technique takes more than one sample for each pixel and combines them to arrive at the resulting pixel value. These sample points are called subpixels. Most antialiasing methods are named by the number of subpixels that are sampled, for example four subpixel antialiasing would take four sample points for each pixel.

What are the benefits of FSAA?

In general the benefit of antialiasing is increased image quality, realism and the reduction of artifacts. Since we do not perceive jaggies or pixel flicker in real life, reducing these artifacts makes simulation applications more realistic.



Figure 4: No antialiasing. Notice the jaggies and sub pixel flicker. Some telephone poles or parts of them are not intersecting the center of a pixel and as a result are not displayed. This is typical of any system that does not have antialiasing.



Figure 5: Four subpixel full scene antialiasing. Notice that more of the telephone poles are complete and some in the distance are appearing.



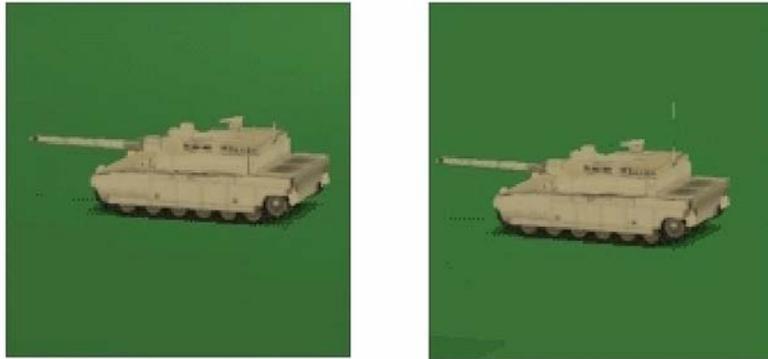
Figure 6: Eight subpixel full scene antialiasing. Notice that all of the telephone poles are complete, even the ones in the far distance.

Reducing aliasing is especially important to simulation applications. Since most of the aliasing is spatial aliasing, simulation users (pilots, gunmen, truck drivers...) end up making assumptions about their environment based on these aliasing artifacts. With aliasing artifacts present on the simulation training display, pilots make

decisions based on the amount of aliasing present or when certain artifacts appear. Since these artifacts do not exist in the real world, they are unable to use the same cues learned in training to make decisions in the real world; this is called negative training. The impacts of negative training can range from subtle to catastrophic. Antialiasing decreases the amount of negative training by more accurately representing the 3D objects.

Although the benefits of increased realism are obvious, to a pilot or gunman it means much more. Being able to quickly determine the correct shape and size of an object will help them determine if an approaching object is friend or foe. Aliasing distorts the true size and shape of objects and handicaps the trainee. Target recognition is crucial to simulations and cannot be performed well on systems with inadequate antialiasing. Aliasing creates many distractions in the scene. High quality antialiasing using FSAA is a requirement for any training applications.

9 Tap Gaussian Antialiasing



Quantum3D Rotated Grid Antialiasing



Figure 7: Notice that the antenna on this tank flickers and disappears these distractions prevent users from having a positive and effective immersive experience and could prevent them from recognizing a target. This same effect will happen on various parts of the tank depending on distance; the cannon will disappear and flicker as well.

What are the drawbacks of full-scene antialiasing?

The technical hurdle when using full-scene antialiasing is the impact on fill rate performance on COTS hardware. Unfortunately there are only two straightforward ways to decrease applications demand for fill rate. The first is to decrease the resolution of the application and the second is to decrease the depth complexity of the database. Unfortunately doing either of these is unacceptable to most visual simulation applications.

This drawback can be overcome by using systems that have the capability to perform antialiasing using dedicated hardware designed specifically with antialiasing in mind. These systems have the graphics horsepower needed to perform as well as, if not better than COTS graphics accelerator even when FSAA is enabled.

Quantum3D systems use multiple graphics chips to implement fullscene antialiasing without compromising performance. Quantum3D's Parallel Rendering Architecture powers graphics systems that are capable of increasing fill rate performance and image quality by coupling graphics chips together. As a result, peak fill rates of nearly a gigapixel enable applications to run at real-time frame rates with FSAA enabled (four or eight subpixel rotated grid) at resolutions as high as 1600x1200.

Edge antialiasing

There are other forms of antialiasing. The most popular technique is to antialias only the edges of objects in the scene. In theory, edge antialiasing will work very well at removing edge jaggies and some instances of pixel flicker, but in practice it is very difficult to use.

Benefits of edge antialiasing

The benefit of edge antialiasing is that it does not result in 75% loss of fill rate performance and if used on only a few objects in the scene, edge antialiasing may not adversely affect application performance. Systems that have good polygon performance may be able to handle the addition load of the added edges.

Drawbacks of edge antialiasing

One of the biggest drawbacks to edge antialiasing is that it will not reduce all of the aliasing artifacts present within a polygon.

