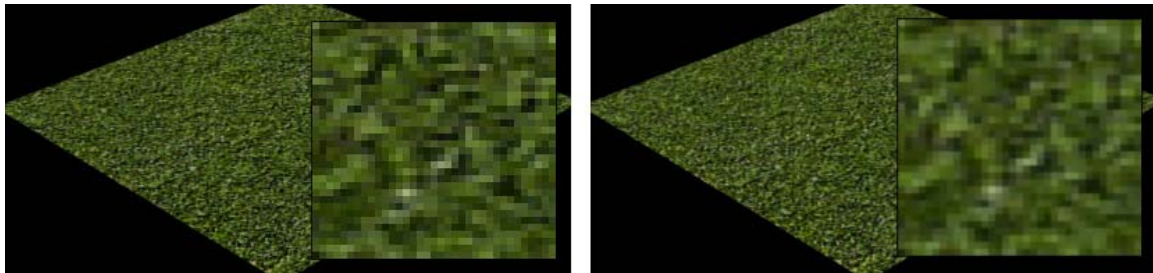


Antialiasing and its Side Effects

In the 3D world, render engines use a process known as SuperSampling to improve the quality of rendered output by decreasing the effect of aliasing. Aliasing is a term to describe imperfections in the rendering process caused by color changes that are too drastic, and that occur over too small an area of screen space to be adequately depicted by the pixels that define that space. In a single still image, aliasing is certainly noticeable and can reduce realism, but if the same scene is set in motion, the results can be downright distracting and can ruin what would otherwise be a great animation. Two examples of these antialiasing side effects are flickering and texture crawling. The following discussion explains this problem further and discusses a unique and effective way of reducing these unwanted effects for any animation. The results are sure to amaze.

Flickering

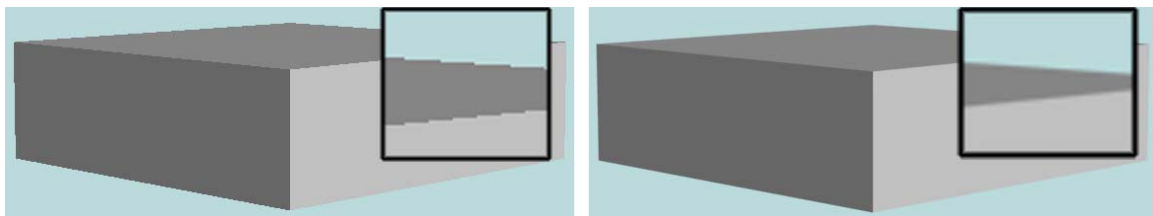
Flickering occurs during animations when materials, such as the grass shown below, have a speckled appearance with drastic changes in color occurring over a small number of pixels. In these cases, the computer cannot accurately determine which color to assign a given pixel, because the edge of two colors exists in the middle of a pixel. In the left image below, supersampling is disabled, and the contrast in adjacent pixels is harsher than in the right image, in which supersampling is enabled. When scenes with these textures are animated, the difference between the two can be dramatic.



Supersampling disabled (left) and enabled (right)

Texture Crawling

Texture crawling occurs in animations in areas along the edges of lines and polygons. This undesirable effect happens because there are not enough pixels available on a typical monitor to properly display mathematically smooth lines and polygon edges. When a 3D scene is transposed onto a monitor's pixel grid, or raster, each pixel is colored according to whether or not it is covered by an object in the scene. Aliasing occurs because the raster system does not properly handle the case in which a pixel is only partially covered. Partially covered pixels occur along the edges of objects and are referred to as edge pixels or fragment pixels. The mishandling of fragment pixels results in harsh, jagged color transitions between an object's edge and the background. Examples of materials in which texture crawling is prevalent are bricks in a wall, pavers in a walkway, and fascia on a roof. In the left image below, anti-aliasing is disabled and the edges are much more jagged than in the right image, in which anti-aliasing is enabled. In animations, these edges appear to crawl whether they are the edges of an object or the edges within a material applied to an object.



Anti-aliasing disabled (top) and enabled (bottom)

Anti-aliasing techniques attempt to smooth jagged edges by properly handling fragment pixels (i.e., adjusting the pixel color according to the amount of pixel coverage). 3ds Max incorporates anti-aliasing filters in the rendering process by default, and although they greatly improve renderings and animations, in some scenes the standard filters aren't enough. Supersampling is an additional anti-aliasing process that samples the

colors around a pixel and uses that information to determine the final color of the pixel. Supersampling can be initiated for specific materials or for the entire scene with the default scanline renderer. Four supersampling methods are available with 3ds Max, but each one slows the rendering process to some degree. Other render engines employ forms of supersampling with numerous settings that never use the term 'supersampling'.

Using SuperSampling can greatly increase render times, often with negligible improvements. Another entirely different way to perform supersampling, and one which we use at 3DAS when time and budget allows for it, is rendering a scene at a higher resolution than what the final output is actually going to be, and then scaling it down to that final output. This is, in effect, a way of performing supersampling yourself. Movie studios use this same process to deliver unbeatable quality with absolutely minimal antialiasing side effects. Some studios render animations out at resolutions well above high definition and the results are stunning. To demonstrate this, we return once again to the animation of the restaurant used in the 3D Site tutorial.

The first image below shows a screen shot of the animation rendered at 640x480 with the **Video** filter. As you can see from the animations, there are numerous areas where both flickering and texture crawling are prevalent. Notice how the shadows from the street lamps almost blink on and off in some areas. The second image shows a screen shot of the same animation rendered with the same render filter (Video). This animation is not only devoid of most of the antialiasing side effects of the first animation, it is far sharper and more aesthetically appealing. The only thing that was done differently with this animation was that it was rendered at a higher resolution (1280x720). Although this obviously increases render times, to get the same quality in a lower resolution, you will have to improve settings in ways that may very well increase render times far more than rendering at a slightly higher resolution. Even then, it may not be possible to achieve this same level of quality regardless of how much you improve your settings.

One additional, and important note, is that you have to pick your antialiasing filter wisely. While using a filter such as **Catmull-Rom** or **Mitchell-Netravali** works great for stills, these filter types are sure to maximize the effects of flickering and texture crawling. When you create animations, you must use other filters that blur the rendered output, thereby, minimizing these effects. Two such filters are the default **Area** filter and the **Video** filter. Remember however that the more blurred a filter, the greater the rendering time. At 3DAS, we try to use the Area filter to create animations (whenever possible) because using the Video filter can easily add 25% to your final rendering times. However, if the Area filter causes too much flickering or texture crawling, we must either use the slower (and more blurry) Video filter or render at a higher output.

[Click to play Quicktime animation \(23MB\) – Animation rendered at 640x480](#)



[Click to play Quicktime animation \(35MB\) - Animation rendered at 1280x720 and scaled down](#)

