



## **Technical Brief**

---

HRAA: High-Resolution  
Antialiasing through  
Multisampling

*N*VIDIA

NVIDIA's GeForce3 graphics processing unit (GPU) introduces radical new technology designed to provide the industry's first solution for high-resolution antialiasing (HRAA). While other antialiasing solutions and their methods have been offered before, only GeForce3 can offer the unique combination of high resolutions and high frame rates in combination with high-quality antialiasing. By implementing hardware support for multisampling, as well as innovating new sampling technology, GeForce3 makes HRAA available for all real-time content.

## Generic Solutions for Antialiasing

One of the key image quality problems that has plagued PC users is aliasing, or the "jaggies". Aliasing is the stair-step effect on the edges of objects and can be extremely distracting for the PC user. Readers who are unfamiliar with aliasing and its causes are referred to Appendix A for a more detailed explanation.

**Higher Resolution** - The only way to combat aliasing is to create the effect of having more pixels on the screen. Historically, increasing the resolution was the best solution to this problem. The size of the "jaggy" or stair-step artifact is never larger than the size of the actual pixel. Hence, reducing the size of the pixel reduces the size of these artifacts. Changing the resolution is not always feasible, however. The user may already be using the maximum resolution supported by the monitor or the application itself may limit the resolution. Beyond these hard limits the only solution is to increase the *effective* resolution. The best way to do this is to use more sophisticated techniques for computing the color of each pixel of the display in a way that simulates having more pixels. These techniques are referred to as "antialiasing."

**Supersampling** - Algorithmic antialiasing techniques involve "sampling" the content of each pixel at multiple locations, meaning that the color is computed at more than one location inside the area covered by the pixel. The results from these "samples" are combined to determine the final color of the pixel. These samples are essentially additional pixels, used to increase the *effective* resolution of the image to be displayed. If the edge of an object falls partially inside the area of a pixel, its color and the color of another object that partially fills the "area" of the pixel can both be used to calculate the final color. The result is smoother transitions from one line of pixels to another line of pixels along the edges of objects, where aliasing is most obvious.

"Supersampling" is an antialiasing technique that is simply a brute force approach and is used in NVIDIA's GeForce2 GPUs and other modern graphics processors. A graphics processor that uses supersampling renders the screen image at a much higher resolution than the current display mode, and then scales and filters the image to the final resolution before it is sent to the display. A variety of methods exist for performing this operation, but each requires the graphics processor to render as many additional pixels as required by the supersampling method. Additionally, because the graphics processor is rendering more actual pixels than will be displayed, it must scale and filter those pixels down to the resolution for final display. This scaling and filtering can further reduce performance.

The degree of scaling in a specific supersampling mode is often identified by the ratio of pixels in the unscaled image to the number of pixels in the final, scaled output. For example, 2x supersampling writes twice as many pixels to the frame buffer as would be required without antialiasing. 4x writes four times as many pixels. As you might guess, supersampling causes a substantial drop in performance as measured

by frame rate. If the graphics processor renders four times as many pixels, then the frame rate will be one-fourth what it was in the standard display mode. In fact, the performance drop can be even worse than the "x" multiple of the supersampling setting because of the scaling step mentioned in the previous paragraph.

The biggest problem with supersampling is performance. Users don't want to see their frame rate fall to one-half or one-fourth of what it should be. Multisampling is a more sophisticated technique than supersampling that involves higher quality output than standard rendering with much higher performance than supersampling--a win/win scenario compared to either alternative. Multisampling requires a more sophisticated GPU, however, so only the newest GPUs available support this technique.