Depending on the graphics hardware that is used, edge antialiasing may or may not be hardware accelerated and may involve drawing additional primitives. For example many systems not only require that you draw the original object, but that you also draw the antialiased edges on top of the objects. This increases the number of primitives that the graphics system must draw and thus reduces performance. Implementing edge antialiasing on each and every object in the scene will double the primitive count and reduce performance by as much as 50%.

The drawing of an outline of alpha-blended antialiased edges around all objects will cause all such objects to grow by a small amount. This has negative effects on target detection/recognition, and may even cause small objects to "squeeze out" small details (such as the gap between a tank's turret and the top of the treads). Nasty special cases will also occur on triangles of extreme aspect ratios (much wider than they are tall, for example). These will cause wedges to appear outside the triangle that will be distracting.

When outlined triangles form a mesh, and the resulting mesh is rendered with partial transparency (using alpha blending, such as in a smoke cloud) the resulting object will display its internal edges with incorrect transparency. The internal edges will be more opaque than the interior of these triangles, due to the summing effects of alpha blending.

Edge antialiasing algorithms cannot predict when objects will interpenetrate each other. Modern Z-buffer architectures allow 3D objects to freely interpenetrate, and implicit edges will be formed along the joint between the two objects. A pure edge antialiasing algorithm cannot smooth these edges.

Perhaps the most complicated problem to solve when using edge antialiasing is that of drawing order. In order to properly antialias the edge of an object, it must be drawn after (on top of) the objects that appear in the image around it. This is required in order to antialias the edge with the other objects in the scene. Ensuring that the edge is drawn on top of the surrounding objects is difficult and must be done either in the

3D object definition or computed automatically by the 3D application software. Performing this ordering in the object definition using a 3D design package is a tedious process at best and will not result in perfect results due to human error.

Other requirements such as the ordering of transparent objects create even more difficulty for artists and database engineers. The only method that can be used to guarantee proper order is for the application software to perform this operation automatically. The application software is then required to sort each object (and possibly each polygon within the object) for each and every frame that it draws. Sorting objects like this is an expensive calculation that robs applications of graphics performance. Because of the cost of sorting it is not possible to edge antialias each and every object in the scene and further reduces the benefits of edge antialiasing.

Which is better, FSAA or Edge AA?

Without a doubt FSAA is the preferred method of antialiasing. It does not suffer from the limitations and performance impacts as edge antialiasing (if using a parallel rendering architecture such as that found in Quantum3D systems). In addition it does not involve any special database or application modifications.¹

Full-scene antialiasing does not have the same performance robbing drawbacks of edge antialiasing (if implemented in hardware). No special work is required by the 3D content developer. The application developer does not need to incorporate complicated sorting mechanisms for ordering the polygons in the database. Full-scene antialiasing operates at the pixel level, so it implicitly works on each and every object in the scene. This means that full-scene antialiasing is not limited to merely reducing the jaggies in a scene. FSAA will reduce jaggies as good as, if not better, than edge antialiasing, but will also reduce subpixel flicker, texture aliasing and moiré patterns that are a result of high-frequency noise in textured images.

At present, edge antialiasing algorithms are only configurable via the width of the edge drawn. There is no way to accurately control the number of samples when using edge antialiasing methods. As a result there are no tunable image quality parameters.

When looking at computer graphics systems offered by the leading manufactures it is clear that hardware accelerated FSAA is where the graphics industry is headed. In the past, utilizing FSAA has required very expensive high-end systems and has been limited to applications with large budgets. As a result of this, applications developers and hardware manufactures supported edge antialiasing methods on lower end systems in order to offer lower cost alternatives.

With the availability of FSAA systems from Quantum3D and other manufactures, FSAA is preferred and edge antialiasing is now rarely used.

FSAA on Quantum3D systems

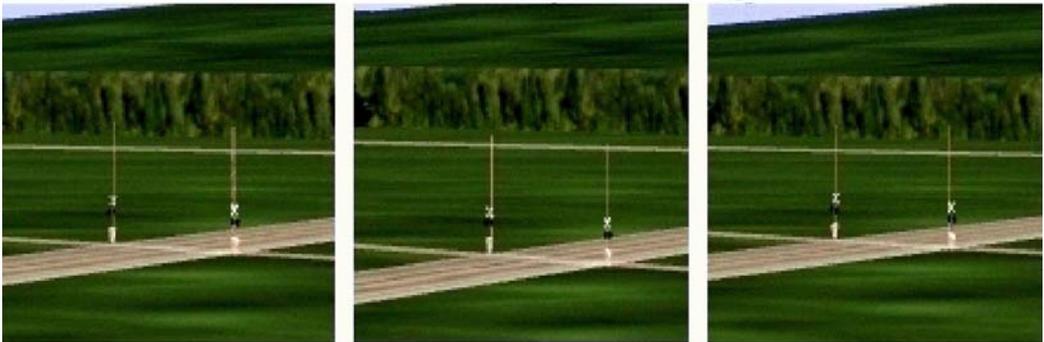
Quantum3D systems have hardware-accelerated FSAA. AAlchemy systems have the ability to change the number of samples per pixel and the location of these samples. This gives one the ability to tune the image quality of the system depending on application requirements. Quantum3D currently offers AAlchemy systems with hardware accelerated four and eight subpixel FSAA. Future systems will offer greater number of subpixels with performance levels higher than AAlchemy.

