King Abdullah University of Science and Technology

KAUST

AMCS / CS 247 – Scientific Visualization
Lecture 9: Iso-Surface Shading, Pt. 3

Markus Hadwiger, KAUST

Reading Assignment #5 (until Sep 22)

Read (required):

- Real-Time Volume Graphics, Chapter 2 (*GPU Programming*)
- Real-Time Volume Graphics, Chapter 5.3 (*Gradient-Based Illumination*)
- Real-Time Volume Graphics, Chapter 5.4.1 (*Blinn-Phong Illumination*)

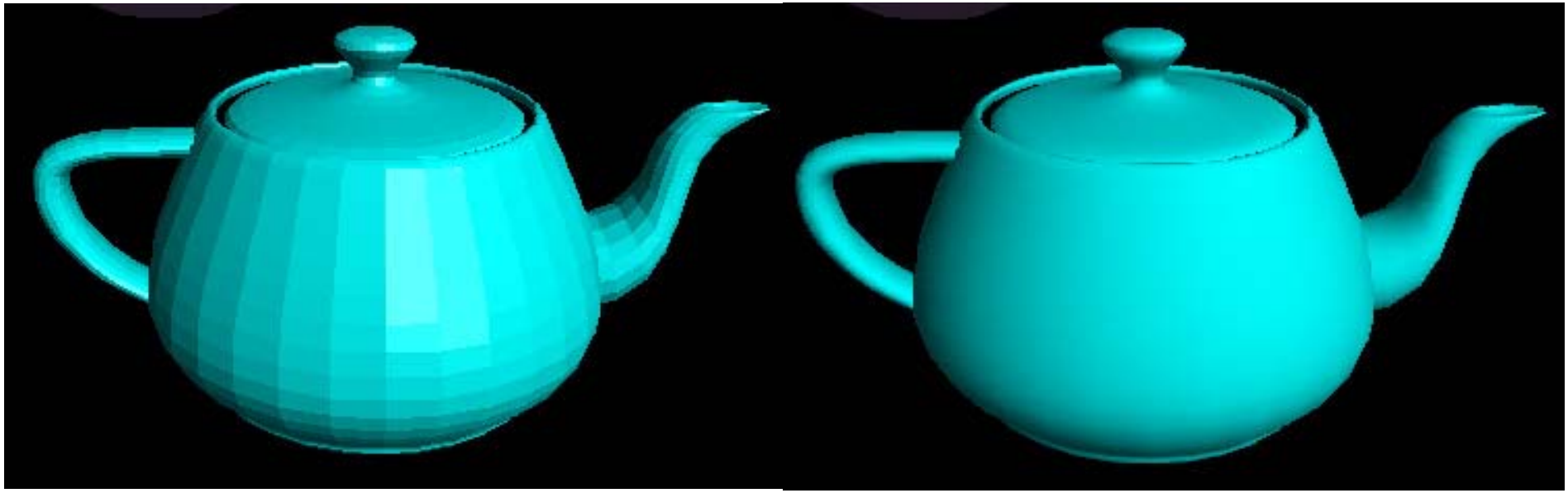
Polygon Shading Methods

Application of illumination model to polygon rendering

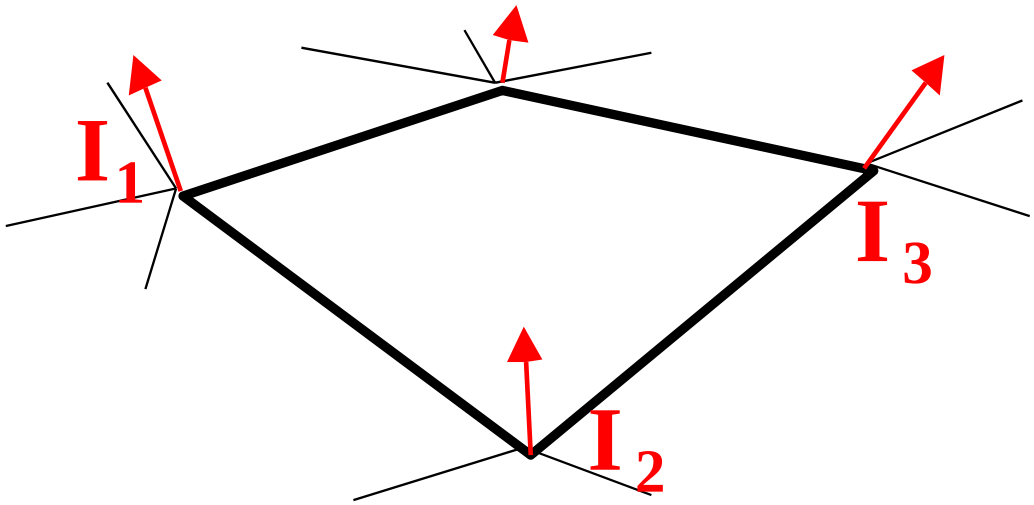
Constant-intensity shading (flat shading)

