Texto construído com base nas seguintes URLs:

1) http://games.tecmundo.com.br/especiais/antialiasing-supersampling_161450.htm
2) http://games.tecmundo.com.br/especiais/antialiasing-multisampling-e-alphatocoverage_161618.htm
3) http://games.tecmundo.com.br/especiais/antialiasing-coverage-sampling_161396.htm
4) http://www.nvidia.com/object/coverage-sampled-aa.html

1 Super-Sampling

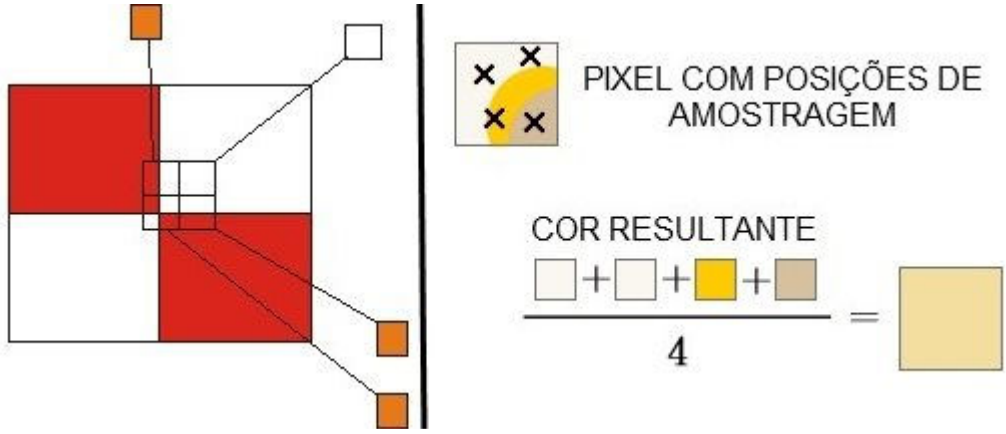
O Super-Sampling Anti-Aliasing — ou simplesmente SSAA — é uma técnica específica de ***anti-aliasing***, como o **Multi-Sampling** (MSAA). É uma forma alternativa, portanto, de aplicar correções de bordas serrilhadas com a suavização de imagens renderizadas (desenhadas) em programas e games de diversas plataformas. O SSAA surgiu pela primeira vez nas séries 4/5 das placas de vídeo conhecidas como **Voodoo** (abaixo, uma 3DFX Voodoo 5 5500).

1.1 Como funciona?

No SSAA, são coletadas amostras de várias instâncias — e não necessariamente do centro, como é o padrão — do pixel manipulado. Em seguida, é calculada uma “média” entre as cores de cada amostra. Isso é obtido pela renderização interna da imagem abordada em uma resolução maior do que aquela em que a figura está sendo visualizada.

Depois disso, a imagem passa pelo processo de *downsampling* — literalmente um “encolhimento” — para o tamanho desejado, e os pixels extras são utilizados para os cálculos. O resultado final é uma série de transições mais suavizadas de uma linha de pixels para outra nas bordas dos objetos.

Os modos específicos de SSAA são nomeados de acordo com a resolução usada internamente. Por exemplo: 2x1 significa que a resolução interna duplicou a resolução horizontal e manteve a resolução vertical. O resultado é que cada pixel final é feito a partir da média de dois pixels. Já no modo 4x4 cada pixel final é criado a partir da média de 16 pixels.



A imagem à esquerda mostra o que ocorre quando a técnica de Super-Sampling — “superamostragem” — é usada em quatro quadrados na posição central. Três dos quatro quadrados estão em ambas as áreas vermelha e branca. Sendo assim, eles serão exibidos com um matiz laranja.

Um dos quadrados está inteiramente localizado na região branca, portanto o pixel será mostrado como puramente branco. Se uma das amostras fosse retirada de uma área inteiramente vermelha, o pixel seria mostrado somente na cor rubra. É óbvio que as amostras são muito mais complexas que isso, possuindo diversas cores para o cálculo da média. No entanto, o princípio é o mesmo. Matematicamente, a cor de cada pixel é calculada.

Como de praxe, o número inicial de amostras coletadas determina a qualidade do resultado. É por isso que é muito comum, em jogos e programas para computadores, encontrar telas de configurações gráficas com diferentes opções de SSAA de acordo com esse indicador de amostragem.

