King Abdullah University of Science and Technology

KAUST

AMCS / CS 247 – Scientific Visualization
Lecture 15: Volume Visualization, Pt. 4

Markus Hadwiger, KAUST

Reading Assignment #8 (until Oct 20)

Read (required):

- Real-Time Volume Graphics, remainder of Chapter 4 (Sec. 4.5-)
- Real-Time Volume Graphics, parts of Chapter 10:
  Secs. 10.1, 10.2, 10.3, 10.6

Quiz #2: Oct 23

Organization

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

Content of questions

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

Isosurface Ray-Casting

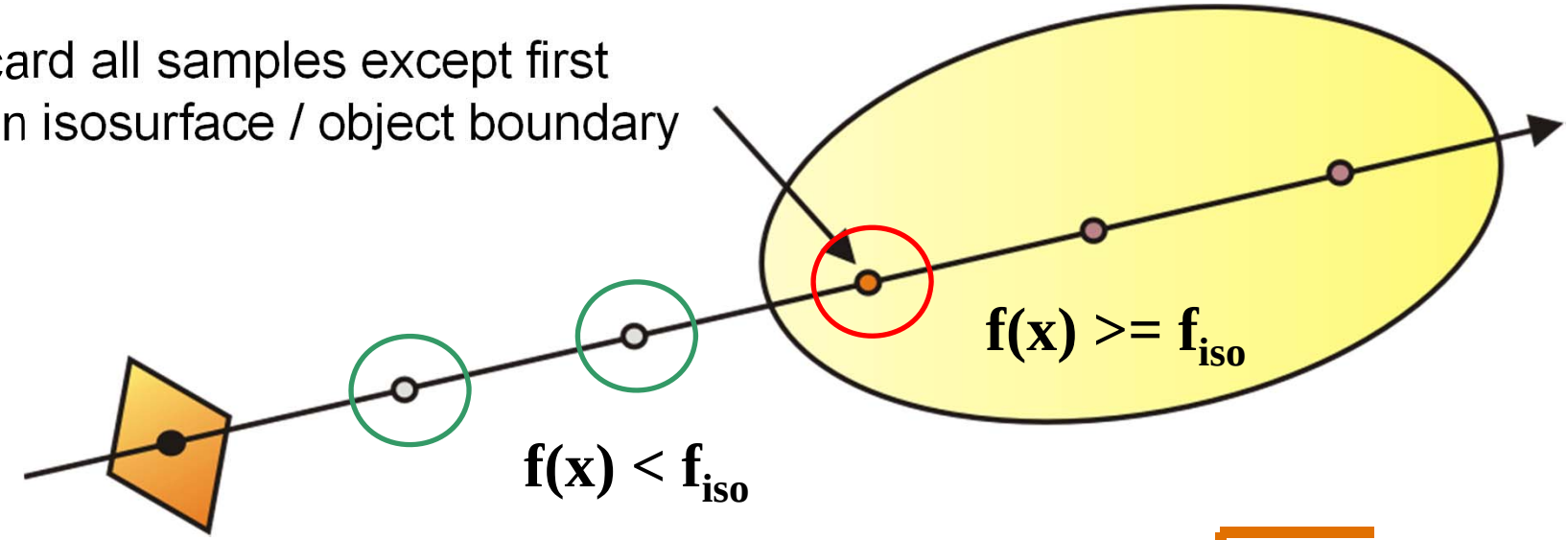
Isosurfaces/Level Sets

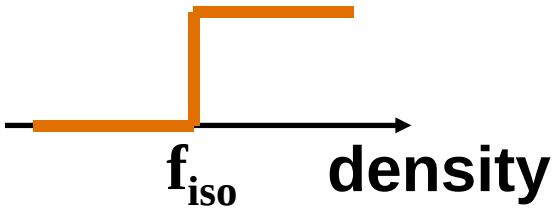
- Scanned data
- Distance fields
- CSG operations

