King Abdullah University of Science and Technology KAUST

CS 380 - GPU and GPGPU Programming
Lecture 15: GPU Texturing 3

Markus Hadwiger, KAUST

Reading Assignment #7+8 (until Oct 28)

Read (required):

- Interpolation for Polygon Texture Mapping and Shading,
  Paul Heckbert and Henry Moreton

`http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.7886`

- MIP-Map Level Selection for Texture Mapping

`http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=765326`

- OpenGL 4.6 Core Specification,
  Chapter 9 (Frame Buffers and Frame Buffer Objects)

`https://www.khronos.org/registry/OpenGL/specs/gl/glspec46.core.pdf`

Next Lectures

Lecture 16: Thursday, Oct 18, 14:30 (room 4138)

next week no lectures!

Lecture 17: Sunday, Oct 28, 13:00 [Quiz #3]

Lecture 18: Wednesday, Oct 31, 13:00

Lecture 19: Thursday, Nov 1, 14:30 (room 4138)

Quiz #3: Oct 28

Organization

- First 30 min of lecture
- No material (book, notes, ...) allowed

Content of questions

- Lectures (both actual lectures and slides)
- Reading assigments
- Programming assignments (algorithms, methods)
- Solve short practical examples

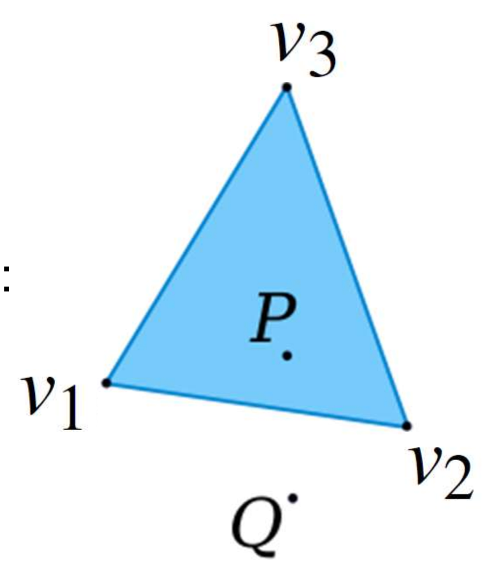
GPU Texturing

Rage / id Tech 5 (id Software)

Linear Interpolation / Convex Combinations

Linear combination:

$$\alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n = \sum_{i=1}^{n} \alpha_i v_i$$



Affine combination: restrict to $(n-1)$-dim. subspace:

$$\alpha_1 + \alpha_2 + \ldots + \alpha_n = \sum_{i=1}^{n} \alpha_i = 1$$

Convex combination: $\alpha_i \geq 0$

(restrict to simplex)

Linear Interpolation / Convex Combinations

The weights α_i are the (normalized) barycentric coordinates

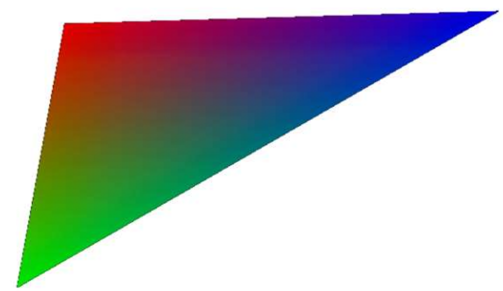
→ linear attribute interpolation in simplex

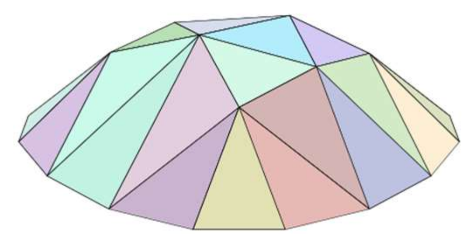
$$\alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n = \sum_{i=1}^{n} \alpha_i v_i$$

