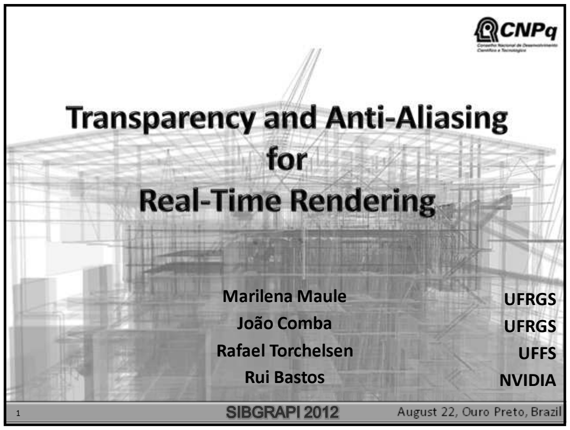
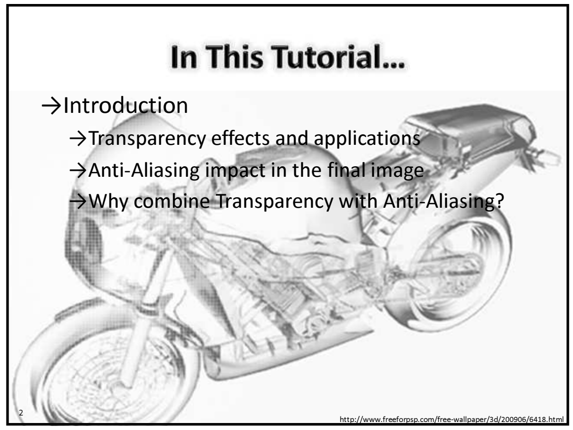
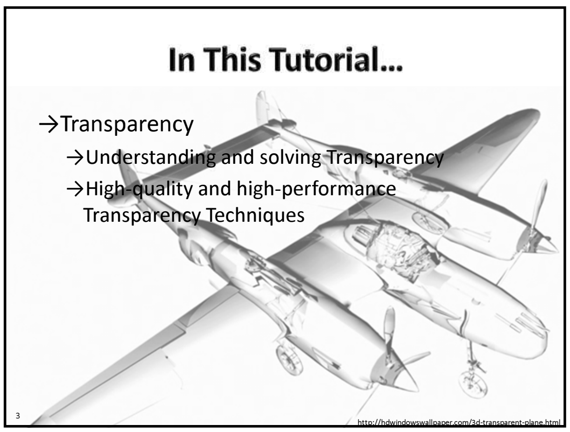
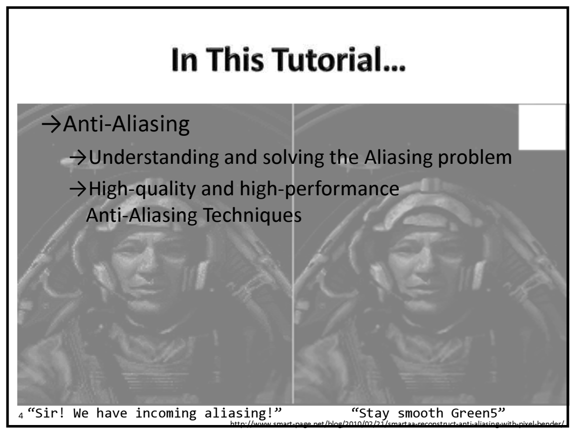
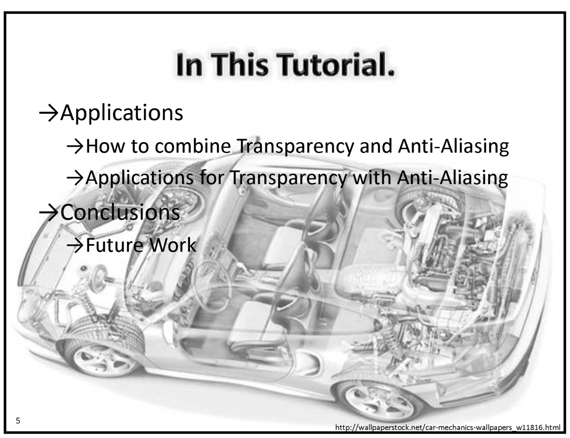










Outline

→Introduction

→Transparency Problem and Basic Approach
→Raster-based Transparency Techniques

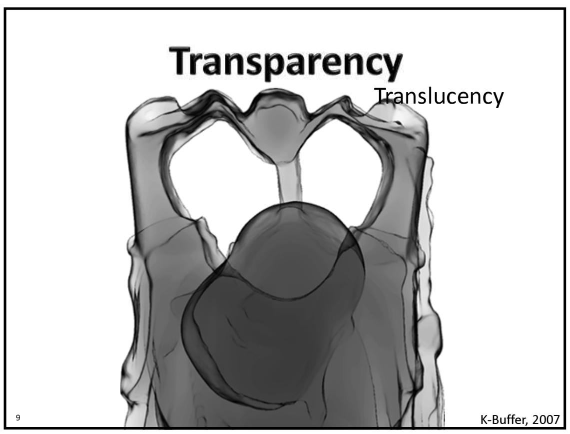
→AA Problem and Basic Approaches
→AA Techniques

→Applications
→Conclusions

6


Transparency
Effects
7
Stochastic Transparency, NVIDIA2010


Transparency
Particles
8
http://www.wallpaper-photo.ru/en/preview.php?hd=15217

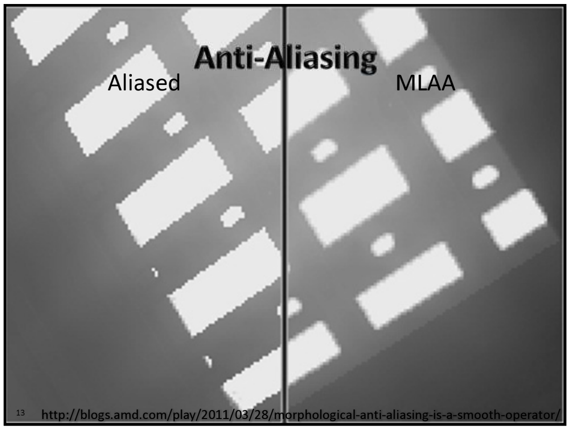

Transparency
Translucency
9
K-Buffer, 2007


Transparency
CAD
10


Anti-Aliasing
Aliased
Anti-Aliased
11
Jimenez, Siggraph Course 2011


Anti-Aliasing
Aliased
4xMSAA
12
http://thesavingthrow.blogspot.com.br/2011/05/ati-vs-nvidia-who-is-belle-of-ball.html

Anti-Aliasing



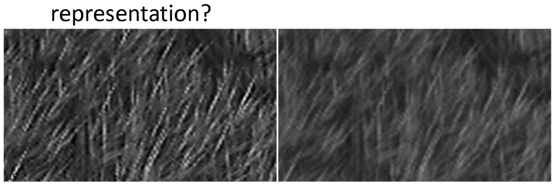
13 http://blogs.amd.com/play/2011/03/28/morphological-anti-aliasing-is-a-smooth-operator/

Anti-Aliasing

14 http://www.humus.name/index.php?page=3D

Transparency + Anti-Aliasing

- Both refer to visibility computation
  - Fragment coverage over pixel area
  - Fragment visibility along depth
- Could both be combined into a single visibility representation?