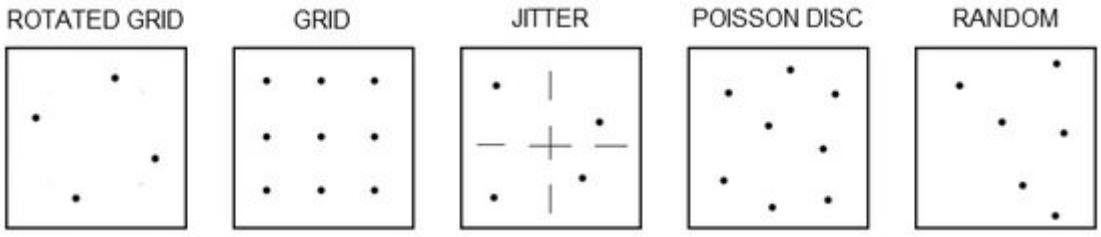
Em games para consoles, entretanto, são as desenvolvedoras dos títulos que decidem qual o nível de SSAA a ser utilizado pela placa de vídeo da plataforma em questão. É claro que é preciso sabedoria e experiência para conciliar desempenho e qualidade visual, embora tenha sido comprovado que o SSAA é um dos métodos menos eficientes (na relação custo/benefício) para o uso em consoles.

1.2 Algoritmos de Super-Sampling

Há vários tipos de algoritmos empregados para a aplicação do SSAA. Todos eles variam de acordo com o local onde as amostras extras são coletadas dentro do pixel.

O OGSS (Ordered Grid Super-Sampling ou simplesmente Grid) é o algoritmo mais simples. O pixel é dividido em vários subpixels e uma amostra é retirada do centro de cada um deles. O Grid é fácil e rápido de ser “implementado”. Mesmo assim, devido à natureza regular da amostragem, bordas serrilhadas podem continuar ocorrendo se um baixo número de subpixels for utilizado.

Já o Random (Aleatório, também chamado de Stochastic Sampling) foge da regularidade do Grid. Contudo, a irregularidade do código faz com que as amostras acabam se tornando desnecessárias em certas áreas do pixel e estejam em falta em outras.



O Poisson Disc também coleta as amostras aleatoriamente, mas marca as duas que não estão tão próximas da área principal. A distribuição das amostras, portanto, é aleatória, porém o resultado final fica uniforme. A desvantagem? O tempo computacional requerido para o uso do algoritmo PD é muito extenso.

O algoritmo conhecido como Jittered é uma modificação do Grid que se aproxima do PD. Aqui, um pixel é dividido em vários subpixels e a amostra não é

retirada do centro de cada um, mas sim de um ponto aleatório dentro de cada subpixel. A “congregação” ainda pode ocorrer, mas em uma escala menor.

Por fim, o Rotated Grid — RGSS — emprega um layout 2x2, mas o padrão de amostragem é girado para evitar que as amostras fiquem alinhadas aos eixos horizontal ou vertical. Isso melhora consideravelmente a qualidade do *anti-aliasing* nos casos encontrados mais comumente no universo gráfico.

1.3 Prós e contras

Em termos de desenvolvimento, o SSAA é uma técnica computacionalmente “cara”, pois requer muito mais memória de vídeo e banda de memória, levando em consideração que o buffer (registro) utilizado é bem maior que o “normal”.

Uma possível solução para esse problema é o Adaptive Super-Sampling, uma técnica que reconhece que são bem poucos os pixels presentes em um limite de objeto e, por causa disso, apenas esses pontos devem passar pelo Super-Sampling.

Com esse método, apenas algumas amostras são retiradas de um pixel. Se os valores dessas amostras forem muito similares, apenas elas serão utilizadas para a determinação da cor final. Caso contrário, mais *samples* são usadas. Sendo assim, um número elevado de amostras é calculado apenas quando necessário, o que melhora o desempenho da aplicação.

De forma geral, o SSAA apresenta uma excelente qualidade na imagem final, suaviza corretamente até mesmo as bordas localizadas dentro de texturas *alpha-blended* — processo de combinação entre cores frontais translúcidas e cores de fundo — e também reduz o Texture Shimmering (efeito no qual o jogador tem a impressão de que as texturas de certos objetos estão “caminhando” junto com a câmera).

Por outro lado, o desempenho é afetado consideravelmente com o uso do SSAA. Enquanto o modo 2x1 faz o processador gráfico “se esforçar” duas vezes mais, o 2x2 demanda quatro vezes o poder de processamento e o 4x4 leva a máquina a trabalhar 16 vezes mais.

