



Technical Brief

NVIDIA Accuview Technology High-Resolution Antialiasing Subsystem

The top left of the page features a large, faint NVIDIA logo. To its right, there is a graphic of concentric, wavy lines in shades of blue and white, resembling a stylized eye or a lens flare, which is part of the Accuvision technology branding.

NVIDIA

Accuvision Antialiasing

A primary visual quality issue for PC users is on-screen aliasing, the “stair-step” effect that is highly visible on computer displays. This stair-step effect is commonly referred to as the “jaggies” because it makes a line that should really be smooth appear jagged. To end users, these jaggies can be extremely distracting, and detract from their overall immersive experience.

While the edges on a single image can look acceptable, visual artifacts become even more readily apparent with moving images due to temporal quantization errors. As the diagonal edge of an object moves from one frame to the next, the portions of the edge that show up can change dramatically. The eye is quick to notice these changes, and it is especially distracting when portions of thin lines pop in and out of view from one frame to the next.

Although there are many antialiasing (AA) techniques available that help diminish the appearance of the jaggies, none are sophisticated enough to solve these visual disturbances while providing both high-level visual quality and superior performance.

The NVIDIA® Accuvision™ Antialiasing subsystem however, tackles these difficult antialiasing problems by providing a unique and flexible technology architecture that delivers high-quality graphics at unbeatable levels of performance. For the first time, end users can choose high-resolution antialiasing as their default display mode, without suffering any performance degradation in their favorite games and applications.

The Antialiasing Challenge

There are many approaches to solving the aliasing problem, including rendering to a higher resolution and sampling each pixel at more locations. These methodologies increase the frequency at which an image is sampled or displayed. The more granularity and detail there is to work with, the less jagged the image will appear.

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 end user may already be using the maximum resolution supported by the monitor, or the resolution may be limited by the application itself. 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.”

Sampling

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 a smoother transition from one line of pixels to another line of pixels along the edges of objects, where aliasing effects are most obvious.

“Supersampling” is a brute force antialiasing technique. 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. Unfortunately, this scaling and filtering process 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 aforementioned scaling process. Supersampling is also highly inefficient, and wastes valuable texture bandwidth fetching texture data for pixels that will never actually be drawn or seen.

Note: End users need to be aware that not all products follow these conventions, as they may actually render less pixels than the antialiasing mode would suggest. End users must judge for themselves whether the product is performing as advertised.

Multisampling

Multisampling is a more sophisticated technique than supersampling. Multisampling involves higher-quality output than standard rendering, with much higher performance than supersampling—a win-win scenario. Multisampling requires a more sophisticated graphics processing unit (GPU), however, so only the most complex GPUs, including the NVIDIA GeForce4™ GPU family, are capable of providing this level of antialiasing support.

The NVIDIA Accuview Multisampling Solution

The basic idea behind multisampling is to embed the antialiasing intelligence (which is inside the GPU core) into hardware. This makes the GPU more complex, but rewards the end user with higher-quality visuals and faster 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 GeForce4 GPU family integrates the NVIDIA Accuview antialiasing subsystem, which provides wider internal data paths to handle these extra virtual pixels without slowing down its standard rendering speed. In fact, the GeForce4 GPUs can compute these “virtual pixels” or additional samples at full speed, with no reduction in engine performance whatsoever. These wider data paths enable the GeForce4 GPUs to use the same texture data for all of the samples in the pixel, significantly reducing the memory bandwidth required to texture all of the AA samples.

Note: 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, please read the NVIDIA “Lightspeed Memory Architecture II” technical brief.

The Accuview Advantage, Improved Quality and Performance

The Accuview engine improves upon NVIDIA's multisampling technology by providing a variety of multisampling modes. These modes include 2x, 4x, Quincunx, and a new 4XS mode that delivers improved subpixel coverage better texture quality. In addition, Accuview incorporates anisotropic filtering for more detailed images, and a patent-pending pipeline to increase overall performance.