Outline

→Introduction

→Transparency Problem and Basic Approach
→Raster-based Transparency Techniques

→AA Problem and Basic Approaches
→AA Techniques

→Applications
→Conclusions

16

Outline

→Introduction

→Transparency Problem and Basic Approach
→Raster-based Transparency Techniques

→AA Problem and Basic Approaches
→AA Techniques

→Applications
→Conclusions

1

Outline

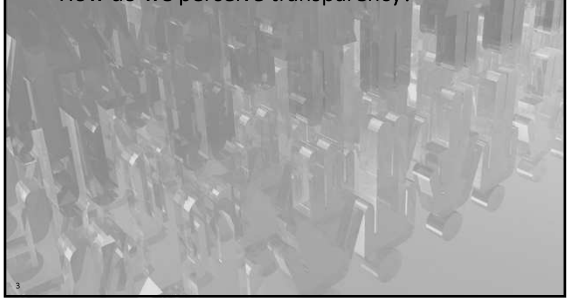
→Transparency Problem and Basic Approach
- →Transparency Fundaments
- →Transparency in Practice
- →Primitive Sorting

→Raster-based Transparency Techniques
- →Buffer-Based
- →Depth Peeling
- →Sort Independent
- →Stochastic

2

Transparency Fundaments

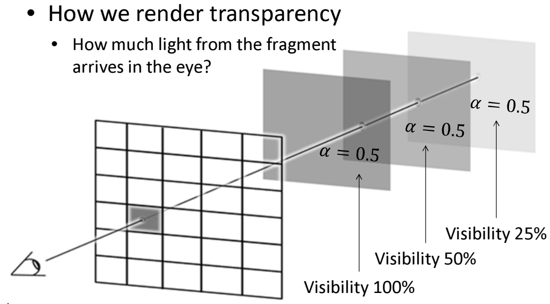
- How do we perceive transparency?



3

Transparency Fundaments

- How we render transparency
  - How much light from the fragment arrives in the eye?



4

Transparency Fundaments

- Opacity (α):
  – how much light the surface transmits?
- Visibility:
  – how much of the transmitted light the eye can see?

5

Transparency Fundaments

- Blending equations [Porter and Duff, 1984]
  – Back-to-Front

$$C'_{acc} = (1 - \alpha_i)C_{acc} + \alpha_i C_i$$

  – A new fragment reduces visibility from the accumulated color

$\boldsymbol{C'_{acc}}$	Accumulated color
C_i	Color from fragment *i*
α_{acc}	Accumulated opacity
α_i	Opacity from fragment *i*

6

Transparency Fundaments

- Blending equations [Porter and Duff, 1984]
  - Front-to-Back

$$C'_{acc} = Ca_{cc} + (1 - \alpha_{acc})\alpha_i C_i$$

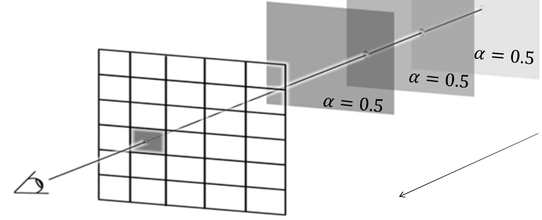
  - A new fragments has its visibility reduces by the accumulated opacity

$\boldsymbol{C'_{acc}}$	Accumulated color
C_i	Color from fragment *i*
α_{acc}	Accumulated opacity
α_i	Opacity from fragment *i*

7

Transparency Fundaments

- Back-to-front blending

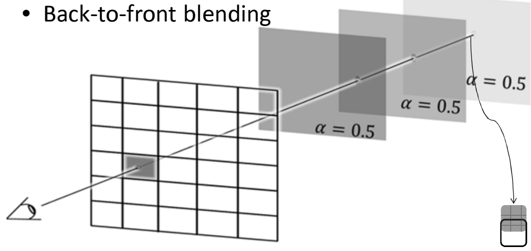


$$C'_{acc} = (1 - \alpha_i)C_{acc} + \alpha_i C_i$$

8

Transparency Fundaments

- Back-to-front blending

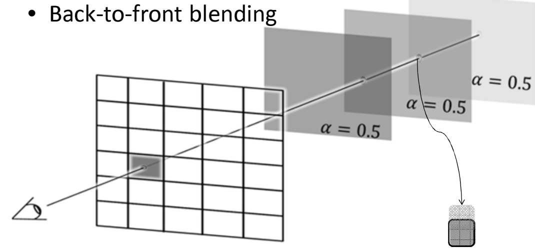


$$(0.5, 0, 0, 0.5) = (1 - 0.5)(0,0,0,0) + 0.5Red$$
$$C'_{acc} = (1 - \alpha_i)C_{acc} + \alpha_i C_i$$

9

Transparency Fundaments

- Back-to-front blending

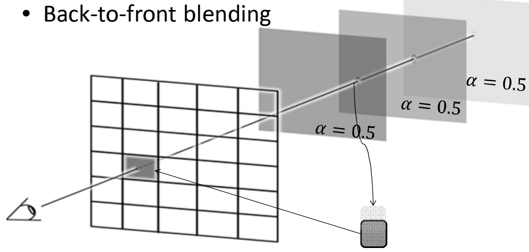


$$(0.25,0, 0.5, 0.75) = (1 - 0.5)(0.5, 0, 0, 0.5) + 0.5Blue$$
$$C'_{acc} = (1 - \alpha_i)C_{acc} + \alpha_i C_i$$

10

Transparency Fundaments

- Back-to-front blending

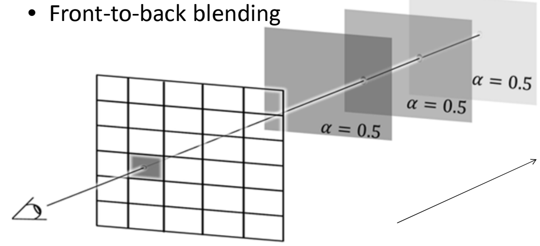


$$(0.125,0.5, 0.25, 0.875) = (1 - 0.5)(0.25,0, 0.5, 0.75) + 0.5Green$$
$$C'_{acc} = (1 - \alpha_i)C_{acc} + \alpha_i C_i$$

11

Transparency Fundaments

- Front-to-back blending

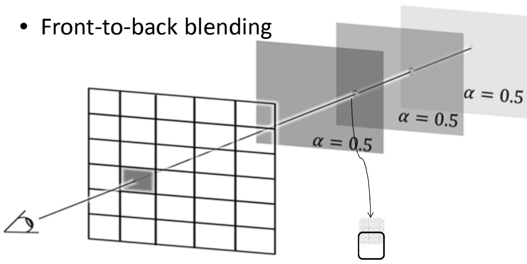


$$C'_{acc} = C_{acc} + (1 - \alpha_{acc})\alpha_i C_i$$

12

Transparency Fundaments

- Front-to-back blending



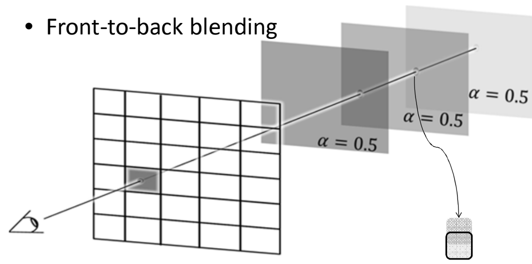
$(0{,}0.5{,}0{,}0.5) = (0{,}0{,}0{,}0) + (1-0)0.5Green$

