Resampling

Dinesh K. Pai

Textbook Chapter 18

Several slides courtesy of M. Kim

1

Today

- Announcements
  - Quiz 3 discussion
  - A4 grading
- Resampling

2

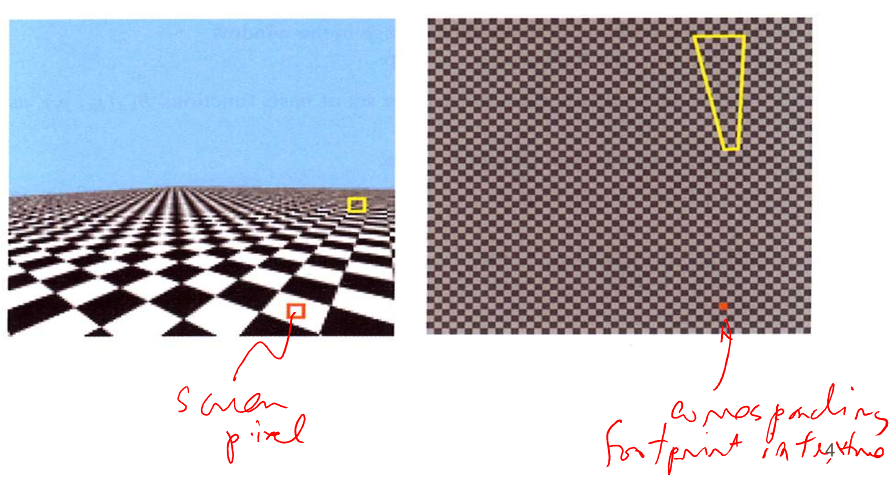
Chapter 18

RESAMPLING

(RECONSTRUCTION+SAMPLING, DISCRETE→CONTINUOUS→DISCRETE)

3

Resampling



4

Resampling

- Let's revisit texture mapping
- We start with a discrete image and end with a discrete image.
- The mapping technically involves both a reconstruction and sampling stage.
- In this context, we will explain the technique of mip mapping used for anti-aliased texture mapping.

Lance Williams

multum in parvo

5

(Textbook description) Resampling equation

- Suppose we start with a texture image (discrete) T[k][l] and apply some 2D warp to this image to obtain an output image I[i][j].
- Reconstruct a continuous texture $T(x_t, y_t)$ using a set of basis functions $B_{k,l}(x_t, y_t)$.
- Apply the geometric wrap (at the view point) to the continuous image.
- Integrate against a set of filters $F_{k,l}(x_w, y_w)$ (a box filter) to obtain the discrete output image.

6

(Textbook description) Resampling equation

- Let the geometric transform be described by a mapping $M(x_w, y_w)$ which maps from continuous window to texture coordinates.
- We obtain:

$$\mathrm{I[i][j]} \leftarrow \iint_{\Omega} F_{i,j}(x_w, y_w)\left(\sum_{k,l} B_{k,l}[M(x_w, y_w)]\mathrm{T[k][l]}\right) dx_w dy_w$$

$$= \sum_{k,l} \mathrm{T[k][l]}\left(\iint_{\Omega} F_{i,j}(x_w, y_w)\left(B_{k,l}[M(x_w, y_w)]\right) dx_w dy_w\right)$$

(we could obtain an output pixel as a linear combination of the input texture pixels.)