- single intensity for each polygon

flat ***Gouraud***

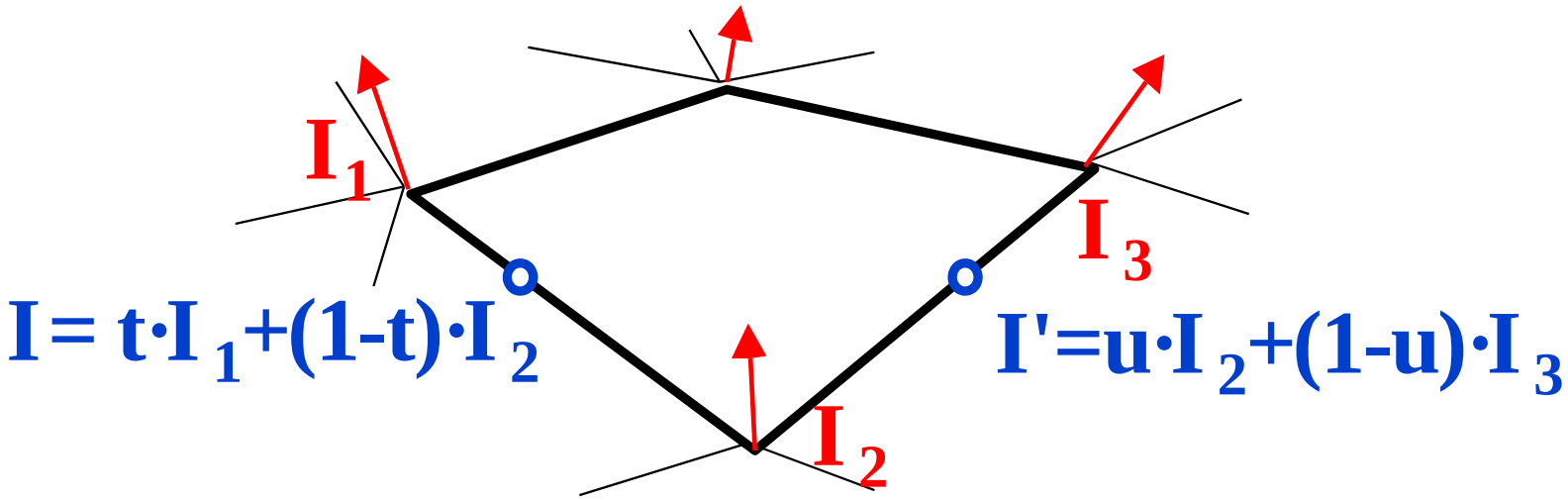


Gouraud Shading



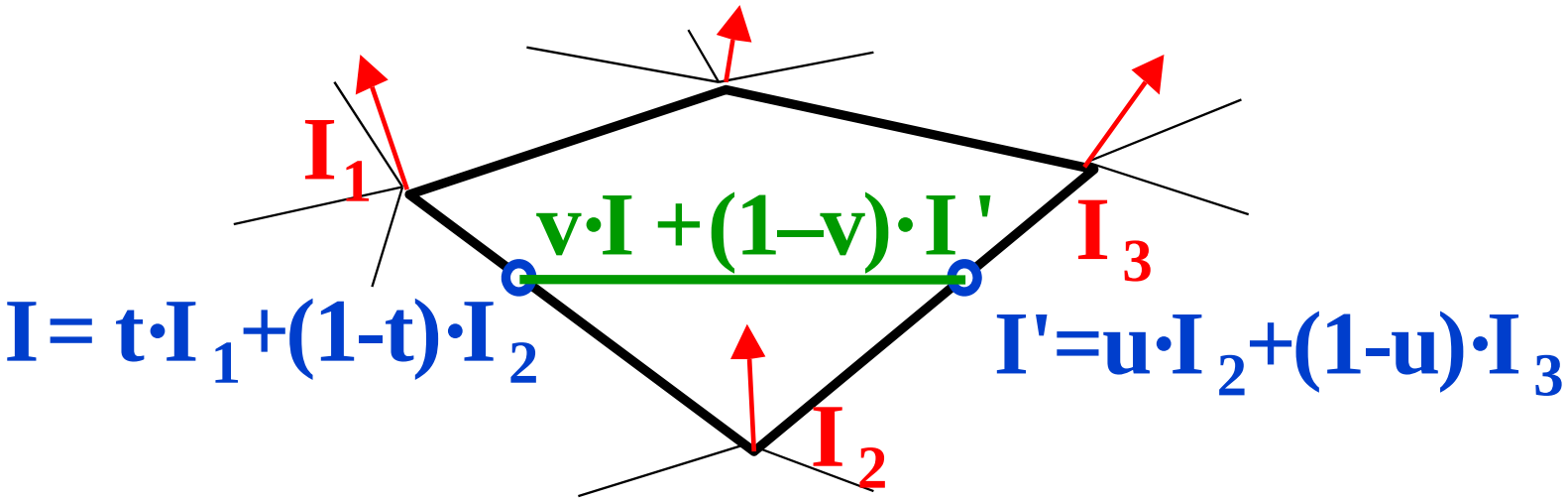
1. find normal vectors at corners and calculate shading (intensities) there: I_i

Gouraud Shading



1. find normal vectors at corners and calculate shading (intensities) there: I_i
2. interpolate intensities along the edges linearly: I, I'