$$C'_{acc} = C_{acc} + (1-\alpha_{acc})\alpha_i C_i$$

13

Transparency Fundaments

- Front-to-back blending



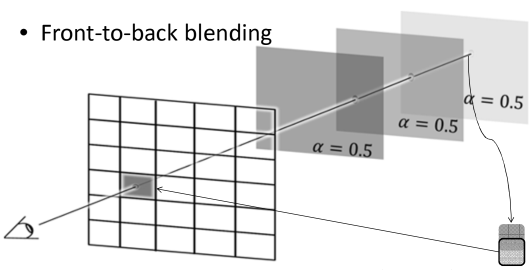
$(0{,}0.5{,}0.25{,}0.75) = (0{,}0.5{,}0{,}0.5) + (1-0.5)0.5Blue$

$$C'_{acc} = C_{acc} + (1-\alpha_{acc})\alpha_i C_i$$

14

Transparency Fundaments

- Front-to-back blending



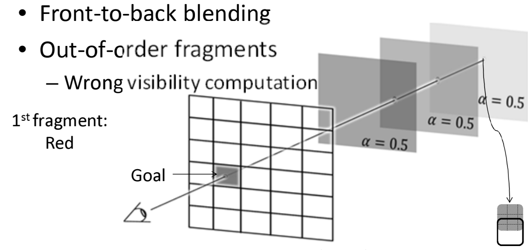
$(0.175{,}0.5{,}0.25{,}0.875) = (0{,}0.5{,}0.25{,}0.75) + (1-0.75)0.5Red$

$$C'_{acc} = C_{acc} + (1-\alpha_{acc})\alpha_i C_i$$

15

Transparency Fundaments

- Front-to-back blending
- Out-of-order fragments
  - Wrong visibility computation



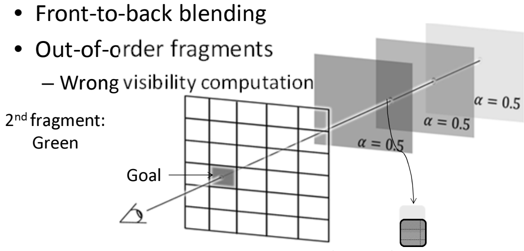
$(0.5{,}0{,}0{,}0.5) = (0{,}0{,}0{,}0) + (1-0)0.5Red$

$$C'_{acc} = C_{acc} + (1-\alpha_{acc})\alpha_i C_i$$

16

Transparency Fundaments

- Front-to-back blending
- Out-of-order fragments
  - Wrong visibility computation



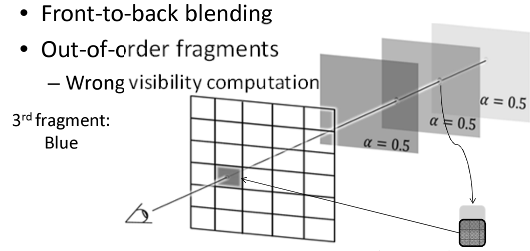
$(0.5{,}0.25{,}0{,}0.75) = (0.5{,}0{,}0{,}0.5) + (1-0.5)0.5Green$

$$C'_{acc} = C_{acc} + (1-\alpha_{acc})\alpha_i C_i$$

17

Transparency Fundaments

- Front-to-back blending
- Out-of-order fragments
  - Wrong visibility computation



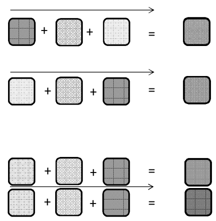
$(0.5{,}0.25{,}0.175{,}0.875) = (0.5{,}0.25{,}0{,}0.75) + (1-0.75)0.5Blue$

$$C'_{acc} = C_{acc} + (1-\alpha_{acc})\alpha_i C_i$$

18

Transparency Fundaments

- Summary
  - Back-to-Front
  - Front-to-Back
  - Out-of-Order
    - FTB blending
    - BTF blending



19

Outline

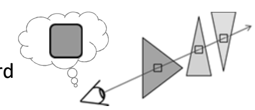
→Transparency Problem and Basic Approach
- →Transparency Fundaments
- →Transparency in Practice
- →Primitive Sorting

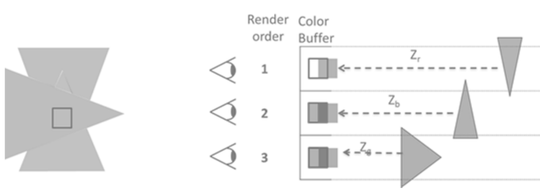
→Raster-based Transparency Techniques
- →Buffer-Based
- →Depth Peeling
- →Sort Independent
- →Stochastic

20

Transparency in Practice

- Depth Test Enabled
  - Incorrect fragment discard

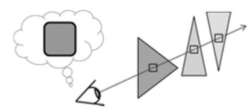


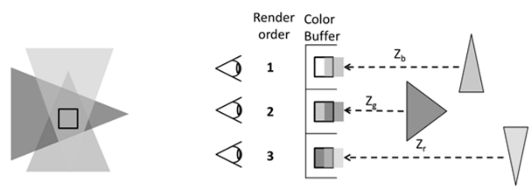


21

Transparency in Practice

- Depth Test Disabled
- Out-Of-Order rendering
  - Incorrect visibility results

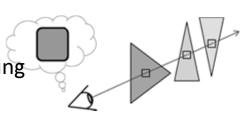


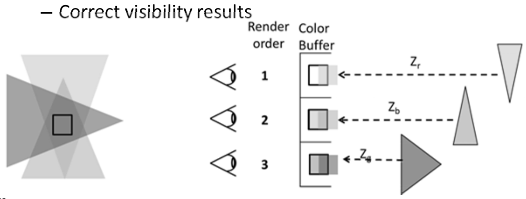


22

Correct Transparency

- Depth Test Disabled
- Depth-Sorted-Order rendering
  - Primitive sorting
    - Assuming no interpenetration
  - Correct visibility results

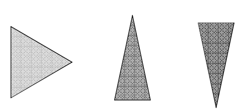




23

Correct Transparency

Primitive Sorting X Fragment Sorting



All fragments have the same depth-sorted order from their primitive.

The fragments are depth-sorted individually.

24

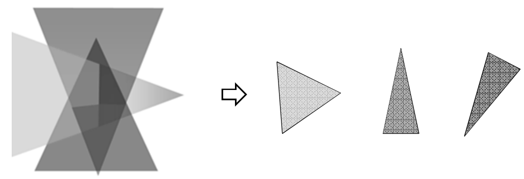
Outline

- →Transparency Problem and Basic Approach
  - →Transparency Fundaments
  - →Transparency in Practice
  - →Primitive Sorting
- →Raster-based Transparency Techniques
  - →Buffer-Based
  - →Depth Peeling
  - →Sort Independent
  - →Stochastic

