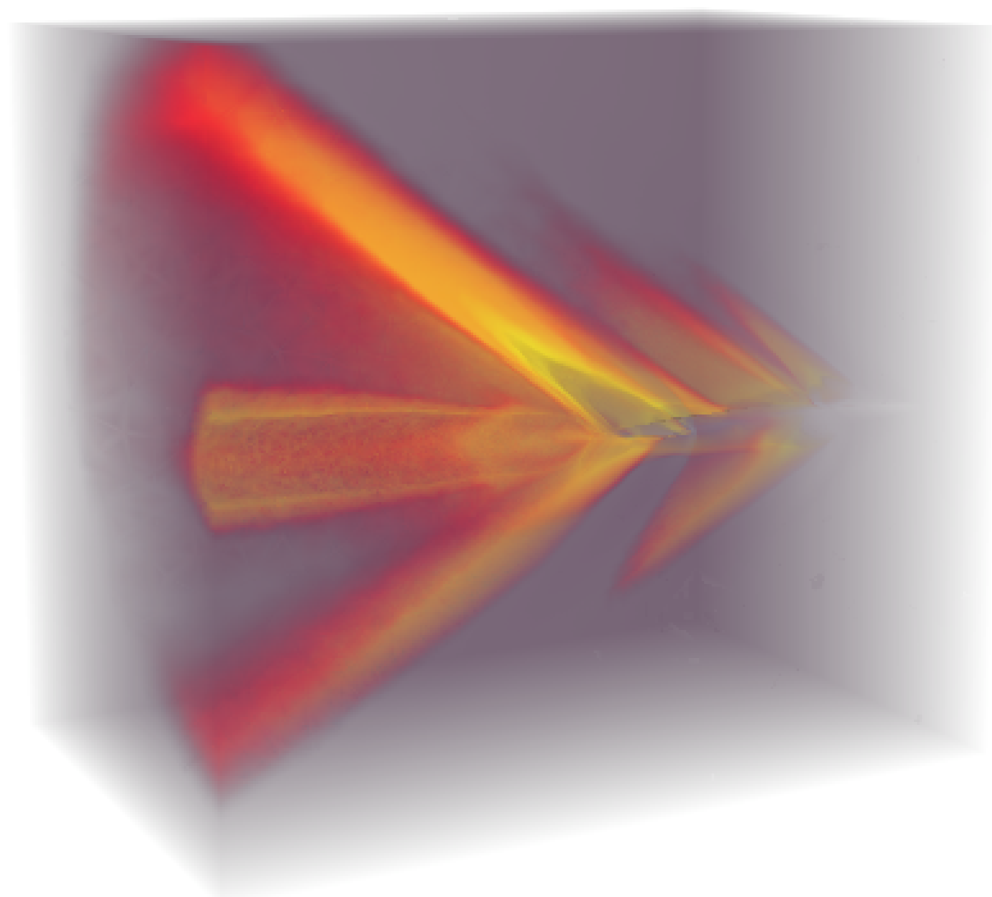
Adaptive Visualization of Dynamic Unstructured Meshes

Steven P. Callahan

Dissertation Defense
in partial fulfillment for a
Ph.D. in Computing

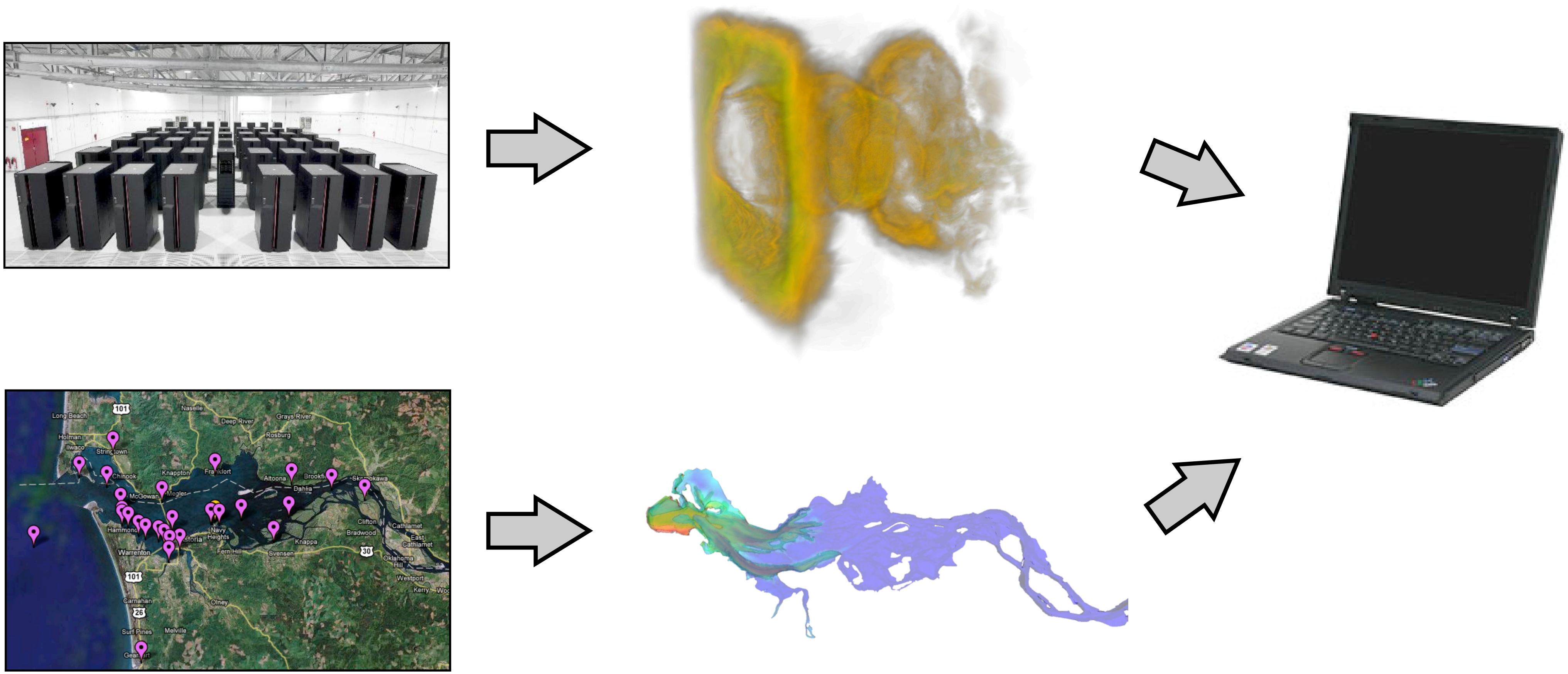
Overview

- What is the problem?
- Where were we then?
- Where are we now?
- How did we get here?
- Where do we go now?

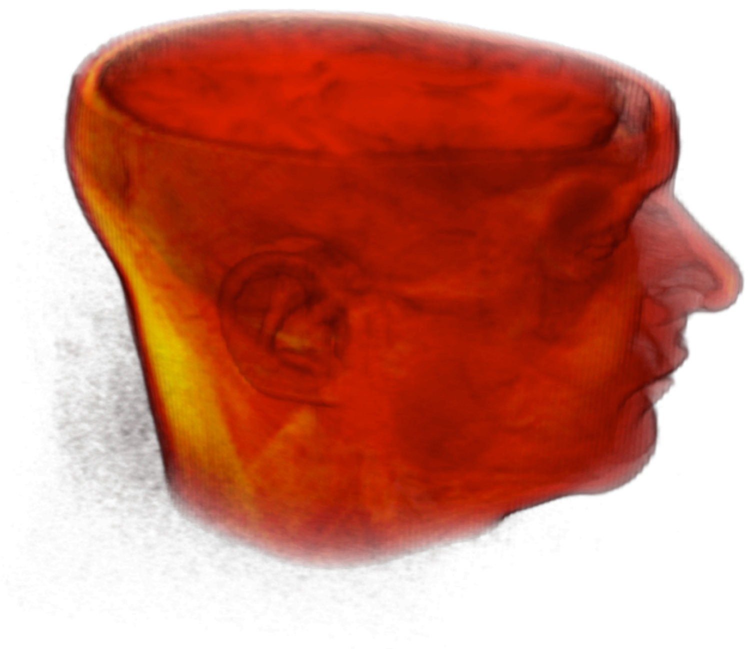


Motivation

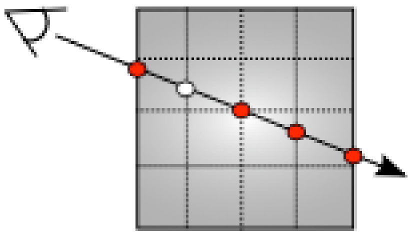
- Volume Rendering is important for analysis
- Visualization is not keeping pace with simulation/measurement



Direct Volume Rendering



- Sampling

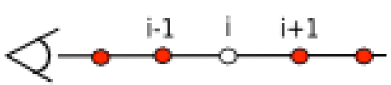


- Classification

$$I(D) = I_0 e^{-\int_0^D \rho(t)A dt} + \int_0^D C(s)\rho(s)A e^{-\int_s^D \rho(t)A dt} ds$$

[Blinn 1982]
[Sabella 1988]
[Max 1995]

- Compositing

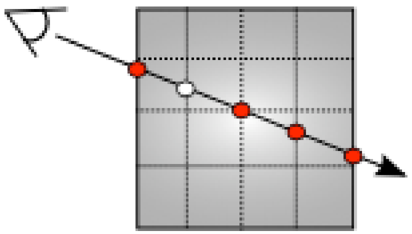


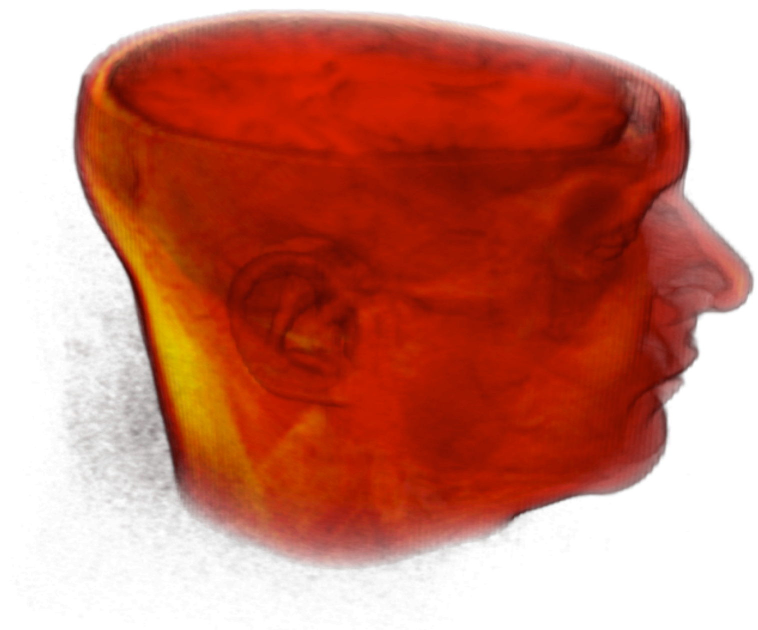
$$\begin{aligned} \mathbf{c}_i &= \mathbf{c}_{i-1} + \mathbf{c}_i \alpha_i (1 - \alpha_{i-1}) \\ \alpha_i &= \alpha_{i-1} + \alpha_i (1 - \alpha_{i-1}) \end{aligned}$$

[Porter and Duff 1990]

Direct Volume Rendering

- Sampling





- Classification

$$I(D) = I_0 e^{-\int_0^D \rho(t) A dt} + \int_0^D C(s)\rho(s) A e^{-\int_s^D \rho(t) A dt} ds$$

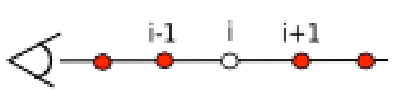
Absorption

[Blinn 1982]

[Sabella 1988]

[Max 1995]

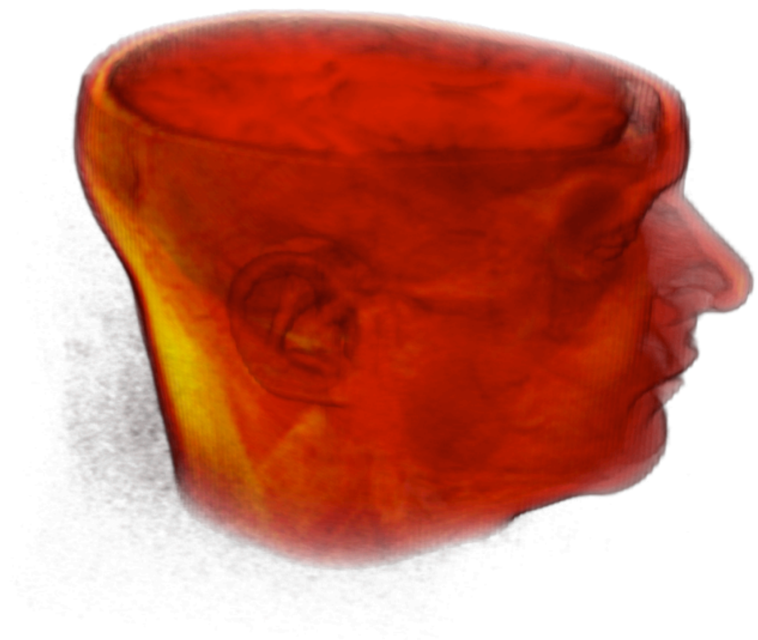
- Compositing



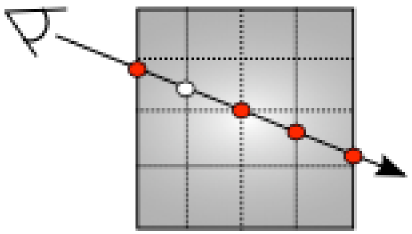
$$\begin{aligned} \mathbf{c}_i &= \mathbf{c}_{i-1} + \mathbf{c}_i \alpha_i (1 - \alpha_{i-1}) \\ \alpha_i &= \alpha_{i-1} + \alpha_i (1 - \alpha_{i-1}) \end{aligned}$$

[Porter and Duff 1990]

Direct Volume Rendering



- Sampling



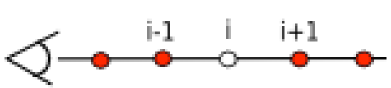
- Classification

$$I(D) = I_0 e^{-\int_0^D \rho(t) A dt} + \int_0^D C(s)\rho(s) A e^{-\int_s^D \rho(t) A dt} ds$$

Absorption   Emission

[Blinn 1982]
[Sabella 1988]
[Max 1995]

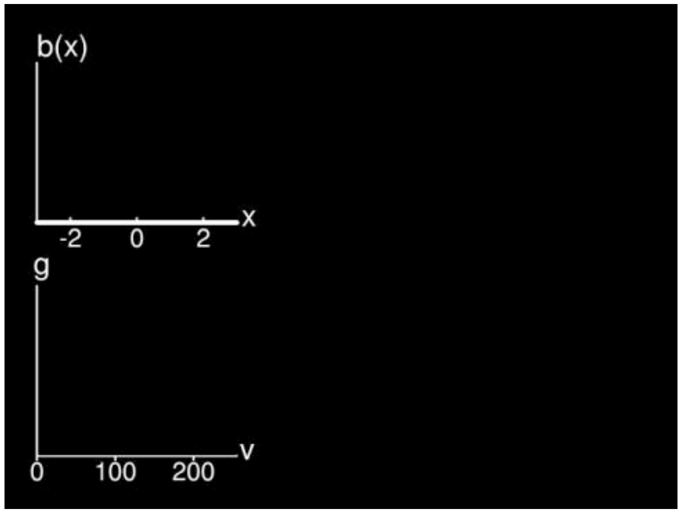
- Compositing



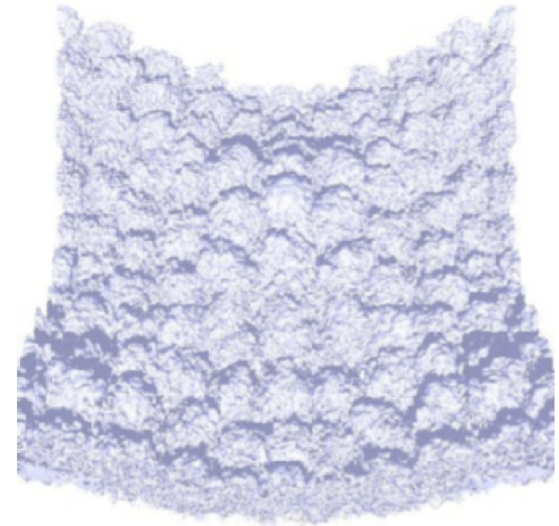
$$\begin{aligned} \mathbf{c}_i &= \mathbf{c}_{i-1} + \mathbf{c}_i \alpha_i (1 - \alpha_{i-1}) \\ \alpha_i &= \alpha_{i-1} + \alpha_i (1 - \alpha_{i-1}) \end{aligned}$$

[Porter and Duff 1990]

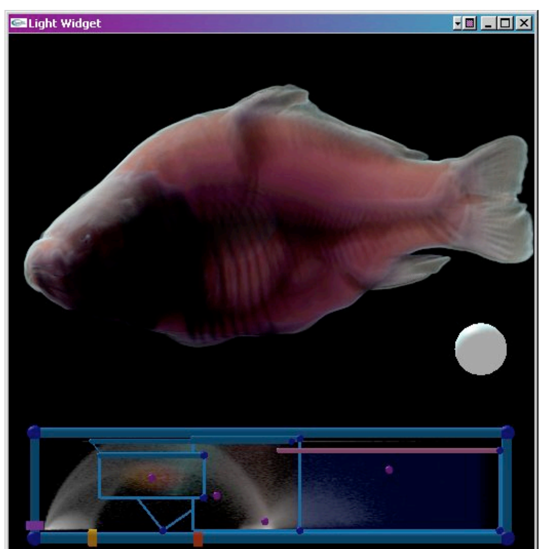
Structured vs. Unstructured



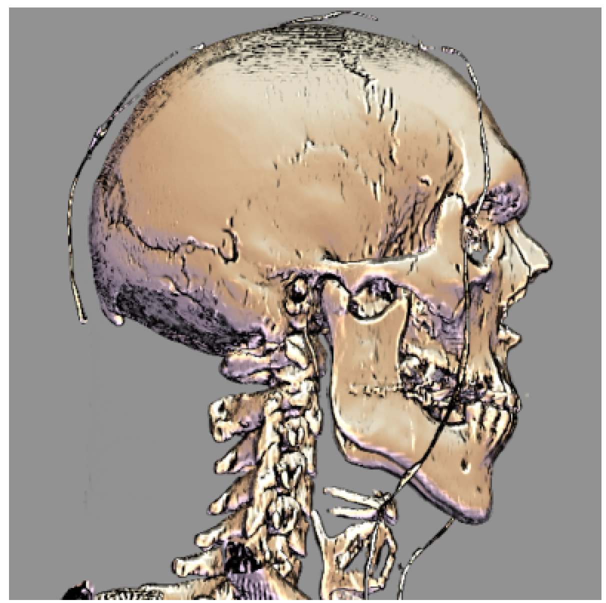
[Kindlmann and Durkin 1998]