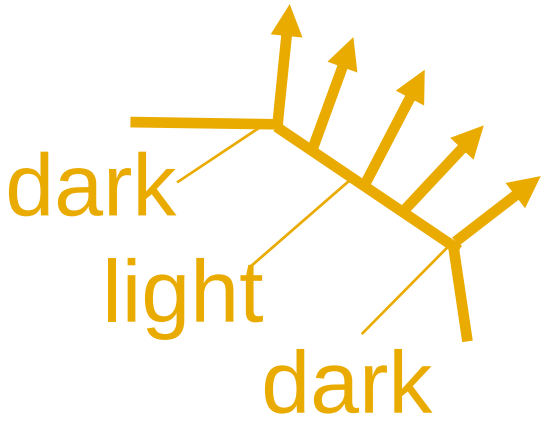
Gouraud Shading



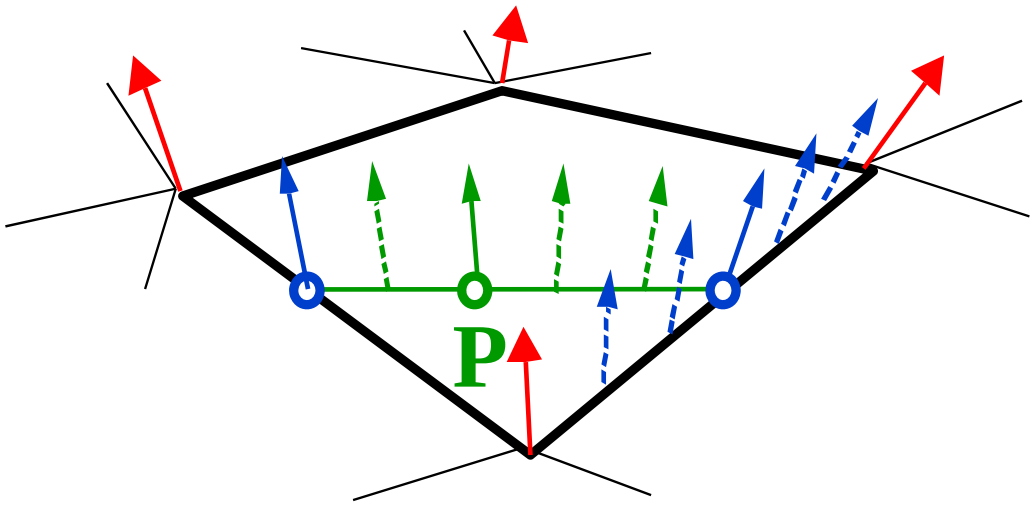
1. find normal vectors at corners and calculate shading (intensities) there: I_i
2. interpolate intensities along the edges linearly: I, I'
3. interpolate intensities along scanlines linearly: I_p

Phong Shading

Instead of intensities the normal vectors are interpolated, and for every point the shading calculation is performed separately



Phong Shading Overview

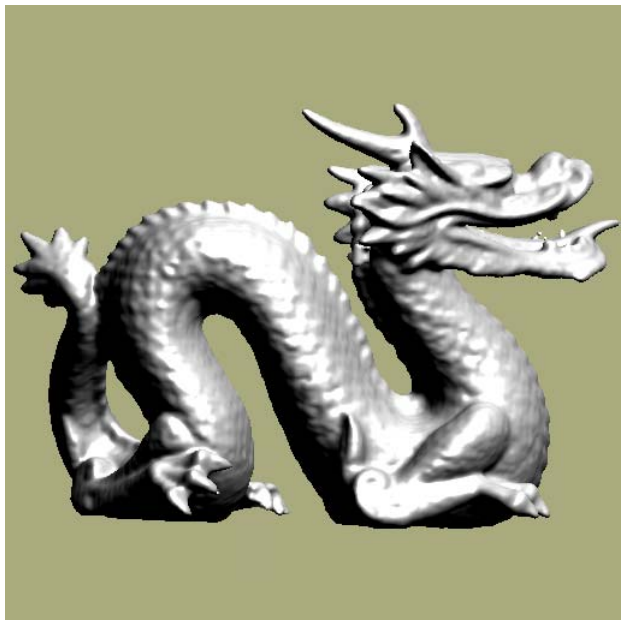


1. normal vectors at corner points
2. interpolate normal vectors along the edges
3. interpolate normal vectors along scanlines
   & calculate shading (intensities) for every pixel

Flat/Gouraud/Phong Comparison

What About Volume Illumination?

Crucial for perceiving shape and depth relationships



this is a scalar volume (3D distance field)!