25

Primitive Sorting

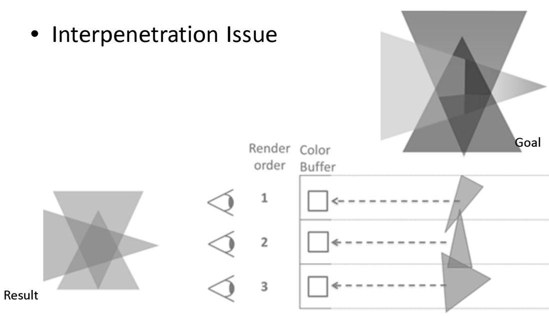
- Interpenetration Issue



26

Primitive Sorting

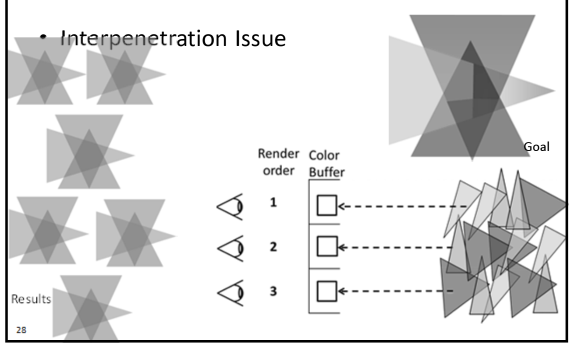
- Interpenetration Issue



27

Primitive Sorting

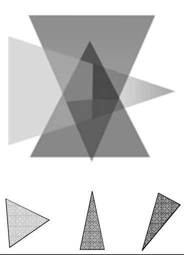
- Interpenetration Issue



28

Primitive Sorting

- Interpenetration
- Spatial data structures
  - BSP
  - Kd-Tree
  - Octree
  - . . .

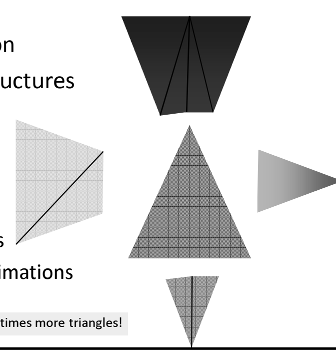


3 triangles...

29

Primitive Sorting

- Interpenetration
- Spatial data structures
  - BSP
  - Kd-Tree
  - Octree
  - . . .
  - Split primitives
  - Rebuild for animations



n times more triangles!

30

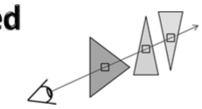
Outline

→Transparency Problem and Basic Approach
- →Transparency Fundaments
- →Transparency in Practice
- →Primitive Sorting

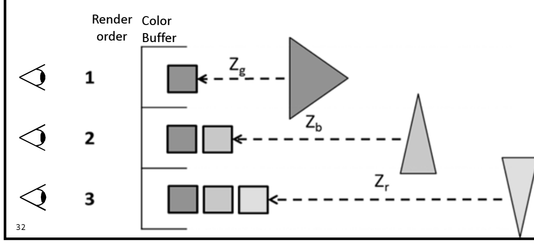
→Raster-based Transparency Techniques
- →Buffer-Based
- →Depth Peeling
- →Sort Independent
- →Stochastic

31

Buffer-based

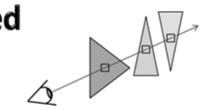


1. Store fragments

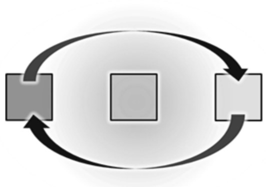


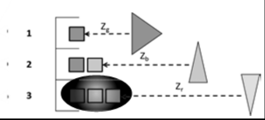
32

Buffer-based



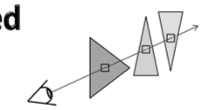
1. Store fragments
2. Sort fragments





33

Buffer-based



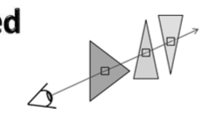
1. Store fragments
2. Sort fragments
3. Blend fragments

Back-to-Front

$$C'_{acc} = (1-\alpha_{frag})C_{acc} + \alpha_{frag}C_{frag}$$

34

Buffer-based



1. Store fragments
2. Sort fragments
3. Blend fragments

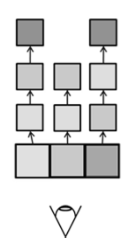
Front-to-Back

$$C'_{acc} = C_{acc} + (1-\alpha_{acc})Cf_{rag}\alpha_{frag}$$

35

Unbounded Buffers

- Store all fragments
  - → A-buffer
    True Linked Lists



36

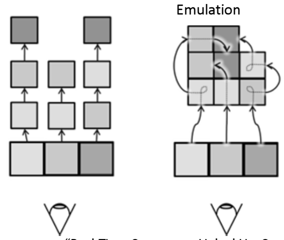
Carpenter, 1984

Unbounded Buffers

- Store all fragments

→ A-buffer
True Linked Lists

→ PPLL
GPU Linked Lists Emulation



37 "Real-Time Concurrent Linked List Construction on the GPU", Yang et al, 2010

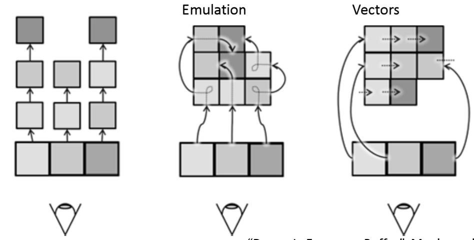
Unbounded Buffers

- Store all fragments

→ A-buffer
True Linked Lists

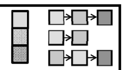
→ PPLL
GPU Linked Lists Emulation

→ DFB
GPU Dynamic Vectors



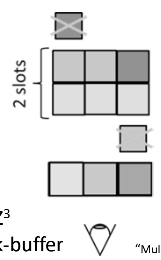
38 "Dynamic Fragment Buffer", Maule et al, 2012

Fixed-Size Buffers



Correct order

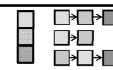
→ 3D buffers
Stores all attributes



→ Z^3
→ k-buffer

"Multi-fragment effects on the GPU using the kbuffer", Bavoil et al, 2007
39 "Z3: an economical hardware technique for high-quality antialiasing and transparency", Jouppi, 1999

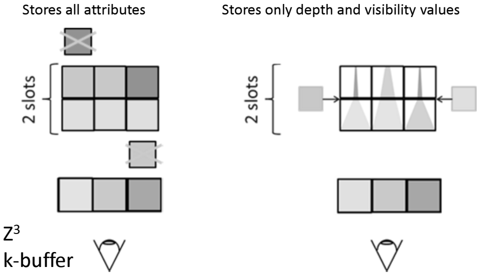
Fixed-Size Buffers



Correct order

→ 3D buffers
Stores all attributes

→ Adaptive Transparency
Stores only depth and visibility values



→ Z^3
→ k-buffer

40 "Adaptive transparency", Salvi et al, 2011

Buffer-Based Summary

- Store all fragments
  - Precise compositing of fragments
  - Unbounded amounts of memory
- Fixed storage
  - Faster but approximate
    - Not all fragments considered in each pixel
    - Approximate visibility contributions

41

Outline

→Transparency Problem and Basic Approach
- →Transparency Fundaments
- →Transparency in Practice
- →Primitive Sorting

→Raster-based Transparency Techniques
- →Buffer-Based
- →Depth Peeling
- →Sort Independent
- →Stochastic

42

Depth Peeling

- Store the farthest unblended layer
  - 1st geometry pass

43

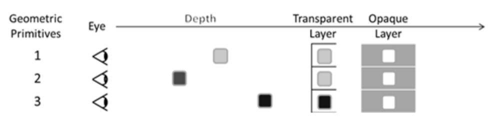
Depth Peeling

- Store the farthest unblended layer
  - 1st geometry pass

44

Depth Peeling

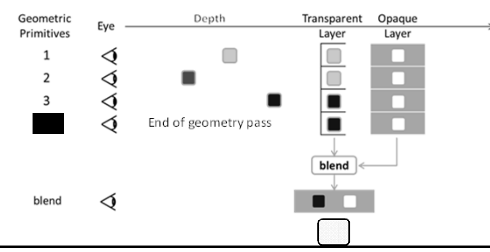
- Store the farthest unblended layer
  - 1st geometry pass