Isosurface Ray-Casting

Opaque isosurfaces:
only one sample contributes per ray/pixel

Discard all samples except first
hit on isosurface / object boundary





Threshold transfer function / alpha test

First hit ray casting

Implementation - Isosurface Ray-Casting

Ray setup

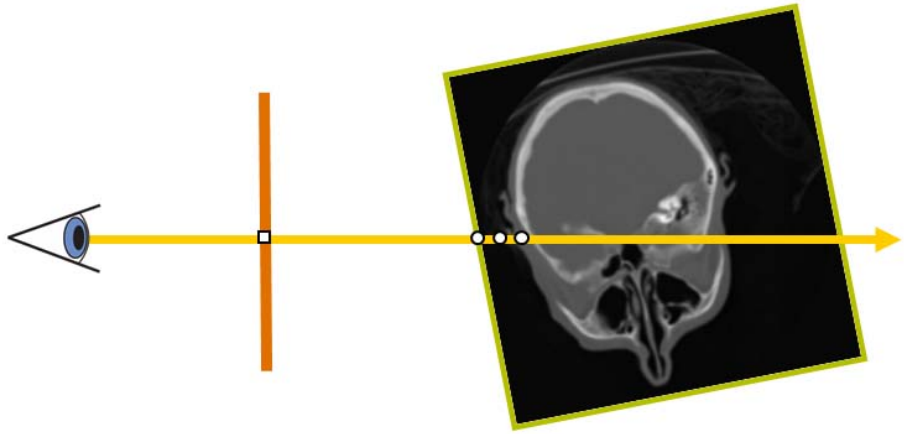
Loop over ray

- Sample scalar field
- Classification
- if sample is opaque (i.e., first hit)
  - break out of the loop

Refine first hit location

Shading

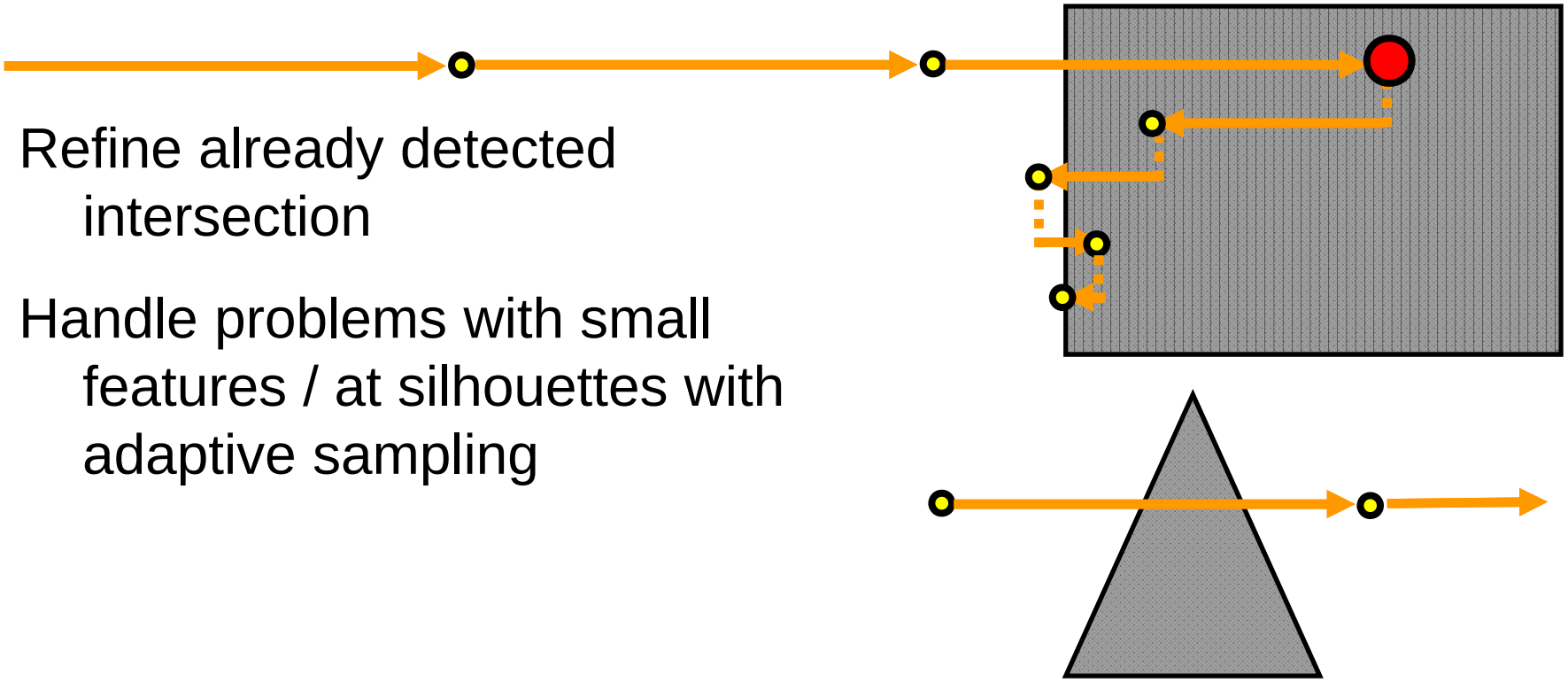
(Compositing not needed)