## Multisampling for High-Resolution Antialiasing (HRAA)

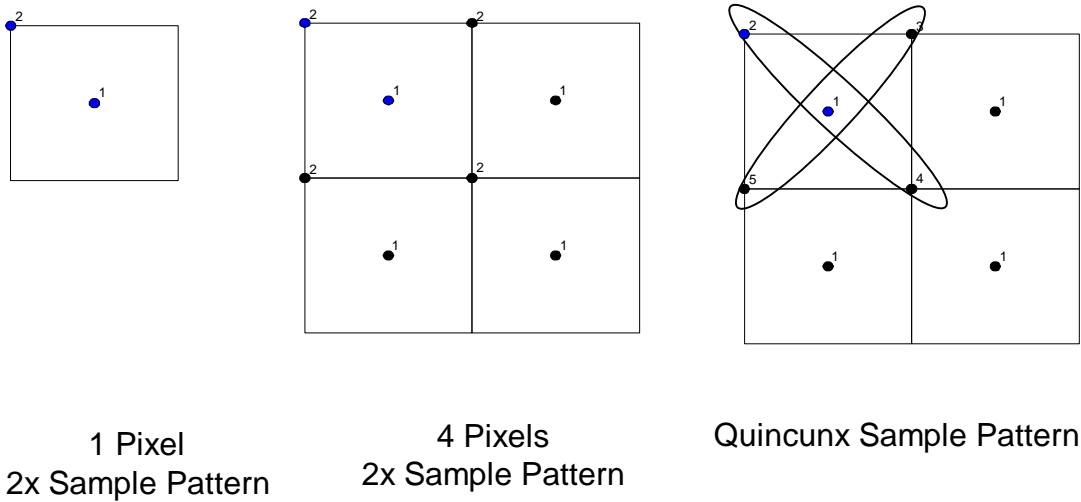
The basic idea behind multisampling is to embed the intelligence behind antialiasing inside the core of the GPU, in hardware. This makes the GPU more complex, but rewards the user with higher quality and higher performance.

Multisampling works because the GPU itself is "aware" that multiple samples will be used to calculate the final pixel color. You can think of these extra samples as extra "virtual pixels." The GeForce3 GPU has wider data paths internally so it can handle these extra virtual pixels without slowing down its standard rendering speed. In fact, the GeForce3 GPU can compute these "virtual pixels" or additional samples at full speed, with no reduction in engine performance whatsoever. These wider data paths enable GeForce3 to use the same texture data for all of the samples in the pixel and significantly reduce the memory bandwidth required to texture all of the AA samples. Memory bandwidth can be the performance bottleneck for high-resolution display modes and becomes even more constraining when antialiasing is used. For more information about memory bandwidth constraints, readers are referred to a separate technical brief from NVIDIA titled "GeForce3: Lightspeed Memory Architecture."

GeForce3 has new technology for graphics hardware antialiasing in the form of multisampling using a new sampling pattern. The locations of the sample points for multisampling enable the use of a "reconstruction filter" that uses data from neighboring pixels to compute the final pixel color. This increases the total amount of work, and therefore slows down the frame rate marginally, but it delivers much higher quality as a reward. This new sampling pattern is called Quincunx, after the name of the pattern of the five dots on the "5" sided of a 6-sided die. This Quincunx pattern, implemented in hardware, offers quality comparable to "4x" modes, with performance similar to that of "2x" modes. The higher quality yielded by using a Quincunx reconstruction filter is a key advancement and offers a fundamental improvement in the quality vs. performance tradeoff of antialiasing.

The Quincunx AA mode works so well because it uses data samples from neighboring pixels instead of requiring the GPU to create additional samples for each pixel in the initial rendering step. Figure 1 shows the sample grid and how samples from neighboring pixels are used.

Figure 1 AA Sampling Patterns



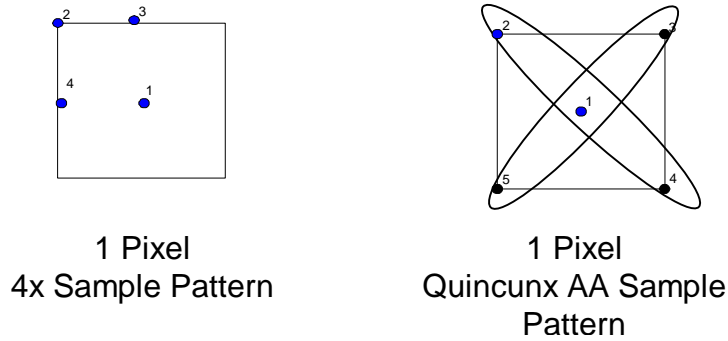
Because the Quincunx AA mode uses samples from neighboring pixels, it balances the image quality benefit of using more input data to reconstruct the final pixel versus the performance degradation of writing more samples per pixel out to the frame buffer. In addition, because the Quincunx method is twice as efficient as traditional 4x, Quincunx AA only requires the smaller memory footprint that 2x modes require, allowing for higher native resolutions without bumping into the limits of graphics card memory. Figure 2 shows the memory footprint of key antialiasing display modes.

