Resampling, Exam prep

Dinesh K. Pai

Textbook Chapter 18

Several slides courtesy of M. Kim

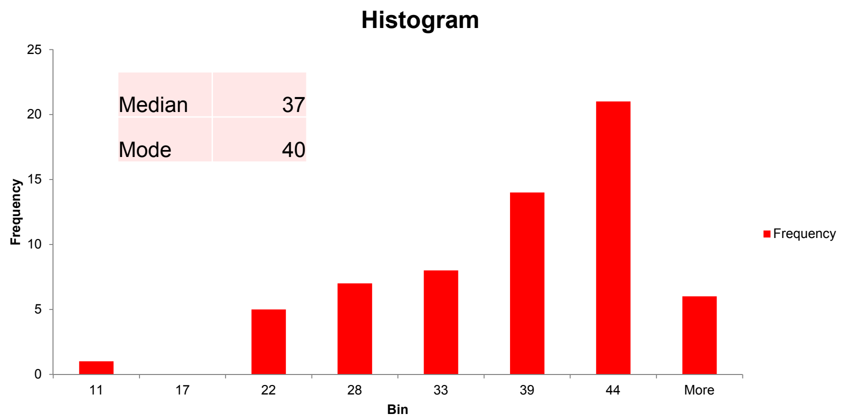
1

Today

- Announcements
  - Face-to-face grading reminder. Must finish this week
- TA evaluation
- Grade statistics, so far
- Exam prep
- Resampling

2

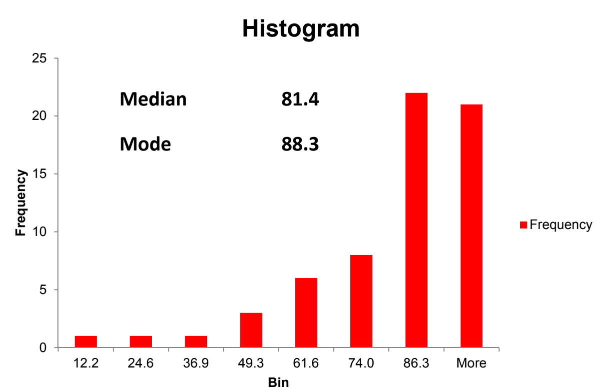
Quiz 3



3

Performance so far (A1:A3, Q1:Q3, 63% of total)

- Very good performance so far, great job!
- But you can still improve your grade
- Final exam is worth 27%



4

Exam Format

- The exam will be similar to quizzes, but longer. Closed book, closed electronic device (laptops, phones, etc. should be out of sight).
- 150 marks (in 150 minutes)
- Three types of questions
  - small questions (fill in the blank, many choices given) "Can you recognize the concepts?"
  - direct questions (write down short answer) "Do you understand the concepts?"
  - problem solving questions "Can you use your knowledge in a new situation?"

5

Exam Format

- The first two question types are meant to be easy. Try to go through them quickly, so that you have time to think about the problem solving questions at the end.
- Some questions may have multiple parts that build on one another. You can get credit for later parts if you show your steps

6

Exam Format (changes)

- New: you may be asked what a small piece of code does.
  - This will be similar to code you have seen in your assignments
- New(ish): You may be asked to write small program fragments. Exact syntax is not important, but conceptual understanding is. E.g., you should know different types of data you can pass to shaders, and how to do that from an Three.js program. Straightforward if you understood what you did in the assignments.

7

Exam Preparation

- Review lecture notes, text, and assignments
- Everything covered in lecture could be on the exam
- Everything covered in listed textbook chapters could be on the exam
- Practice:
  - Take a look at the quizzes, and make sure you understand the techniques, not just the answers. Try to do them as a test. Some questions will be variations of quiz questions.
  - I will provide some practice problems for parts after Quiz3 today. Try the questions yourself first. Solutions posted in a week on Piazza.
- Extra office hours: 1-2pm April 15, 16, 17, in 005 lab

8

Textbook reading

- Read ALL of Chapters 1-18 and Appendix A, except as noted below
  - Skip all of Chapters 7,8,13
  - Ch 2: skip Eq. 2.5
  - Ch 10,11: focus mainly on lecture notes. Use text for clarification. skip 10.3.2, 10.3.3, 11.2.1,
  - Ch 12: skip 12.2, 12.4
  - Ch 18: Understand concepts. No need to memorize the resampling equation. See lecture notes.