Skip

7

Key Intuition in 1D

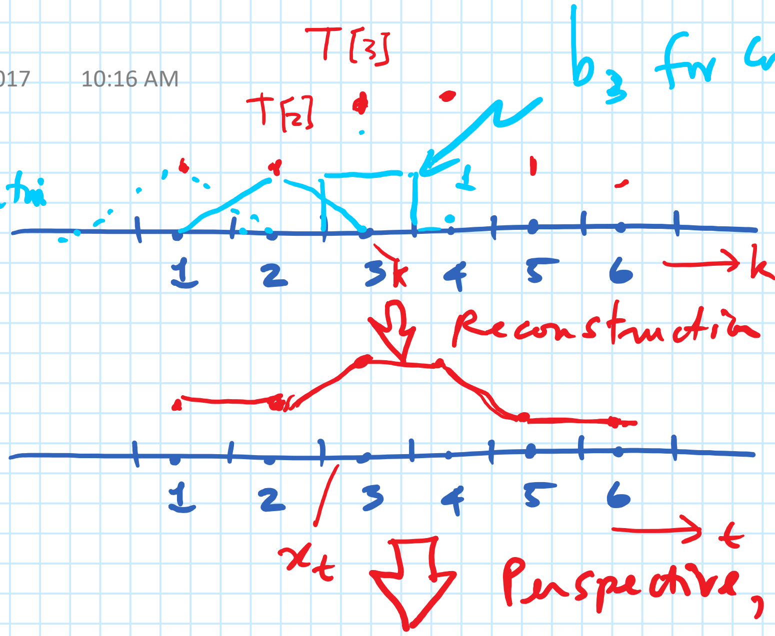
- Switch to tablet

8

November 24, 2017 10:16 AM

b_2 for linear interpolation

b_3 for constant interpolation

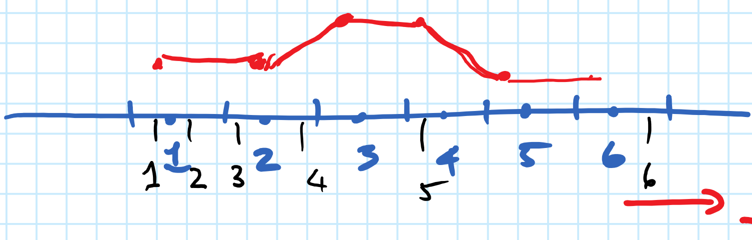


Texture Image $T[k]$

Linear

$$T(x_t) = \sum_k b_k(x_t)\, T[k]$$

Perspective, View, Viewport



→ w window coord

Magnification

Interpolation

OpenGL

- NEAREST (constant interpolation) — 1 texel access
- LINEAR — 4 texels access

Minification

Interpolation

Filter & Sample

MIP maps

Prefilter the texture

8 texels

- Nearest
- Linear (Tri)

Magnification

- We tell OpenGL to do this using the call glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR).
- In Three.js set Texture.magFilter to THREE.LinearFilter (default)
- For a single texture lookup in a fragment shader, the hardware needs to fetch 4 texture pixels and blend them appropriately.

9

Minification

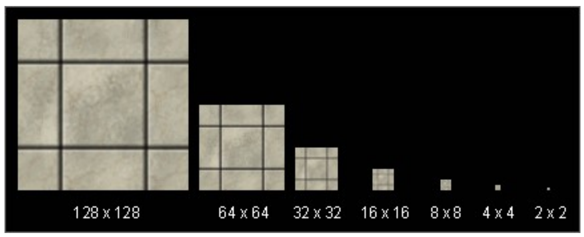
- In the case that a texture is getting shrunk down, then, to avoid aliasing, the filter component should not be ignored.
- Unfortunately, there may be numerous texture pixels under the footprint of $M(\Omega_{i,j})$, and we may not be able to do our texture lookup in constant time.

10

Mip mapping

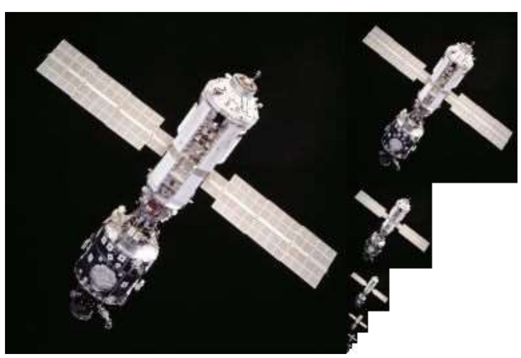
- In mip mapping, one starts with an original texture T^0 and then creates a series of lower and lower resolution (blurrier) texture T^i.
- Each successive texture is twice as blurry. And because they have successively less detail, they can be represented with ½ the number of pixels in both the horizontal and vertical directions.

for anti-aliasing?



11

Mipmap example

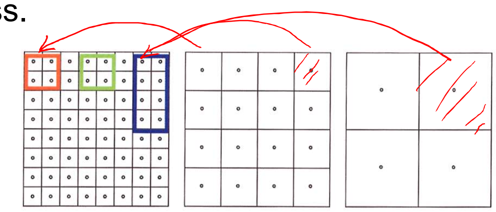


Source: wikipedia

12

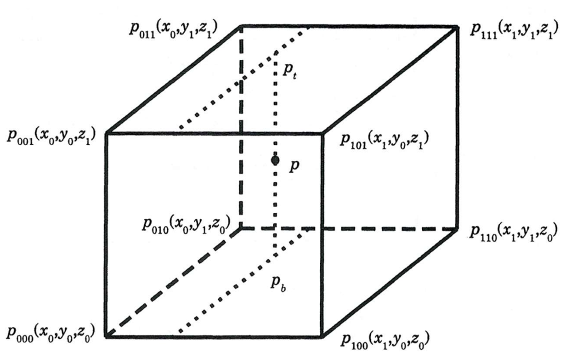
Mip mapping

- In OpenGL/WebGL Mip mapping with trilinear interpolation is specified with the call glTexParameteri(GL TEXTURE 2D, GL TEXTURE MIN FILTER,GL LINEAR MIPMAP LINEAR)
- In Three.js set Texture.minFilter to THREE.LinearMipMapLinearFilter
- Trilinear interpolation requires OpenGL to fetch 8 texture pixels and blend them appropriately for every requested texture access.