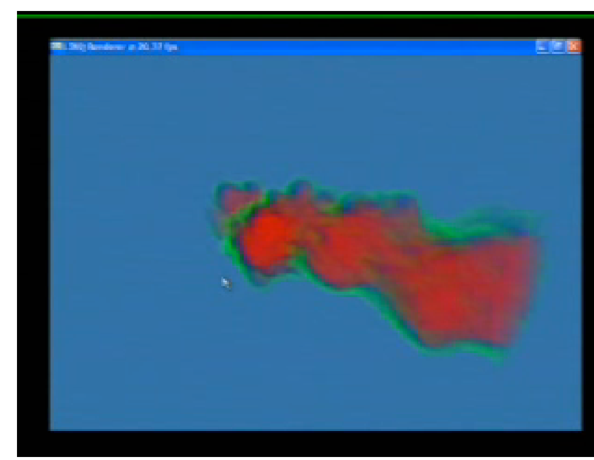
[Demarle et al. 2003]



[Kniss et al. 2002]

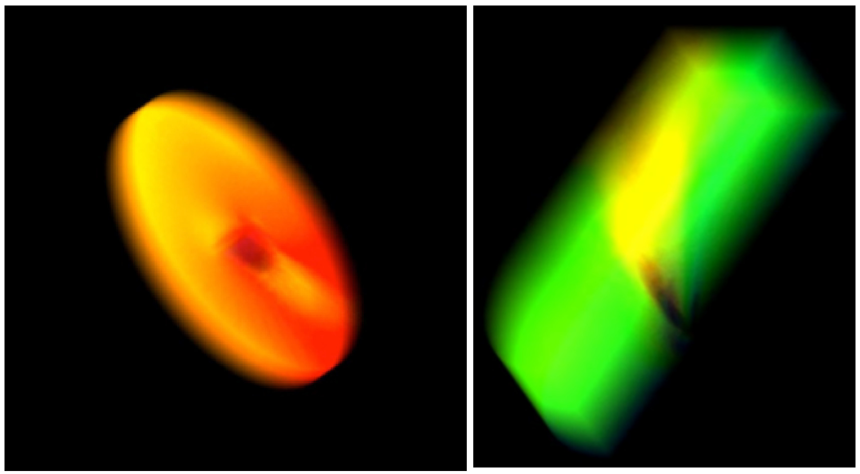


[Kindlmann et al. 2003]

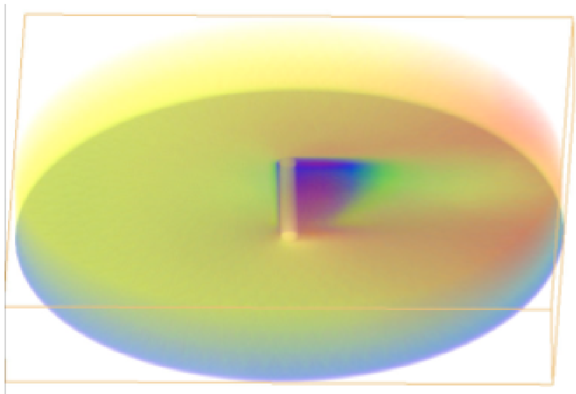


[Schneider and Westermann 2003]

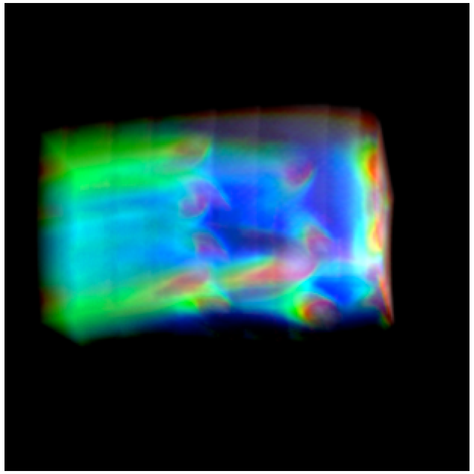
Structured vs. Unstructured



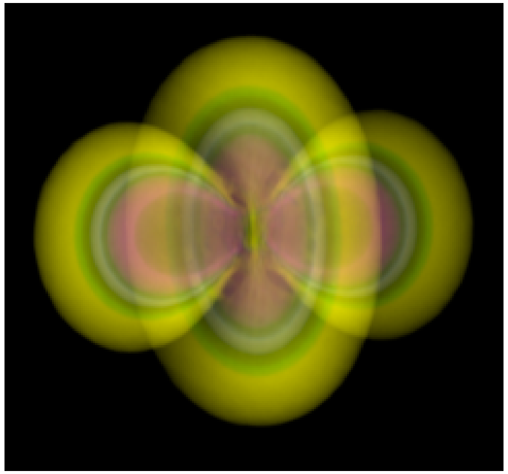
[Farias et al. 2000]



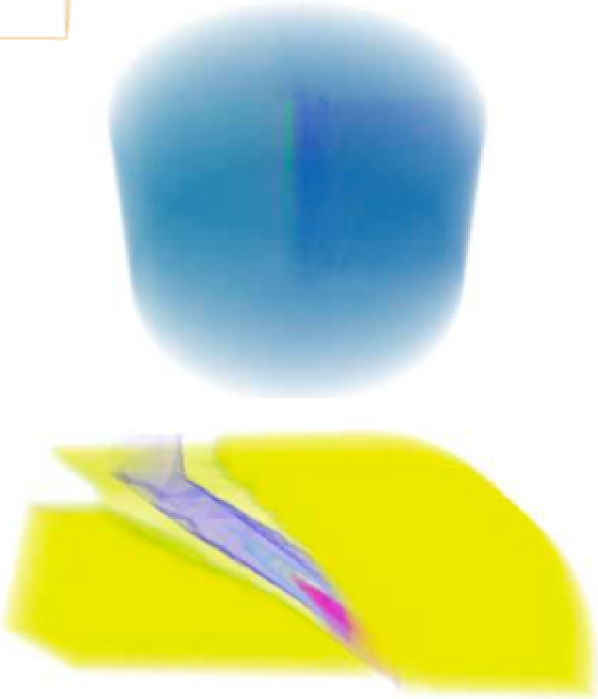
[Wylie et al. 2002]



[Bunyk et al. 1997]



[Weiler et al. 2002]

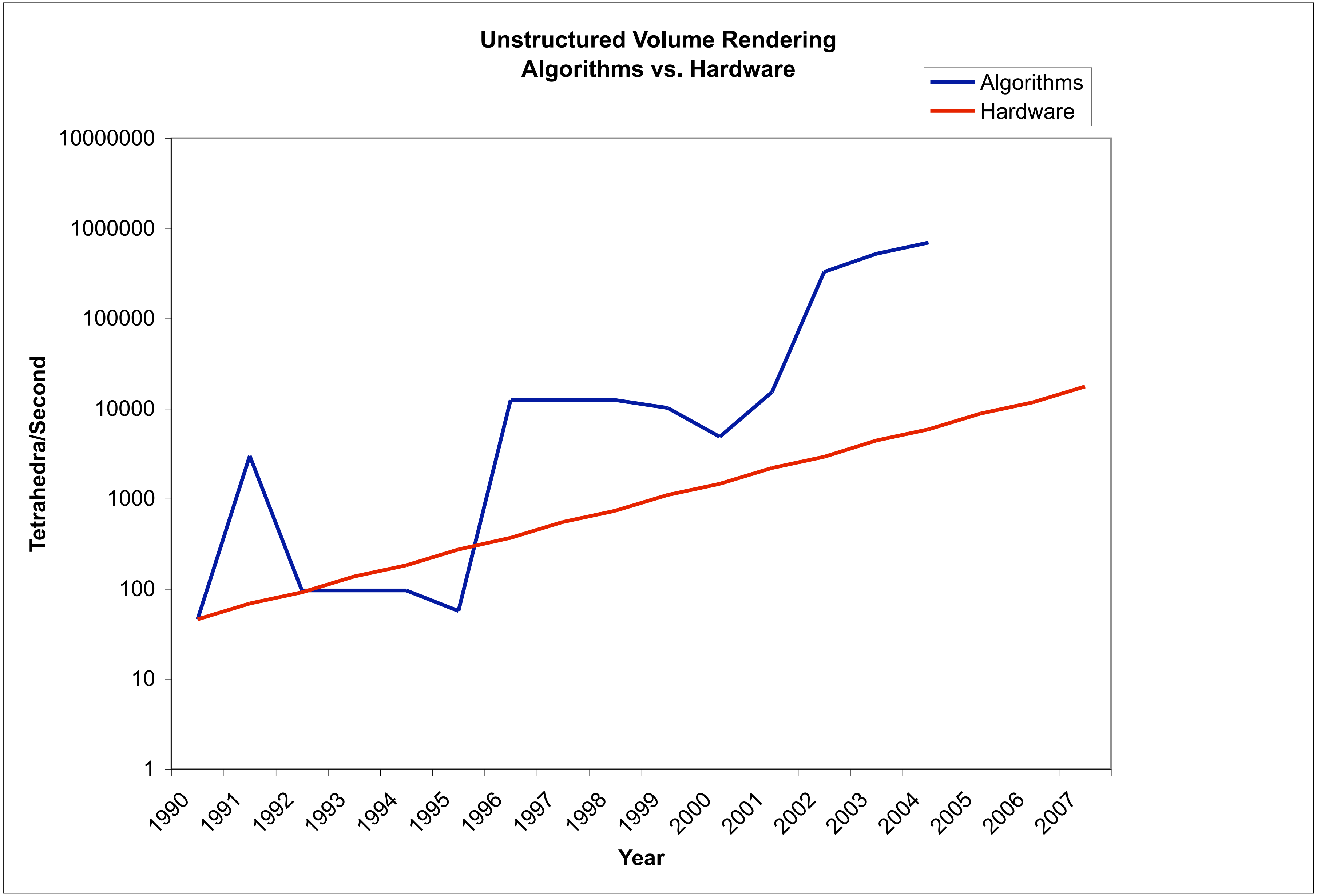


[Weiler et al. 2003]

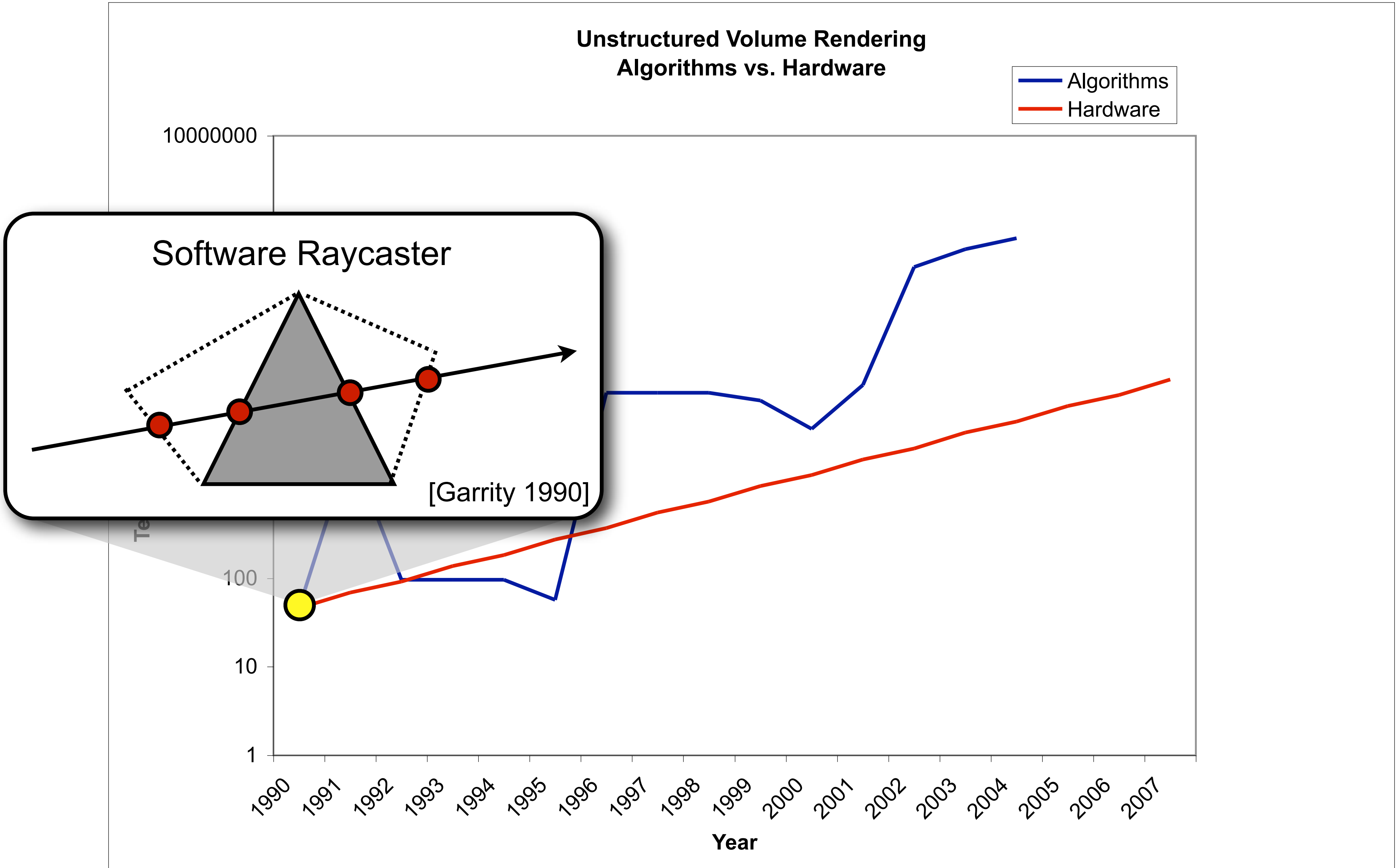
Unstructured Volume Rendering

- Limitations of existing algorithms
  - Interactivity
  - Large data
  - Dynamic data