Quantum3D AAlchemy systems offer performance that surpasses commodity graphics cards with FSAA enabled. This is due not only to the fill-rate performance of the underlying graphics chipset, but also because AAlchemy employs a patented Parallel Rendering Architecture that utilizes multiple graphics chips in order to provide increased fill rate and subpixel antialiasing. Quantum3D's Rotated Grid selection of subpixels results in superior image quality when compared to a regular grid algorithm.

4 Sub pixel with 9 tap gaussian



Quantum3D 4 Sub pixel rotate grid



No antialiasing



Figure 8: Each column of images above is an identical frame, each row of images represents a certain method of antialiasing. Notice that Quantum3D 4 Subpixel Rotated Grid antialiasing significantly reduces jaggies as well as subpixel flicker. Although antialiasing methods use four subpixels the Rotated Grid selection of subpixels results in significantly improved image quality when compared to the aligned grid method. Enabling this feature on COTS graphics cards requires four times the graphics bandwidth. As a result fill rate performance is dropped to 25% of peak fill rate. Since the jaggies are still very noticeable, crawling, jittering and flickering aliasing remain in the image even with 9 tap Gaussian enable. This is even more noticeable when moving through a 3D model.

Determining antialiasing performance and quality

All standard PC graphics cards that support antialiasing give the user control over enabling antialiasing. Some allow different techniques to be used for antialiasing. For example on Quantum3D AAlchemy systems use: [StartMenu/Settings/ControlPanel/Display/Settings/Advanced/AA5 8000](#) to access all driver controls. Quantum3D Obsidian systems have antialiasing controls under [StartMenu/Settings/ControlPanel/Display/Settings/Advanced/Quadro DCC/Additional Properties](#).

In order to determine the image quality of various forms of antialiasing, it is possible to use almost any application. A trained eye can easily assess the image quality. Copying the screen and zooming in on areas can be used to magnify the pixels and see how a particular antialiasing method works.

Determining performance will typically involve two runs of an application, one with antialiasing enabled and one with antialiasing disabled. Ideally benchmark data is saved during each run to make comparisons easily. However, one must remember that there is no free lunch. Unlike on Quantum3D AAlchemy, when FSAAs are enabled on most PC graphics accelerators you will experience a considerable drop in performance. Typically this loss of performance is due to decrease fill rate performance. On nVidia architectures performance will drop to 25% of peak performance when using a four subpixel antialiasing method. This drop in performance is because the same chip must effectively calculate four times as many pixels.

Below are a few examples of applications and benchmarks that may be used to assess image quality and determine performance.

OpenGVS preview of Onyxtown database

Use the following link to download this demo program. Size is approximately TBD MB.

<http://www.quantum3d.com/downloads/gvstown.zip>

This demo allows you to easily fly through the SGI Onyx database as shown in the snapshots above. Use [townogl.bat](#) for any OpenGL based graphics system and [towng3.bat](#) for AAlchemy graphics. When in the application, use F1 for help. Move through the database looking for a variety of aliasing. Jaggies and subpixel flicker are common in the database (signs, telephone poles, goal posts) as well as interior texture aliasing on the fence in town. Make notes during your aliasing/antialiasing runs and compare your notes. Taking snapshots of the screen might also be useful.

OpenGVS triangle test application

The triangle test program may be used to help determine the number of moving models and polygons a system can manage. Use this test with antialiasing off and then compare your results with antialiasing on. Use the following link to download this demo program. Size is approximately TBD MB.

<http://www.quantum3d.com/downloads/gvstritest.zip>

This demo allows you to continually add a moving model to on the display. Each model is 1000 polygons and has eight articulated parts. Each is intersected to the ground, uses a unique display list and is drawn on a simple background that contains 3 layers of polygons covering the entire screen on each layer (i.e. depth complexity 3 background and increased where the models are drawn). Use [tritestogl.bat](#) for any OpenGL based graphics system and [tritestaa5.bat](#) for AAlchemy graphics. Use F1 for help.

OpenGVS OIQ

The OpenGVS Object Image Quality program will give the user a mechanism to run the same application on two PC systems. The systems will be synchronized using the network to be at precisely the same location and viewpoint in the image quality test. This test is for object recognition of ground and air vehicles and can be downloaded from: <http://www.quantum3d.com/downloads/gvstritest.zip>

Resources:

- Computer Graphics Principles and Practice, Foley, van Dam Feiner, Hughes
- OpenGL Programming Guide, Scene Antialiasing p 313
- The Accumulation Buffer: Hardware Support for High-Quality Rendering by Paul Haeberli and Kurt Akeley (SIGGRAPH 1990 Proceedings).