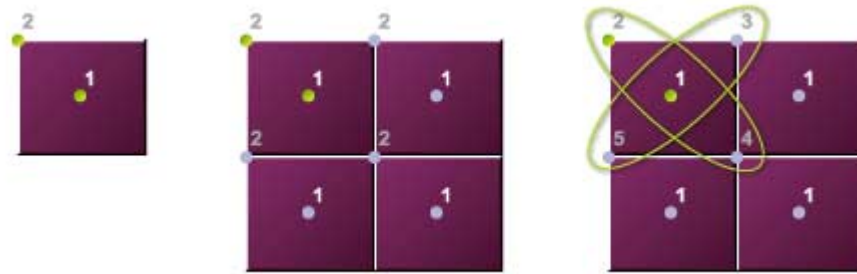


Figure 1. 2x and Quincunx AA sampling patterns

This approach was revolutionary in its balanced approach to quality and performance, but Accuview takes this technology one step further. In multisampling, the texel taken for a given subpixel sample is calculated and referenced as if it were to be placed in the center of the pixel. So, while Location 1 in Figure 1 contains a texel that exactly matches its location, Location 2 is off by some distance from the texel that is retrieved. This results in a possible color error; depending on the scale of the texture to primitives (texel-to-pixel ratio), this error could be large.

Accuview improves antialiased image quality by moving where the subpixel samples are taken from (Figure 2). This movement of subpixel sample points results in better antialiasing. Both sample locations now contain a smaller amount of error (as opposed to one containing no error, and one containing a larger amount of error). Multisampling with this type of distributed coverage is generally more accurate than the pattern mentioned previously. The result is a better antialiased image:

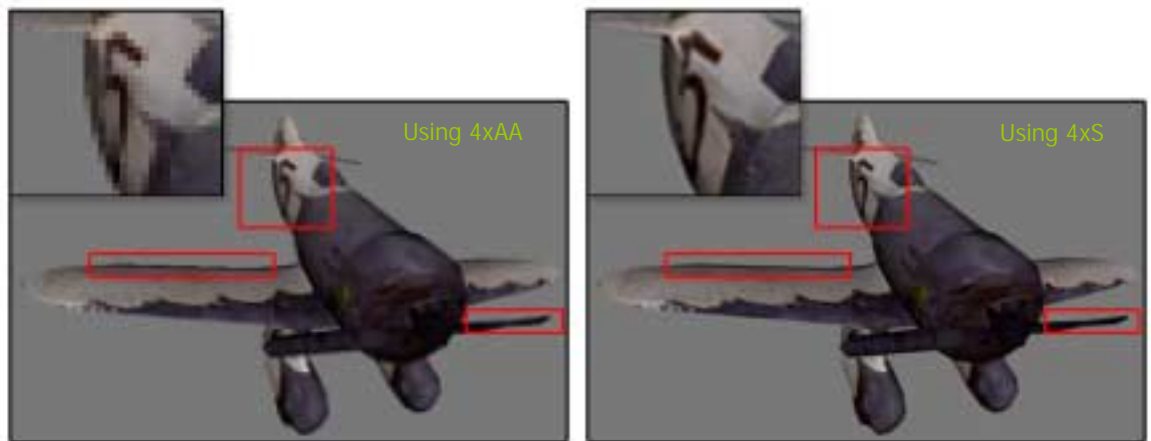


Figure 2. Accuview Shifted AA Sampling Patterns

4x and 4XS Modes: Higher Level of Texture Quality

4XS mode is a new high-quality mode that delivers improved subpixel coverage and a higher level of texture quality.

When the final color of a pixel is determined, all of the subpixels that can contribute to the final pixel color are tested. They are either “in” or “out” (that is, they contribute by being part of the object and that object covers the subpixel, or they don’t). The final pixel color is then reconstructed by implementing a weighted summation of all of the subpixels. 4XS mode delivers 50 percent more subpixel coverage than previous modes. This translates into a finer gradation in final pixel color values at the edge of an object, resulting in a smoother antialiased edge. Additionally, 4XS mode delivers a higher-quality image by delivering more texture samples per pixel.



The picture on the left uses 4x AA mode. The picture on the right uses 4XS mode. Notice how jagged the blow-up on the left plane is and how smooth it appears on the right. By using increased gradations in color, 4xS makes the lines and edges look crisp when viewed at screen resolution. The difference is even more noticeable in actual gameplay as the jaggies become more apparent when objects begin to move.