Além disso, há limitações na resolução-alvo. A resolução interna máxima é 4096 pixels em ambos os eixos (vertical e horizontal). Portanto, o modo 4x4 pode ser usado apenas em 1024x768 ou menos. O limite para placas de vídeo NVIDIA GeForce 8, por exemplo, é 2048x1536. Outro aspecto negativo é que certos modos SSAA só ficam disponíveis com o uso do Direct3D e não aparecem em programas com base na tecnologia OpenGL.

É importante considerar que, para boa parte dos jogos eletrônicos, a queda no desempenho gerada pelo SSAA não é aceitável se a resolução utilizada for 1280x1024 ou mais. No entanto, usar certos modos de SSAA durante a experiência com um game mais antigo — com o auxílio de um potente chip gráfico, obviamente — pode gerar uma melhoria incrível na qualidade visual sem diminuir drasticamente as taxas de quadro por segundo (fps).

2 Multi-Sampling e Alpha-to-Coverage

O Multi-Sampling é uma das técnicas de *anti-aliasing* (o processo que suaviza serrilhados na imagem). Ele foi introduzido com a série GeForce 3 de placas de vídeo da NVIDIA, sendo considerado como de segunda geração, e atualmente é utilizado pela maioria dos jogos.

O MSAA funciona da seguinte maneira: a placa de vídeo processa a imagem, verifica em que pontos é necessário eliminar serrilhados (geralmente nas bordas dos objetos) e aplica o filtro a eles, ignorando boa parte do restante.

Para que isso ocorra, algoritmos calculam a diferença entre a profundidade dos polígonos e determinam, por exemplo, quais porções da imagem estão contidas em um mesmo objeto, quais pixels e valores precisam ser filtrados e assim por diante. O processo é muito mais inteligente que o do Super-Sampling Anti-Aliasing (SSAA).

Para fins de comparação, o SSAA renderiza toda a imagem em uma resolução altíssima, não se importando com questões como desempenho ou necessidade de suavização de serrilhados. Se um ponto está na imagem, ele é processado individualmente e depois reescalonado para a resolução do monitor.

O Multi-Sampling, por sua vez, reconhece pixels idênticos em uma região e os filtra como um único conjunto, aplicando o mesmo valor de cor a todos. O resultado imediato é um ganho enorme de desempenho, que pode ser aproveitado pela placa para rodar com folga cenas mais complexas — seja em termos de geometria dos cenários, qualidade de texturas ou até mesmo melhoras nos efeitos visuais.

Embora alguns jogos não tragam opções para a ativação do Multi-Sampling Anti-Aliasing, os jogadores podem recorrer diretamente aos drivers das placas de vídeo e ativar o recurso manualmente.

2.1 Amostras, amostras...

Como funciona a filtragem da imagem? Uma vez definida a região em que deve ser aplicada a suavização de serrilhados, inicia-se o processo de filtragem. São

colhidas amostras de cores a partir dos pixels da imagem. Essas amostras são calculadas e recombinadas, formando transições mais graduais entre as diferentes cores. Como resultado, o jogador não vê serrilhados, mas sim uma linha suave.

O número de amostras colhidas é determinado pelo jogador (no caso de jogos para computador) ou pelos desenvolvedores (em consoles), e é diretamente proporcional à qualidade da imagem. Isso explica porque há 4xMSAA, 2xMSAA, entre outros.

Em uma explicação mais técnica, as amostras colhidas a partir do centro de um pixel têm mais peso do que aquelas obtidas a partir dos cantos dele, para a criação dos novos valores de cor. De acordo com documentos da NVIDIA, o centro equivale a 4/8 do cálculo, enquanto cada um dos cantos do pixel tem peso de 1/8 para a equação.

2.2 Alpha-to-Coverage: uma solução elegante

O MSAA não é isento de falhas. Por limitar a filtragem apenas a áreas externas dos polígonos, as texturas internas não recebem nenhum tipo de tratamento. Isso não é problema quando o objeto recriado com os polígonos é sólido, tal como uma pedra ou uma parede de cenário.

No entanto, quando desenvolvedores utilizam texturas transparentes (chamadas *alpha*) para criar coisas como cercas metálicas, folhagem, árvores e portões, o serrilhado persiste, incomodando os olhos do jogador e reduzindo consideravelmente a qualidade de imagem. É como se essas texturas caminhassem junto com a câmera, criando um defeito conhecido como Texture Shimmering.