Intersection Refinement (1)

Fixed number of bisection steps

Virtually no impact on performance

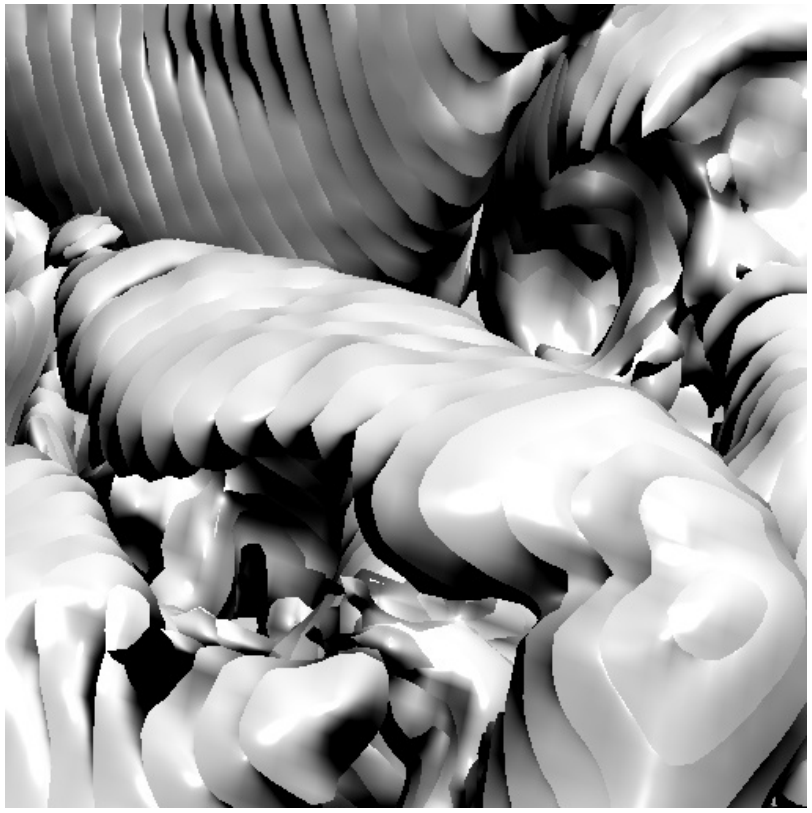


Refine already detected intersection

Handle problems with small features / at silhouettes with adaptive sampling

Intersection Refinement (2)

without refinement



with refinement

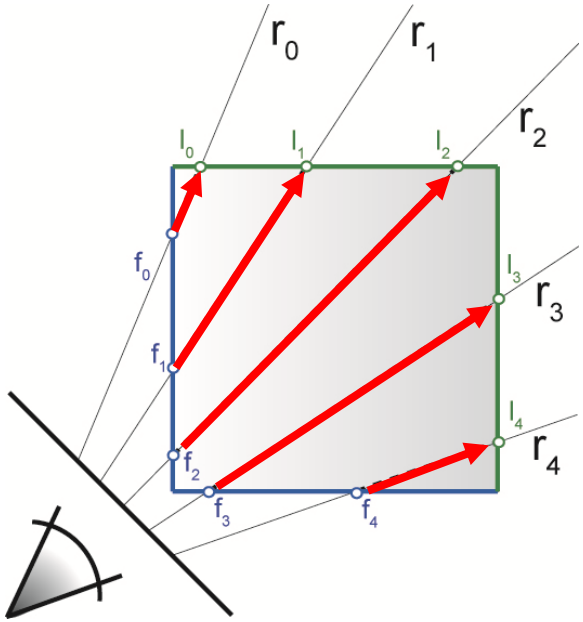
sampling distance 5 voxels (no adaptive sampling)