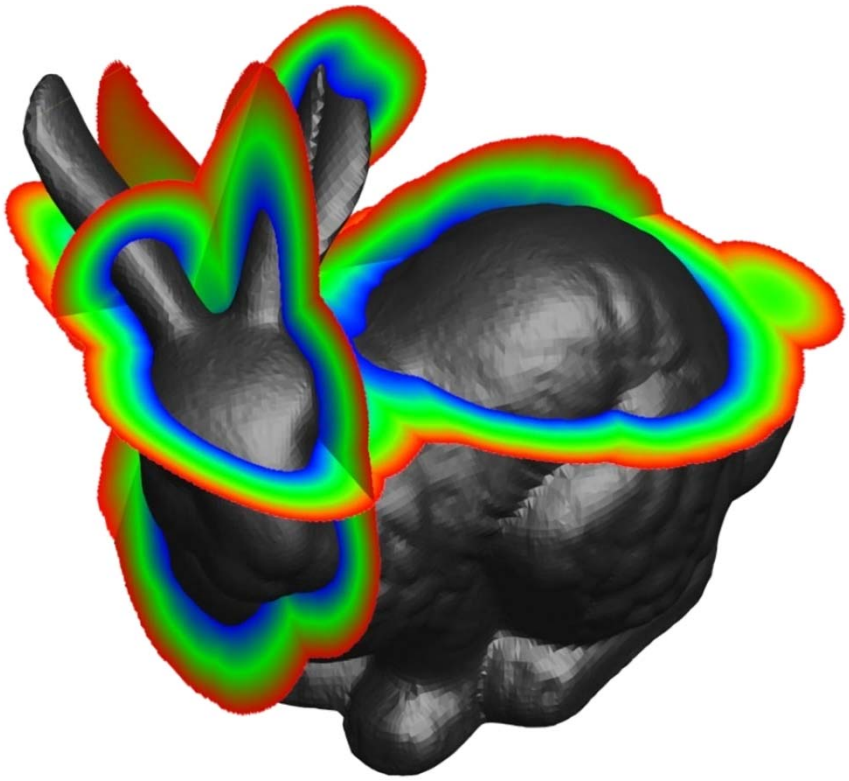
Signed Distance Fields

Special case of general volume / scalar field for objects

Volume discretizes embedding

$$\phi : \mathbb{R}^3 \mapsto \mathbb{R}$$

Signed distance to surface



Implicit surface:
zero level set

$$S = \{\mathbf{x} | \phi(\mathbf{x}) = 0\}$$

Local Illumination in Volumes

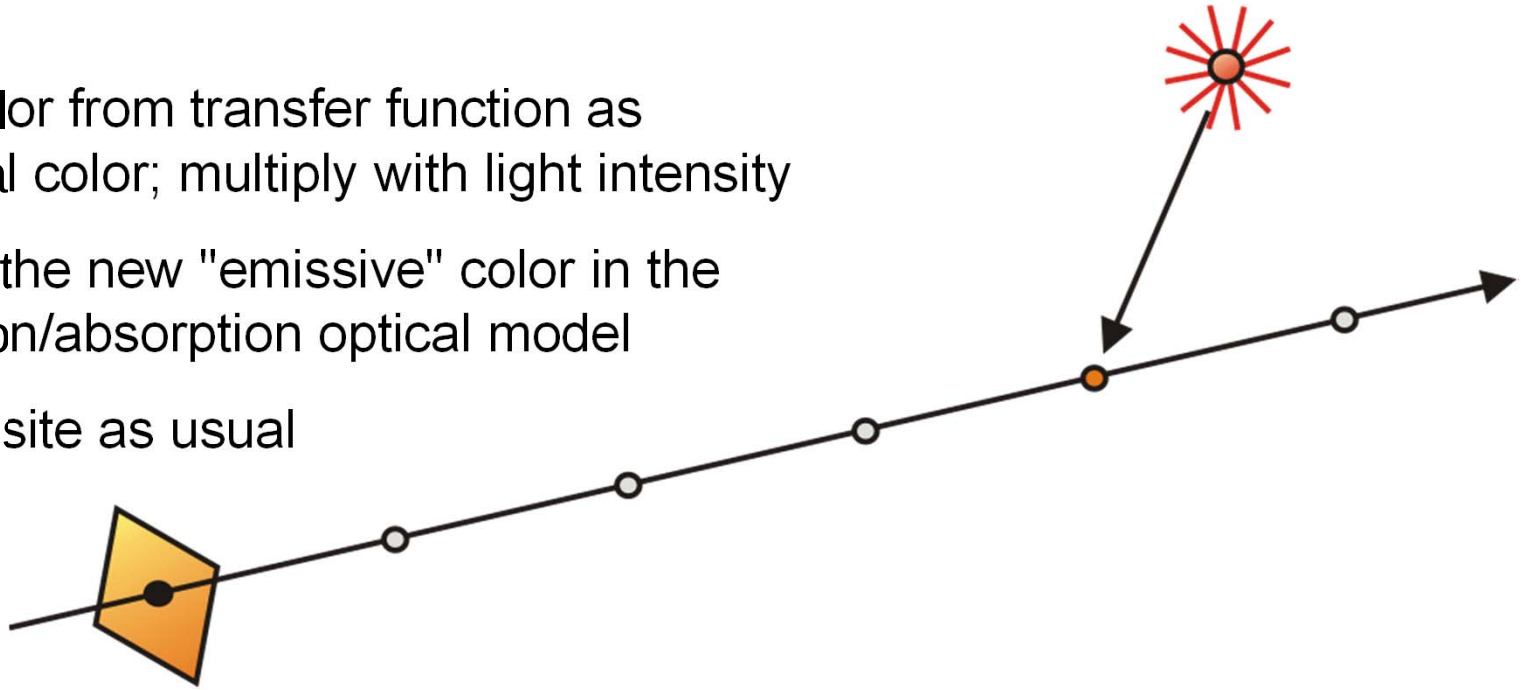
Interaction between light source and point in the volume

Local shading equation; evaluate at each point along a ray

Use color from transfer function as material color; multiply with light intensity