Para que fique claro, os objetos citados acima são muitas vezes construídos apenas por texturas para poupar o poder de processamento das placas, em vista da complexidade bem reduzida. Exemplos do problema são ilustrados na imagem abaixo. Reparem especialmente nas pontas dos galhos das árvores e na folhagem, que ganham definição e perdem o visual pontilhado:

A solução está em estender a aplicação do MSAA para as texturas, através do Alpha-to-Coverage ("Alfa para Cobertura"). De um modo simplificado, o Alpha-to-Coverage faz a amostragem das texturas transparentes (assim como para um polígono) e interpreta o quanto elas cobrem um pixel da imagem. O resultado é

recombinado e aplicado à cena.

O processo, tecnicamente, é bem mais complexo, envolvendo a criação de uma série de máscaras de filtragens e a recombinação delas na imagem final. Ainda que esta filtragem das texturas tenha um peso considerável, ela ainda é muito mais leve do que aplicar SSAA aos cenários.

O resultado também é quase impecável (ficando as texturas um pouco mais borradas em suas bordas, em comparação à aplicação de SSAA). Para corrigir isso, alguns desenvolvedores optam por acrescentar um multiplicador de contraste para essas texturas, na tentativa de torná-las mais visíveis.

De um modo sucinto, o MSAA é uma técnica de suavização de serrilhados inteligente, que reconhece quais porções da imagem necessitam da aplicação do recurso, apresentando desempenho muito superior ao método SSAA tradicional.

A falha do MSAA na leitura de texturas alpha pode ser corrigida por meio do uso da técnica Alpha-to-Coverage, resultando em uma qualidade de imagem equiparável à apresentada pelo SSAA, ainda que com um desempenho vastamente superior.

3 Coverage Sampling

Coverage Sampling Anti-Aliasing (CSAA) é uma técnica de anti-aliasing — como o MSAA e Alpha-to-Coverage —, ou seja, um processo de suavização dos serrilhados na imagem. O CSAA é praticamente uma variação do MSAA, salvo pelo número de amostras de cobertura — que no Coverage Sampling está desatrelado dos dados de cor e profundidade.

Como resultado, o CSAA oferece imagens com qualidade similar às tratadas com MSAA 8x ou 16x. Além disso, por utilizar um número menor de amostras, o CSAA também exige menos desempenho da máquina.

O funcionamento do CSAA não é muito diferente do MSAA: a placa de vídeo processa a imagem e aplica o filtro nos pontos em que é necessário eliminar serrilhados. Isto significa que o processo ignora boa parte da imagem e se concentra somente nos elementos relevantes (as bordas de objetos).

Para tanto, o algoritmo do Coverage Sampling analisa diferenças de valores de cor dos polígonos, determinando assim quais porções da imagem estão contidas em um mesmo objeto. O CSAA reconhece pixels idênticos em uma região e os filtra como um único conjunto, aplicando o mesmo valor de cor a todos.

As minúcias técnicas do processo estão nos cálculos das amostras colhidas da imagem. O CSAA atribui novos valores de cor aos pixels, que na imagem final aparecem com bordas suavizadas. O jogador, ou os desenvolvedores, designam a quantidade de amostras que serão colhidas, sendo que estas são diretamente proporcionais à qualidade final da imagem.

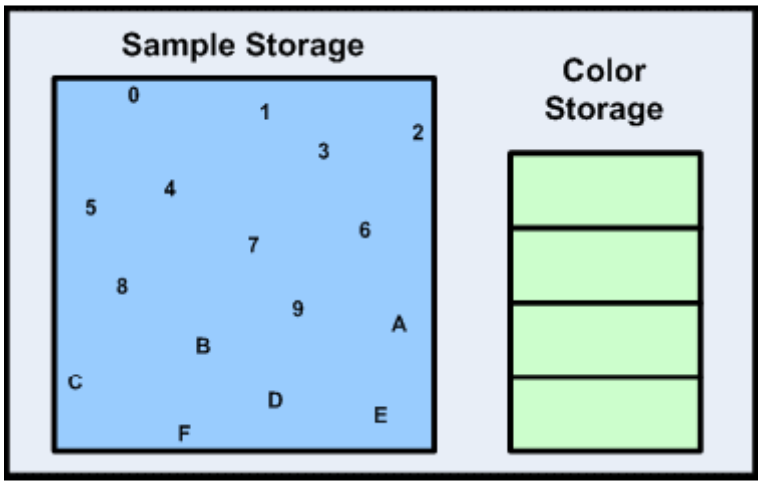
As amostras de cores/Z/estêncil são traduzidas em trechos a serem “corrigidos”. O esquema é menos complicado do que parece: basicamente o algoritmo busca por padrões de cor, profundidade (z) e forma (estêncil). Quando dizemos CSAA 8X, isto significa que serão utilizadas oito amostras para identificar a imagem, enquanto o CSAA 16x terá 16 amostras.

