CENTRO UNIVERSITARIO
DE TECNOLOGÍA Y ARTE DIGITAL

Estudio sobre sistemas de anti-aliasing e implmenetación de anti-aliasing temporal

Hugo Ferrando Seage

U-Tad

septiembre 2018

Tutor: Alberto Sánchez Campos

Resumen

Debido a la resolución limitada de las pantallas al rasterizar gráficos 3D se deforman ciertas lineas y curvas. Este fenómeno se llama aliasing. La demanda de gráficos cada vez más realistas ha propiciado la creación de diferentes técnicas y algoritmos para disimular estos defectos, sin tener que recurrir necesariemente a pantallas de mayor resolución.

El antialiasing se ha vuelto una técnica crucial para mejorar la calidad de imagen en el software con gráficos tridimensionales. Este campo de la computación gráfica lleva años desarrollandose, con nuevas técnicas publicadas continuamente.

Este proyecto tiene como objetivo el análisis de las distintas técnicas existentes en el estado del arte y la implementación de un algoritmo de temporal antialiasing, compatible con motores con deferred shading y comparable en calidad al multisampling, junto a otras propiedades.

Abstract

When rasterizing 3D graphics using computer screens, due to their limited resolution some lines and curves will become deformed. This phenomena is called aliasing. The ever increasing demand for more realistic graphics has driven the creation of new techniques and algorithms to hide these artifacts, without necessarily using higher resolution screens.

Antialiasing has become a crucial technique to advance image quality in graphics software. This field has been explored for years, with new approaches being developed continually.

This projects aims to analyze different antialiasing techniques developed and the implementation of a temporal antialiasing technique, compatible with deferred shading engines and with similar quality to mutlisampling, amongst other benefits.

Índice general

1. Introducción **11**

1.1. ¿Qué es el aliasing? 11

1.2. ¿Qué es el antialiasing? 11

2. Planteamiento del problema **13**

2.1. Aliasing en bordes de geometría 13

2.2. Aliasing de texturas 13

2.3. Specular Aliasing 13

2.4. Motion Aliasing 13

3. Objetivos **15**

4. Estado del Arte **17**

4.1. Filtrado 17

4.2. Supersampling Antialiasing 17

4.2.1. Ordered Grid Supersampling 19

4.2.2. Rotated Grid Supersampling 19

4.2.3. Sparse Grid Supersampling 19

4.3. Multisampling 19

4.4. Post Processing Antialiasing 19

4.4.1. Fast Approximate Anti-aliasing 19

4.4.2. Morphological Antialiasing 20

4.4.2.1. Enhanced Subpixel Morphological Antialiasing 20

4.4.3. Temporal Antialiasing 20

5. Desarrollo **21**

5.1. Jitter 21

5.2. Motion Vectors 21

5.3. Combinación de texturas 21

5.4. Neighborhood Clamping 21

5.5. Sharpening kernel 21

6. Resultados **23**

7. Conclusión **25**

Glosario

OpenGL	API estándar para el desarollo de software con gráficos 2D o 3D con aceleración por hardware[8].
rasterizar	Conversión de una imagen descrita con alguna primitiva (curvas, vectores, triangulos) a un conjunto de pixels.

Capítulo 1

Introducción

1.1. ¿Qué es el aliasing?

El antialiasing es la deformación de ciertos elementos gráficos al ser rasterizados y plasmados en una pantalla con una resolución finita[1]. Normalmente los bordes de la geometría que tengan ángulos que no se alineen perfectamente con los pixels presentaran bordes de sierra abruptos, que no es fiel a la escena como se vería de forma natural.

Este proyecto se centra en aliasing dentro del campo de la computación gráfica, pero afecta a otras disciplinas, como el procesamiento de señales (por ejemplo al digitalizar señales de audio analógicas). En general, al representar valores continuos en algún medio discreto siempre habrá algún defecto de este tipo[1].

Existen otros tipos de aliasing, como en el interior de texturas o al mover la camera rápidamente.

1.2. ¿Qué es el antialiasing?

El antialiasing es el conjunto de técnicas cuyo objetivo es el de disimular o eliminar estas imperfecciones.

Capítulo 2

Planteamiento del problema

2.1. Aliasing en bordes de geometría

imagen

2.2. Aliasing de texturas

Posibles soluciones: Mip Maps, LoD

2.3. Specular Aliasing

2.4. Motion Aliasing

Explicar motion blur en cámaras

Accumulation motion blur (ps2+) Per pixel motion blur (ps3+) Per object motion blur (ps3+)

Capítulo 3

Objetivos

Este proyecto tiene dos objetivos principales:

1. Analizar el estado del arte en el campo de la computación gráfica en relación al anti-aliasing. Explorar los problemas que presenta el aliasing en gráficos 2D y 3D, la evolución de las soluciones y los desarrollos en curso.