Procedural Ray Setup/Termination

- Everything handled in the fragment shader / CUDA kernel
- Procedural ray / bounding box intersection

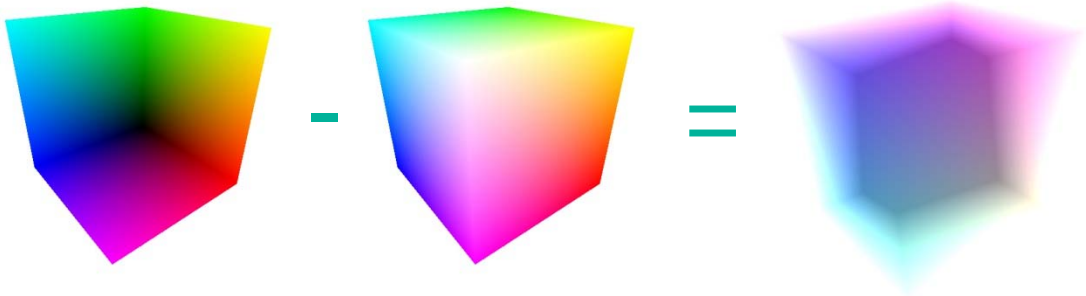
- Ray is given by camera position and volume entry position
- Exit criterion needed

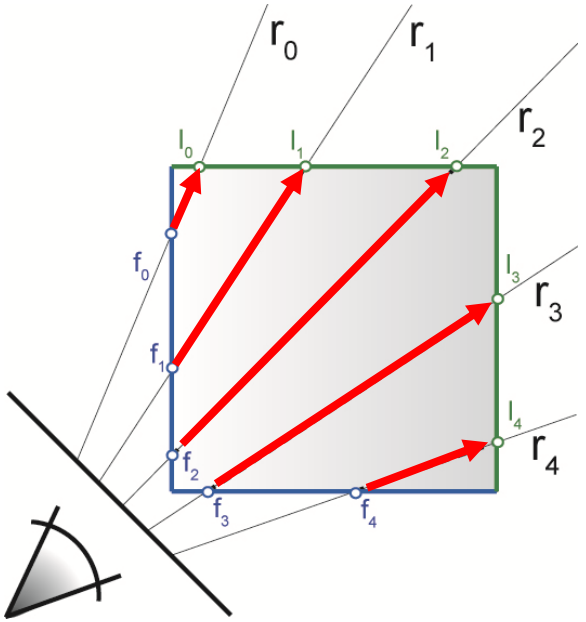
- Pro: simple and self-contained
- Con: full computational load per-pixel/fragment



Rasterization-Based Ray Setup

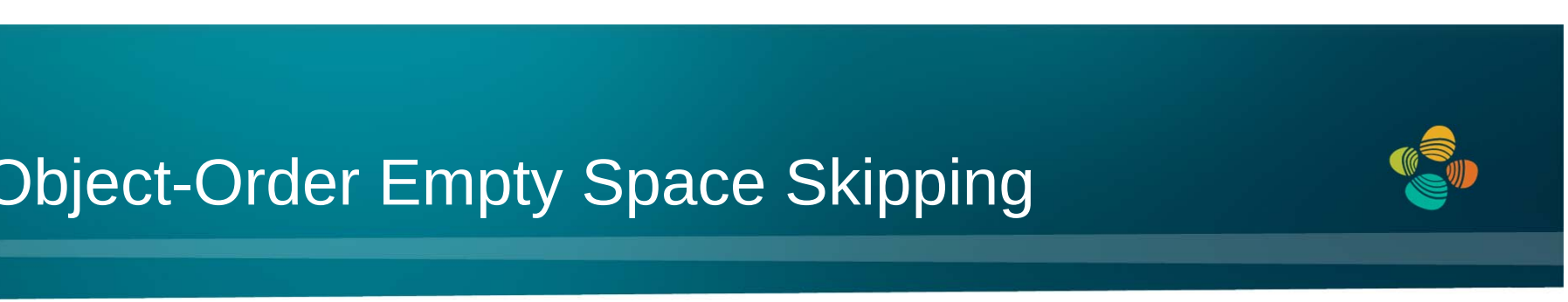
- Fragment == ray
- Need ray start pos, direction vector
- Rasterize bounding box