Nas placas de vídeo NVIDIA GeForce 8800, os modos CSAA apresentam os seguintes valores:

CSAA	Amostra de Cores/Z/Estêncil	Amostras de Cobertura
8x	4	8
8xQ	8	8
16x	4	16
16xQ	8	16

Como no Multi Sampling, o Coverage Sampling reduz o impacto da operação no processamento gráfico ao separar as amostras sombreadas de cor (Color Sample) e de cobertura (Coverage Sample). Assim, mesmo, com menos instruções de renderização (shaders), o processo não perde amostras de cobertura cor/z/estêncil.

No modo CSAA 16x, as cores e a cobertura estão dispostas dentro do pixel da seguinte forma:



Resumindo, o CSAA comprime as informações de cor/z/estêncil “dentro” do rastro de memória e banda utilizada pelo MSAA 4x. Em seus padrões de alta qualidade (CSAA 8xQ e 16xQ), o Coverage Sampling comprime estas informações nas “pegadas” do MSAA 8x.

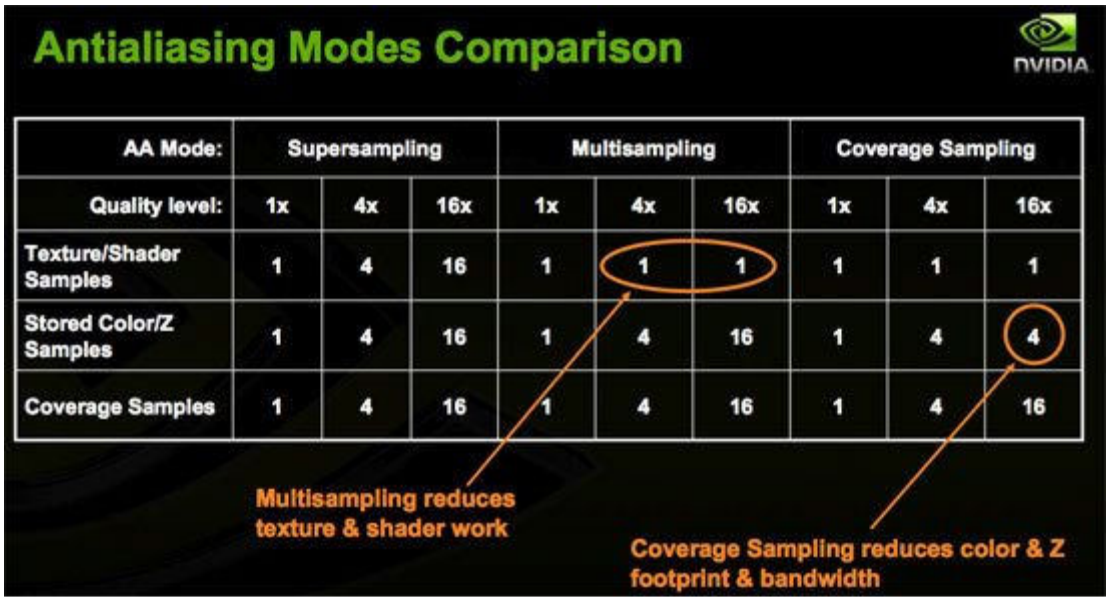
3.1 .Desempenho

O grande trunfo do CSAA em relação ao MSAA é o seu baixo “custo” de processamento. Apesar das similaridades com o Multi Sampling — já que utilizam a mesma contagem de amostra de cor/Z/estêncil —, o Coverage Sampling é mais eficiente.