Figure 2 Memory Storage Requirements for AA Display Modes

Horizontal Resolution	Vertical Resolution	Color Depth (bits)	Frame Buffer Storage Required in MB			
			No AA	2x AA	Quincunx AA	4x AA
640	480	32	3.6	6	6	10.8
800	600	32	5.625	9.375	9.375	16.875
1024	768	32	9.216	15.36	15.36	27.648
1280	1024	32	15.36	25.6	25.6	46.08
1600	1200	32	22.5	37.5	37.5	67.5
2048	1536	32	36.864	61.44	61.44	110.592

Figure 3 provides a comparison of the data sample points used to reconstruct each pixel for the Quincunx AA mode compared to a standard 4x mode. The locations of these sample points apply specifically to NVIDIA's GeForce3. Other GPUs may take pixel samples at different locations within the pixel area.

Figure 3 Quincunx AA Sample Pattern



## Results

The true test of any antialiasing solution is the balance of performance and image quality. NVIDIA's Quincunx AA is the ultimate balance of these two qualities. Figure 4 shows magnified samples from actual *Quake3*<sup>TM</sup> screen shots to highlight the visual differences between GeForce3's antialiasing modes. Note the similar complexity of the Quincunx AA and the 4x examples. The larger number of intermediate colors is driven by the larger number of samples that both Quincunx AA and 4x use to reconstruct the pixel compared to the standard 2x mode.