9

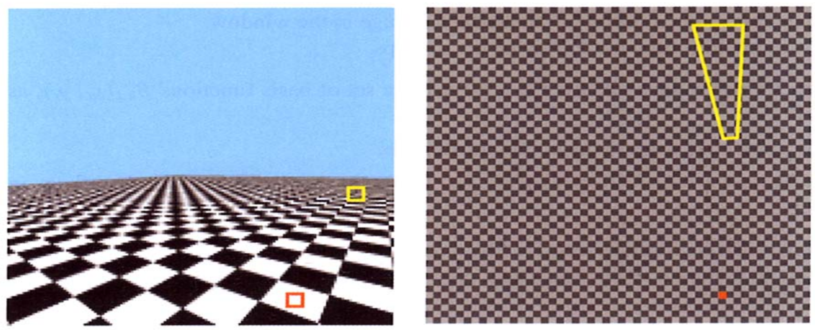
Chapter 18

RESAMPLING

(RECONSTRUCTION+SAMPLING, DISCRETE→CONTINUOUS→DISCRETE)

10

Resampling



11

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.

multum in parvo

many small

12

(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.

13

(Textbook description) Resampling equation

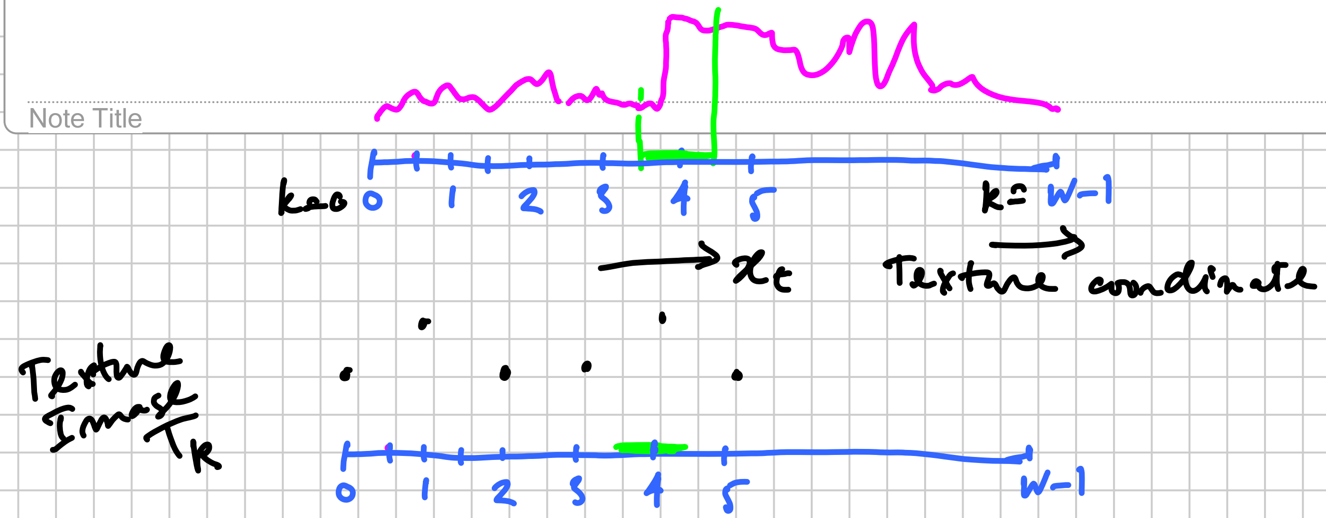
- Let the geometric transform be described by a mapping $M(x_w, y_w)$ which maps from continuous window to texture coordinates.
- We obtain:

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

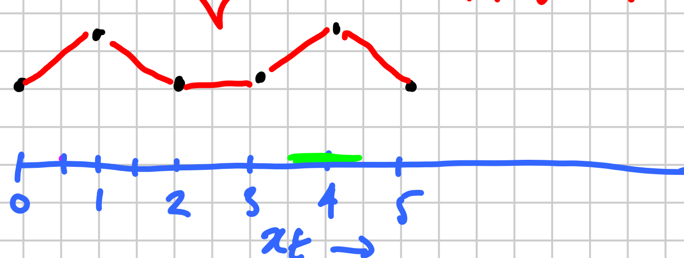
$$= \sum_{k,l} \mathrm{T[k][l]} \left(\iint_{\Omega} F_{i,j}(x_w, y_w) \left(B_{k,l}[M(x_w, y_w)] \right) dx_w dy_w \right)$$

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

14

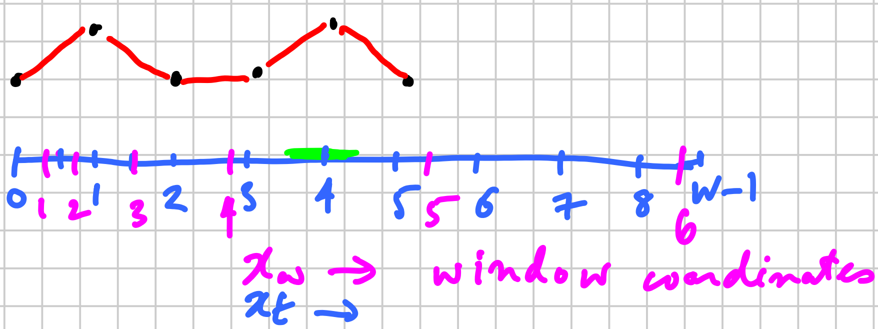


RECONSTRUCTION



$$T(x_t) = \sum b_k(x_t)\, T_k$$

Viewport, Perspective, etc



Magnification — an interpolation problem
- Nearest — 1 texel access
- Linear — 4 texel access

Minification
- Interpolation
  - Nearest
  - Linear
- Filter + Sample
  - Mipmaps
  - Pre Filter the texture
    - Nearest
    - Linear — 8 texel access

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.

16

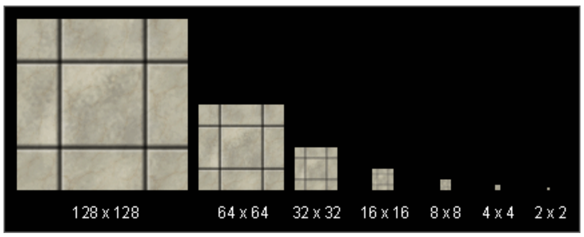
Minification

- 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.

17

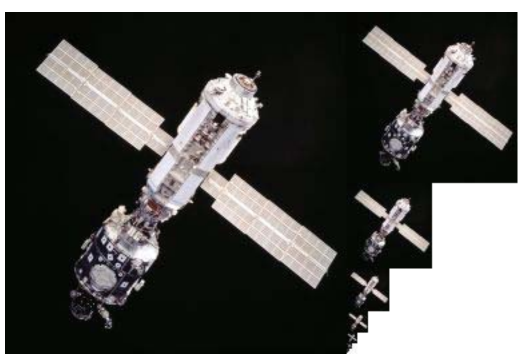
Mip mapping

- 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.



18

Mipmap example

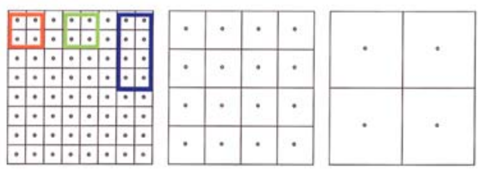


Source: wikipedia

19

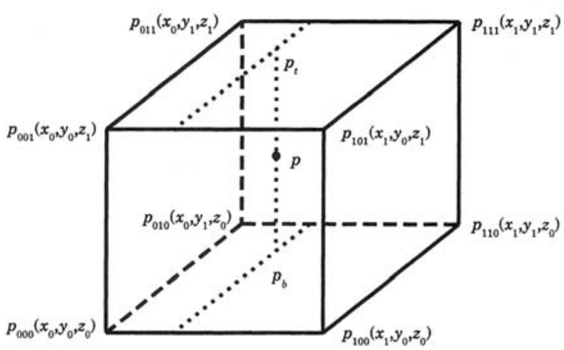
Mip mapping

- 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.



20

Trilinear interpolation



21

Properties

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