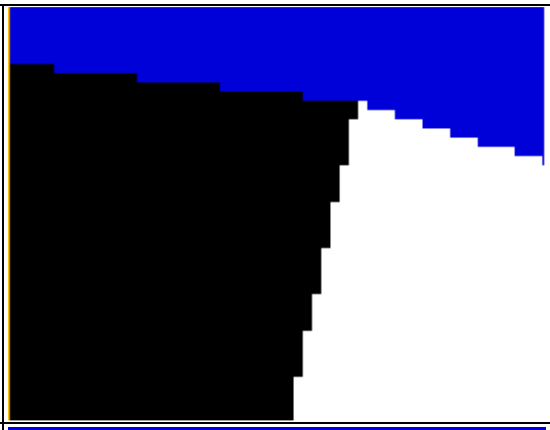
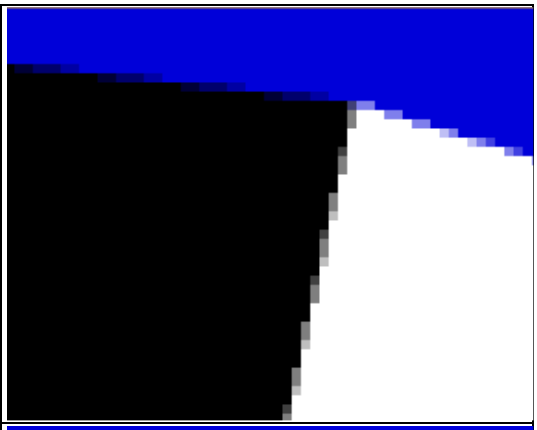
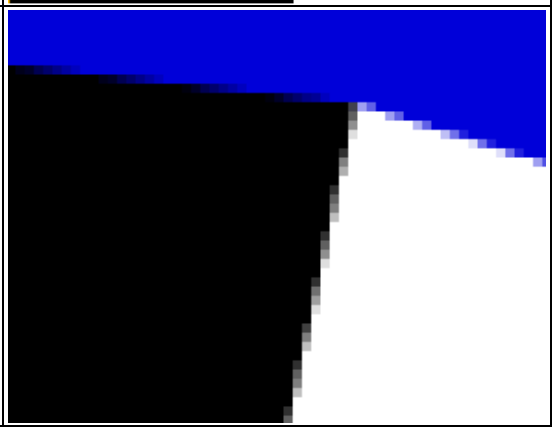
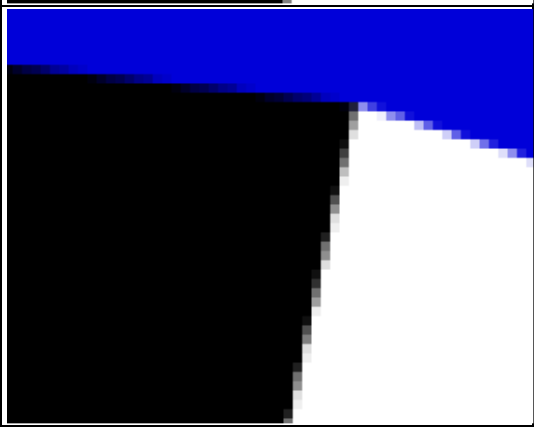
Os modos 8x e 16x do CSAA (que utilizam quatro amostras de cor/Z/estêncil) tem o mesmo comportamento do MSAA 4x. Ou seja, em termos de armazenamento, o CSAA apresenta uma relação menos dispendiosa, pois as amostras de cobertura não necessitam de muito espaço de memória.

Todavia, existem controvérsias sobre os valores reais de amostragem do CSAA. A tabela de valores indica claramente que o CSAA 8xQ é na verdade um MSAA 8x — pois ambos utilizam oito amostras de cor/Z/estêncil. Diferentemente do CSAA 8x que utiliza apenas quatro amostras, mas emprega oito amostras de cobertura.

Mesmo assim, não há como negar a eficiência do CSAA, especialmente em relação aos resultados apresentados e seu baixo impacto no desempenho da máquina. O quadro a seguir compara o número de amostras de coletadas, para diferentes informações, pelos algoritmos de anti-aliasing Supersampling, Multisampling e Coverage Sampling.



As imagens a seguir mostram as diferenças na qualidade de imagens quando usando os algoritmos MSAA e CSAA, comparados à imagem original sem anti-aliasing. Observe a melhoria na qualidade da imagem quando se usa o algoritmo 16x CSAA que, em termos de desempenho computacional, é equivalente ao 4x MSAA.

No AA		4x MSAA	
16x CSAA		16xQ CSAA