Figure 3: **4x and 4XS AA Modes**

Anisotropic Filtering: Advanced Visual Techniques

Another feature of Accuvision is its support for anisotropic filtering, an advanced texture-filtering technique that improves image quality for scenes with objects that extend from the foreground deep into the background. Anisotropic filtering provides the ability to choose the scale between a texture map and the primitive it is projected on to.

In 3D graphics, the texture map for a given primitive is chosen based on the size of the primitive and the resulting scale. Ideally, the scale is 1:1, so that each pixel of a primitive receives one texel. If the texture map is too large, there will be quality issues selecting the right texel, which results in an aliased image. Conversely, if the texture map is too small, the texture mapped on to the primitive will look blocky and

posterized. The generic solution for this is to create multiple maps (called mip maps) of a texture. Mip maps of the texture are created at various sizes and resolutions. Depending where in 3D space the primitive is rendered and displayed on the screen, the texture that “fits and looks the best “ is chosen.

This works for many cases. However, in situations where the polygon is large or at sharp angles, normal mip maps do not deliver the best quality. Anisotropic filtering forces a larger texture map onto a primitive in situations where it helps image quality. The result is a higher-quality, more detailed image, as seen in Figure 4.

See how the image on the left blurs as it goes farther off in the distance. The image on the right is the same 3D rendering with anisotropic filtering applied. Notice how the brick texture is crisp and clear.



Figure 4. **Anisotropic Filtering**

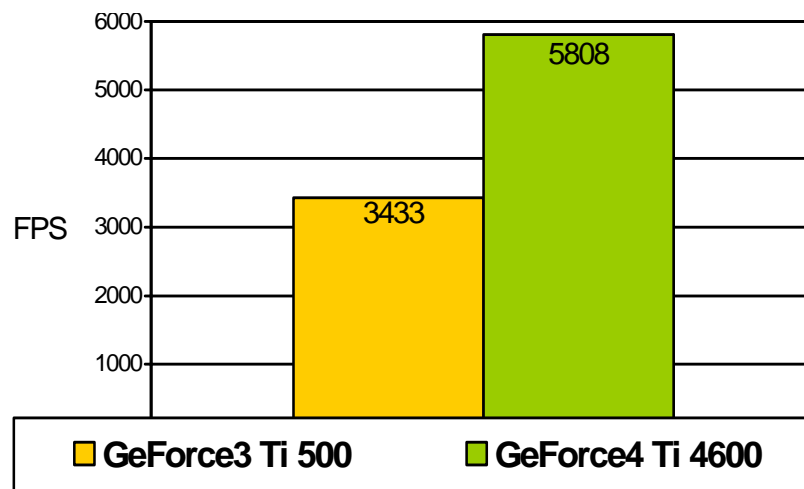
The secret of Accuvue lies in its flexibility in its use of anisotropic filtering. With Accuvue, anisotropic filtering can be used in conjunction with Bi-Linear or Tri-Linear filtering schemes. The result is a high-quality image that, combined with Accuvue’s multisampling technology, runs at blistering frame rates. This freedom allows the end user to receive the highest possible quality with exceptionally fast speed.

Optimized Pipeline for Increased Performance

The Accuvue subsystem substantially increases performance by optimizing the way pixels move through the graphics pipeline to create antialiased images.

1. Subpixels are rendered in parallel (thanks to multisampling technology) to a back buffer. This back buffer is a factor that is larger than the final display resolution.
2. The image is filtered and written out to a front frame buffer.
3. The frame buffer is sent to the display.