Figure 4 Magnified Screen Shots of GeForce3's AA Modes

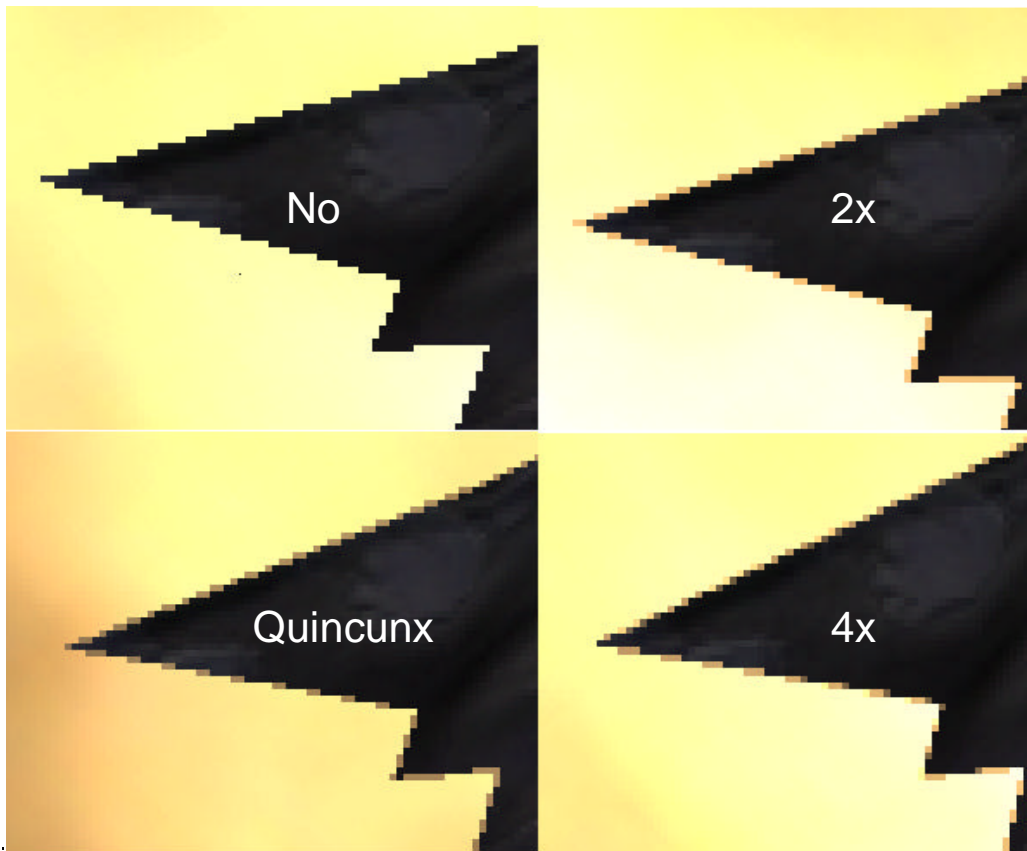


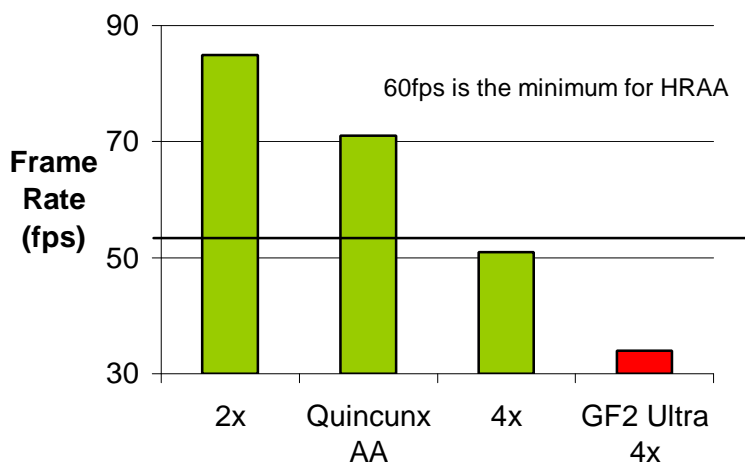
Figure 5 shows a larger section of the screenshots for three of the four settings in figure 4: Off, Quincunx AA and 4x.

Figure 5 Samples of GeForce3's Antialiasing Modes



GeForce3 delivers these antialiasing modes with incredible performance levels. This is the first GPU that delivers frame rates above 60fps in high-resolution modes. For this reason, GeForce3 is the first GPU to deliver HRAA, or high-resolution antialiasing, to the end user. Lesser GPUs can technically support antialiasing, but users simply turn it off because the frame rates are too low for serious game playing. HRAA requires a minimum of 60fps at 1024x768 with full 32-bit color. Figure 6 shows the relative performance levels for several AA modes.

Figure 6 – AA Performance in Quake3 at 1024x768x32



## Conclusion

The GeForce3 GPU is the first product ever to deliver HRAA at a consumer price point. The hardware multisampling capabilities of the GeForce3 GPU offer an unprecedented level of performance and flexibility, giving end users more and better mode choices for antialiasing, and helping them tap the full potential of this groundbreaking graphics processor.

## Appendix A

### Causes and Effects of Aliasing

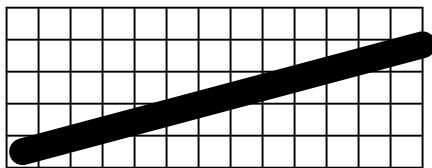
Aliasing is the stair-step effect that is highly visible on computer displays when a line is drawn at an angle close to vertical or close to horizontal, but not exactly vertical or horizontal. This stair-step effect is often referred to as the "jaggies" because it makes lines that should be smooth appear jagged. The height (or width) of each stair step is equal to a pixel on the computer display. Figure 1A shows a screen shot that shows several image defects cause by aliasing.

Figure 1A

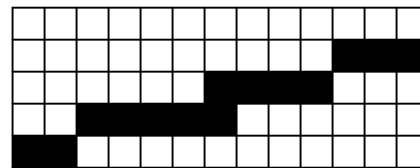


Aliasing can also make a line appear thicker than it should be. Both of these image "defects" are shown in Figure 2 which includes two versions of a line: one near perfect line and the other heavily aliased.

Figure 2A



Smooth Line



Aliased Line

(Note: these lines were magnified to clearly show the issues.)

What causes aliasing? In the simplest terms, aliasing is caused by the "grid" of pixels in a computer display. Virtually every computer display is what is called a "raster" display, meaning that pixels are located in exact rows and columns. Aliasing occurs when a line crosses those rows and columns at a slight angle to either the row or the column. The result is that a specific point on a line that should be 1-pixel



wide might be half on one row and half on a neighboring row. Using the near-horizontal line from Figure 2 as the example, this leaves 3 choices for which pixels get filled in whenever the line is crossing from one row on the display to another: only the pixel on the bottom row, only the pixel on the top row, or both pixels. Each choice involves trade-offs. If both pixels are filled in, the line becomes 2 pixels wide, if only one pixel is filled in for any column of pixels, then the line narrows to almost nothing where the line transitions from one row of pixels to a neighboring row. Both of these "defects" are shown in Figure 2.