This is the new "emissive" color in the emission/absorption optical model

Composite as usual

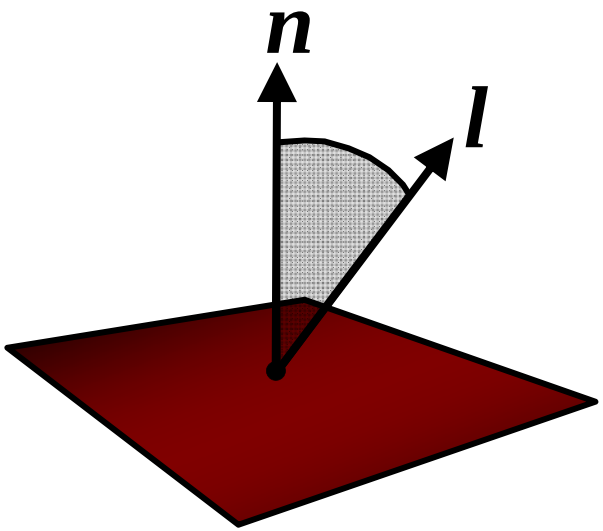


Local Shading Equations

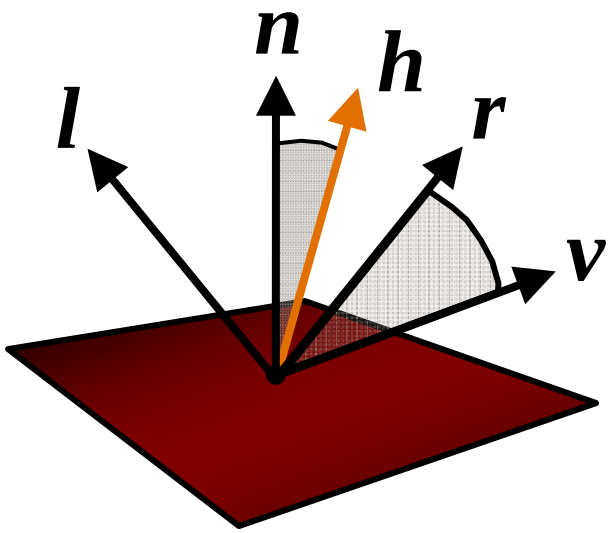
Standard volume shading adapts surface shading

Most commonly Blinn/Phong model

But what about the "surface" normal vector?



diffuse reflection



specular reflection

The Gradient as Normal Vector

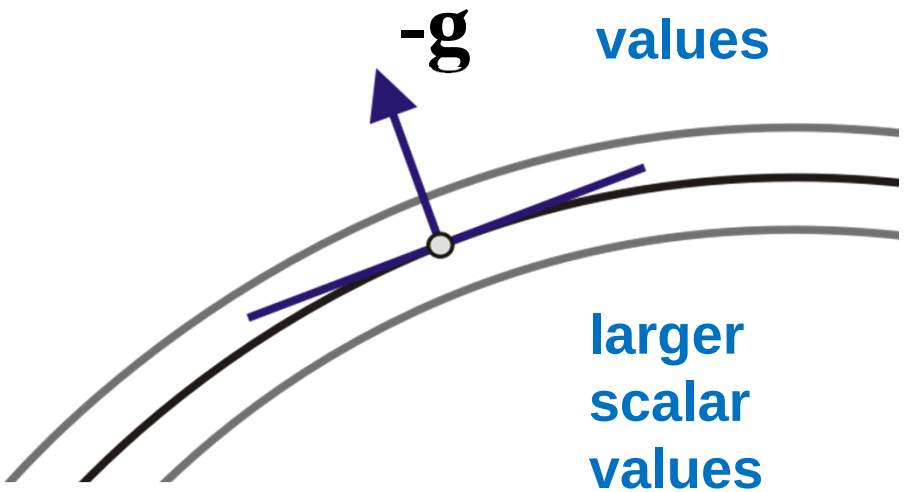
Gradient of the scalar field is direction of highest change

$$\mathbf{g} = \nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right)^{\mathbf{T}}$$

Local approximation to isosurface at any point:
tangent plane = plane orthogonal to gradient

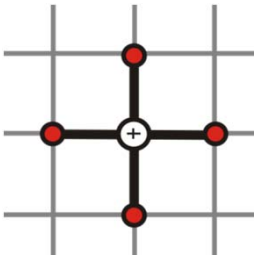
Normal of this isosurface:
normalized gradient

$$\mathbf{n} = -\mathbf{g}/|\mathbf{g}|$$



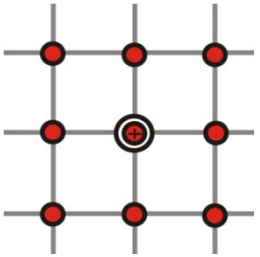
Gradient Reconstruction

We need to reconstruct the derivatives of a continuous function given as discrete samples



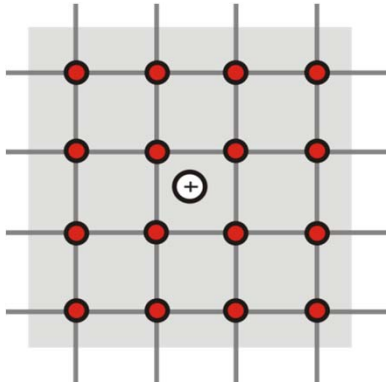
Central differences

- Cheap and quality often sufficient (2+2+2 neighbors in 3D)