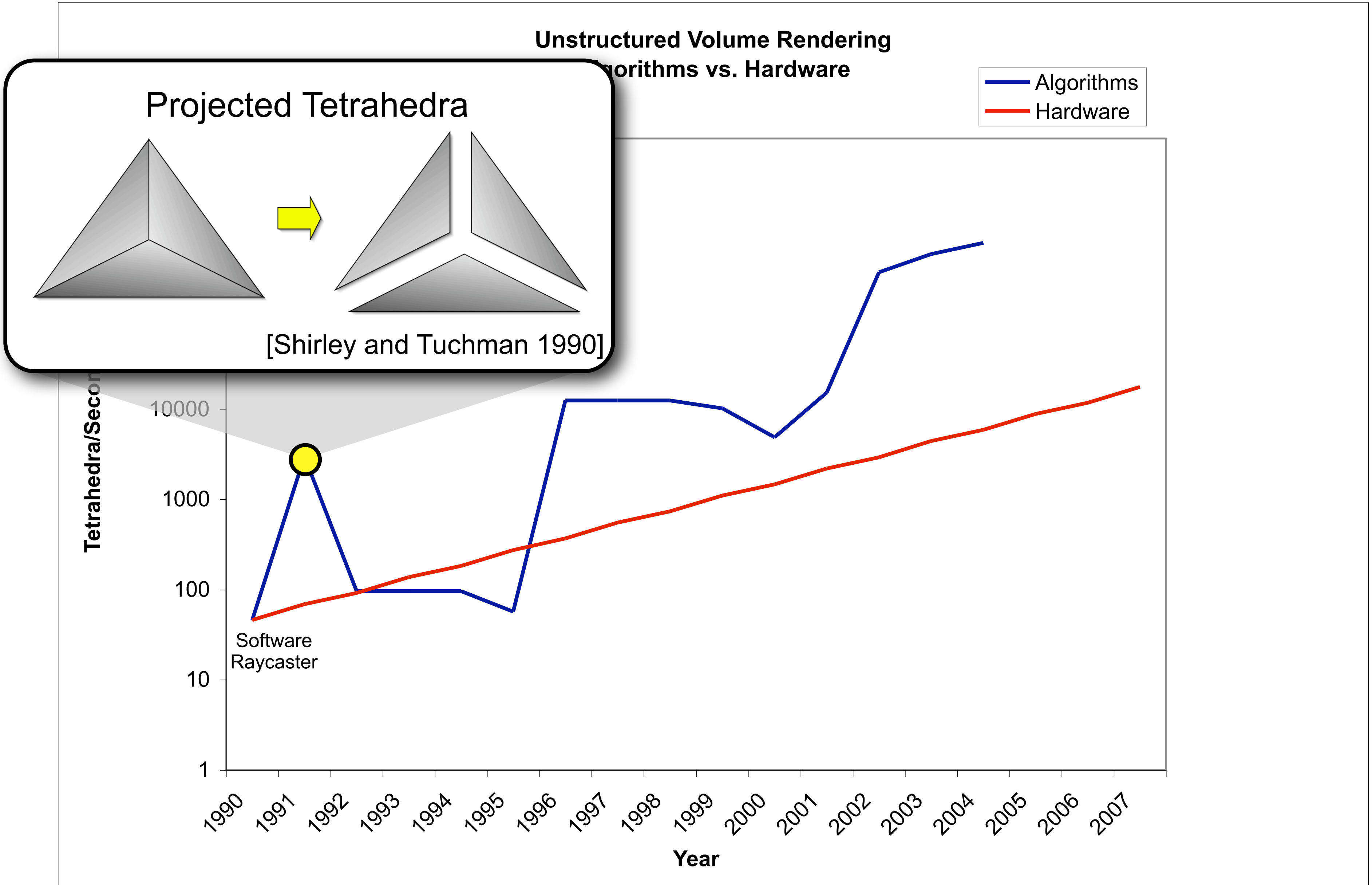
A Historical Perspective



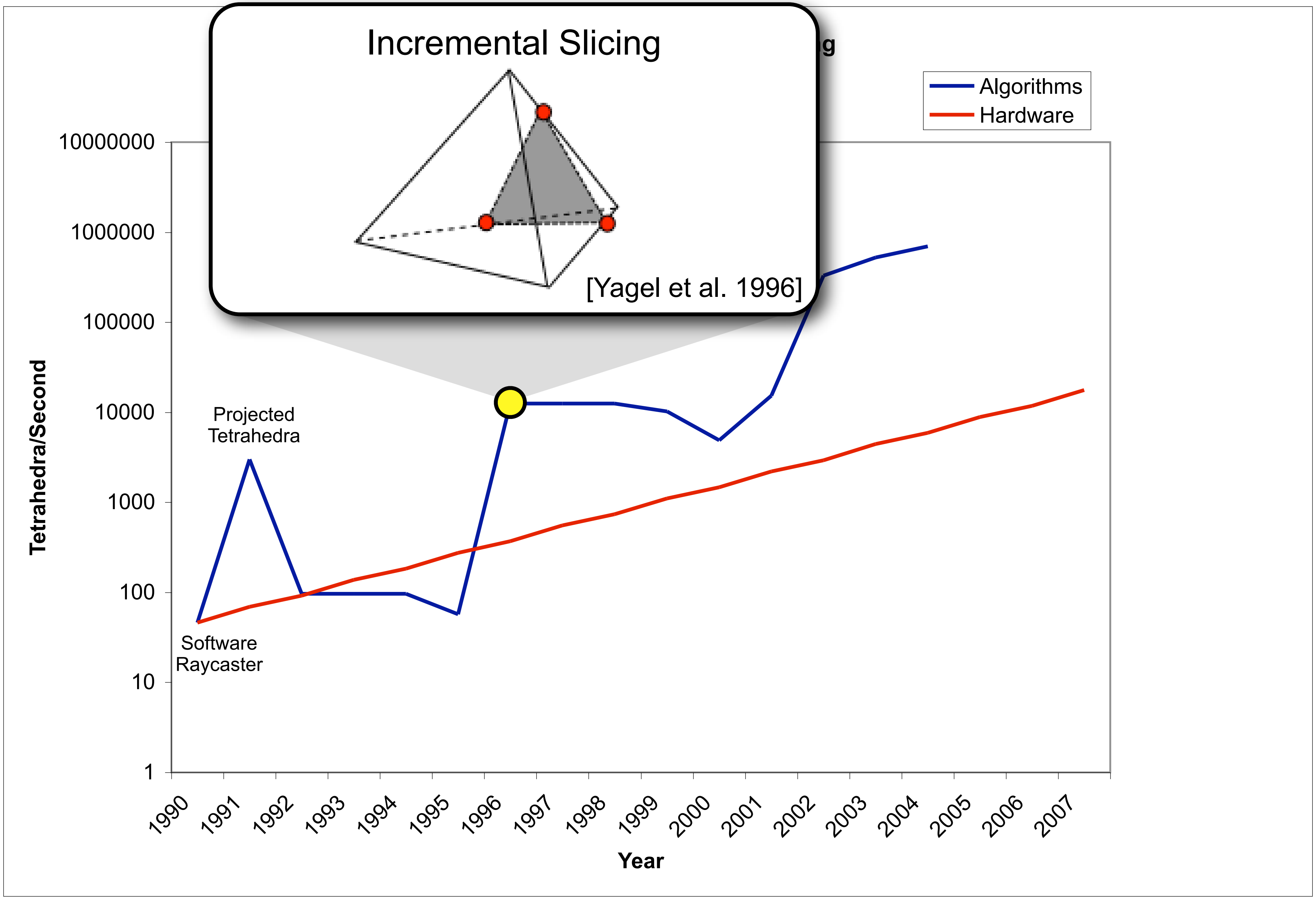
A Historical Perspective



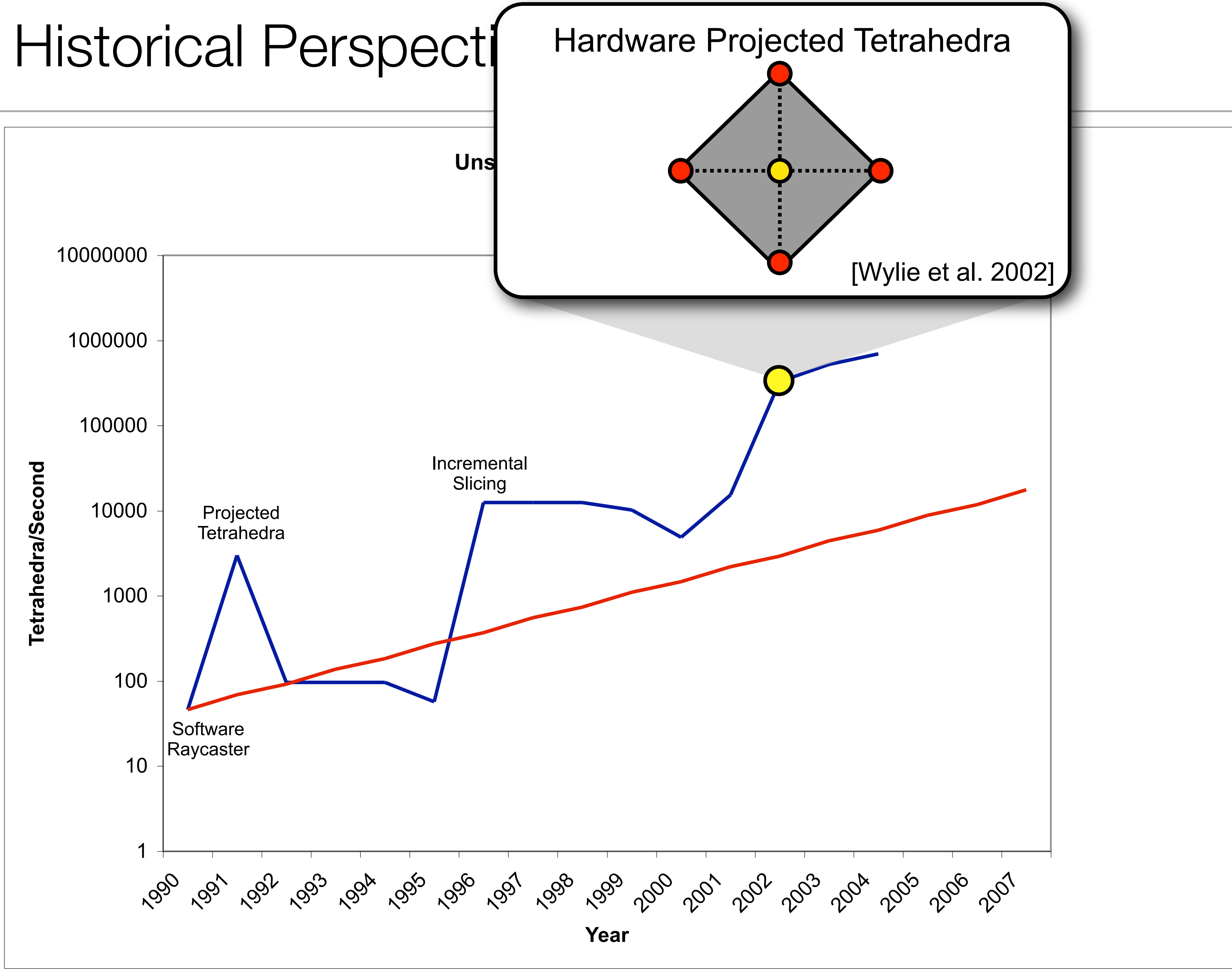
A Historical Perspective



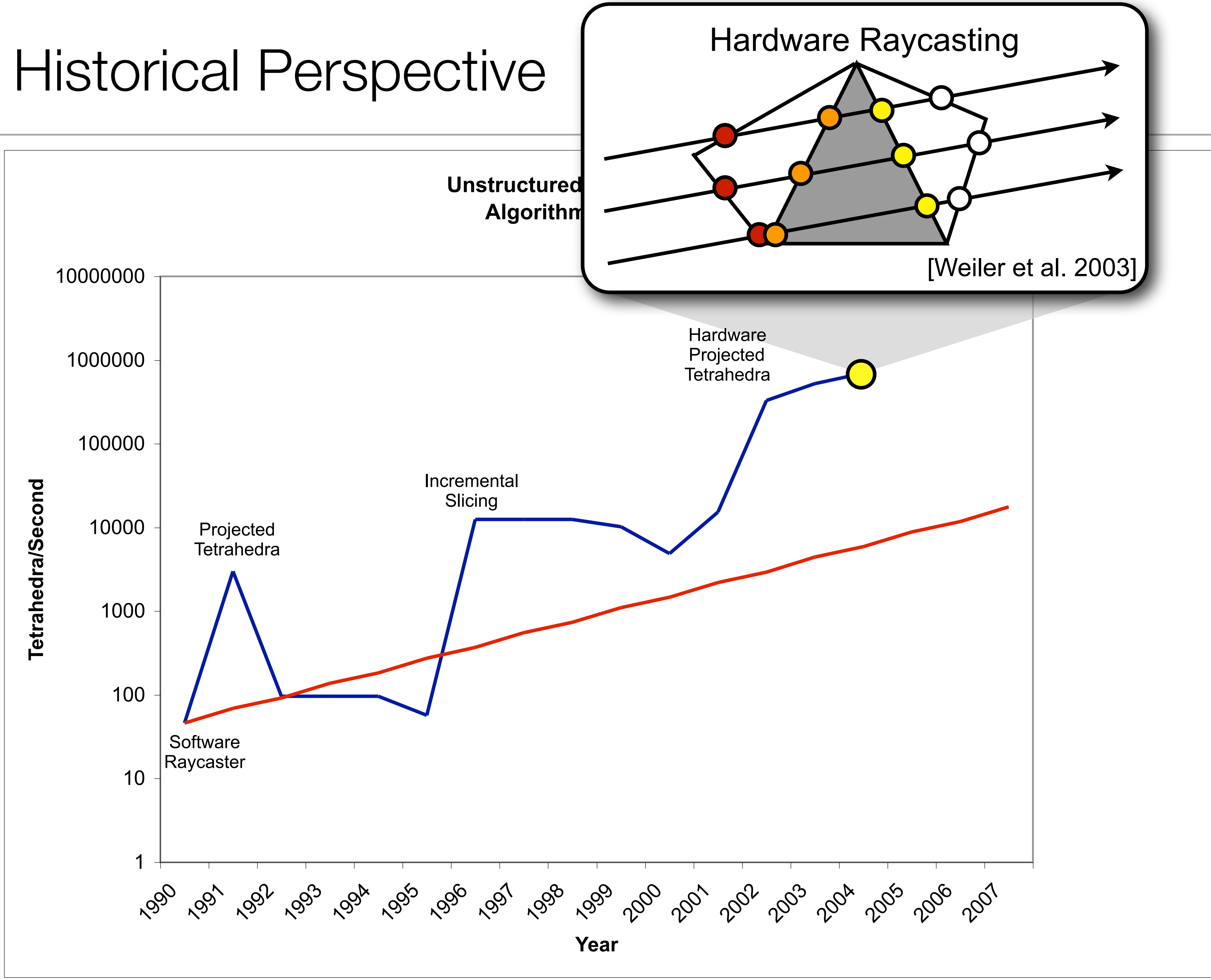
A Historical Perspective



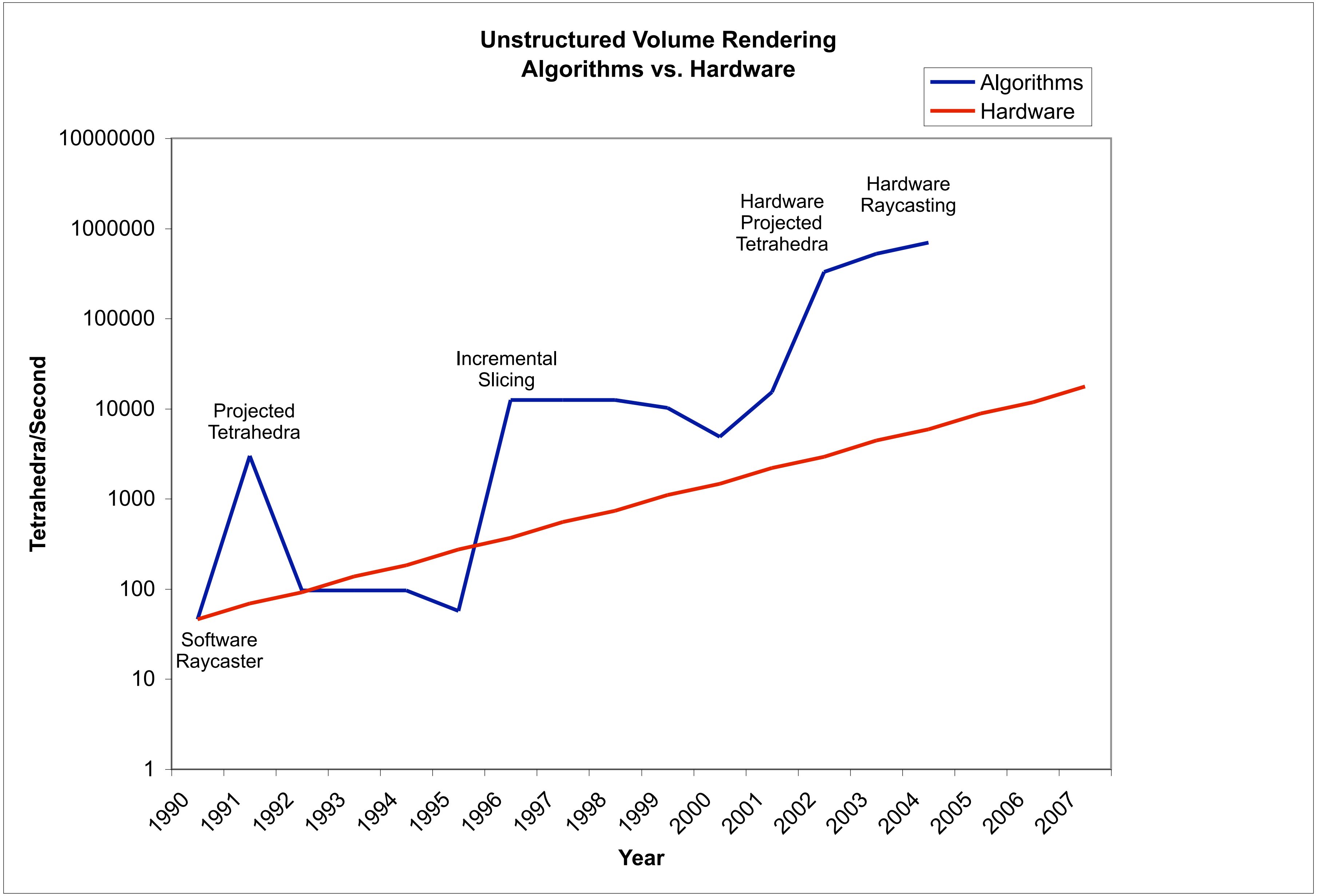
A Historical Perspecti



A Historical Perspective



A Historical Perspective



Thesis Statement

Interactive volume rendering of dynamic unstructured grids requires a combination of novel software algorithms and frameworks that efficiently amortize recent hardware configurations

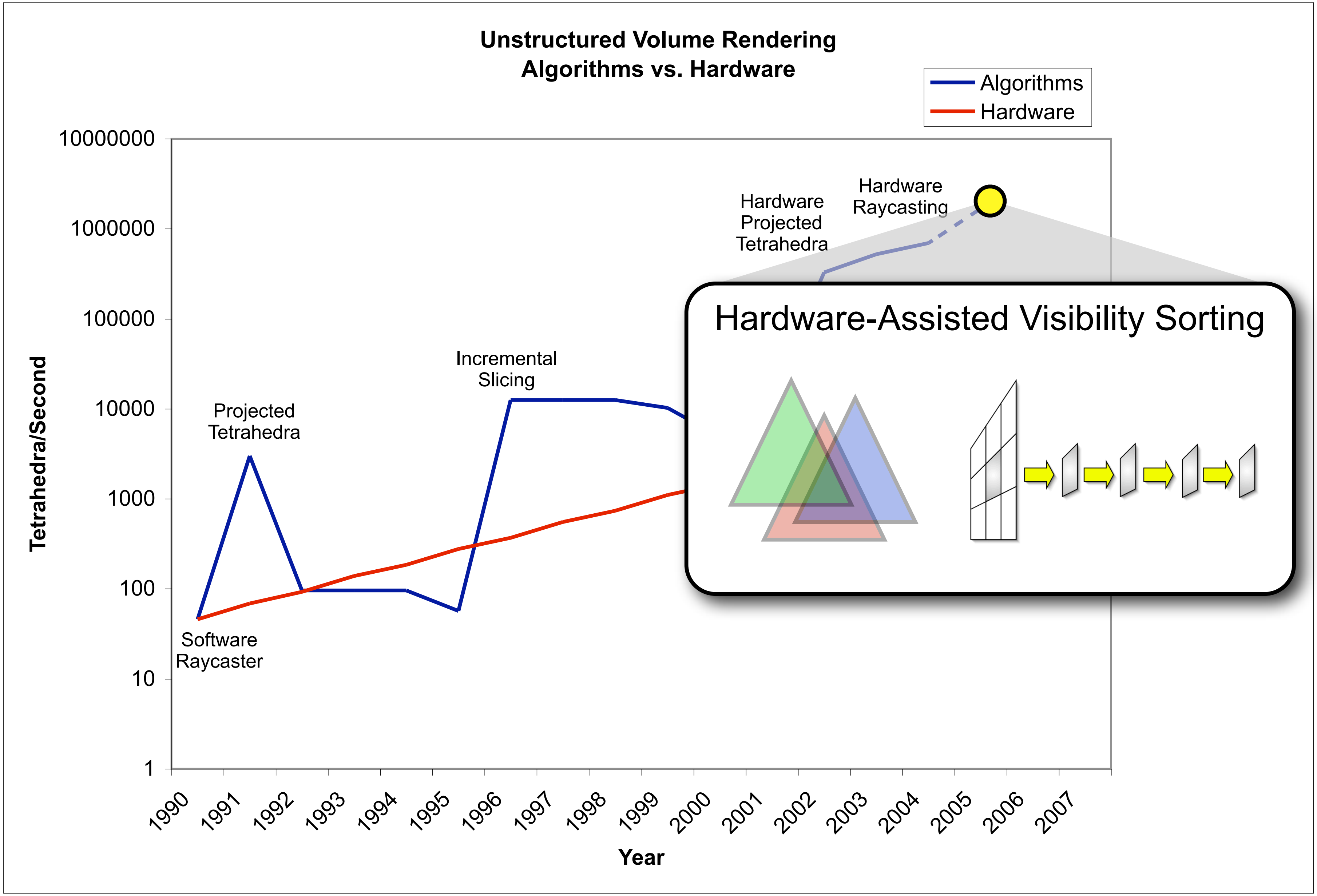
Contributions

- Improved interactive volume rendering
  - Object-space acceleration (Chapter 3)
  - Image-space acceleration (Chapter 4)
- Increased limits on data size
  - Progressive volume rendering (Chapter 5)
- Extended support for dynamic data
  - Time-varying scalar field volume rendering (Chapter 6)
- Created support for exploration of large dynamic volumes
  - Transfer function specification (Chapter 7)

