

Os efeitos do Aliasing e os principais métodos de soluções Anti-Aliasing

Gabriel Campelo Gomes¹, Vívian Leite Sousa¹

¹Faculdade de Engenharia de Computação – Universidade Federal do Pará (UFPA)
Rua Itaipú, 38 – Vila Permanente – Tucuruí – PA – Brazil

{vivian.eng15@gmail.com, campelojon@gmail.com}

Resumo. Introdução: Descreve detalhes de como as imagens digitais são formadas por pixels e que a resolução dessas imagens causa o efeito de Aliasing. **Referencial Teórico:** Nessa seção é detalhada a base teórica sobre os conceitos de Aliasing e Anti-Aliasing, fazendo uma breve síntese sobre os temas abordados. **Principais Métodos de Anti-Aliasing:** Já nesta seção são apresentados os algoritmos de anti-aliasing, seus desempenhos e diferenças. Também é explícito um levantamento de estudo de casos comparativos entres esses métodos, esclarecendo seus prós e contras. **Conclusões:** O estudo demonstrou como os algoritmos de Anti-Aliasing são importantes nos efeitos visuais de jogos atualmente e que todos os métodos apresentados possuem suas vantagens e desvantagens, a escolha de um método específico depende de como se adapta ao processador do computador utilizado e dos efeitos visuais pretendidos.

Abstract. Introduction: Describes details about as digital images are formed by pixels and are a way of causing the effect of Aliasing **Theoretical Reference:** In this section the theoretical basis on the concepts of Aliasing and Anti-Aliasing is described, giving a brief synthesis on the topics covered. **Main Anti-Aliasing Methods:** Already in this section are presented the anti-aliasing algorithms, their performances and differences. It is also explicit a survey of comparative case study between these methods, clarifying their pros and cons. **Conclusions:** The study demonstrated how Anti-Aliasing algorithms are important in visual effects of actual games and that all presented methods have their advantages and disadvantages, the choice of a specific method depends on how it adapts to the processor of the computer used and visual effects required.

1. Introdução

Quando os primeiros títulos em 3D foram produzidos pelas desenvolvedoras, a quantidade de polígonos era limitada e dava um visual quadriculado aos jogos, porém mesmo que o poder computacional atual consiga gerar uma quantidade de polígonos maior, ainda há como notar os modelos 3D com pequenos quadrados. Isso porque todas as imagens digitais reproduzidas pelos computadores, celulares ou consoles, são uma coleção de pontos de cores. Esses pontos, normalmente, são quadrados e em número tal que constituem a resolução da imagem, eles são chamados de pixels. “Os pixels são às vezes caracterizados como amostras de área, mas como cada pixel é representada por um único valor (por canal de cor), a caracterização da amostra da área não possui uma formalização muito precisa” [Goss and Wu 2000].

A imagem a ser amostrada por estes pequenos quadrados é composta de linhas e polígonos projetados para o plano de visão. Qualquer aresta viva, como uma aresta de polígono ou linha, contém não-zero amplitude em qualquer frequência arbitrariamente alta. Isto implica que não é possível usar amostras para representar exatamente imagens que tenha bordas afiadas (GOSS e WU, 2000). Quando elementos gráficos arredondados ou diagonais são analisados com detalhe na tela, é nítido o serrilhado em suas bordas. Para corrigir este serrilhado temos a opção de aumentar a resolução em pixel, reproduzindo valores de amostra que ocupem uma aproximação razoável da imagem ideal. Porém “Quanto maior for a resolução gráfica da textura, mais pixel dessa imagem precisão ser encaixados em cada pixel da tela e maior será a arbitrariedade da escolha do representante. Texturas de grande definição produzem, dessa forma, imagens com vários problemas de Aliasing”[Cuzziol 2007].