Discrete convolution filters on grid

- Image processing filters; e.g. Sobel (3x3x3 neighbors)



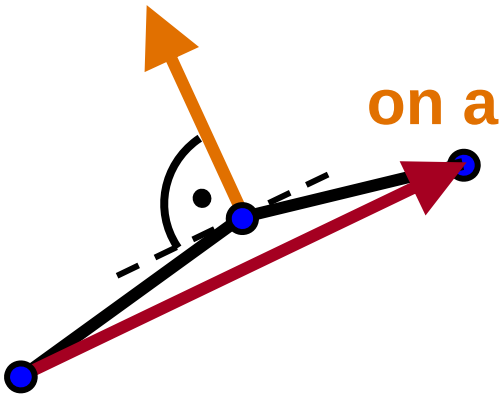
Continuous convolution filters

- Derived continuous reconstruction filters
- E.g., the cubic B-spline and its derivatives (4x4x4 neighbors)

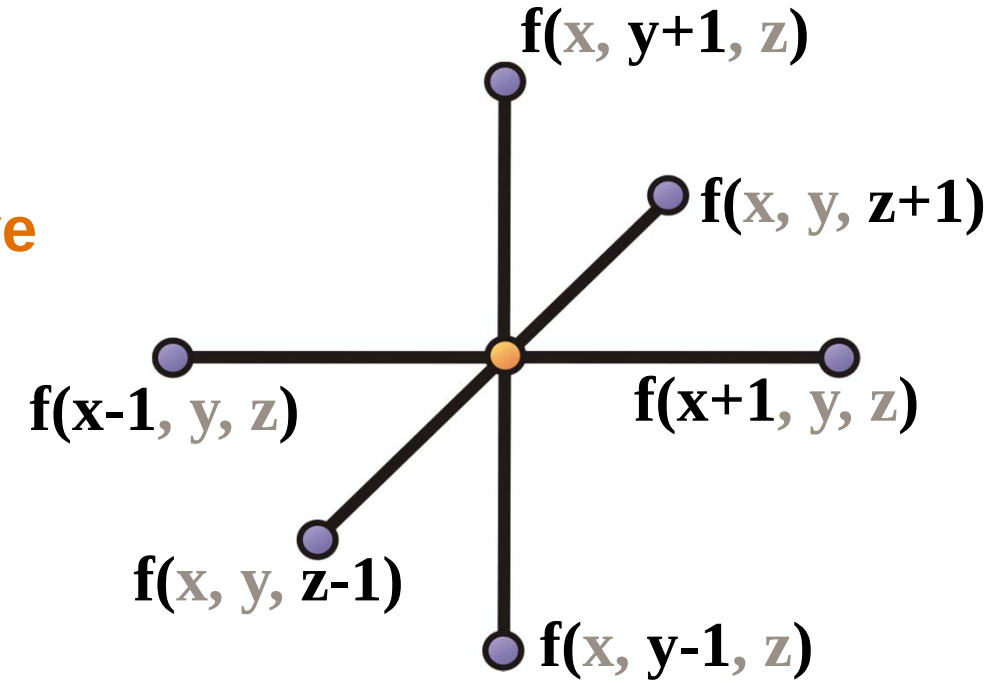
Central Differences

Need only two neighboring voxels per derivative

Most common method



on a curve



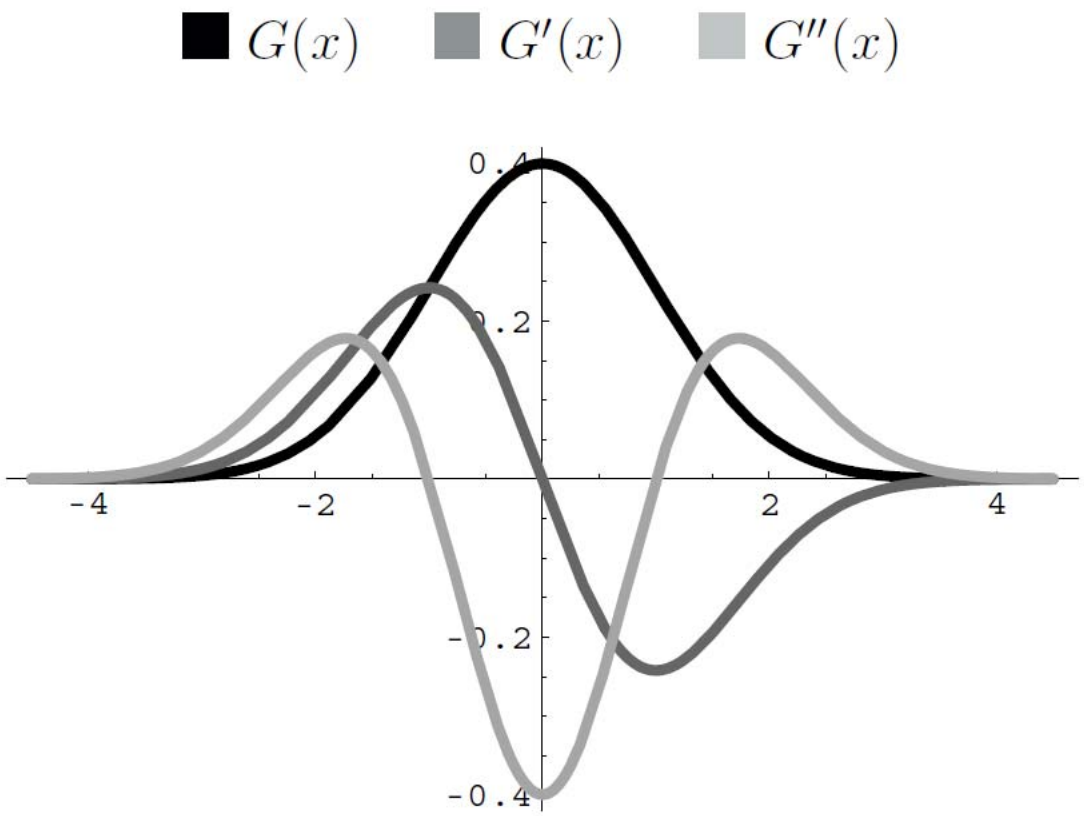
g_x = 0.5(f(x+1, y, z) – f(x-1, y, z))