13

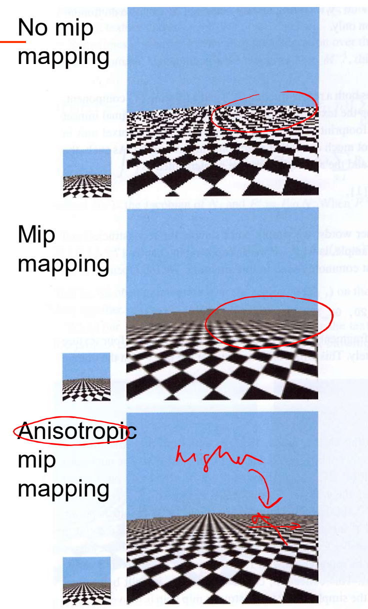
Trilinear interpolation



14

Properties

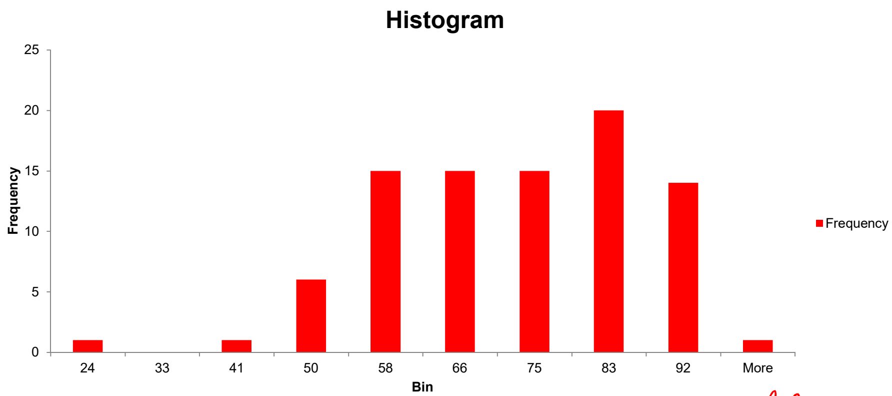
- It is easy to see that mip mapping works reasonably well, but has limitations that can be addressed by more advanced methods.



- Next class: Course review

16

Quiz 3 performance*



Mean	68.8
Median	70
Mode	82.2
Maximum	100

will add 2 point (not [illegible])

* Raw scores, missing 2 entries

17

Quiz 3 solutions

- Q1: 9,16,10,5,2,18,15
- Q2: T,F,T,T

18

Quiz 3 solutions

- Q3 (these example answers. Equivalent statements are acceptable.
  a. key point is that the output is still in clip coordinates, and you have to do the “perspective divide” yourself
  b. (from Lecture)

deg 3 | $(1-t)^3$ $3(1-t)^2 t$ $3(1-t)t^2$ t^3

19

- Q3 continued
  - c. from L27, the viewport matrix:

$$\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} W/2 & 0 & 0 & (W-1)/2 \\ 0 & H/2 & 0 & (H-1)/2 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix}$$

substitute W=512 H=256 (Ok if you exchanged W and H)

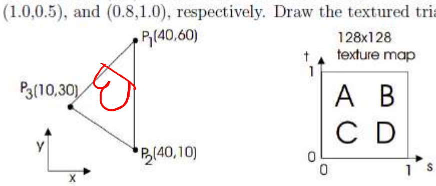
20

- Q3 continued
  - d. pixel A: 3/8 = ¾*0.5, B: 0
    many forgot to multiply by intensity (0.5)
    B = 1 acceptable if assumption of background stated
  - e. Key point is that the resolution of the depth buffer (z buffer) is limited, and what you are seeing is the aliasing in sampling the depth coarsely.

21

- Q4

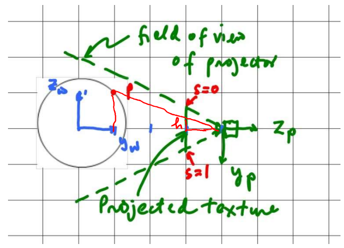
triangle. The (s, t) texture coordinates of P_1, P_2, and P_3 are (0.5,0.5), (1.0,0.5), and (0.8,1.0), respectively. Draw the textured triangle.



most people got this right.
-1 for not flipping B

22

- Q6 $\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & -4 \\ 0 & 0 & 1 \end{bmatrix}$ Not Viewport
  - a. [0 -1 0; 1 0 -4; 0 0 1] (for 2D view matrix) many did not have a rotation part. Some didn't have the position oriented properly
  - B. answer = 1/6. Many drew figure but didn't have right logic.



$\frac{h}{1} = \frac{1}{3}$

Simpler explanation than in class: Observe that the edge of the slide marked "s=0" is at height 0.5. So the distance from there the intersection of the ray is s= 1/2 - 1/3 = 1/6