Coverage, compositing and the alpha channel

Dinesh K. Pai

Textbook Chapter 16

Several slides courtesy of M. Kim

1

Today

- Coverage and alpha
- Multisample anti-aliasing
- Compositing

2

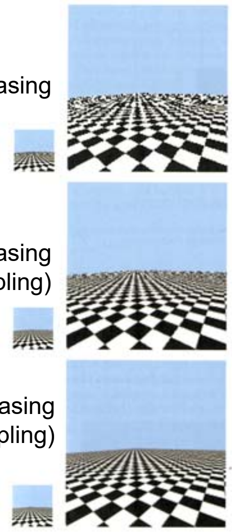
Recap: Aliasing and anti-aliasing

Different types of Oversampling

Aliasing

Anti-aliasing (multi-sampling)

Anti-aliasing (super-sampling)



Recap: Sampling in 1D

Image in 1D

November 15, 2017 10:45 AM

Continuos $I(x)$

Approximate with $I[x]$

Pixel → 4 ← Samples

Box filter: Average over a pixel

Oversampling: Approximate

$$\frac{\int I(x)}{\text{length}} \simeq \frac{\sum_{\text{samples in pixel}} I[x]}{N \text{ (number of samples)}}$$

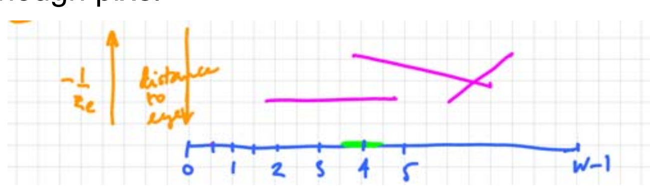
Coverage

- Rapid changes in color could be due to
  - Texture
  - Shading
  - Depth discontinuities
- Supersampling deals with all at once, but at great cost
- It may be more efficient to separately handle each of the source of color change

5

Coverage

- Texture => Pre-filtered textures, "mip mapping"
- Shading => generally changes slowly, except at edges of triangles
- Depth discontinuities => check if discontinuity passes though pixel



- Estimate partial coverage of pixel by triangle fragment
- Fraction of pixel covered is denoted alpha (α).

6

Multi-sampling

- *During the rasterization* of each triangle, "coverage" and z-values are computed at "high resolution".
- But for efficiency, the fragment shader is only called **only once per final resolution pixel.**
  - This color data is shared between all of the samples hit by the triangle in a single (final resolution) pixel.
- Once rasterization is complete, groups of these high resolution samples are averaged together.

7

Multi-sampling

- Multisampling can be an effective anti-aliasing method since, without texture mapping, colors tend to vary quite slowly over each triangle, and thus they do not need to be computed at high spatial resolution.
- To deal with aliasing that occurs during texture mapping, we have the advantage of possessing the texture image in hand at the outset of the rendering process.
- This leads to specialized techniques such as *mip mapping*.

8

Compositing?

- Example of demo reel
  http://vimeo.com/72617082

9

Overview of Compositing

- Generalize idea of anti-aliasing to representing the “coverage” of each pixel by an object
- Essential for multi-pass rendering, requiring combination of images
- Historically, related to “matte”s in film, now done using the “alpha” channel in RGBA color images
- Importance increasing due to increasing availability of digital imagery
- Widely used: Visual Effects, “Sprites” in games, etc. Natively supported in most OS’s for GUI

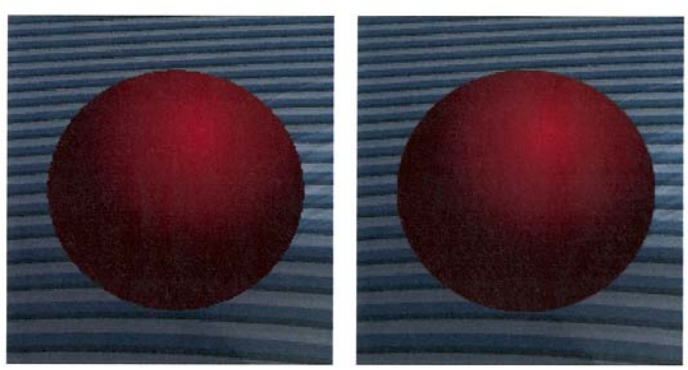
10

Image compositing

- Given two discrete images, a foreground, I^f , and background, I^b, that we want to combine into one image I^c .
- Simple: in composite, use foreground pixels where they are defined. Else use background pixels.
- This will give us a jagged boundary.
- Real image would have "boundary" pixels with blended colors.
- But this requires using "sub-pixel" information.

11

Image compositing



12

Alpha blending

- Associate with each pixel in each image layer, a value, $\alpha[\mathrm{i}][\mathrm{j}]$, that describes the overall *opacity* or *coverage* of the image layer at that pixel.
  - An alpha value of 1 represents a fully opaque/occupied pixel, while a value of 0 represents a fully transparent/empty one.
  - A fractional value represents a partially transparent (partially occupied) pixel.
- Alpha will be used during compositing.

13

Alpha definition

- More specifically, let $I(x,y)$ be a continuous image, and let $C(x,y)$ be a binary valued (x,y) *coverage function* over the continuous domain, with a value of 1 at any point where the image is “occupied” and 0 where it is not.
- Let us store in our discrete image the values:

$$\mathrm{I}[\mathrm{i}][\mathrm{j}] \leftarrow \iint_{\Omega_{i,j}} I(x,y)C(x,y)dxdy$$

$$\alpha[\mathrm{i}][\mathrm{j}] \leftarrow \iint_{\Omega_{i,j}} C(x,y)\,dx\,dy$$

14

Over operation

Note: a technical term

- To compose $I^f[i][j]$ over $I^b[i][j]$, we compute the composite image colors, $I^c[i][j]$, using

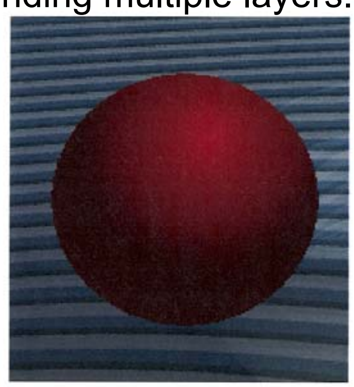
  $$I^c[i][j] \leftarrow I^f[i][j] + I^b[i][j]\,(1 - \alpha^f[i][j])$$

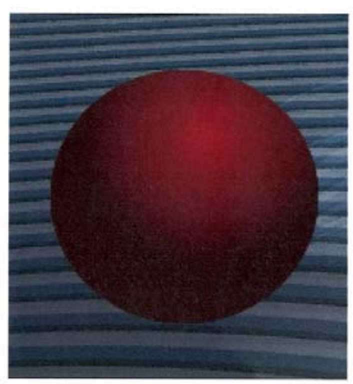
  That is, the amount of observed background color at a pixel is proportional to the transparency of the foreground layer at that pixel.
- Likewise, alpha for the composite image can be computed as:

  $$\alpha^c[i][j] \leftarrow \alpha^f[i][j] + \alpha^b[i][j]\,(1 - \alpha^f[i][j])$$

Over operation

- If background is opaque, so the composite pixel is opaque.
- But we can model more general case as part of blending multiple layers.





16

Over properties

- This provides a reasonable approximation to the correctly rendered image.
- One can easily verify that the over operation is associative but not commutative. That is,

$$I^a \text{ over } (I^b \text{ over } I^c) = (I^a \text{ over } I^b) \text{ over } I^c$$

$$I^a \text{ over } I^b \neq I^b \text{ over } I^a$$

17