2. El segundo objetivo es el de integrar una solución de anti-aliasing temporal a un motor gráfico 3D existente desarrollado durante la asignatura de ÄPIs tridimensionalesd̈el Máster en Computación Gráfica y Simulación cursado.

Capítulo 4

Estado del Arte

4.1. Filtrado

En gráficos raster (bidimensionales) el anti-aliasing se realiza mediante filtros.Para entender el filtrado en imágenes es necesario pensar en pixels no como pequeños cuadrados en una pantalla, si no en muestras de una función (normalmente 3 muestras, una por cada canal RGB)[2]. Los filtros (o kernels) transforman las funciones para intentar suavizar los cambios bruscos de color.

En la figura 4.1 se muestra una comparación de las diferentes formas en las que cada filtro interpola valores de una imagen con un solo canal. Cada punto representa la muestra de la función que se visualizará en la pantalla.

4.2. Supersampling Antialiasing

El supersampling consiste en minimizar el aliasing usando imágenes con una resolución mayor a la usada para su visualización (antialiasing espacial). Cada pixel en la pantalla será representado por más de un pixel en la imagen. Para el color final normalmente se hace la media del valor de los colores de los múltiples pixels.

Estas técnicas consumen más ancho de banda y memoria, ya que los buffers tienen que guardar más información por pixel que la imagen original.

La calidad del supersampling viene dado por la cantidad de samples por pixel y la técnica para saber que pixels usar en la imagen con mayor resolución para cada pixel de la pantalla.

Figura 4.1: Comparación de interpolación usando diferentes filtros[7]

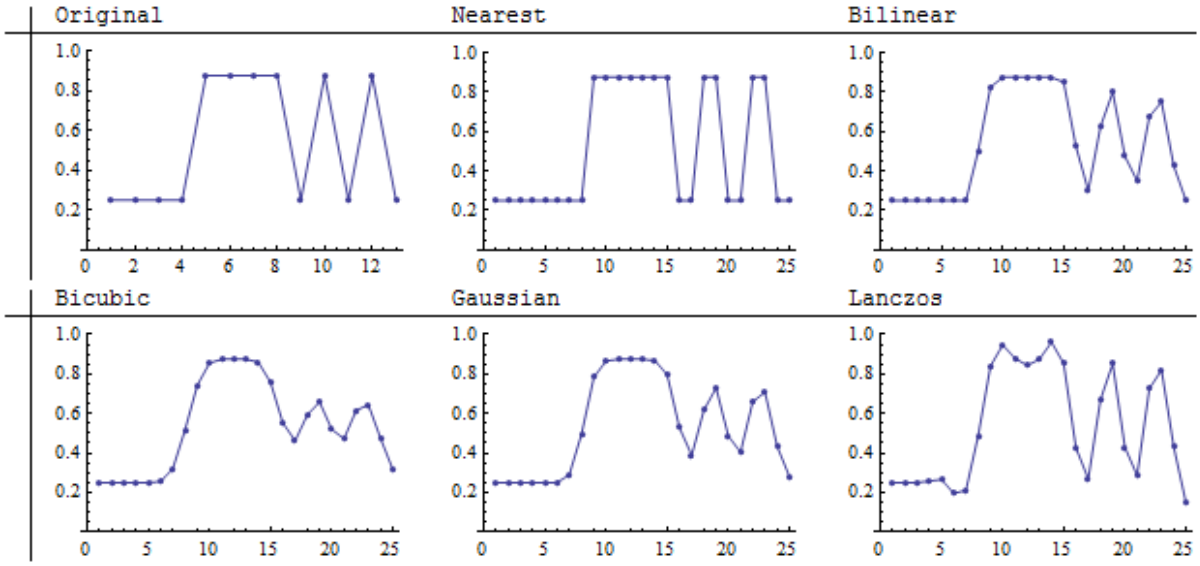


Figura 4.2: Comparación de resultados de diferentes filtros[6]

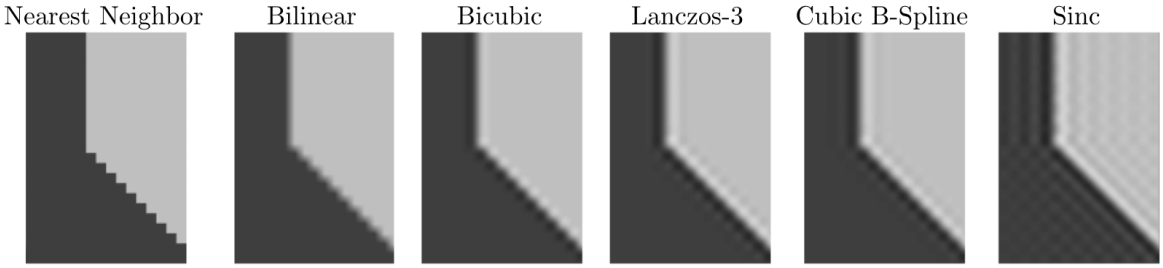


Figura 4.3: Ordered Grid Super-Sampling[3]