45

Depth Peeling

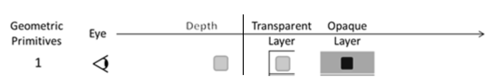
- Store the farthest unblended layer
  - Blend with the opaque layer



46

Depth Peeling

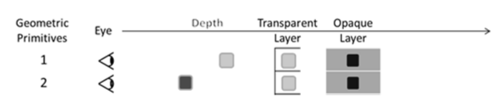
- Compose layers over several geometry steps
  - 2nd geometry pass



47

Depth Peeling

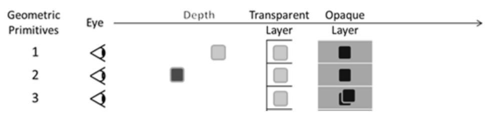
- Compose layers over several geometry steps
  - 2nd geometry pass



48

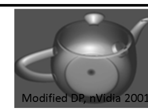
Depth Peeling

- Compose layers over several geometry steps
  - 2nd geometry pass

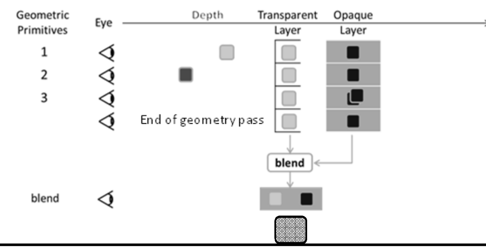


49

Depth Peeling



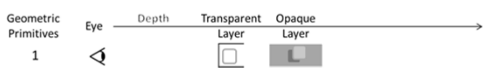
- Compose layers over several geometry steps
  - Blend with the opaque layer



50

Depth Peeling

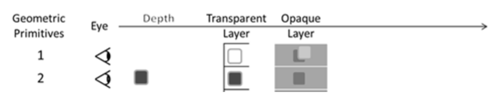
- Compose layers over several geometry steps
  - 3rd geometry pass



51

Depth Peeling

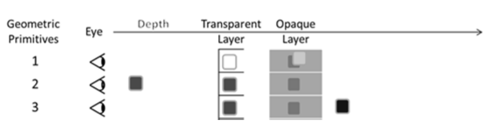
- Compose layers over several geometry steps
  - 3rd geometry pass



52

Depth Peeling

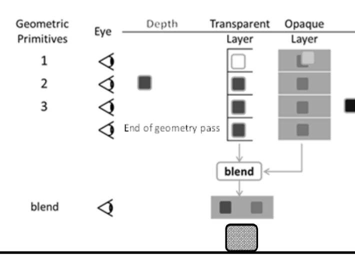
- Compose layers over several geometry steps
  - 3rd geometry pass

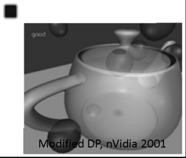


53

Depth Peeling

- Compose layers over several geometry steps
  - Blend with the opaque layer

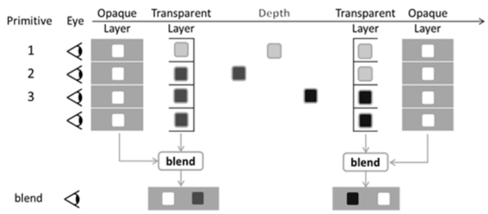




54

Dual Depth Peeling

- Compose two layers per geometry step
  - Front-most and back-most



55

Depth-Peeling Summary

- Fixed and small amount of memory
- Slow geometry multipass

56

Outline

→Transparency Problem and Basic Approach
- →Transparency Fundaments
- →Transparency in Practice
- →Primitive Sorting

→Raster-based Transparency Techniques
- →Buffer-Based
- →Depth Peeling
- →Sort Independent
- →Stochastic

57

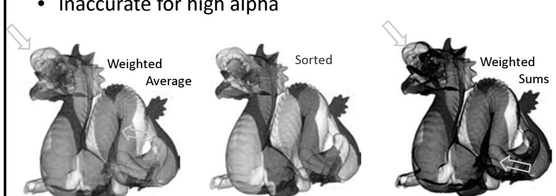
Sort-Independent Transparency

- 2 Geometry passes
  - Alpha accumulation
- Good approximation for low alpha
- Inaccurate for high alpha

58

Sort-Independent Transparency

- 2 Geometry passes
  - Alpha accumulation
- Good approximation for low alpha
- Inaccurate for high alpha



59

BM08, Meshkin07

Sort-Independent Transparency

- 2 Geometry passes
  - Alpha accumulation
- Good approximation for low alpha
- Inaccurate for high alpha

60

BM08, Meshkin07

Outline

→Transparency Problem and Basic Approach
- →Transparency Fundaments
- →Transparency in Practice
- →Primitive Sorting

→Raster-based Transparency Techniques
- →Buffer-Based
- →Depth Peeling
- →Sort Independent
- →Stochastic

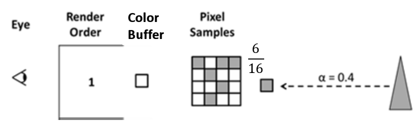
61

Stochastic Transparency

- Super resolution screen-door transparency
  - Alpha and visibility translate to sample individual probability of coverage

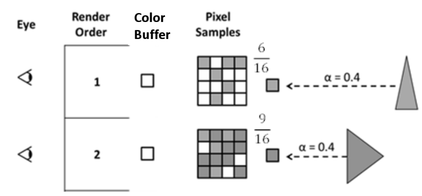
62

Stochastic Transparency



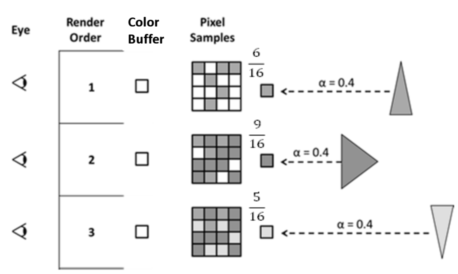
63

Stochastic Transparency



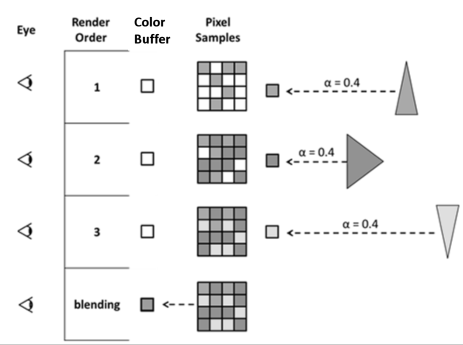
64

Stochastic Transparency



65

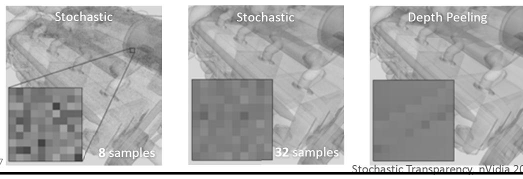
Stochastic Transparency



66

Stochastic Transparency

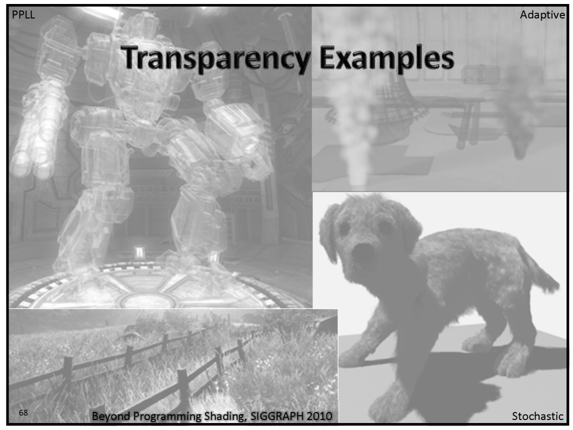
- MSAA samples to represent visibility
  - Pre-passes to estimate visibility
- Small number of samples per pixel leads to noise