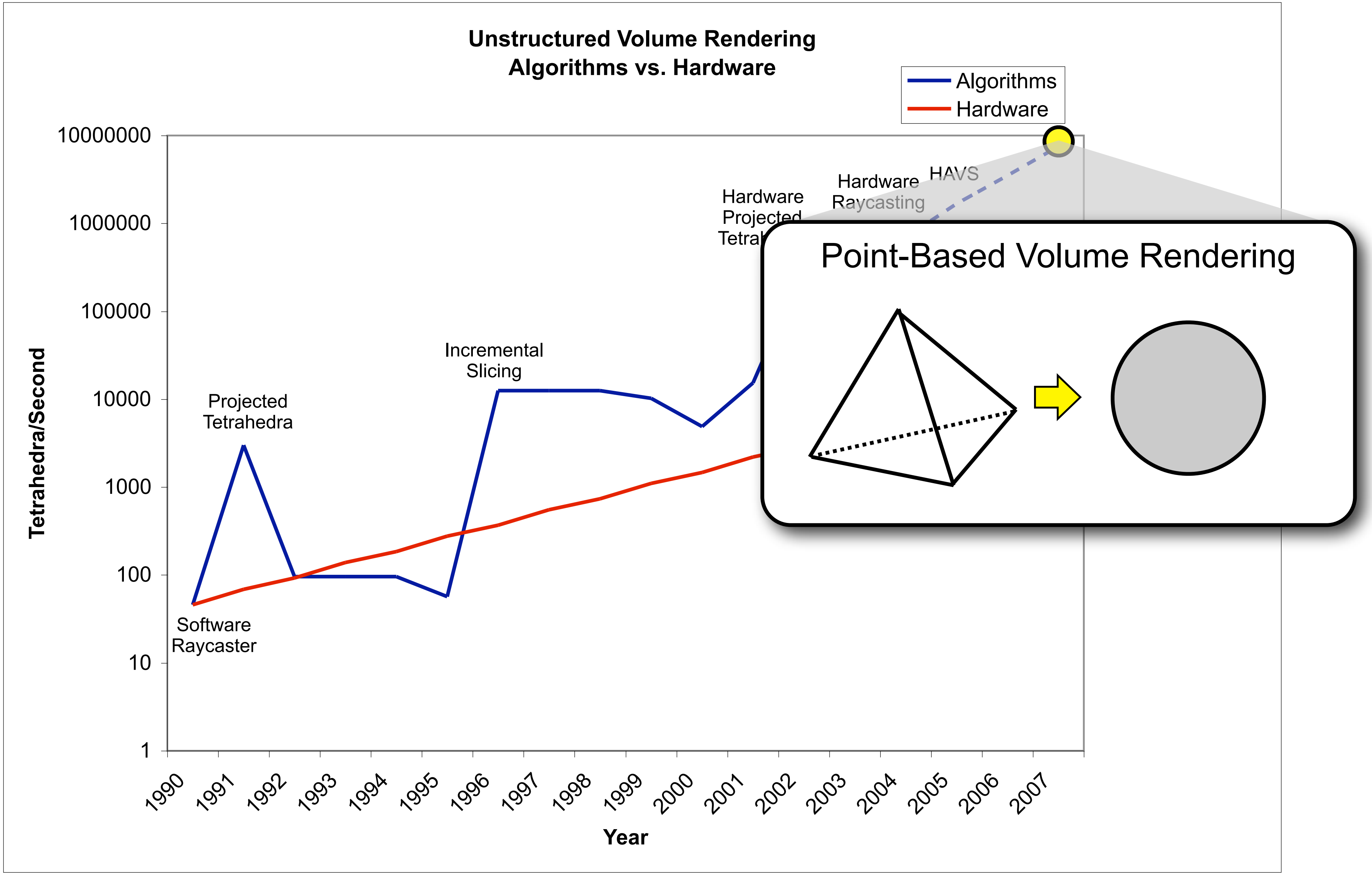
Contributions

- Journal Publications
  - Hardware-Assisted Visibility Sorting for Unstructured Volume Rendering. TVCG, 2005
  - A Survey of GPU-Based Volume Rendering of Unstructured Grids. RITA, 2005
  - Progressive Volume Rendering of Large Unstructured Grids. TVCG, 2006
  - Streaming Simplification of Tetrahedral Meshes. TVCG, 2007
  - An Adaptive Framework for Visualizing Unstructured Grids with Time-Varying Scalar Fields. Parallel Computing, 2007
  - Direct Volume Rendering: A 3D Plotting Technique for Scientific Data. Computing in Sci. and Eng., 2008
- Conference Publications
  - Hardware Accelerated Simulated Radiography. Vis, 2005
  - Interactive Rendering of Large Unstructured Grids Using Dynamic Level-Of-Detail. Vis, 2005
  - Interactive Volume Rendering of Unstructured Grids with Time-Varying Scalar Fields. EGPGV, 2006
  - Multi-Fragment Effects on the GPU using the k-Buffer. i3D, 2007
  - iRun: Interactive Rendering of Large Unstructured Grids. EGPGV, 2007
  - Hardware-Assisted Point-Based Volume Rendering of Tetrahedral Meshes. SIBGRAPI, 2007
- Unpublished Manuscripts
  - Interactive Transfer Function Specification for Direct Volume Rendering of Disparate Volumes. SCI Tech Report, 2007
  - Image-Based Acceleration for Direct Volume Rendering of Unstructured Grids using Joint Bilateral Upsampling. Submitted, 2008

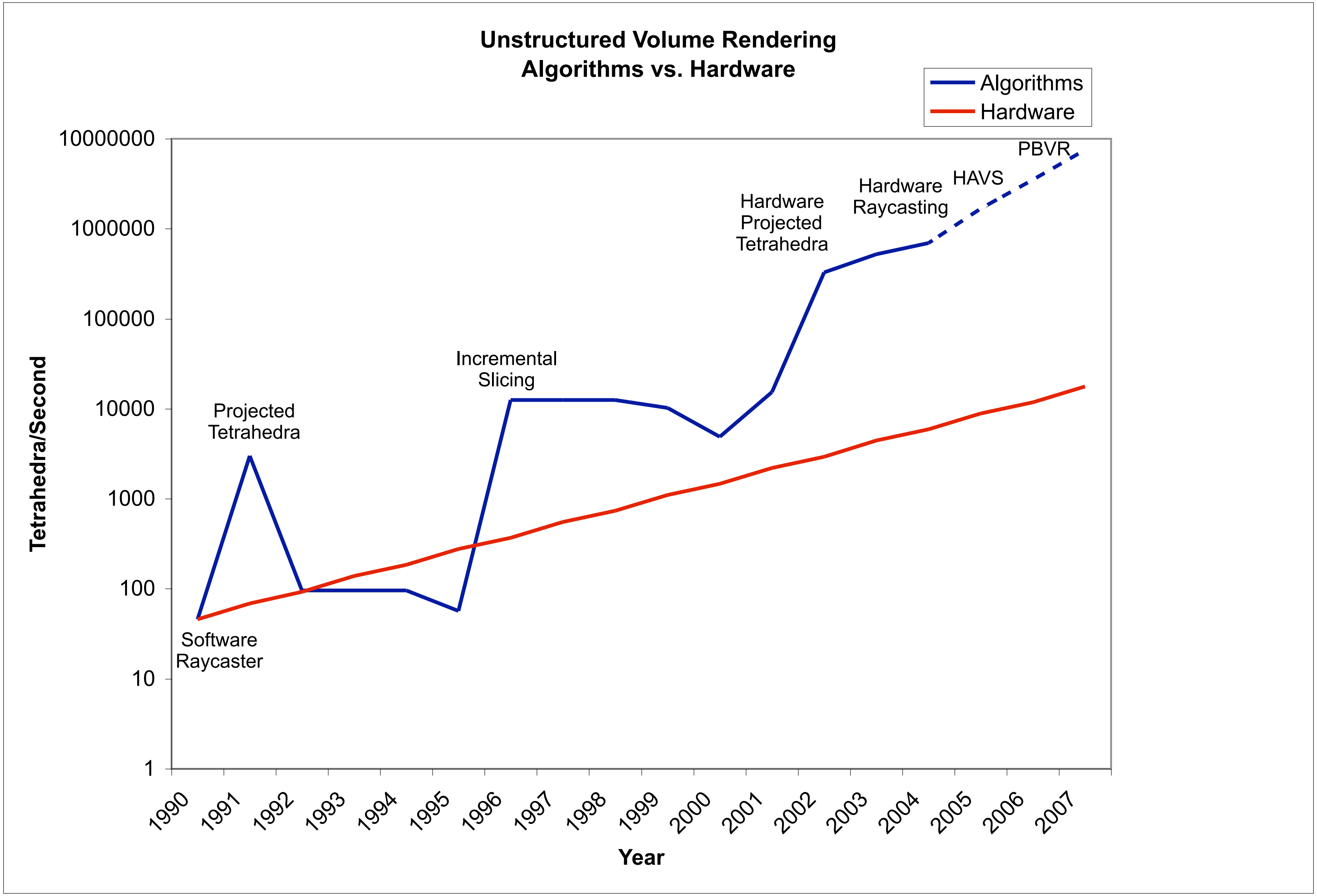
Dissertation Outcome



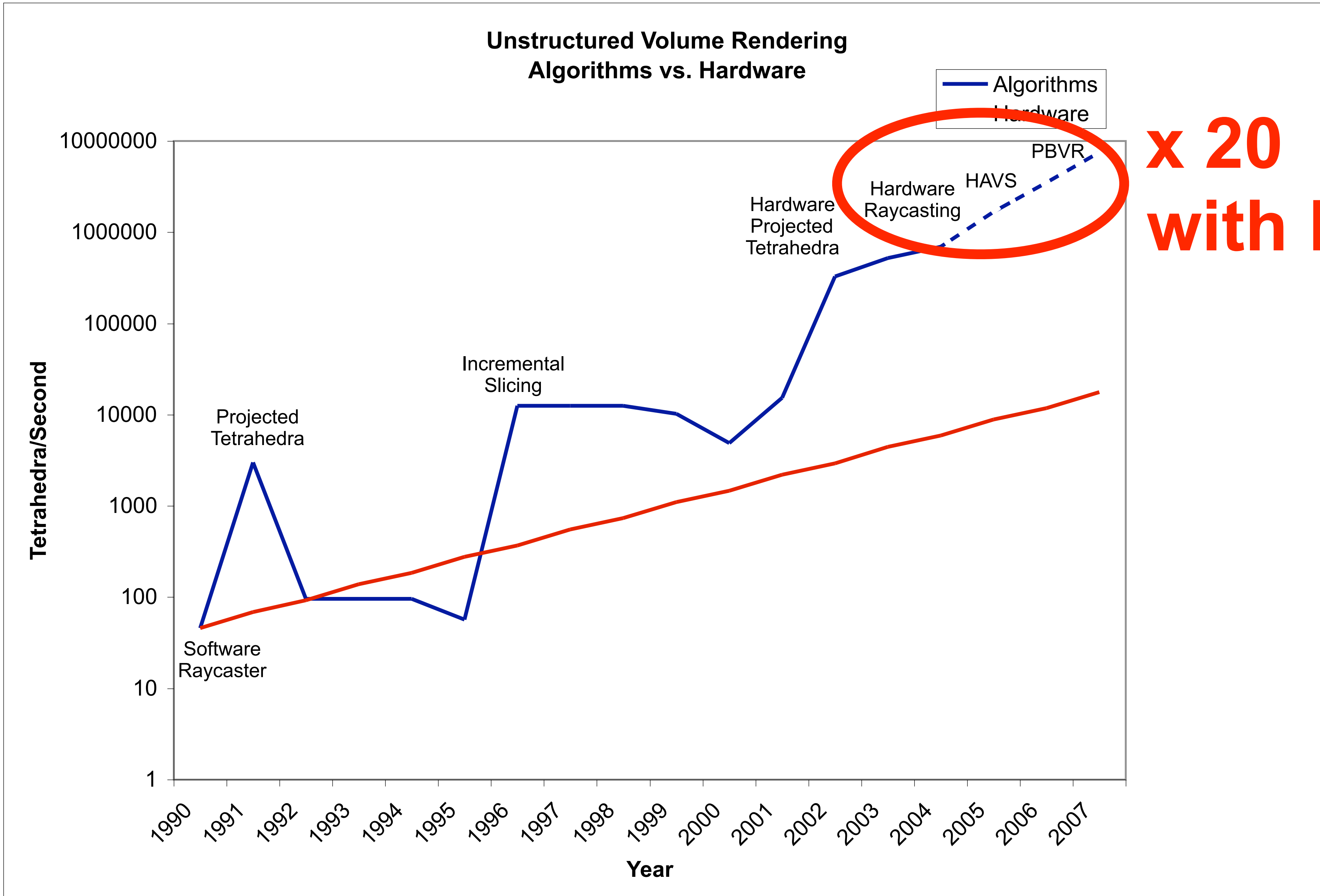
Dissertation Outcome



Dissertation Outcome

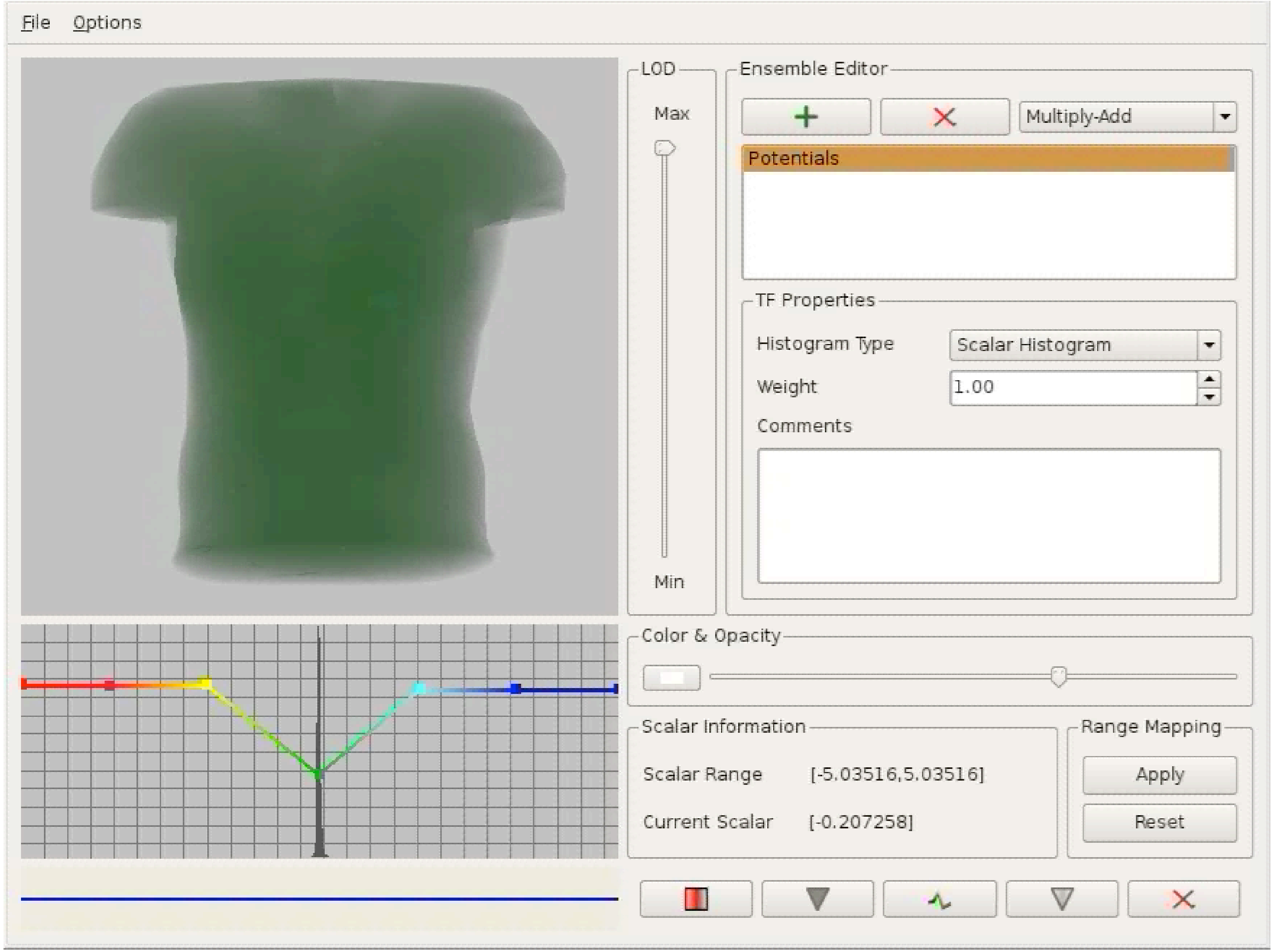


Dissertation Outcome



x 20 with LOD

Dissertation Outcome



Contributions

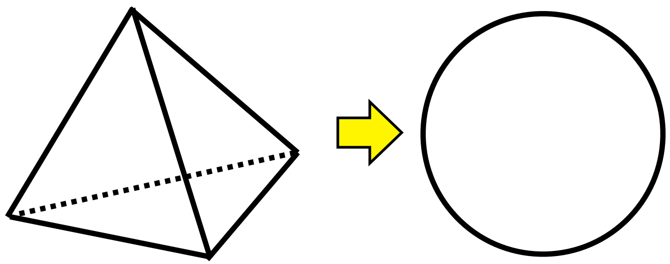
- Improved interactive volume rendering
  - Object-space acceleration (Chapter 3)
  - Image-space acceleration (Chapter 4)
- Increased limits on data size
  - Progressive volume rendering (Chapter 5)
- Extended support for dynamic data
  - Time-varying scalar field volume rendering (Chapter 6)
- Created support for exploration of large dynamic volumes
  - Transfer function specification (Chapter 7)