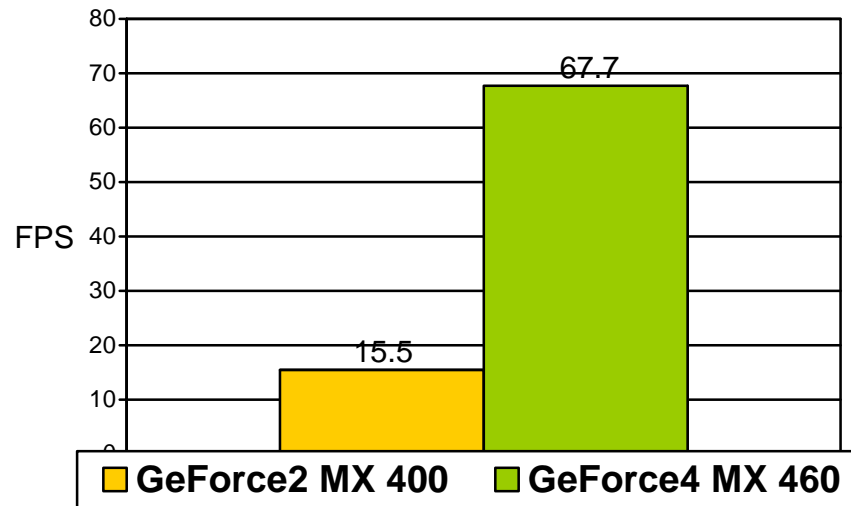
Accuvue's patent-pending technology optimizes this pipeline and reduces and eliminates steps by performing tasks in parallel. The result is a huge improvement in performance as computation time and bandwidth requirements are substantially reduced. Figures 5 and 6 show the performance benefits of the Accuvue technology:



System Configuration:

- P4-2.0GHz, Win XP, 128MB Memory, 12x10x32 resolution
- GeForce3 Ti 500 at 240/520 DDR
- GeForce4 Ti 4600 at 300/650 DDR

Figure 5. 3DMark 2001, 12x10x32 with Quincunx

**System Configuration:**

- P4-2.0GHz, Win XP, 128MB Memory
- GeForce2 MX 400 at 200/166 SDR
- GeForce4 MX 460 at 300/550 DDR

Figure 6. **Quake 3, 12x10x32 with 2X FSAA**

Conclusion

With the integrated Accuvision antialiasing subsystem, the GeForce4 GPU family sets a new standard in visual quality and performance. Accuvision advances antialiasing techniques to such a degree that antialiased operations should now be the default mode of all PC users. By offering a variety of multisampling modes, support for anisotropic filtering, and an optimized texture pipeline, Accuvision delivers unprecedented visual quality and performance without compromise.

The top left of the page features the NVIDIA logo in a large, light, serif font. To its left is an abstract graphic consisting of concentric, wavy lines in shades of blue and white, resembling a stylized eye or a lens flare. A thin green vertical line runs down the right side of the page.

Glossary

Bit Depth

The bit depth refers to the number of bits of precision for the color and z-values associated with each pixel on the screen. More bits of precision improve the visual realism and accuracy of the rendered frame. The two most common bit depths in modern graphics hardware are 16-bit and 32-bit. Each of these values can be associated with color or Z-values. Color that is 32 bit (for example) typically is used to represent red, green, blue and alpha (or transparency) values with up to 8 bits per component, or 256 “values” for each of those components. A 32-bit z-value is typically allocated as 24 bits of Z precision (or depth precision) and 8 bits of stencil or “mask” precision.

Depth Complexity

Depth complexity is a measure of the complexity of a scene. It refers to the number of times any given pixel must be rendered before the frame is done. For example, a rendered image of a wall has a depth complexity of one. An image of a person standing in front of a wall has a depth complexity of two. An image of a dog behind the person but in front of the wall has a depth complexity of three, and so on. As depth complexity increases, more rendering horsepower and bandwidth is needed to render each pixel or scene. The average depth complexity of today’s graphics applications is two to three, meaning that for every pixel you end up seeing, it gets rendered two or three times by the graphics processor.

Fill Rate

Fill rate is the rate at which pixels are drawn into the screen memory. Fill rate is a common measure used to illustrate the pixel processing capabilities of today’s 3D graphics processors. Fill rate is usually measured in millions of pixels/sec. (Mpixels/sec.) In 1997, 50-70 Mpixels/sec. was considered state of the art. In 2002, the leading 3D graphics processors will be capable of more than 1200 Mpixels/sec. While this improvement is an incredible achievement, it is still barely enough to create a compelling 3D environment. Rendering pixels at such a high rate consumes enormous amounts of memory bandwidth.