67

Stochastic Transparency, nVidia 2010

Transparency Examples



68

Beyond Programming Shading, SIGGRAPH 2010

Outline

→Introduction

→Transparency Problem and Basic Approach
→Raster-based Transparency Techniques

→AA Problem and Basic Approaches
→AA Techniques

→Applications
→Conclusions

69

Outline

- →Introduction
- →Transparency Problem and Basic Approach
- →Raster-based Transparency Techniques
- →AA Problem and Basic Approaches
- →AA Techniques
- →Applications
- →Conclusions

1

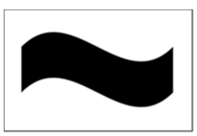
Outline

- →AA Problem and Basic Approaches
  - →The Aliasing Problem
  - →Single Sample Problem
  - →Super Sampling
  - →Sample Distribution
- →AA Techniques
  - →FSAA
  - →IAA
  - →GBAA

2

The Aliasing Problem

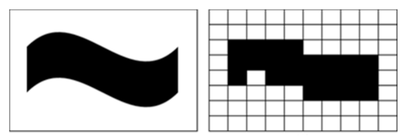
- I want to draw a shape...



3

The Aliasing Problem

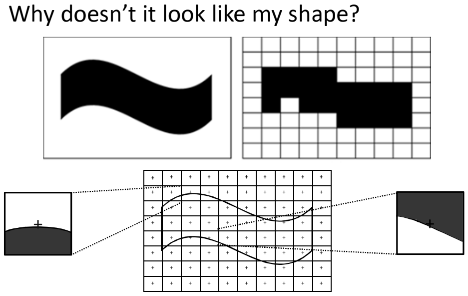
- But what appears doesn't look like my shape!



4

The Aliasing Problem

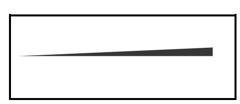
- Why doesn't it look like my shape?



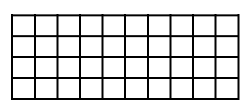
5

The Aliasing Problem

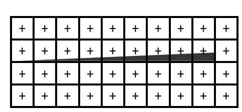
- Thin primitives



Thin primitive



Single sample per pixel result

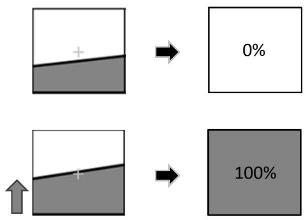


Single sample per pixel

6

The Aliasing Problem

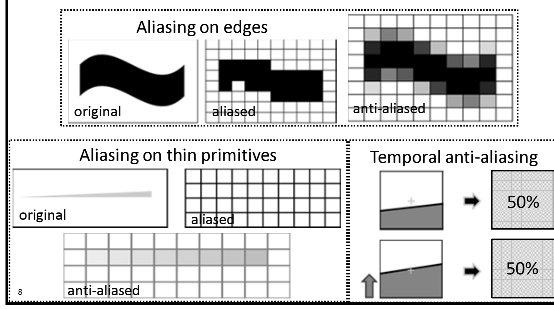
- Single-Sample Problem: Temporal Aliasing



7

The Aliasing Problem

- If only I could estimate the pixels coverage...



8

Outline

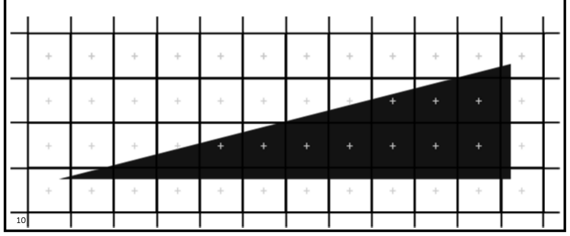
→AA Problem and Basic Approaches
- →The Aliasing Problem
- →Single Sample Problem
- →Super Sampling
- →Sample Distribution

→AA Techniques
- →FSAA
- →IAA
- →GBAA

9

Single Sample Problem

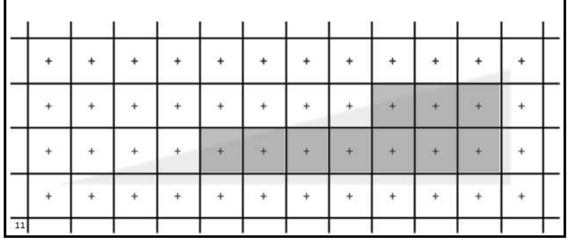
- Rendering into a normal resolution



10

Single Sample Problem

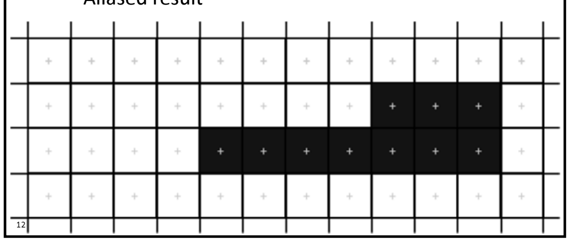
- Rendering into a normal resolution
  - Center sampling



11

Single Sample Problem

- Rendering into a normal resolution
  - Center sampling
  - Aliased result



12

Outline

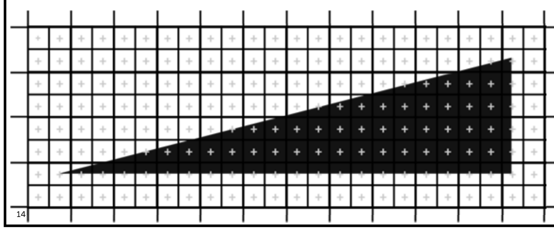
→AA Problem and Basic Approaches
- →The Aliasing Problem
- →Single Sample Problem
- →Super Sampling
- →Sample Distribution

→AA Techniques
- →FSAA
- →IAA
- →GBAA

13

Super Sampling

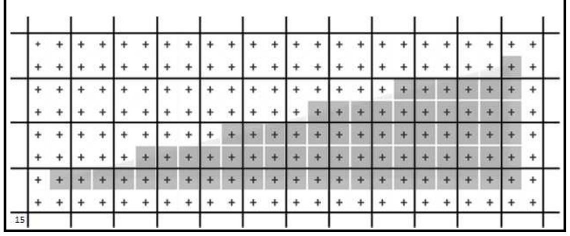
- Rendering into 4x-higher resolution



14

Super Sampling

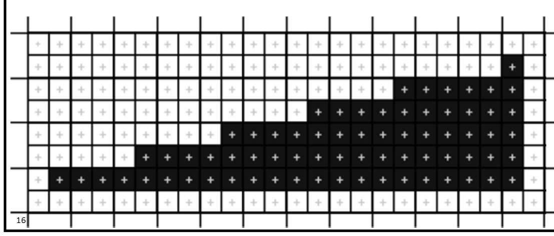
- Rendering into 4x-higher resolution
  – Center sampling