Improving Interactivity

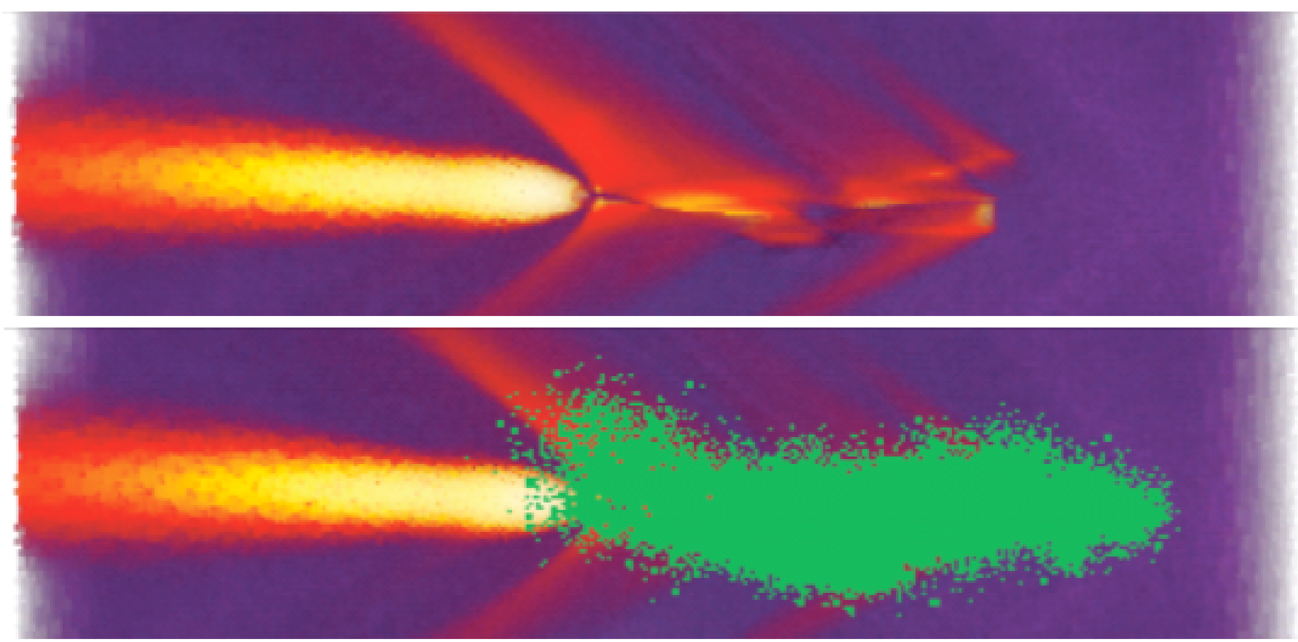
Object-Space Acceleration:
Point-Based Volume Rendering

Object-Space Acceleration

- Points are more flexible and require less data to represent

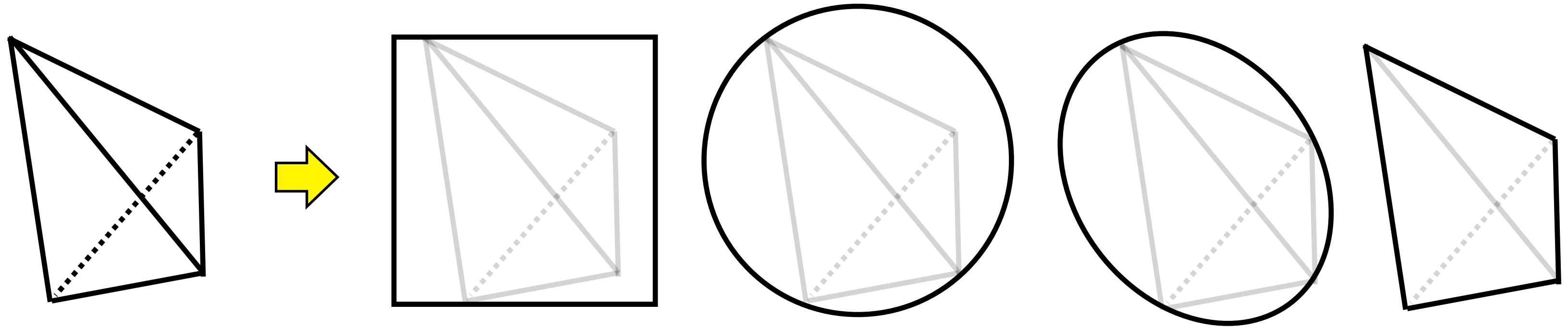


- Large volumes have subpixel-sized geometry

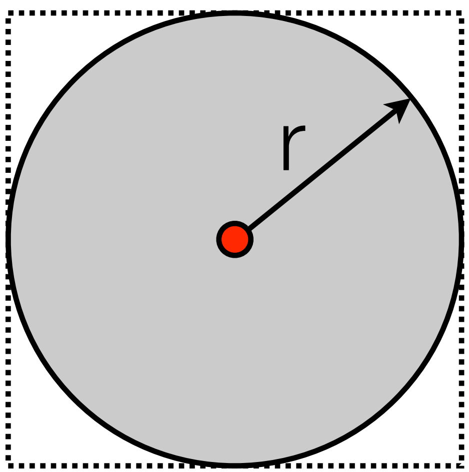


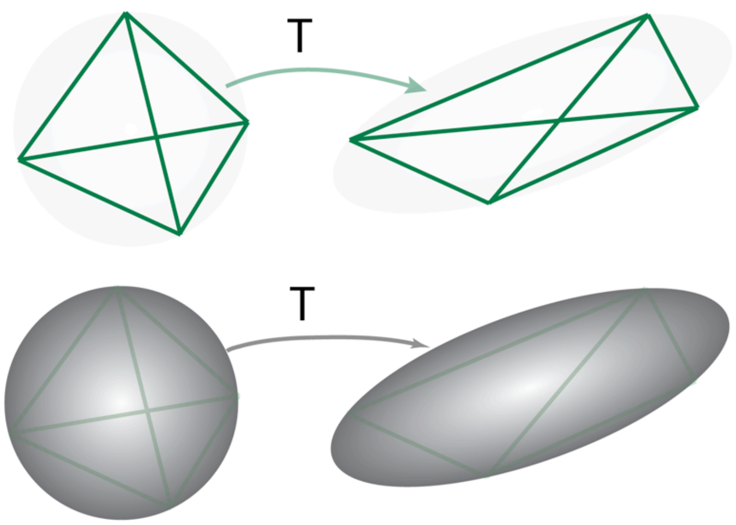
Object-Space Acceleration

- Error minimized by reshaping points



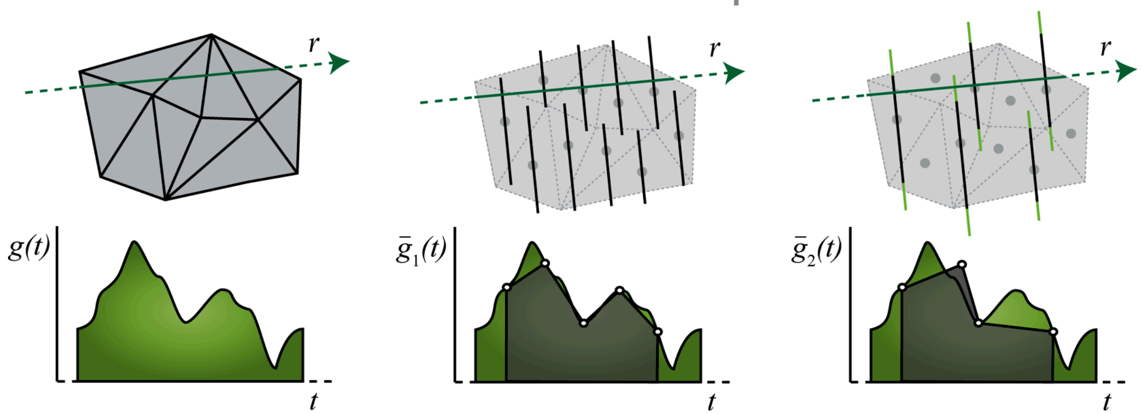
- Cull fragments in fragment shader



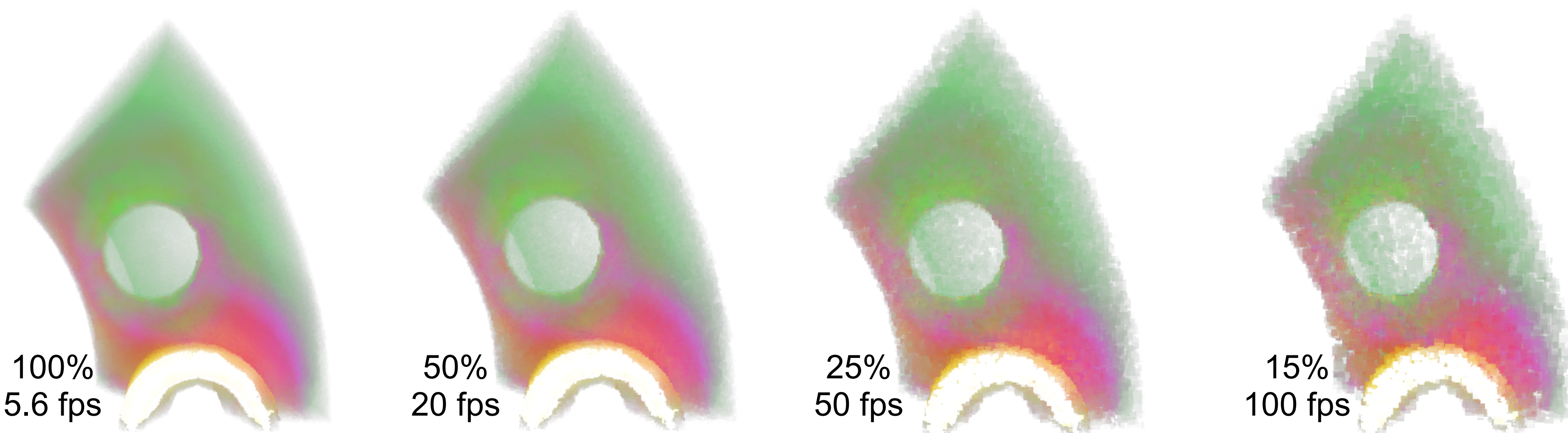


Object-Space Acceleration

- Fragments distances are classified and composited



- Sample-based level-of-detail used for interactivity



Improving Interactivity

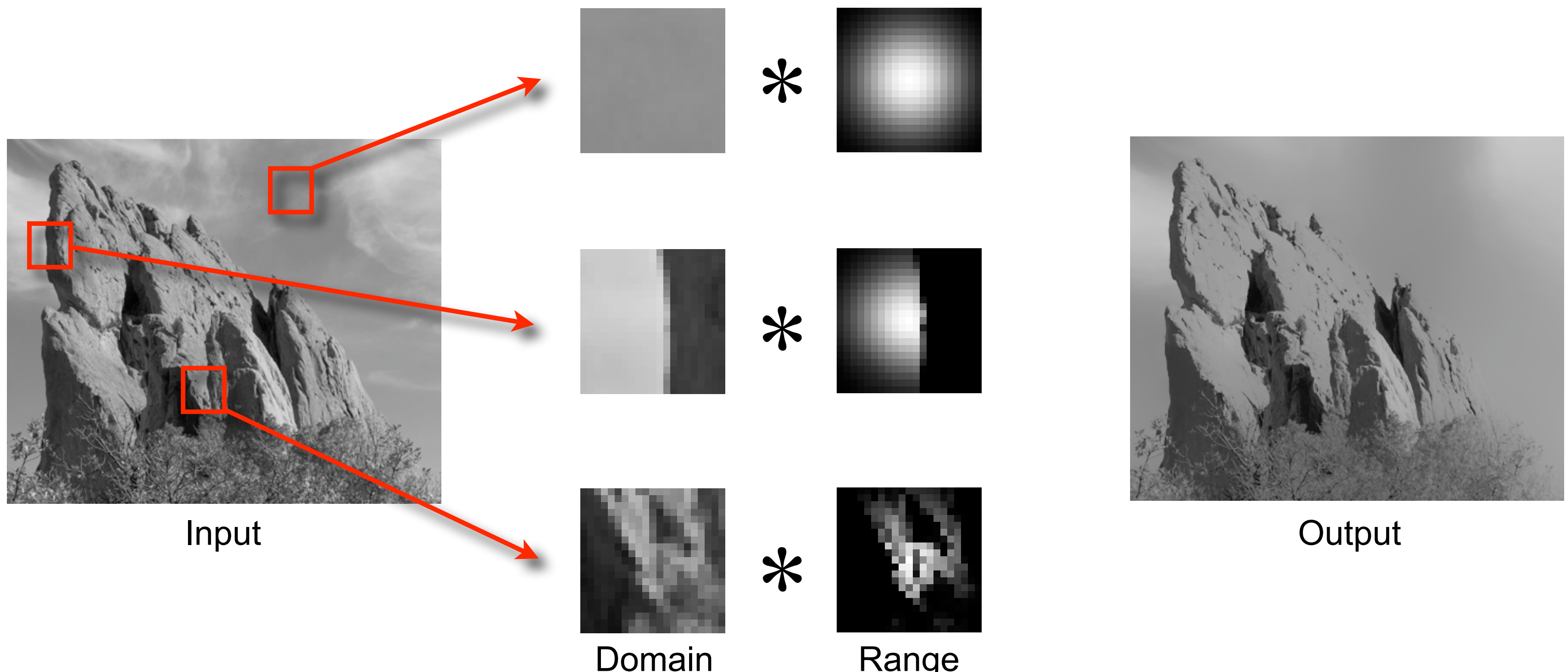
Image-Space Acceleration:
Joint Bilateral Upsampling
for Volume Rendering

Image-Space Acceleration

- Bilateral filter for image denoising

$$J_p = \frac{1}{k_p} \sum_{q \in \Omega} I_q f(\| p - q \|) g(\| I_p - I_q \|)$$

I = Image
R = Reference Image
f = spatial filter
g = range filter
p = position of center pixel
k = normalization term
Omega = spatial support



[Tomasi and Manduchi 1998]

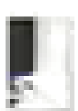
Example from Siggraph 2007 Tutorial by Sylvain Paris

Image-Space Acceleration

- Joint bilateral upsampling for efficient image enhancement

$$S_p = \frac{1}{k_p} \sum_{q_\downarrow \in \Omega} R_{q_\downarrow} f(\| p_\downarrow - q_\downarrow \|) g(\| I_p - I_q \|)$$

I = Image
R = Reference Image
f = spatial filter
g = range filter
p = position of center pixel
k = normalization term
Omega = spatial support

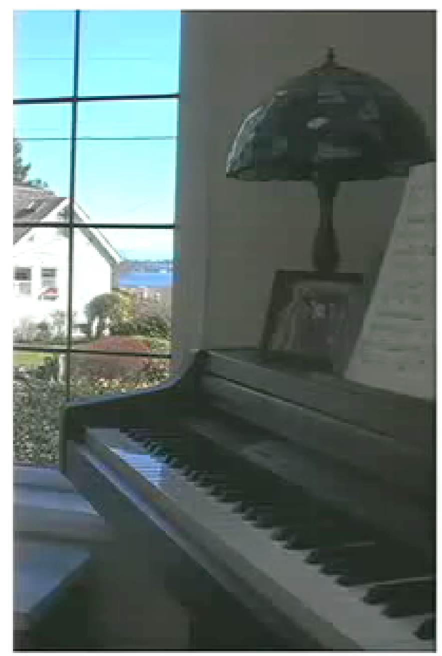


Reference Image (R)

*

Original Image (I)

=



Solution Image (S)

[Kopf et al. 2007]

Example from Siggraph 2007 presentation by Johannes Kopf

Image-Space Acceleration

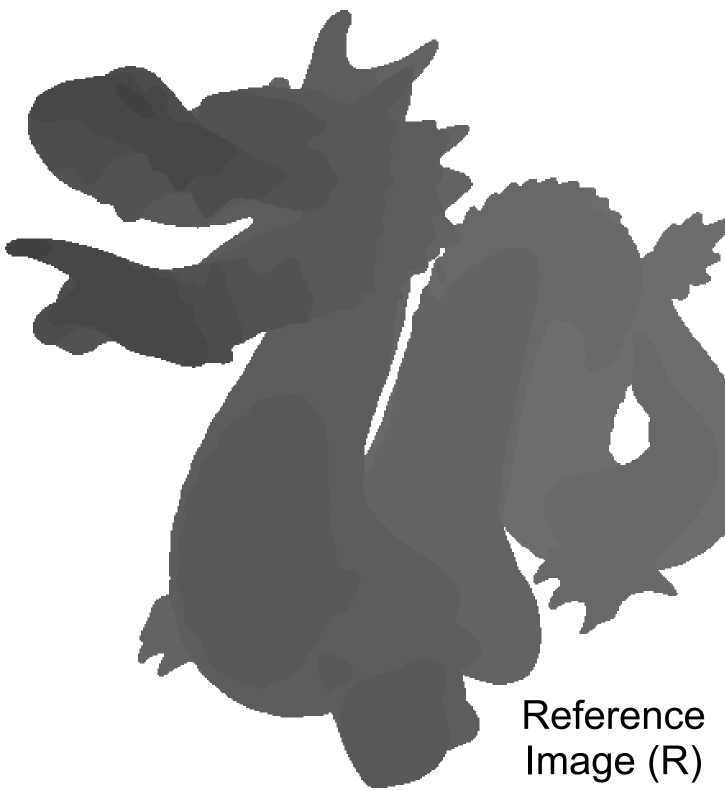
- Joint bilateral upsampling for accelerating rendering

$$S_p = \frac{1}{k_p} \sum_{q_\downarrow \in \Omega} I_{q_\downarrow} f(\| p_\downarrow - q_\downarrow \|) g(\| R_p - R_q \|)$$

I = Image
R = Reference Image
f = spatial filter
g = range filter
p = position of center pixel
k = normalization term
Omega = spatial support