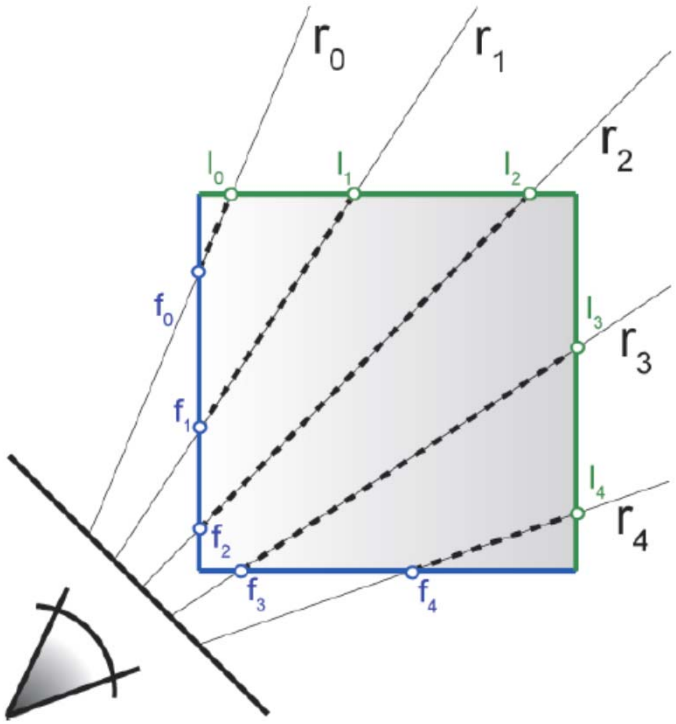


- Identical for orthogonal and perspective projection!