15

Super Sampling

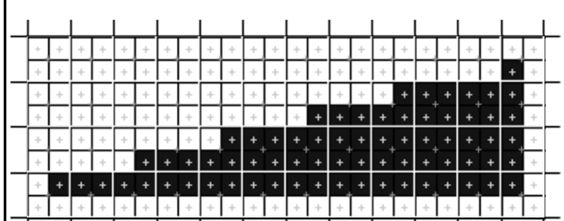
- Rendering into 4x-higher resolution
  – Center sampling high-resolution result



16

Super Sampling

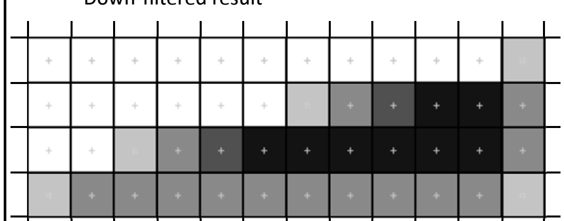
- Rendering into 4x-higher resolution
  – 2x2 box filter



17

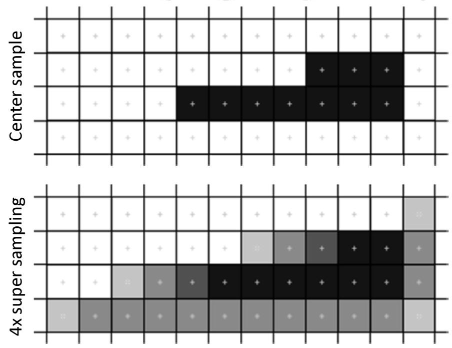
Super Sampling

- Rendering into 4x-higher resolution
  – 2x2 box filter
  – Down-filtered result



18

Central Sampling x Super Sampling



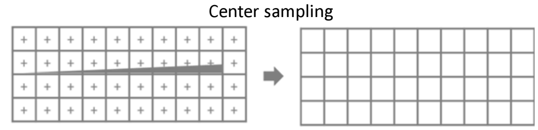
19

Central Sampling x Super Sampling

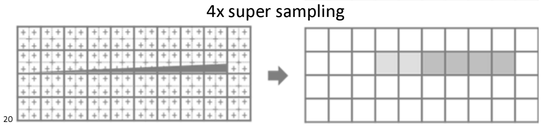
- Thin Primitives

Rasterization process
Result in the display

Center sampling



4x super sampling

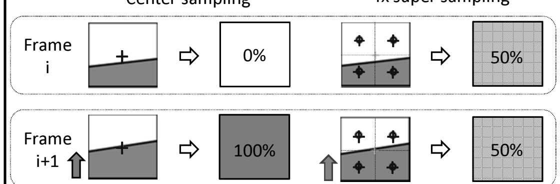


20

Central Sampling x Super Sampling

- Temporal Aliasing

Center sampling
4x super sampling



21

Outline

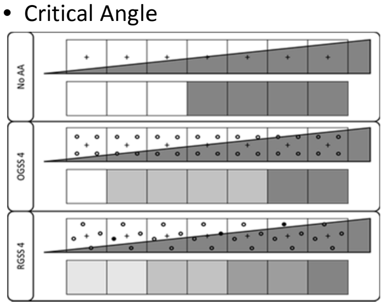
→AA Problem and Basic Approaches
- →The Aliasing Problem
- →Single Sample Problem
- →Super Sampling
- →Sample Distribution

→AA Techniques
- →FSAA
- →IAA
- →GBAA

22

Sample Distribution

- Critical Angle



Critical angle:

Nearly horizontal and nearly vertical edges

Critical angle rotated:

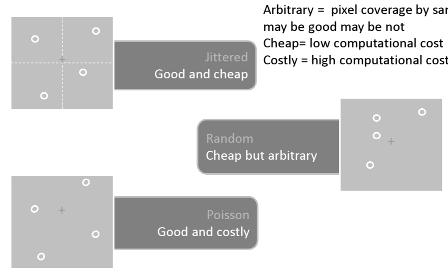
Less disturbing

23

Sample Distribution

- Regularity

Good = pixel area well covered by samples
Arbitrary = pixel coverage by samples may be good may be not
Cheap= low computational cost
Costly = high computational cost



24

Outline

→AA Problem and Basic Approaches
- →The Aliasing Problem
- →Single Sample Problem
- →Super Sampling
- →Sample Distribution

→AA Techniques
- →FSAA
- →IAA
- →GBAA

25

Full-Scene Anti-Aliasing

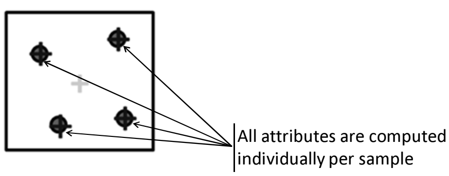
- Input: 3D scene
  - No selection
    - Entire scene is processed
  - More than one sample per pixel
    - Buffers
    - Filtering
    - Downsampling
- Output: anti-aliased image

26

Super Sampling Anti-Aliasing

SSAA

- Per-sample color evaluation

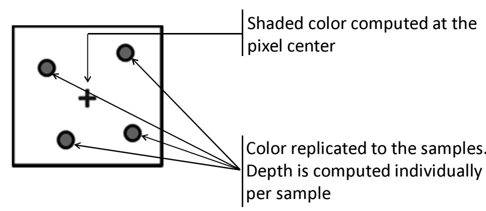


27

Multi-Sampling Anti-Aliasing

MSAA

- Single color evaluation
  - Once per pixel

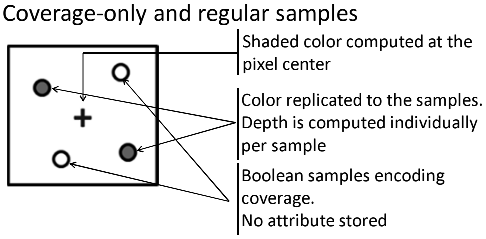


28

Coverage-Sample Anti-Aliasing

CSAA

- Also EQAA
- Single color evaluation
- Coverage-only and regular samples

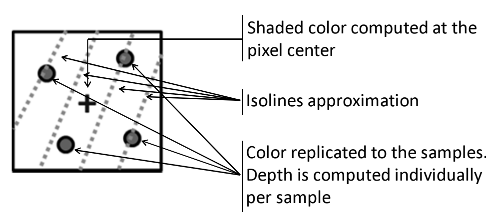


29

Directionally-Adaptive Edge AA

DAEAA

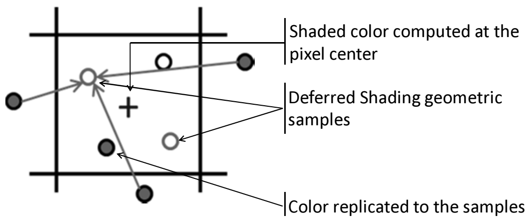
- Single color evaluation



30

Subpixel Reconstruction AA
SRAA

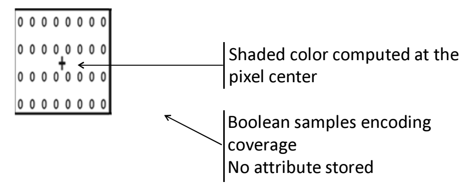
- Single color evaluation
- Samples reconstruction