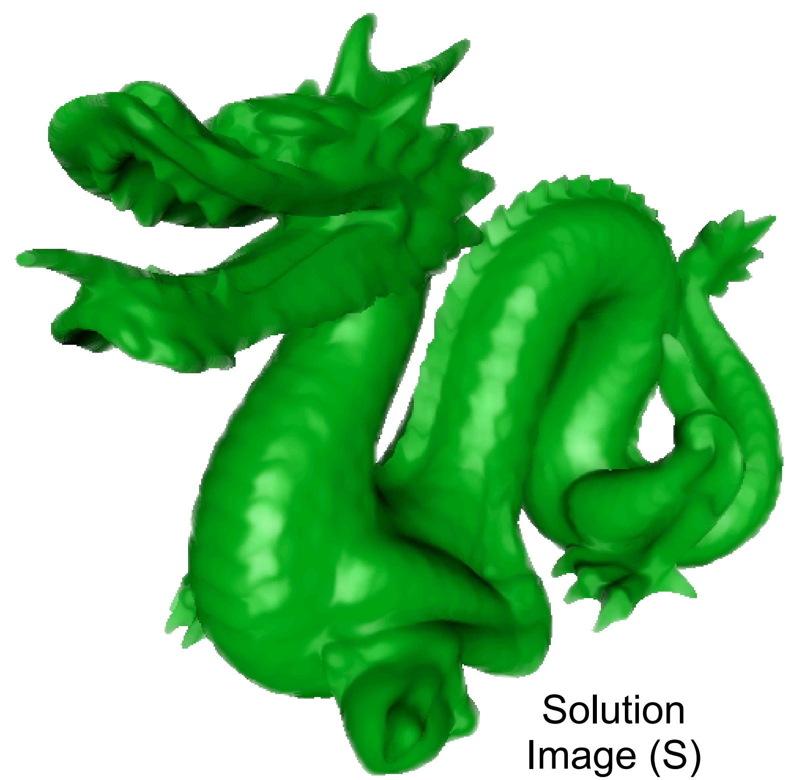
Image (I)

*



Reference Image (R)

=



Solution Image (S)

Image-Space Acceleration

- Effect similar to smoothing the geometry

Full Resolution

Joint Bilateral Upsampled x4

Image-Space Acceleration

- Overview
  - Render image into small offscreen buffer (I)
  - Render boundaries as n depth layers into large offscreen buffers ($R_1 \ldots R_n$)
  - Bind I and $R_1 \ldots R_n$ as textures and render large image (S) using joint bilateral upsampling

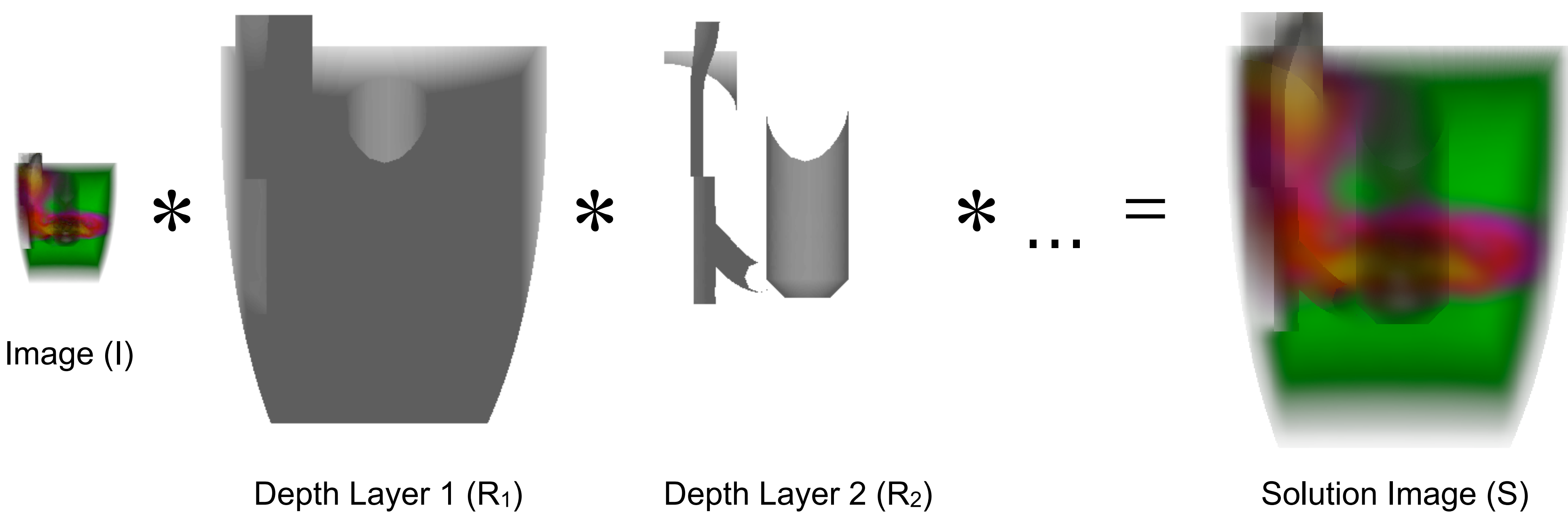
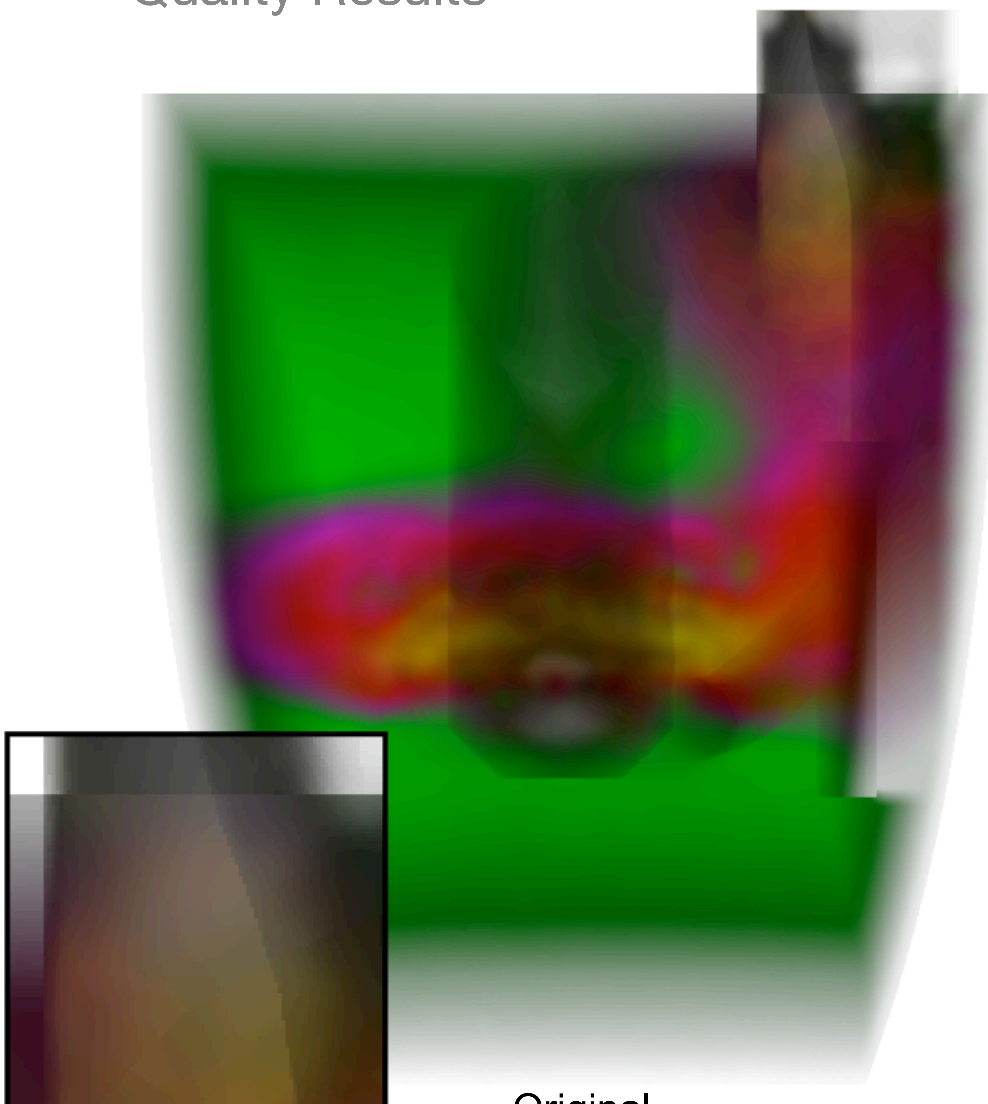
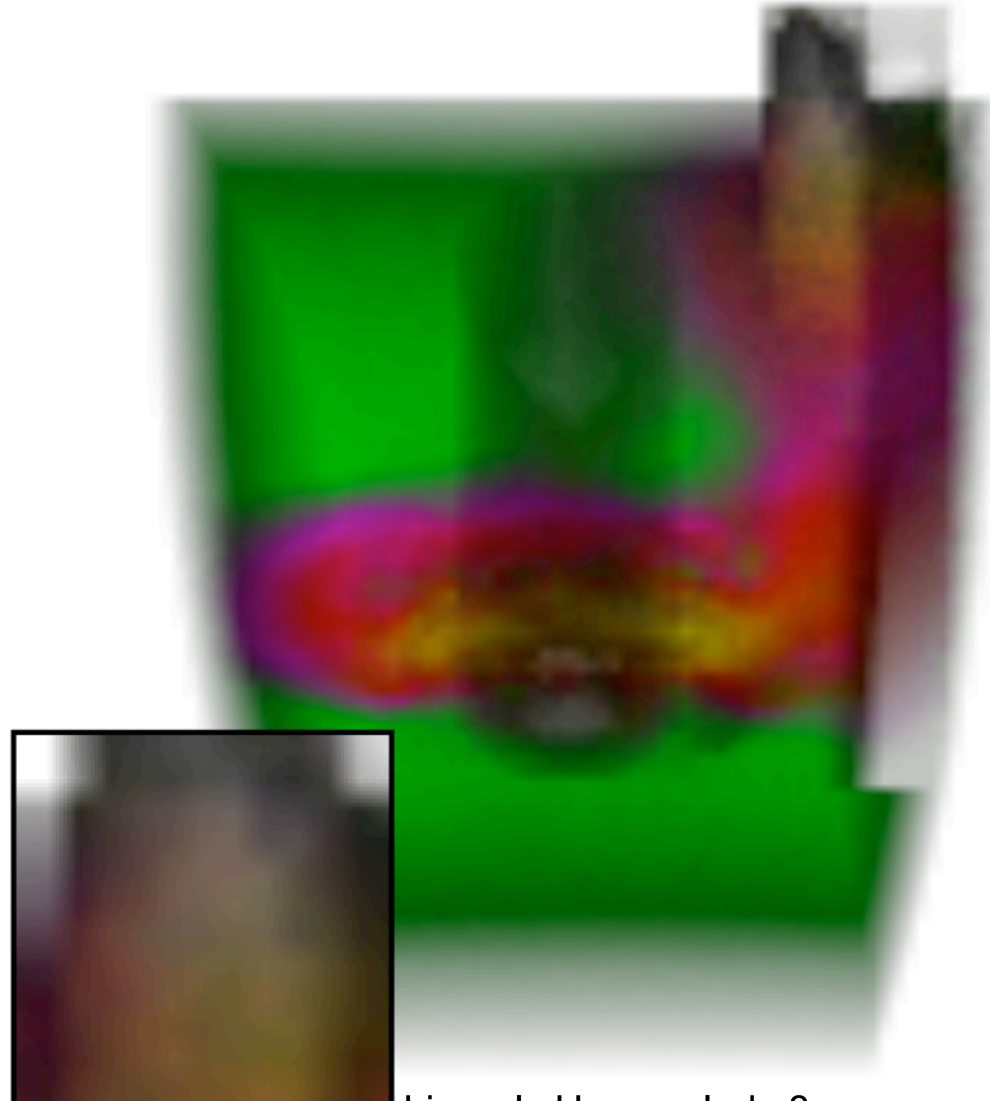