Frames per Second

Frames per second (fps), or frame rate, refers to how many times per second the scene is updated by the graphics processor. Higher frame rates yield smoother, more realistic animation. It is generally accepted that 30fps provides an acceptable level of animation, but increasing the performance to 60fps results in significantly improved interaction and realism. Beyond 75fps it is difficult to detect any performance improvement. Displaying images faster than the refresh rate of the monitor results in wasted graphics computing power, because the monitor is unable to update its phosphors (or display) that fast, wasting frame rate beyond its refresh rate.

Memory Bandwidth

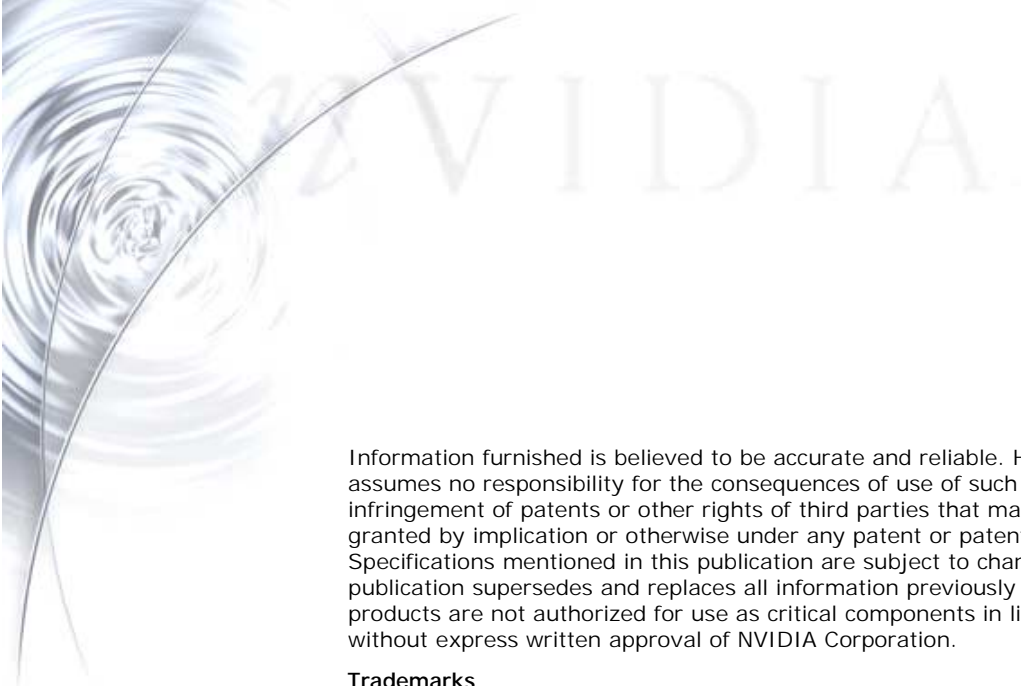
Memory bandwidth refers to the rate at which data is transferred between the graphics processor and graphics memory. Memory bandwidth limitations are one of the key bottlenecks that must be overcome to deliver truly realistic 3D environments. To deliver truly stunning 3D requires high-resolution, 32-bit color depth at high frame rates, with rich geometry, sophisticated texture mapping, and complex vertex and pixel shading.

Resolution

Resolution is the number of pixels on a screen. Higher resolutions can create a more realistic 3D environment because more scene detail can be displayed. Most modern displays are capable of at least 1280 horizontal pixels x 1024 vertical pixels, while many larger or more expensive displays are capable of 2048x1536 pixels. Most graphics applications support a variety of resolutions, allowing the end user to run at higher resolutions (and hence higher level of detail) with the trade-off being increased load on the graphics processing system.

Texture Mapping

Texture mapping is the technique of projecting a 2D image (typically a bitmap) onto a 3D object. Texture mapping allows substantial increases in visual detail without significant increases in polygon count. Because of the improved realism that can be obtained with a very small increase in computational cost, texture mapping is one of the most common techniques for displaying realistic 3D objects. In order to render a texture-mapped pixel, the texture data for that pixel needs to be read into the graphics processor, consuming memory bandwidth.



Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. NVIDIA Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

NVIDIA, GeForce4, and the NVIDIA logo are trademarks of NVIDIA Corporation.

Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

Copyright NVIDIA Corporation 2002.



NVIDIA Corporation
2701 San Tomas Expressway
Santa Clara, CA 95050
www.nvidia.com