g_y = 0.5(f(x, y+1, z) – f(x, y-1, z))

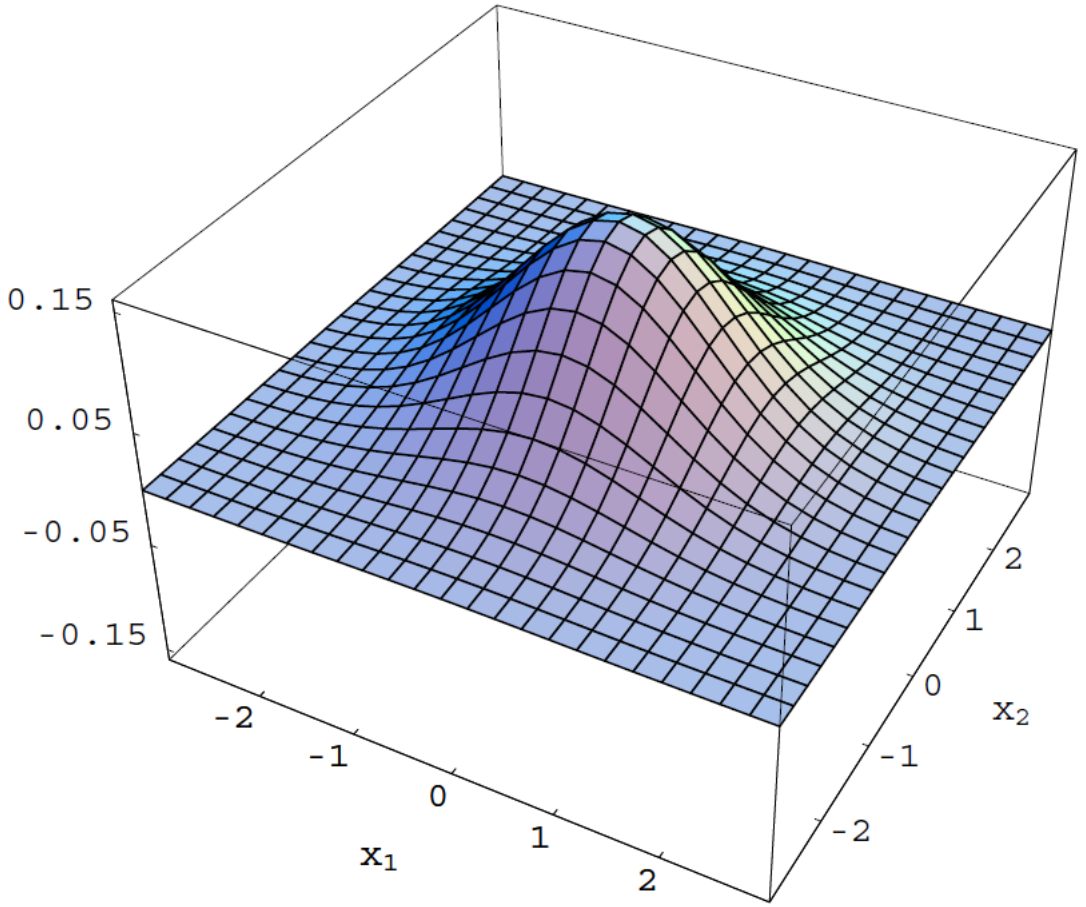
g_z = 0.5(f(x, y, z+1) – f(x, y, z-1))

in a volume

Example: Gaussian Kernel (1D)