Para isso existem métodos de correção presente em qualquer placa de vídeo, recursos que amenizam o serrilhamento utilizando ilusão ótica e aplicados diretamente nos pixels na tela, resultando em imagens que pareçam mais nítidas, esses métodos são chamados de Anti-Aliasing. Este artigo irá apresentar os efeitos negativos do Aliasing e os principais métodos de Anti-Aliasing utilizados, o documento foi dividido em três partes: A primeira apresenta os conceitos de Aliasing e de Anti-Aliasing. A segunda parte explicita os principais métodos Anti-Aliasing e suas performances. E por fim a terceira parte é uma breve conclusão sobre esses métodos de forma comparativa

2. Referencial Teórico

2.1. Aliasing e seus efeitos

“O aliasing é um dos problemas mais antigos na computação gráfica, produzindo artefatos claros em imagens estáticas(aliasing espacial) e introduzindo animações de cintilação aliasing(temporal) [JIM]”. Uma vez que a divisão mínima num monitor é de pixels, ao se desenhar uma reta inclinada surge o aparecimento do efeito de serrilhamento ao longo da reta, este efeito é conhecido como Aliasing. “Um fenômeno muito comum em amostragem é o fenômeno de aliasing. Ele acontece quando realizamos uma amostragem muito pobre da função contínua original e quando vamos reconstruí-la obtemos resultados inconsistentes com o esperado[Scuri 2002].”

Podemos notar este fenômeno em algumas situações, segundo Goss e Wu,[2000, p.3] “Nos gráficos de varredura, o aliasing é mais notável como bordas irregulares de animações quando os padrões da escada rastejam ao longo dessas bordas. O aliasing também aparece como Mudando padrões moiré durante a animação de texturas periódicas finas ou estruturas geométricas”.

O efeito Aliasing é muito comum durante o desenho de primitivas. A memória de vídeo da placa gráfica pode ser vista como uma imagem e uma primitiva a ser desenhada como um objeto real a ser amostrado. Como as primitivas possuem bordas abruptas (alta frequência) aliasing aparece naturalmente[Scuri 2002].

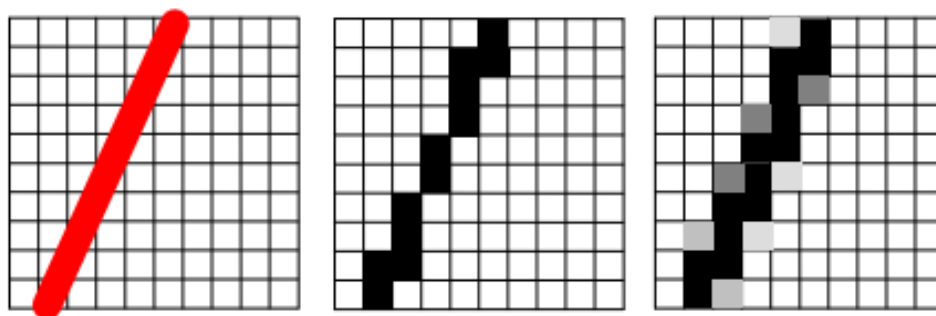


Figure 1. Anti-Aliasing em Desenho de Primitivas

Fonte:[Scuri 2002]

2.2. Soluções Anti-Aliasing

Os processos que reduzem os efeitos de Aliasing são conhecidos como Anti-Aliasing. Para superar esse problema, é proposto desfocar seletivamente os pixels que estão preocupados com o Aliasing. O processo de difusão está entre as melhores estratégias encontradas na literatura para desfocar seletivamente uma imagem [Ziou and Horel 2011]. Basicamente, a solução de minimizar o efeito de serra trabalhando com os pixels vizinhos da área afetada da imagem de forma a criar uma suavização entre as cores na área, alterando as tonalidades das cores dos pixels onde se é necessário.

No entanto, aplicar este tipo de tratamento a imagem com muitos fotogramas por segundo não é tão simples, e neste sentido surgiu vários métodos Anti-Aliasing para melhorar a qualidade dessas imagens, mas que demandam de mais poder computacional e tempo de processamento. E neste artigo abordaremos alguns métodos e os impactos que eles tem sobre a performance de um computador.