$$\alpha_1 + \alpha_2 + \ldots + \alpha_n = \sum_{i=1}^{n} \alpha_i = 1$$

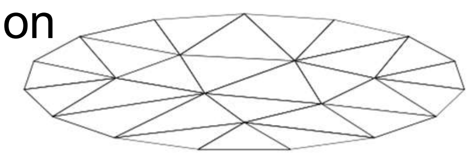
$$\alpha_i \geq 0$$

attribute interpolation





spatial position interpolation



wikipedia

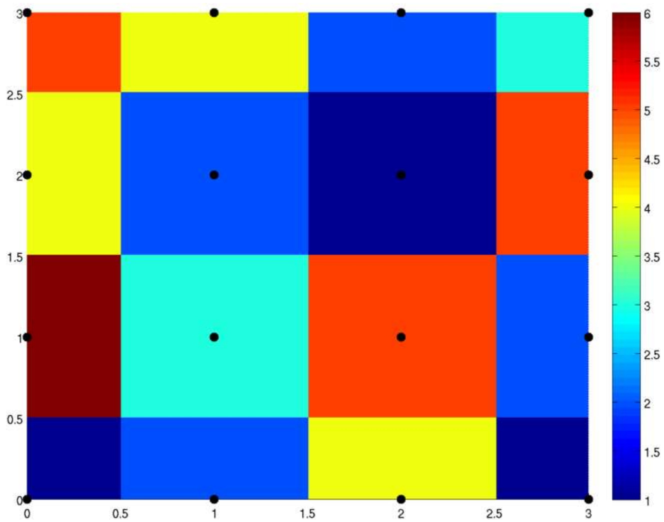
Magnification (Bi-linear Filtering Example)

Original image

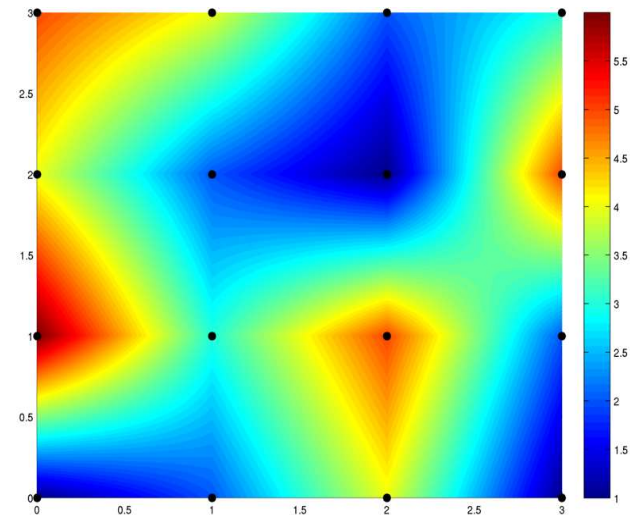
Nearest neighbor

Bi-linear filtering

Nearest-Neighbor vs. Bi-Linear Interpolation

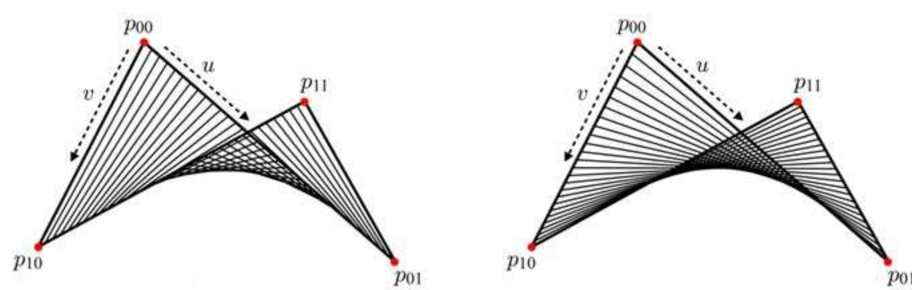


nearest-neighbor



bi-linear

wikipedia



Bilinear patch (courtesy J. Han)

Bi-Linear Interpolation

Consider area between 2x2 adjacent samples (e.g., pixel centers):