Image-Space Acceleration

- Quality Results



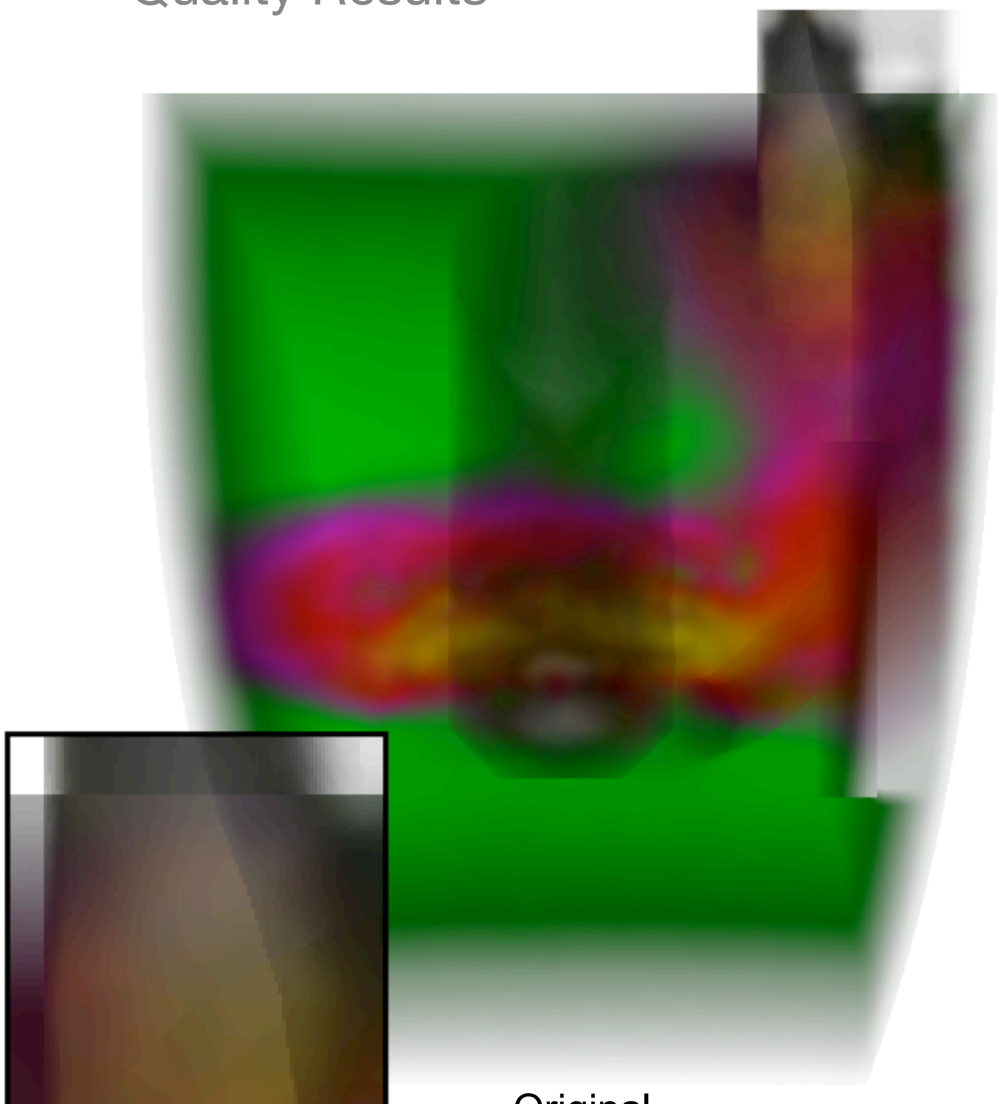
Original



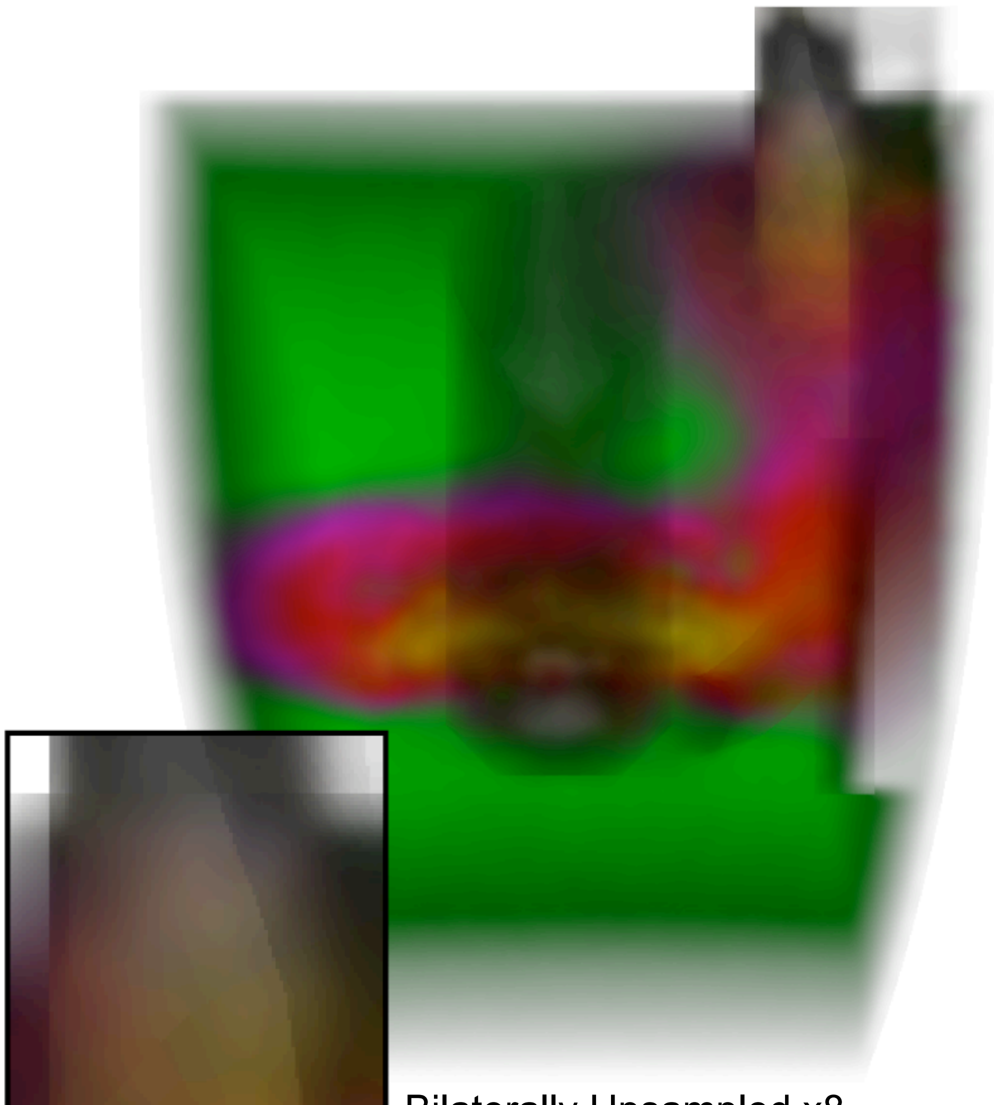
Linearly Upsampled x8

Image-Space Acceleration

- Quality Results



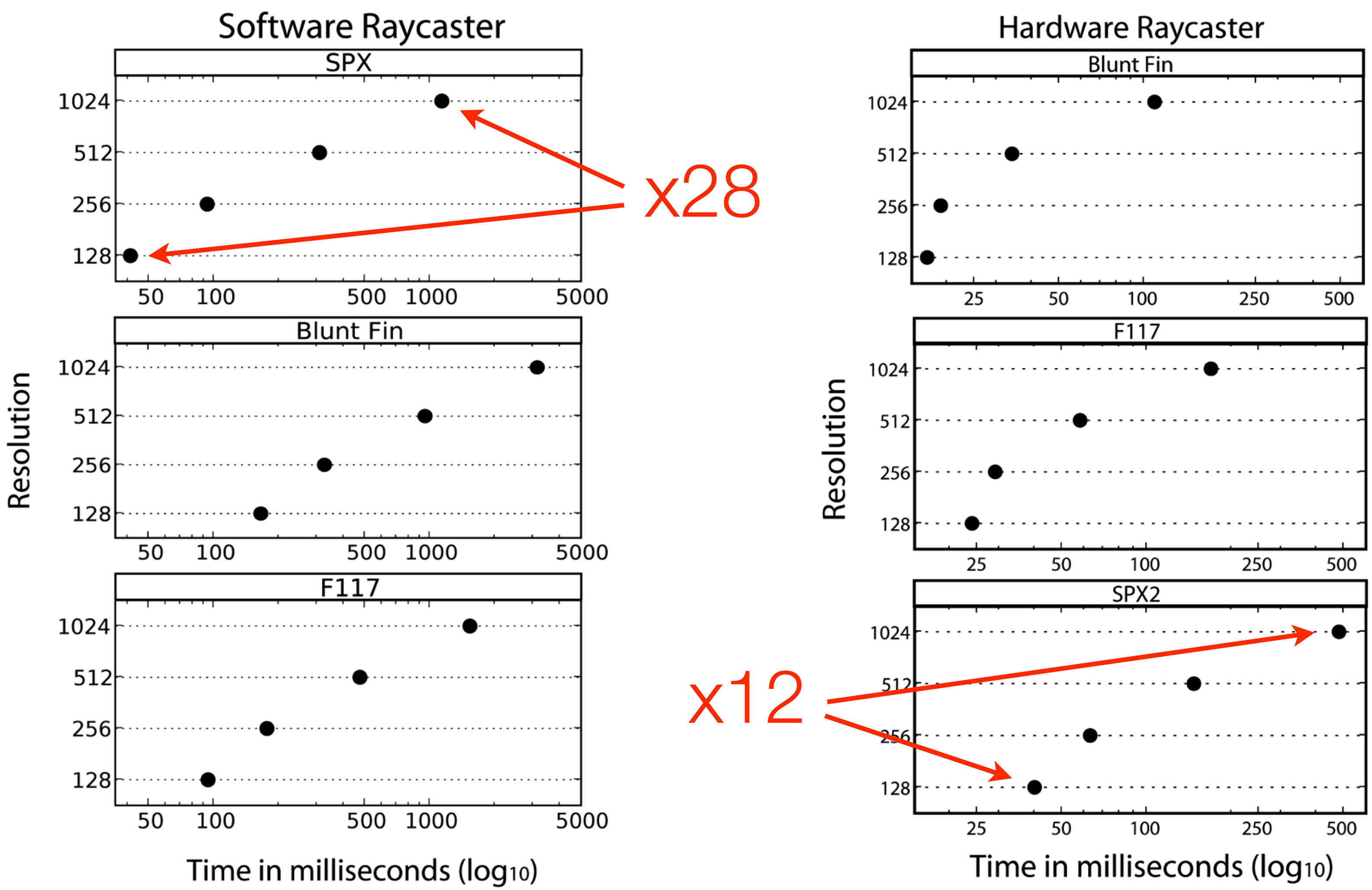
Original



Bilaterally Upsampled x8

Image-Space Acceleration

- Performance Results

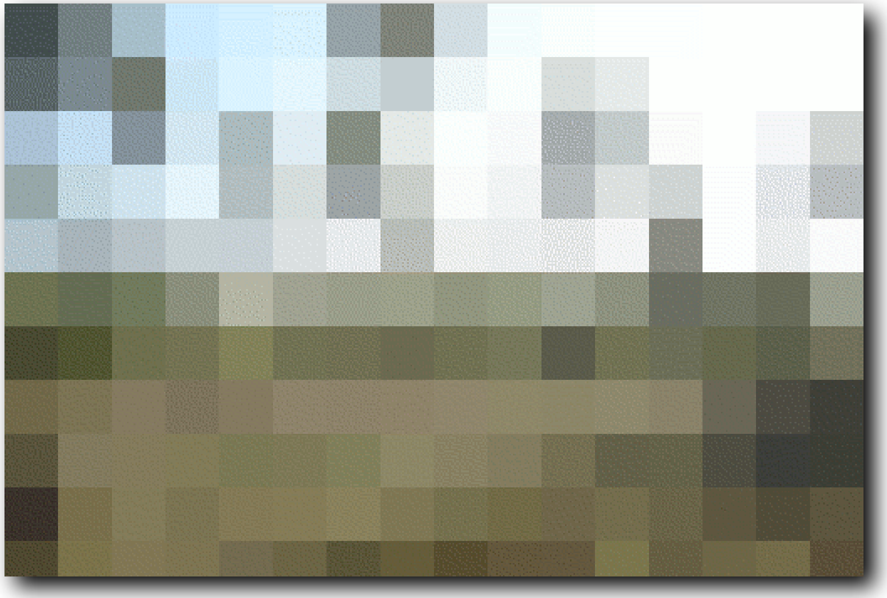


Increasing Allowable Data Size

Progressive Volume Rendering

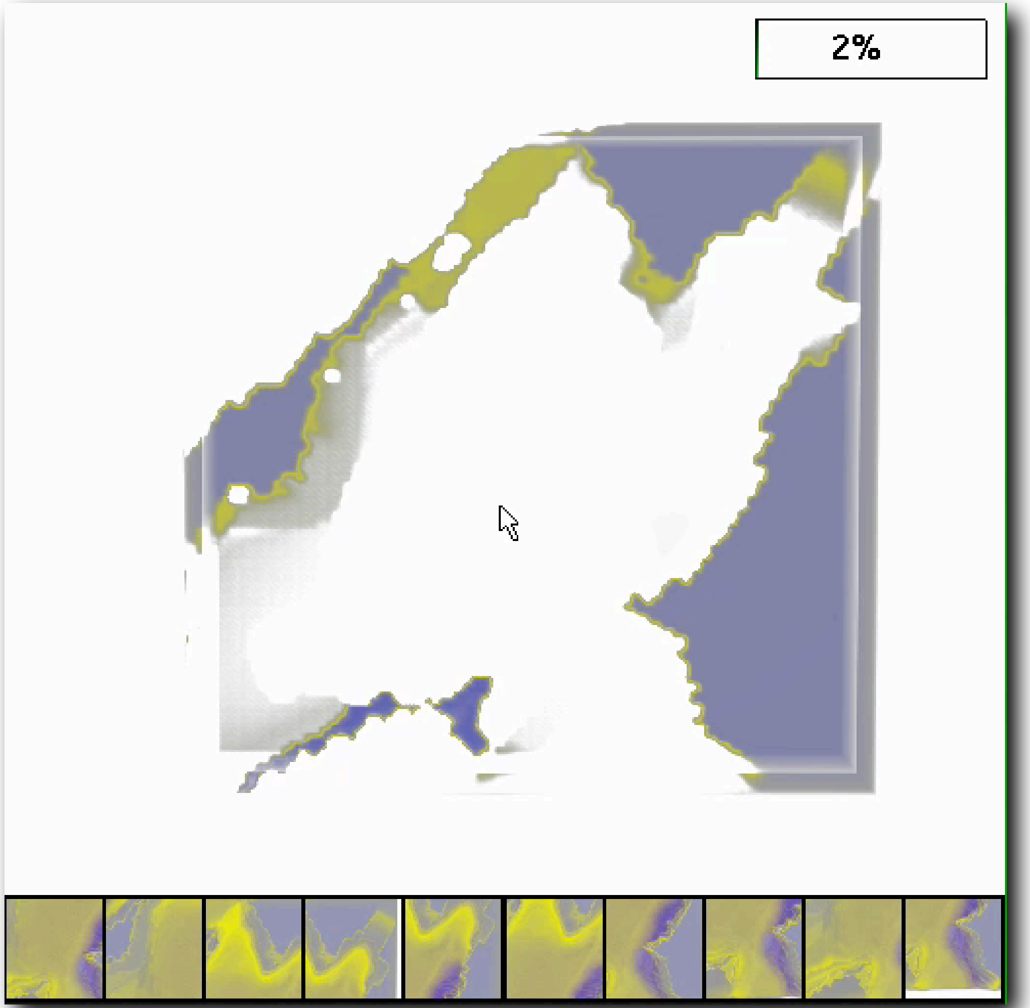
Progressive Volume Rendering

- Data too large to...
  - Render at once
  - Fit in memory
  - Render locally
- Render incrementally
  - Decompose (server)
  - Transmit (network)
  - Accumulate (client)



Progressive Volume Rendering

- Modes
  - Interactive
    - Boundaries
  - Progressive
    - Some internal finished
    - Some internal approximated
  - Completed
    - Save the image
- Configurations
  - Thin client
  - Robust client



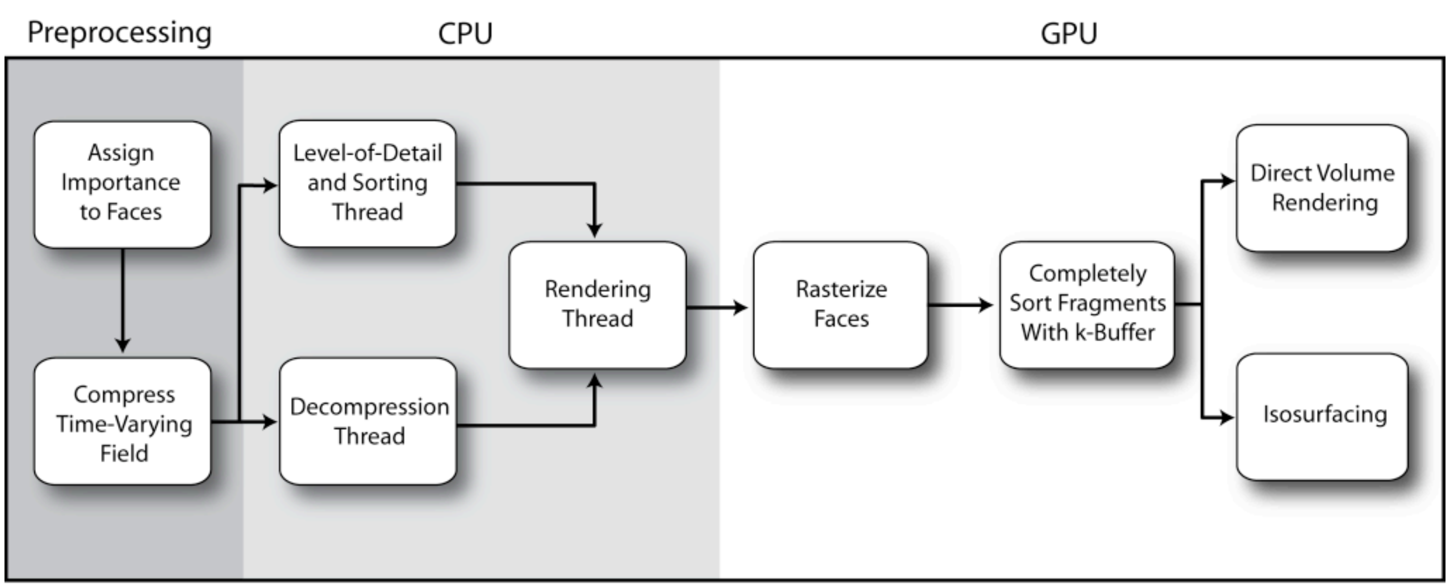
Handling Dynamic Data

Volume Rendering

Time-Varying Scalar Fields

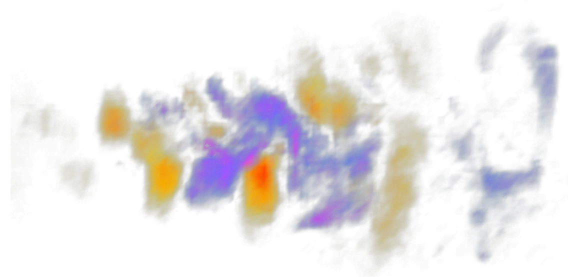
Time-Varying Scalar Fields

- Volume rendering
- Dynamic level-of-detail
- Compression and data transfer
- Parallel processing

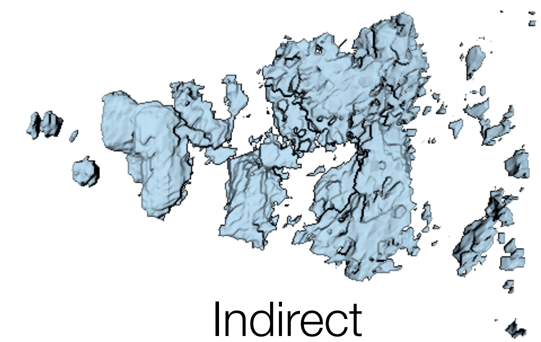


Time-Varying Scalar Fields

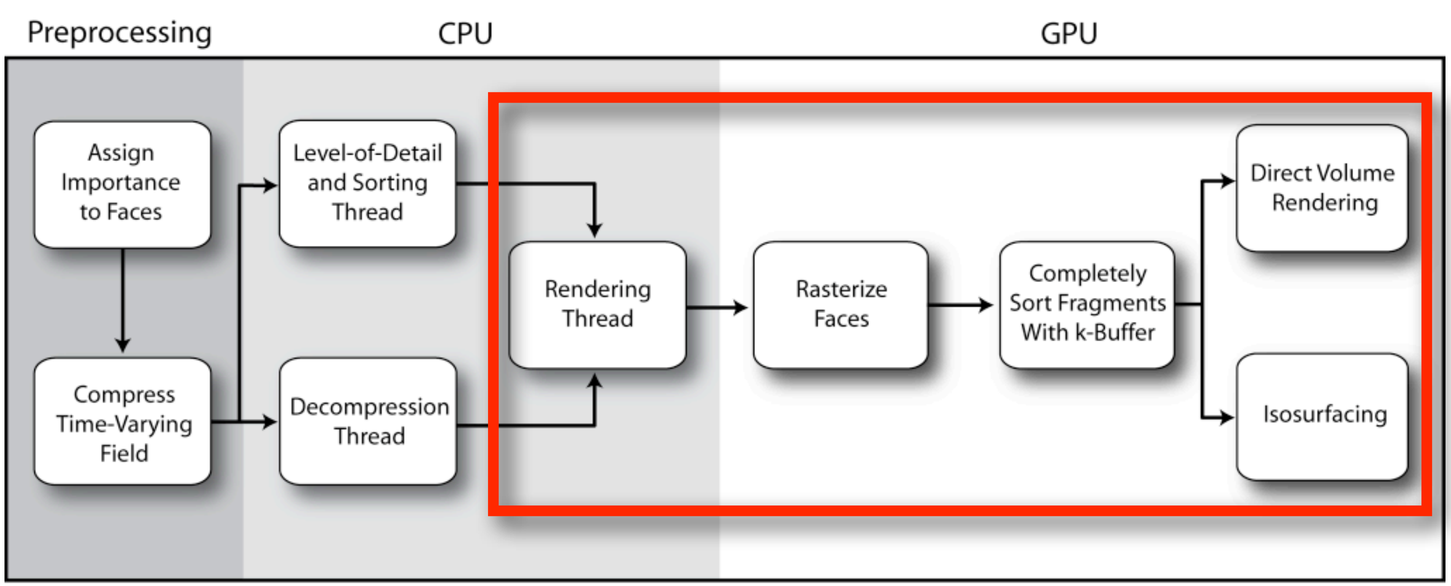
- Volume rendering
- Dynamic level-of-detail
- Compression and data transfer
- Parallel processing