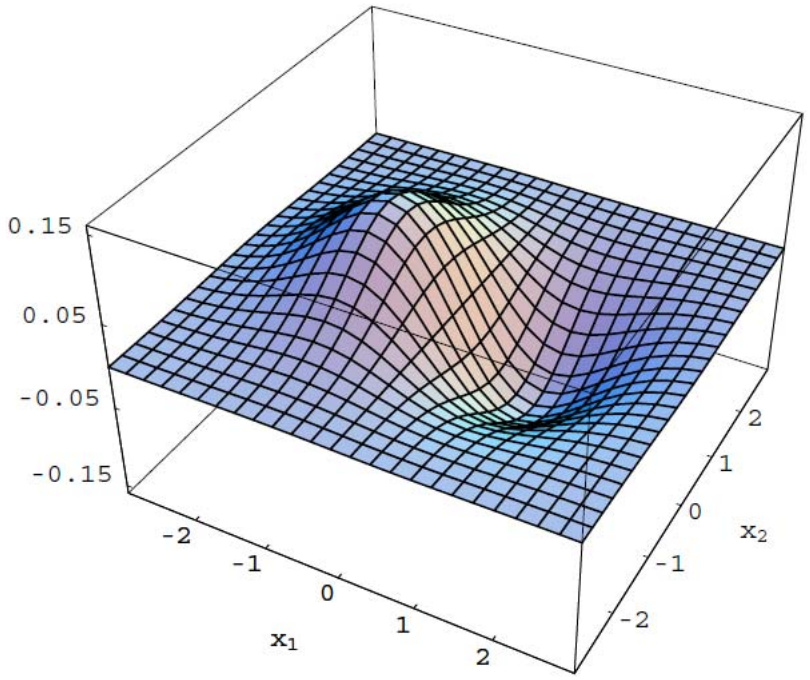


2D Gaussian Kernel

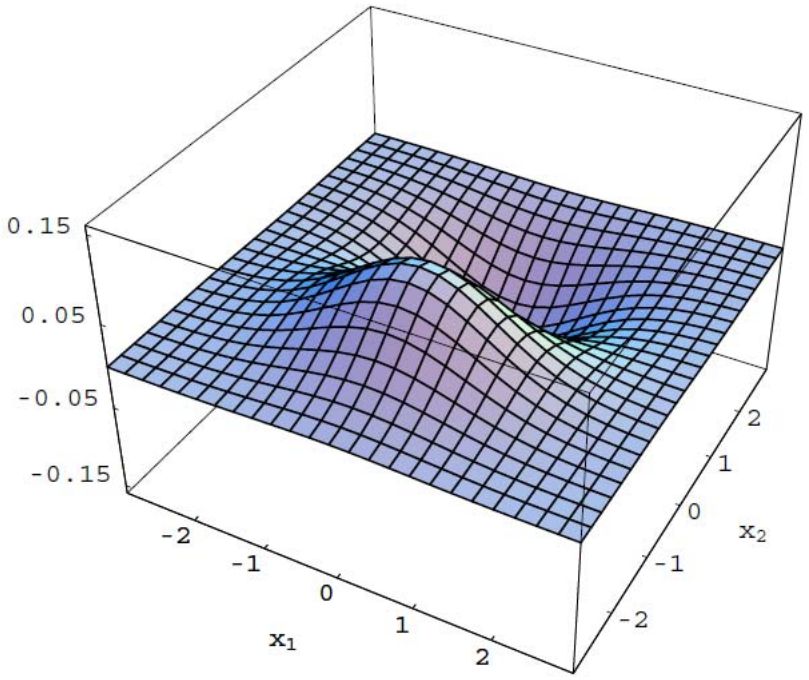


$$G(x_1, x_2)$$

2D Gaussian Derivative Kernels



$\partial G(x_1, x_2)/\partial x_1$



$\partial G(x_1, x_2)/\partial x_2$

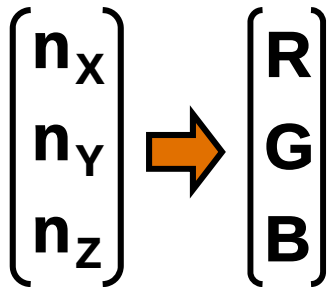
Pre-Computed Gradients

Pre-compute gradients at grid points with any method

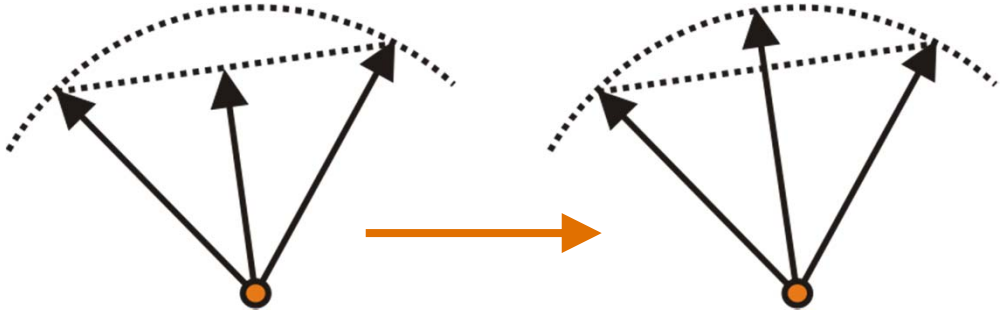
Store normalized gradient directions in RGB texture

Sample gradient texture in fragment shader: interpolation

Re-normalize after fetch!



RGB gradient texture



lerp of texture filter

renormalize!

Thank you.

Thanks for material

- Helwig Hauser
- Eduard Gröller
- Daniel Weiskopf
- Torsten Möller
- Ronny Peikert
- Philipp Muigg
- Christof Rezk-Salama