31

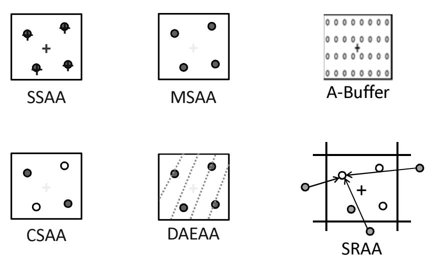
A-Buffer

- Single color evaluation
- Boolean coverage mask
- Max and min depth values



32

FSAA Techniques Summary



33

FSAA Summary

- Pros
  - High image quality
  - Temporal stability
  - Subpixel anti-aliasing
- Cons
  - High memory consumption

34

Outline

→AA Problem and Basic Approaches
- →The Aliasing Problem
- →Single Sample Problem
- →Super Sampling
- →Sample Distribution

→AA Techniques
- →FSAA
- →IAA
- →GBAA

35

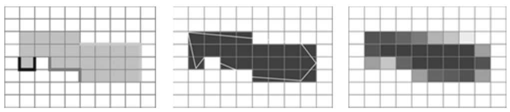
Image Post-Processing AA

- Input: digital image
  - Search for aliased pixels
  - Selection of aliased pixels
  - Intelligent blur of aliased pixels
- Output: anti-aliased image

36

Morphological Anti-Aliasing

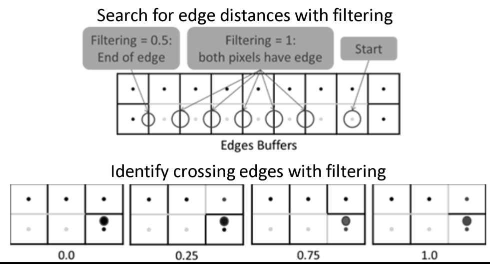
- Search for aliased patterns
  - Create list of found patterns
- Rebuild approximated geometry
- Blur



37

Practical MLAA (Jimenez's)

- Parallel MLAA
  - MLAA in real time using GPU filtering

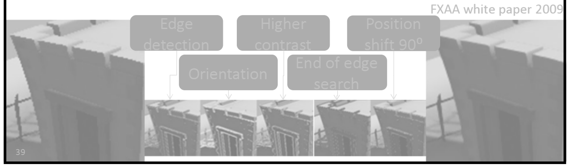


38

Fast approXimate AA

- Search for aliased pixels
  - Local luma contrast
    - 4-neighbors
- Directional blur filter
  - Perpendicular to luma gradient

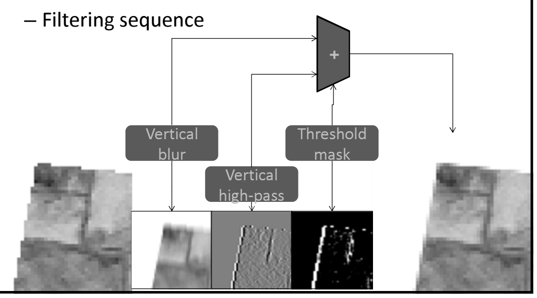
FXAA white paper 2009



39

Directionally Localized AA

- Photoshop prototype
  - Filtering sequence



40

Image Post-Processing Summary

- Pros
  - Very fast
  - Low resource consumption
- Cons
  - Does not adequately handle subpixels
  - Temporal instability

41

Outline

→AA Problem and Basic Approaches
- →The Aliasing Problem
- →Single Sample Problem
- →Super Sampling
- →Sample Distribution

→AA Techniques
- →FSAA
- →IAA
- →GBAA

42

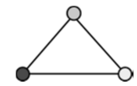
Geometric Anti-Aliasing

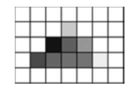
- Input: 3D scene
  - Extra information to edge pixels
  - Post-processing blur
- Output: anti-aliased image

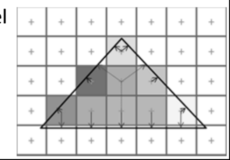
43

Distance-to-Edge Anti-Aliasing

- Pixel distance to triangle edge
  - Distance approximated by color
    - Each vertex receives one channel setup
  - Interpolation
- Distance-guided blur
  - Only if smaller than one pixel



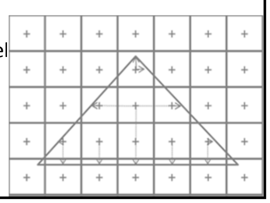




44

Geometry-Buffer Anti-Aliasing

- Pixel distance to triangle edge
  - Computed analytically
    - Encoded into RGB texture
    - Major direction
- Distance-guided blur
  - Only if less than half pixel



45

Geometric Anti-Aliasing Summary

- Pros
  - Image quality
  - Temporal stability
- Cons
  - Does not resolve thin primitives

46

Outline

→Introduction

→Transparency Problem and Basic Approach
→Raster-based Transparency Techniques

→AA Problem and Basic Approaches
→AA Techniques
→FSAA →Image Post-Processing →Geometric AA

→Applications
→Conclusions

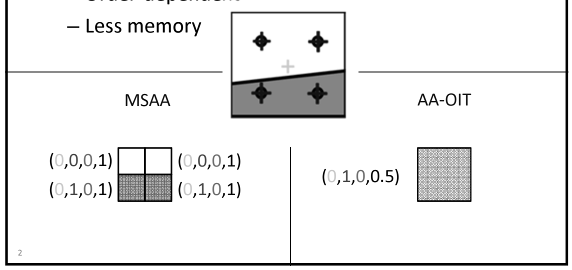
47

Outline

- →Introduction
- →Transparency Problem and Basic Approach
- →Raster-based Transparency Techniques
- →AA Problem and Basic Approaches
- →AA Techniques
- →Applications
- →Conclusions

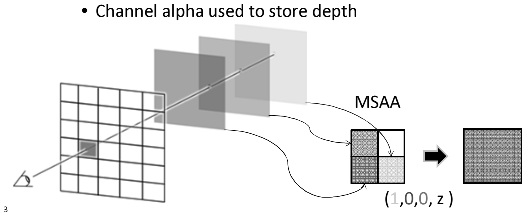
AA using Hardware for Transparency

- Coverage is used to weight alpha
  - Order-dependent
  - Less memory

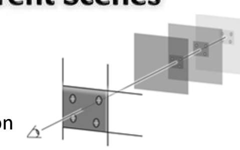


Transparency using Hardware for AA

- Multisampled texture used to store fragments
  - Post-processed sorting and blending
  - Uniform alpha
    - Channel alpha used to store depth



AA for Transparent Scenes



- FSAA
  - High quality
  - High memory consumption
- GAA
  - High quality
  - Geometric borders only
- IAA
  - Low quality
  - Low memory consumption

Outline

- →Introduction
- →Transparency Problem and Basic Approach
- →Raster-based Transparency Techniques
- →AA Problem and Basic Approaches
- →AA Techniques
- →Applications
- →Conclusions

Conclusions

- OIT and AA are still open problems
  - Combining them is challenging
  - Combining them with Deferred Shading is even more challenging
- There is no default solution
  - Application dependent
    - Quality x Performance x Memory

Future…

- OIT and AA with Deferred Shading
- Deferred Shading
  - G-buffers
  - Shading as a post-processing stage
- +OIT
  - Overburden G-buffers
- +AA
  - Overburden G-buffers
  - Decoupling of sample from fragment
    - MSAA impractical

7