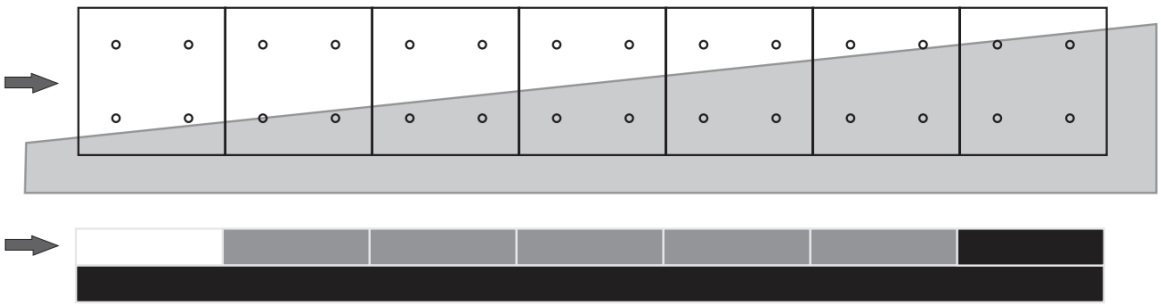
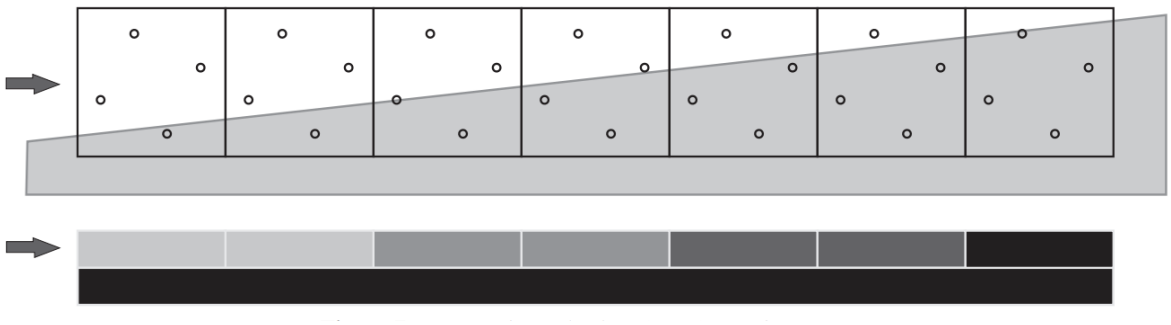


Figura 4.4: Rotated Grid Super-Sampling[3]



4.2.1. Ordered Grid Supersampling

4.2.2. Rotated Grid Supersampling

4.2.3. Sparse Grid Supersampling

4.3. Multisampling

Multisampling es una optimización del supersampling. Se sigue usando subpixels, pero en vez de resolver el color para cada subpixel el color se resuelve una única vez por pixel. Después de resolver el color se calcula el stencil buffer y depth buffer con subpixels, y dependiendo de la cantidad de subpixels que estén contenidos dentro del triangulo se usará un porcentaje del valor del color calculado. Esto reduce drásticamente el impacto en el rendimiento de la aplicación, ya que solo hay que acceder una vez por pixel a la textura, pero solo soluciona aliasing en los bordes de la geometría.

En el centro de un polígono todos los subpixels van a estar contenidos dentro, pero el color solo se calcula una vez (por ejemplo en el centro del pixel), por lo que se usará el 100 % del color, pero puede contener aliasing. En cambio, en los bordes, algún subpixel estará fuera de la geometría, por lo que se usará un porcentaje del color calculado, suavizando el borde.

4.4. Post Processing Antialiasing

Este tipo de antialiasing ha ganado popularidad durante los últimos años, debido en parte al uso del deferred shading en los últimos motores gráficos.

4.4.1. Fast Approximate Anti-aliasing

El FXAA[5] es un sistema de anti-aliasing de post procesado desarrollado por NVIDIA.

4.4.2. Morphological Antialiasing

mlaa

4.4.2.1. Enhanced Subpixel Morphological Antialiasing

smaa

4.4.3. Temporal Antialiasing

El temporal antialiasing es un tipo de antialiasing de post procesado, pero que amortiza el coste computacional usando múltiples samples en múltiples frames. Esto presenta unas complicaciones a la hora de escoger samples en frames anteriores en imágenes no estáticas.

Capítulo 5

Desarrollo

Para este proyecto se ha integrado un sistema de anti-aliasing temporal en un motor de gráficos 3D que usa OpenGL com API gráfica.