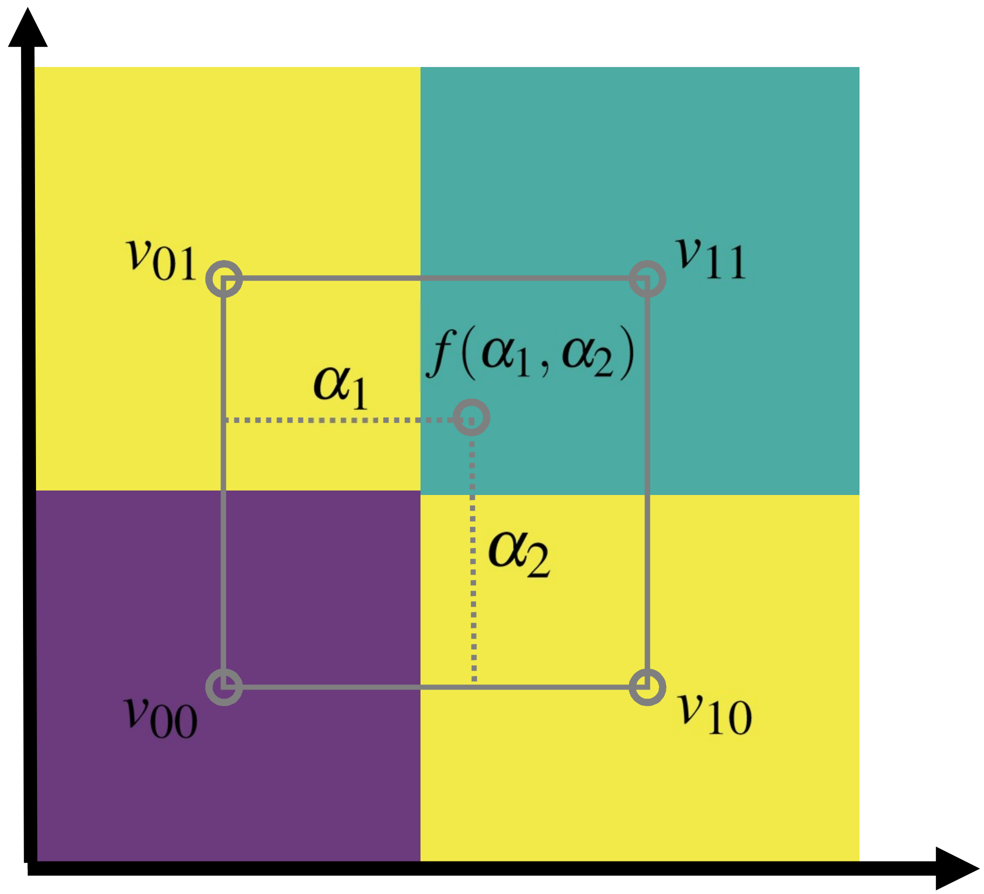
Given any (fractional) position

$$\alpha_1 := x_1 - \lfloor x_1 \rfloor \qquad \alpha_1 \in [0.0, 1.0)$$
$$\alpha_2 := x_2 - \lfloor x_2 \rfloor \qquad \alpha_2 \in [0.0, 1.0)$$

and 2x2 sample values

$$\begin{bmatrix} v_{01} & v_{11} \\ v_{00} & v_{10} \end{bmatrix}$$

Compute: $f(\alpha_1, \alpha_2)$



Bi-Linear Interpolation

Consider area between 2x2 adjacent samples (e.g., pixel centers):