3. Principais Métodos de Anti-Aliasing

3.1. MSAA(Multisampling AntiAliasing)

Antes de apresentar o Multisampling, é preciso comentar sobre um método antigo de anti-aliasing chamado Supersampling. Ele trabalha com um conceito simples: em vez de aplicar a cor detectada no centro do pixel, várias amostras de cores dentro desse pixel são calculadas. Apesar de ter ótimos resultados, é uma técnica cara em termos de processamento e largura de banda de memória, pois a placa gráfica está fazendo todo o trabalho necessário para renderizar cada quadro em uma resolução mais alta.

Então o Multisampling (MSAA) surgiu como uma alternativa entre um equilíbrio entre a qualidade e a carga de trabalho do processamento. 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. Este processo é mais inteligente que o Super-Sampling, pois a placa de vídeo processa a imagem, verifica em que pontos é necessário eliminar serrilhados (geralmente nas bordas dos objetos) e aplica



Figure 2. Comparação de efeitos MSAA

Fonte:[Bigogno]

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. [Woligroski]

3.2. FXAA(Fast approximate Anti-Aliasing)

O método FXAA (Fast approximate Anti-Aliasing) é um algoritmo desenvolvido pela NVIDIA para cobrir as deficiências do MSAA, ele faz uma aproximação de maior desempenho dos efeitos do Anti-Aliasing multi-amostragem tradicional. “ É um sombreador de pixel de passagem única que é executado no estágio de pós-processamento do pipeline de renderização do jogo de destino. Ele foi projetado para ser mais rápido e ter uma pegada de memória menor do que o MSAA tradicional, mas com alguns compromissos de precisão e qualidade” [Warner].

O FXAA apresenta vantagens como a redução de aliasing especular, mantendo a nitidez dentro de um custo para processar um quadro de 1920x1200 no GTX480 que é inferior a 1ms para o pré ajuste FXAA2. E além de reduzir aliasing de um pixel único também faz redução de aliasing de subpixel. Também pode fornecer vantagem de memória sobre o MSAA, principalmente em alvos de renderização estéreo ou de vários monitores ou buffers de retorno [Lottes 2011].

3.2.1. SMAA (Sub-pixel Morphological Anti-Aliasing)

O SMAA (Sub-pixel Morphological Anti-Aliasing) é um método de anti-aliasing pós-processamento, assim como o FXAA, e os dois métodos quando aplica-



Figure 3. Comparação de efeitos FXAA

Fonte:[Bigogno]

dos não aparentam tanta diferença, mas o SMAA tira da imagem o efeito de desfoque e neblina. o SMMA é “uma técnica que aborda todos os pontos fracos restantes em soluções de anti-aliasing baseadas em filtros[...] combinar um filtro baseado técnica de anti-aliasing com multi padrão/abordagens supersampling e reprojeção temporal. Esta nova combinação de estratégias melhoradas de MLAA com as variáveis espaciais e temporais”[JIM].

Este processo pode ser aplicado em três modalidades: SMAA baixo(1x), SMAA médio (2tx) e SMAA alto (4x). O SMAA 1x oferece gradientes muito precisos, estabilidade temporal e robustez, ao introduzir uma sobrecarga mínima, uma boa escolha para configurações low-end. Já o SMAA T2x oferece uma atractiva tradeoff para qualquer tipo de motor de renderização, e por fim o SMAA 4x é a melhor opção em relação à qualidade de imagem.

3.2.2. TXAA (Temporal Approximate Anti-Aliasing)

O TXAA foi projetado para ser integrado diretamente aos mecanismos de jogos. Além disso, combina a força bruta do MSAA com um filtro temporal e sofisticados filtros de resolução semelhantes aos usados em filmes CG. Quando a velocidade do objeto é maior que a do frame-rate do dispositivo, isso faz com que alguns frames sejam perdidos, dando a sensação de que esse objeto parece que está pulando na cena, quando não está. O TXAA resolve esse problema interpolando os frames perdidos na tela, dando a sensação de continuidade e não perdendo a informação. O resultado final é uma imagem mais suave



Figure 4. Comparação de efeitos SMAA

Fonte:[Bigogno]

e realista que qualquer outra técnica comparável.