5.1. Jitter

5.2. Motion Vectors

5.3. Combinación de texturas

5.4. Neighborhood Clamping

5.5. Sharpening kernel

Capítulo 6

Resultados

Figura 6.1: Anti-aliasing off

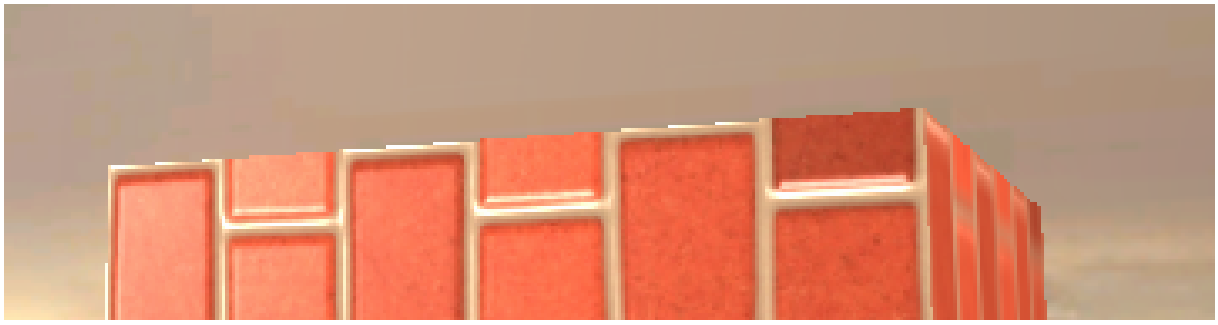


Figura 6.2: Temporal anti-aliasing con 8 samples

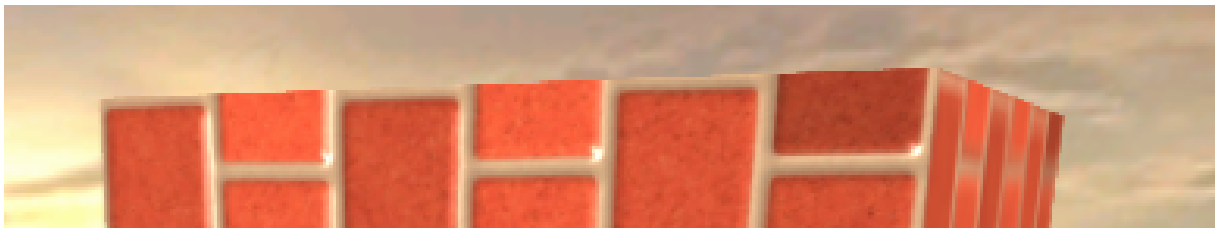


Figura 6.3: Temporal anti-aliasing con 16 samples

Capítulo 7

Conclusión

Bibliografía

[1] D. P. Mitchell y A. N. Netravali, «Reconstruction Filters in Computer-graphics», *SIGGRAPH Comput. Graph.*, vol. 22, n.º 4, págs. 221-228, jun. de 1988, ISSN: 0097-8930. DOI: 10.1145/378456.378514. dirección: http://doi.acm.org/10.1145/378456.378514.

[2] A. R. Smith, «A Pixel Is Not A Little Square, A Pixel Is Not A Little Square, A Pixel Is Not A Little Square! (And a Voxel is Not a Little Cube)», Technical Memo 6, Microsoft Research, inf. téc., 1995.

[3] K. Beets y D. Barron, «Super-sampling Anti-aliasing Analyzed», 2000.

[4] J. Sachs, «Image Resampling», inf. téc., 2001.

[5] T. Lottes, «FXAA. NVIDIA White Paper», inf. téc., 2009.

[6] P. Getreuer, «Linear Methods for Image Interpolation», vol. 1, sep. de 2011.

[7] *What is Lanczos resampling useful for in a spatial context?*, 2011. dirección: https://gis.stackexchange.com/questions/10931/what-is-lanczos-resampling-useful-for-in-a-spatial-context.

[8] J. Kessenich, G. Sellers y D. Shreiner, *OpenGL®Programming Guide: The Official Guide to Learning OpenGL®, Version 4.5 with SPIR-V*, 9.ª ed. Addison-Wesley Professional, 2016, ISBN: 9780134495491.

Índice de figuras

4.1. Comparación de interpolación usando diferentes filtros[7] 18
4.2. Comparación de resultados de diferentes filtros[6] 18
4.3. Ordered Grid Super-Sampling[3] 18
4.4. Rotated Grid Super-Sampling[3] 19

6.1. Anti-aliasing off 23
6.2. Temporal anti-aliasing con 8 samples 24
6.3. Temporal anti-aliasing con 16 samples 24

Índice de cuadros