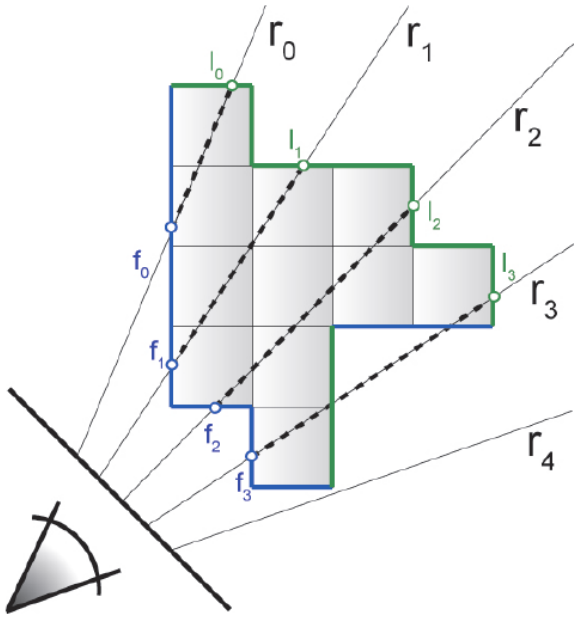
Object-Order Empty Space Skipping



- Modify initial rasterization step



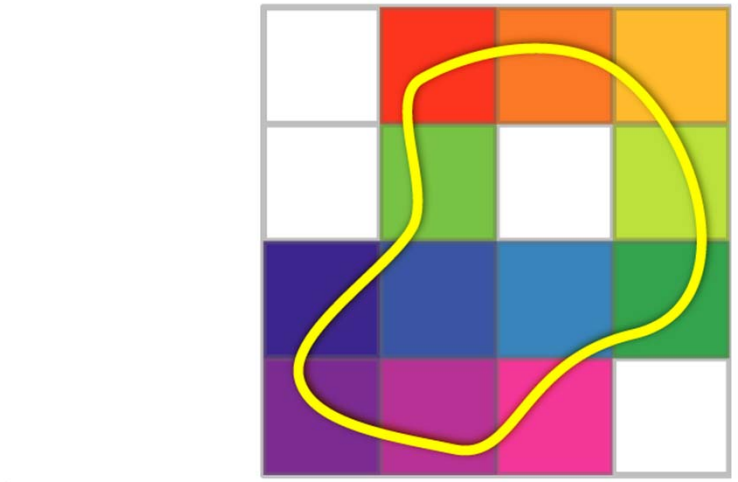
rasterize bounding box

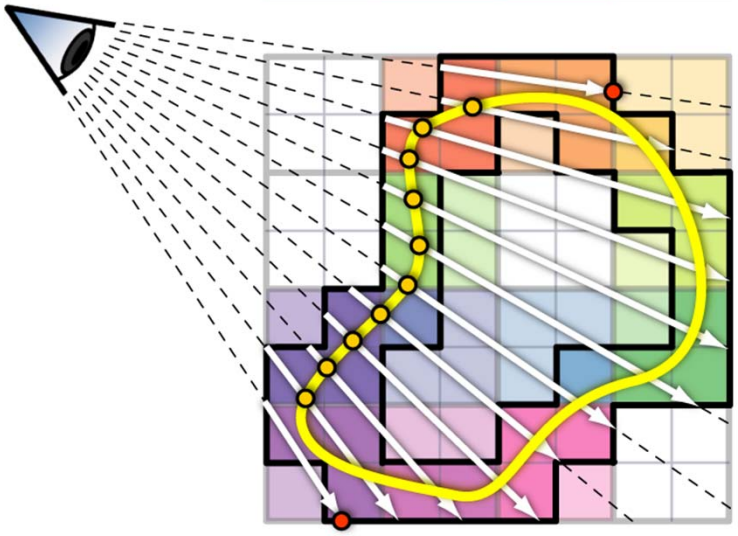


rasterize “tight" bounding geometry

Object-Order Empty Space Skipping

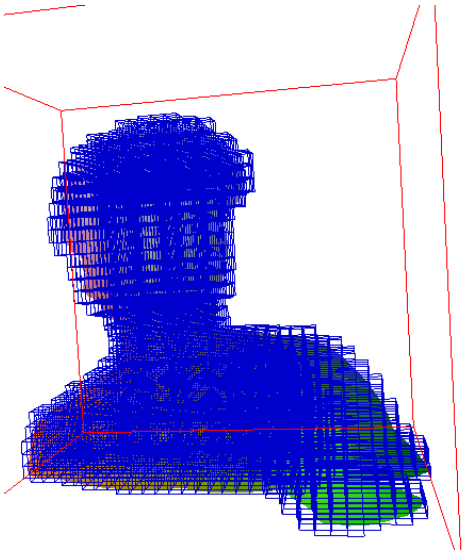
- Rasterize front and back faces of active min-max bricks
- Start rays on brick front faces
- Terminate when
  - Full opacity reached, or
  - Back face reached

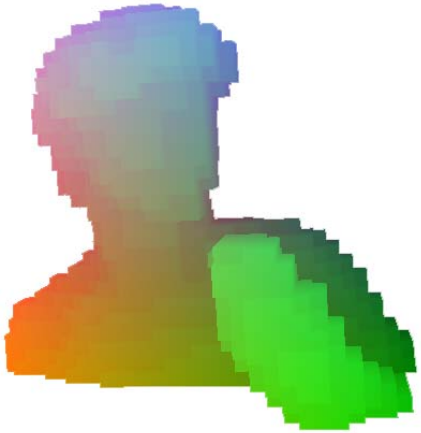




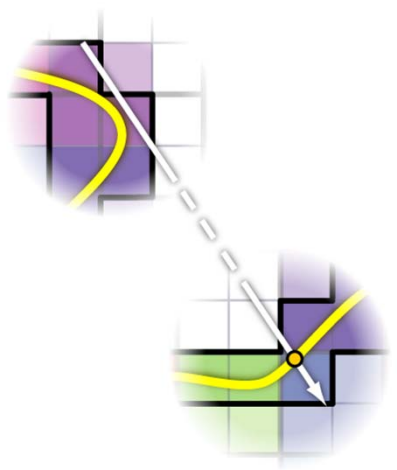
Object-Order Empty Space Skipping

- Store min-max values of volume blocks
- Cull blocks against transfer function or iso value
- Rasterize front and back faces of active blocks





Object-Order Empty Space Skipping



- Not all empty space skipped
  - Holes in the volume
  - Wrong active bricks

Thank you.

Thanks for material

- Helwig Hauser
- Eduard Gröller
- Daniel Weiskopf
- Torsten Möller
- Ronny Peikert
- Philipp Muigg
- Christof Rezk-Salama
- Joe Kniss, Gordon Kindlmann