3.3. Levantamento de estudos de casos com os métodos anti-aliasing

Para análise dos métodos aqui apresentados foi feito um levantamento bibliográfico de estudos de casos, onde foi previsto o impacto tanto em processamento e em qualidade de imagens depois de aplicado o método para a correção. Segundo a análise de Marcos Armão Ferreira [Ferreira 2014], o FXAA é um método anti-aliasing obtido quase sem penalizações nas performances da placa gráfica, no entanto, está longe de ser eficaz e produz um efeito na imagem de desfocagem apesar de não ser algo extremamente notório, funcionando bem apenas em imagens estáticas. Já o SMAA não cria distorções nas imagens, mas possui uma maior penalização na performance da placa gráfica comparado ao FXAA, mas que entre suas modalidades, o SMAA alto (4x) é o mais eficaz de todos os três métodos apresentados dentro de suas modalidades. E o MSAA é um método anti-aliasing bem eficaz e sem qualquer distorção de imagens, porém apresenta penalizações nas performances e possui problemas nas zonas de transparências, a não ser quando utilizadas nas versões otimizadas pela Nvidia ou ATI. Por fim, o TXAA não diminui o desempenho da placa e funciona bem tanto com imagens estáticas e dinâmicas. Mas devido à imprecisão da imposição temporal, especialmente na dinâmica, a imagem é desfocada e requer processamento adicional de imagem pelo filtro de Nitidez. Embora este seja um efeito pós, é muito mais caro do que o mesmo algoritmo SMAA devido à necessidade de renderizar o Motion Vectors Buffer para todos os objetos na imagem.

4. Conclusões

Este artigo apresentou uma abordagem teórica sobre os efeitos negativos do Aliasing e os métodos de correções chamado Anti-Aliasing. Apesar da variância de algoritmos que realizam o processo de anti-aliasing, na seção 3 foram apresentados os três principais métodos mais utilizados, especialmente em jogos, e foi feito um levantamento



Figure 5. Comparação de efeitos TXAA

Fonte:[Bigogno]

bibliográfico sobre o estudo de casos de cada um deles, avaliando como eles se sobressaem em relação ao impacto que causam na performance da placa gráfica e a eficácia de seus processos.

Todos os métodos apresentados possuem seus prós e contras, a escolha de um método específico depende como se adapta ao processador da máquina utilizada e dos efeitos visuais pretendidos, mas é válido resumir os aspectos que se destacam de cada um deles. Se a placa gráfica do computador utilizado não possuir bom desempenho, então o mais indicado é o algoritmo FXAA. Mas caso o sistema for mais potente e o MSAA não estiver disponível, pois renderizado a textura, o SMAA é uma escolha a se considerar, é colocado em evidência sua modalidade SMAA 4x como melhor opção. Se a placa gráfica for rápida e tiver lidado com a técnica SMAA, vale a pena escolher o TXAA.

5. Referências

References

31(2).

Bigogno, M. Anti-aliasing: conheça o efeito responsável por melhorar sua jogatina.

Cuzziol, M. F. (2007). Games3d: Aspectos de desenvolvimento. *Universidade de São Paulo: Escola de Comunicação e Artes*, 1(1):1–95.

Ferreira, M. A. (2014). Os tipos de anti-aliasing existente e a sua diferença.

Goss, M. E. and Wu, K. (2000). Study of supersampling methods for computer graphics hardware antialiasing. *Visual Computing Department Hewlett-Packard Laboratories*, 1(1):1–37.

Lottes, T. (2011). *FXAA*. Nvidia, 1th edition.

Scuri, A. E. (2002). *Fundamentos da Imagem Digital*. Tecgraf/PUC-Rio, 1th edition.

Warner, M. Nvidia's new fxaa antialiasing technology.

Woligroski, D. Análise do anti-aliasing, parte 1: configurações e surpresas.

Ziou, D. and Horel, A. (2011). Reducing aliasing in images: a pde-based diffusion revisited. *Elsevier*, 1(1):1–15.