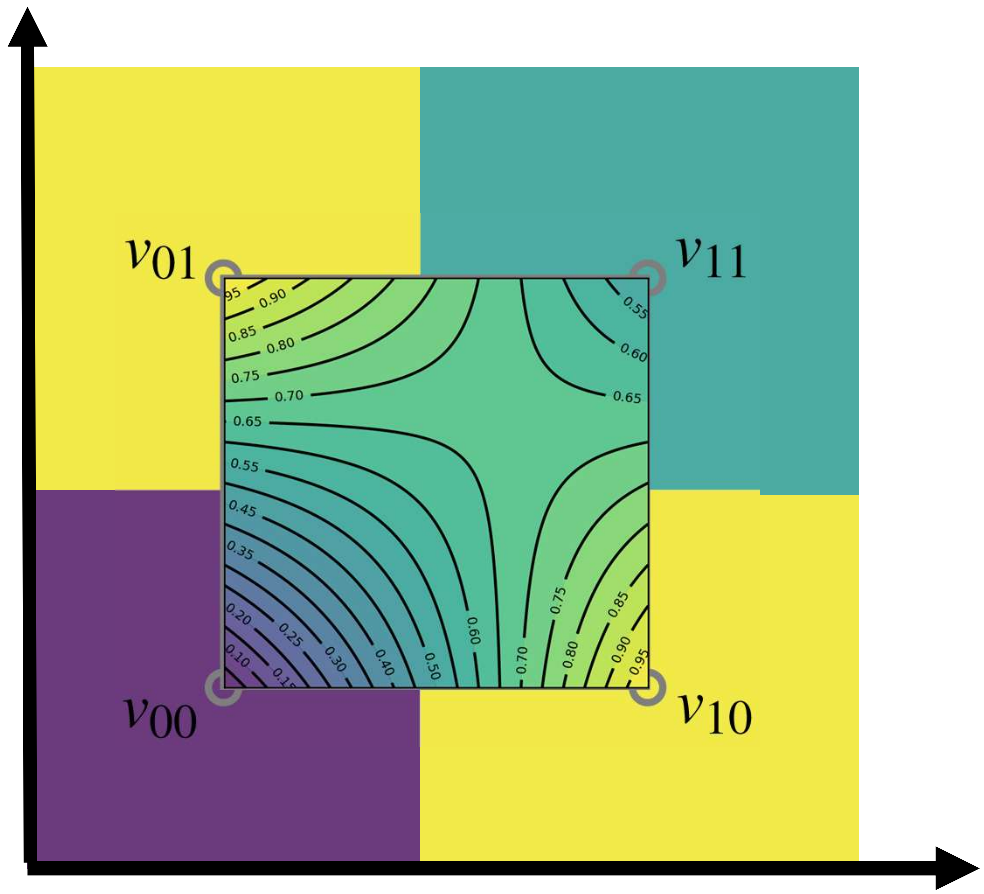
Given any (fractional) position

$$\alpha_1 := x_1 - \lfloor x_1 \rfloor \qquad \alpha_1 \in [0.0, 1.0)$$
$$\alpha_2 := x_2 - \lfloor x_2 \rfloor \qquad \alpha_2 \in [0.0, 1.0)$$

and 2x2 sample values

$$\begin{bmatrix} v_{01} & v_{11} \\ v_{00} & v_{10} \end{bmatrix}$$

Compute: $f(\alpha_1, \alpha_2)$



Bi-Linear Interpolation

Weights in 2x2 format:

$$\begin{bmatrix} \alpha_2 \\ (1-\alpha_2) \end{bmatrix} \begin{bmatrix} (1-\alpha_1) & \alpha_1 \end{bmatrix} = \begin{bmatrix} (1-\alpha_1)\alpha_2 & \alpha_1\alpha_2 \\ (1-\alpha_1)(1-\alpha_2) & \alpha_1(1-\alpha_2) \end{bmatrix}$$

Interpolate function at (fractional) position (α_1, α_2) :

$$f(\alpha_1, \alpha_2) = \begin{bmatrix} \alpha_2 & (1-\alpha_2) \end{bmatrix} \begin{bmatrix} v_{01} & v_{11} \\ v_{00} & v_{10} \end{bmatrix} \begin{bmatrix} (1-\alpha_1) \\ \alpha_1 \end{bmatrix}$$

Bi-Linear Interpolation

Interpolate function at (fractional) position (α_1, α_2) :

$$f(\alpha_1, \alpha_2) = \begin{bmatrix} \alpha_2 & (1-\alpha_2) \end{bmatrix} \begin{bmatrix} v_{01} & v_{11} \\ v_{00} & v_{10} \end{bmatrix} \begin{bmatrix} (1-\alpha_1) \\ \alpha_1 \end{bmatrix}$$

$$= \begin{bmatrix} \alpha_2 & (1-\alpha_2) \end{bmatrix} \begin{bmatrix} (1-\alpha_1)v_{01} + \alpha_1 v_{11} \\ (1-\alpha_1)v_{00} + \alpha_1 v_{10} \end{bmatrix}$$

$$= \begin{bmatrix} \alpha_2 v_{01} + (1-\alpha_2)v_{00} & \alpha_2 v_{11} + (1-\alpha_2)v_{10} \end{bmatrix} \begin{bmatrix} (1-\alpha_1) \\ \alpha_1 \end{bmatrix}$$

Bi-Linear Interpolation

Interpolate function at (fractional) position (α_1, α_2) :

$$f(\alpha_1, \alpha_2) = \begin{bmatrix} \alpha_2 & (1-\alpha_2) \end{bmatrix} \begin{bmatrix} v_{01} & v_{11} \\ v_{00} & v_{10} \end{bmatrix} \begin{bmatrix} (1-\alpha_1) \\ \alpha_1 \end{bmatrix}$$

$$= (1-\alpha_1)(1-\alpha_2)v_{00} + \alpha_1(1-\alpha_2)v_{10} + (1-\alpha_1)\alpha_2 v_{01} + \alpha_1 \alpha_2 v_{11}$$

$$= v_{00} + \alpha_1(v_{10} - v_{00}) + \alpha_2(v_{01} - v_{00}) + \alpha_1 \alpha_2(v_{00} + v_{11} - v_{10} - v_{01})$$

REALLY IMPORTANT:

this is a different thing (for a different purpose) than the linear (or, in perspective, rational-linear) interpolation of texture coordinates!!

Thank you.