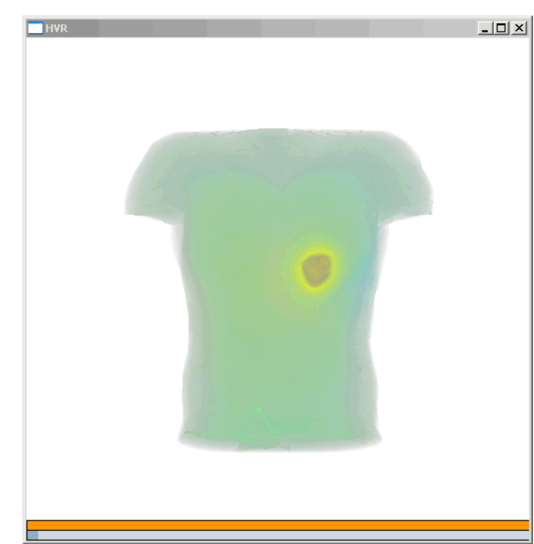


Direct



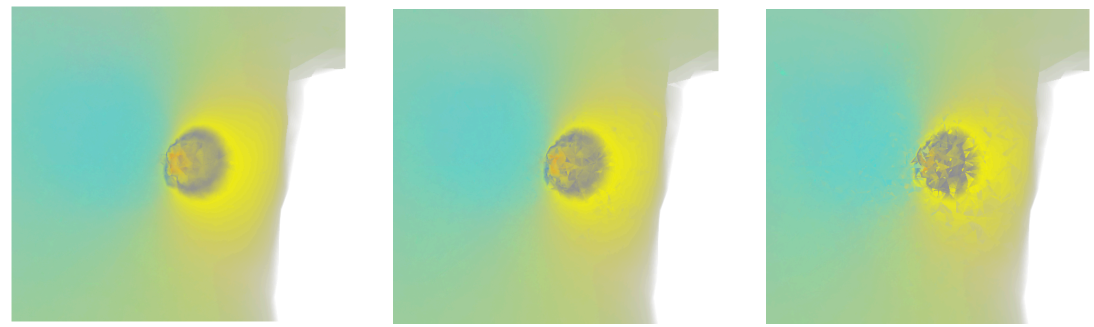
Indirect



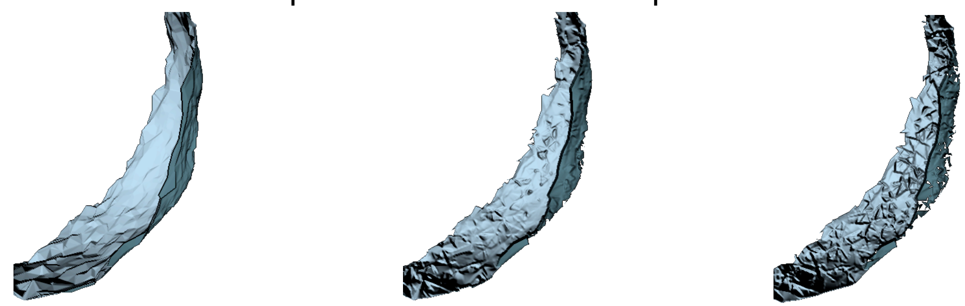


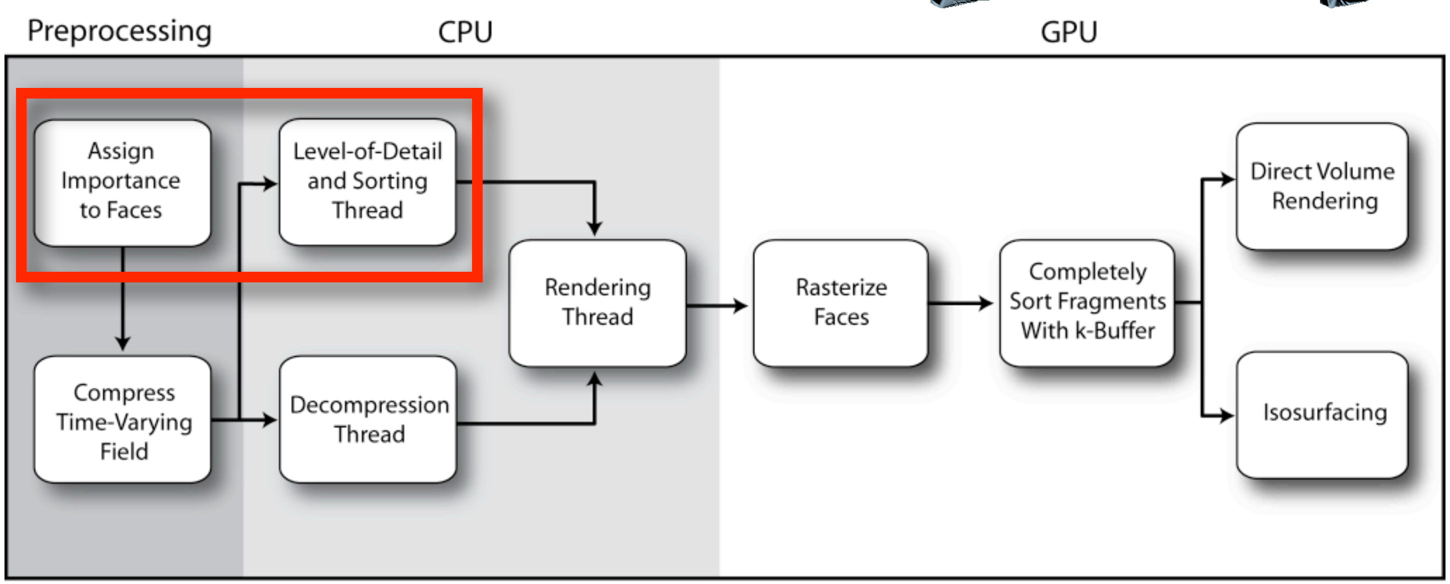
Time-Varying Scalar Fields

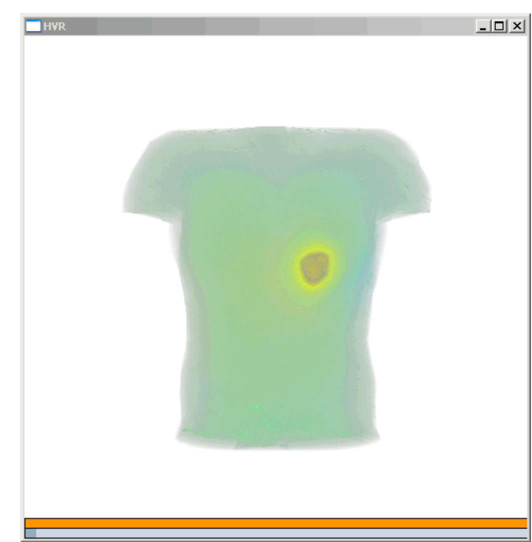
- Volume rendering
- Dynamic level-of-detail
- Compression and data transfer
- Parallel processing



100% at 18 fps   25% at 63 fps   10% at 125 fps





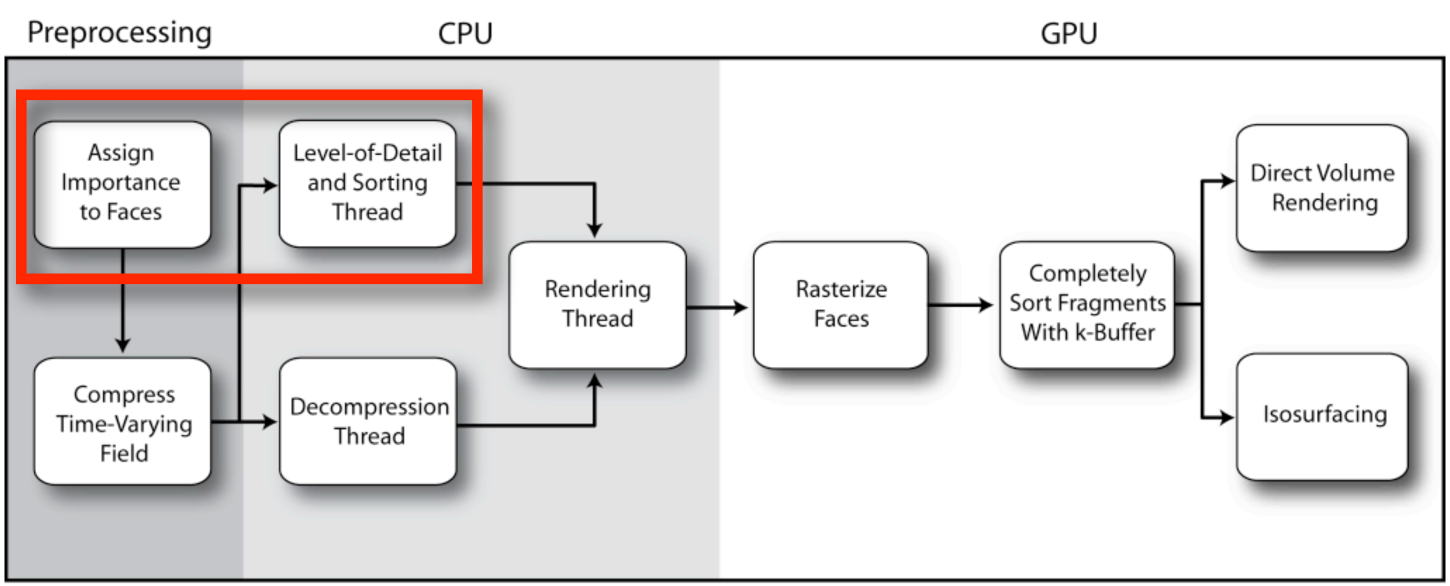


Time-Varying Scalar Fields

- Volume rendering
- Dynamic level-of-detail
- Compression and data transfer
- Parallel processing

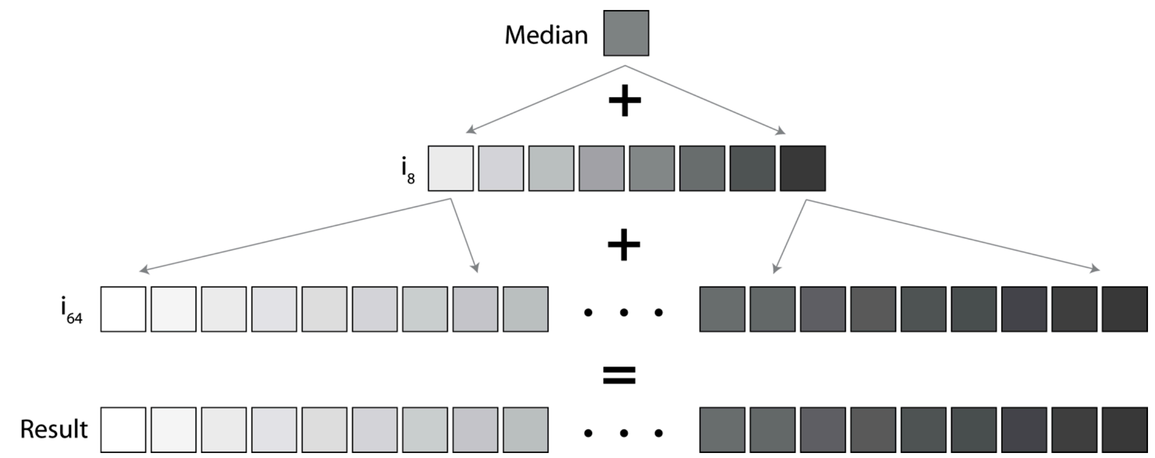
- Sampling strategies:
  - Statistically dynamic data
    - Local sampling
  - Statistically static data
    - Global Sampling

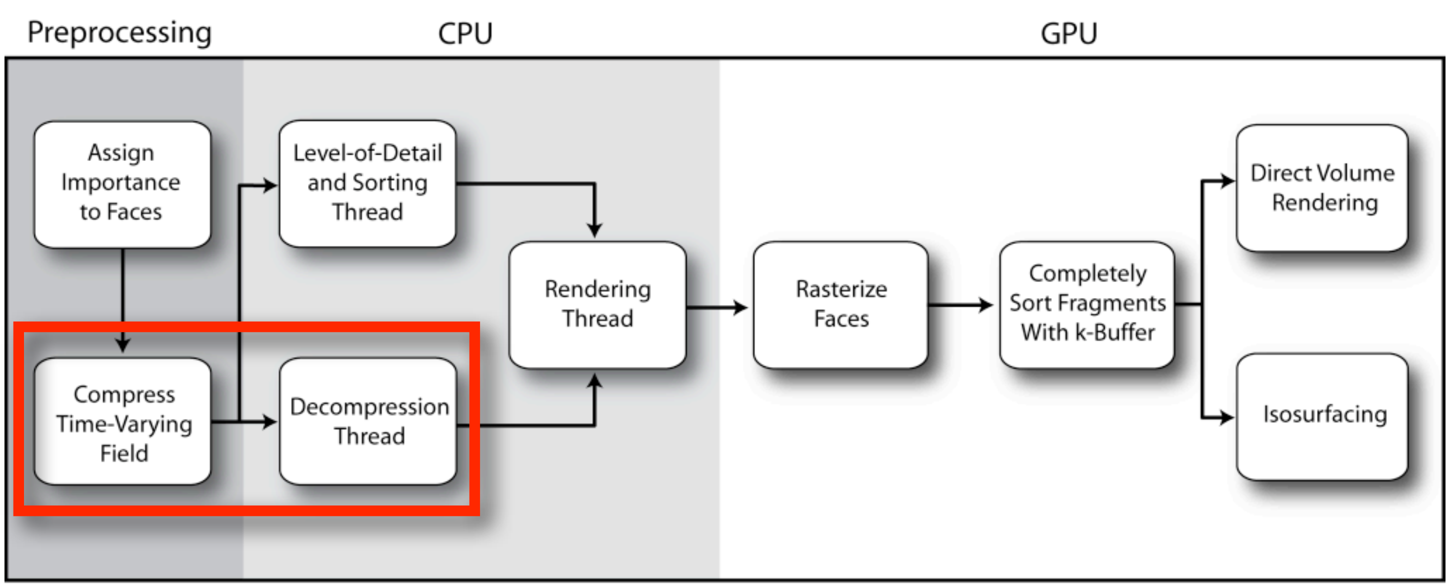
$$C_v(t) = \sum_{i=1}^{3} \frac{\sqrt{Var[X_{t(i)}]}}{E[X_{t(i)}]}$$



Time-Varying Scalar Fields

- Volume rendering
- Dynamic level-of-detail
- Compression and data transfer
- Parallel processing

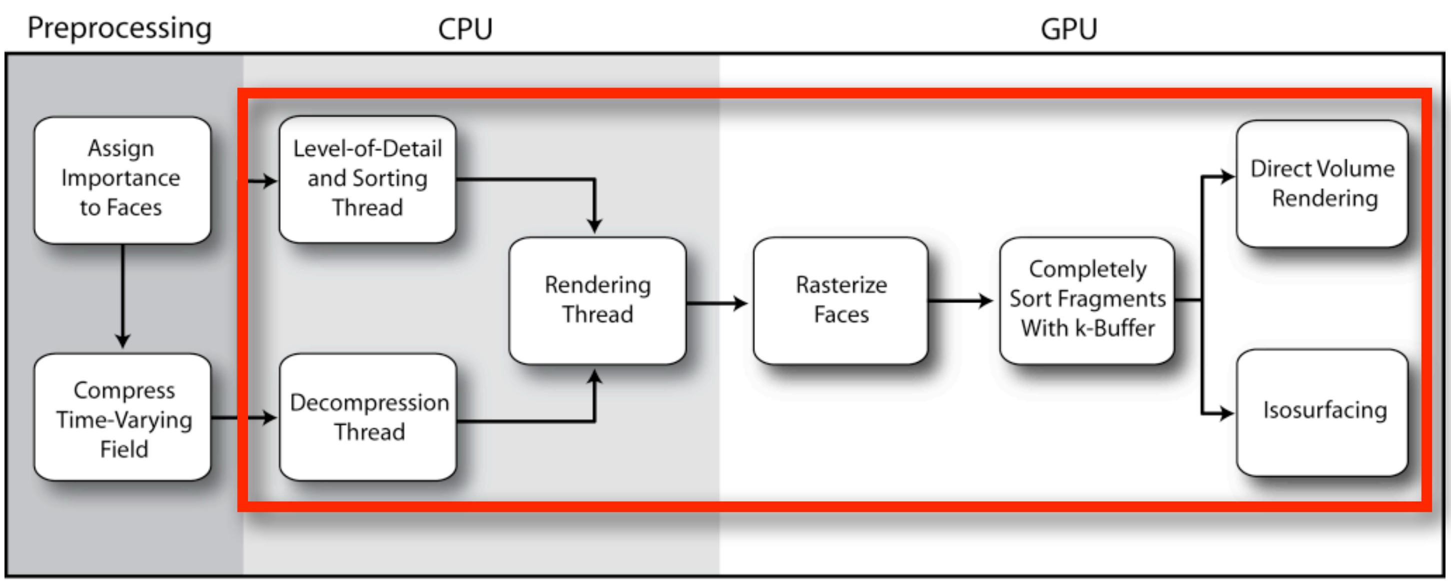




Time-Varying Scalar Fields

- Volume rendering
- Dynamic level-of-detail
- Compression and data transfer
- Parallel processing

- Manage all resources with threads
  - Level-of-Detail and Sorting
  - Decompression
  - Rendering

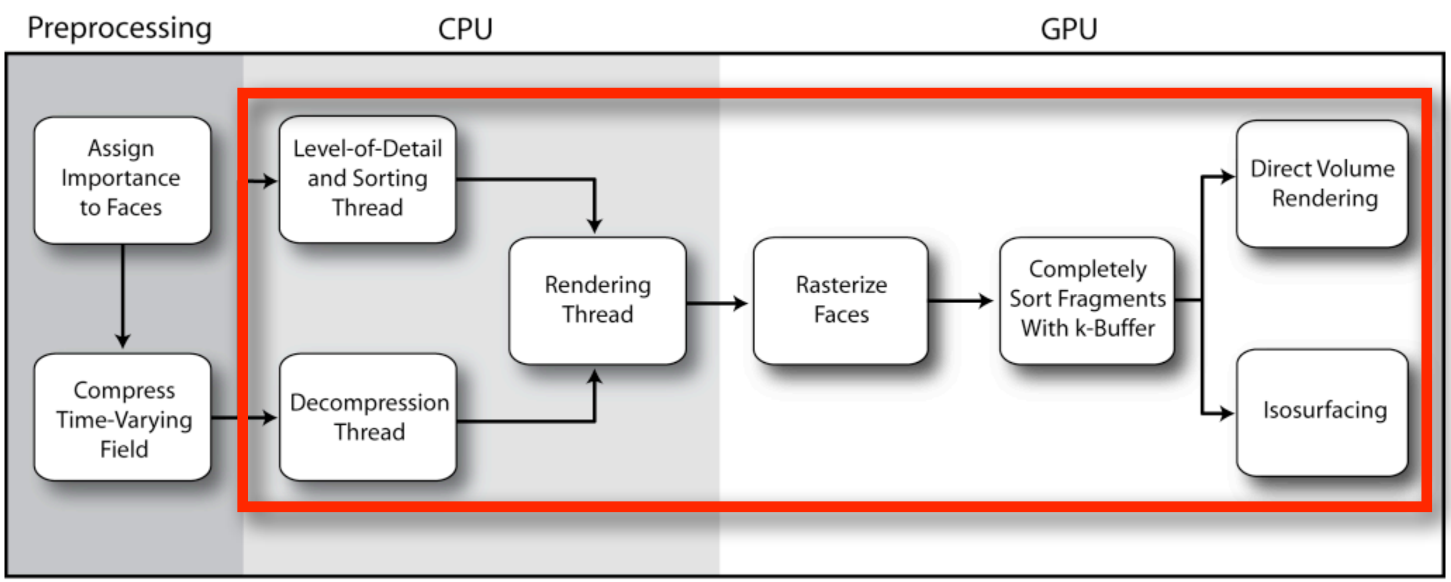


Time-Varying Scalar Fields

- Volume rendering
- Dynamic level-of-detail
- Compression and data transfer
- Parallel processing

- Manage all resources with threads
  - Level-of-Detail and Sorting
  - Decompression
  - Rendering

Results: dynamic data is only 5% slower than static data



Enabling Volume Exploration

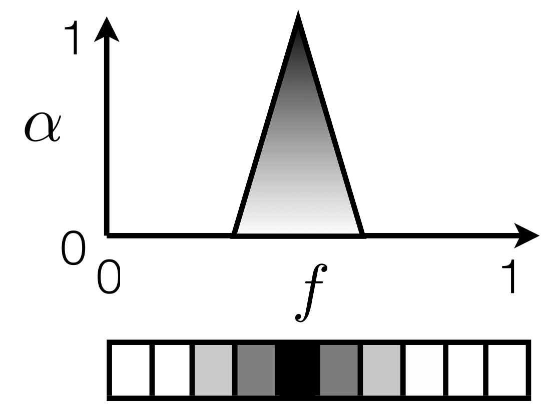
Transfer Function Specification

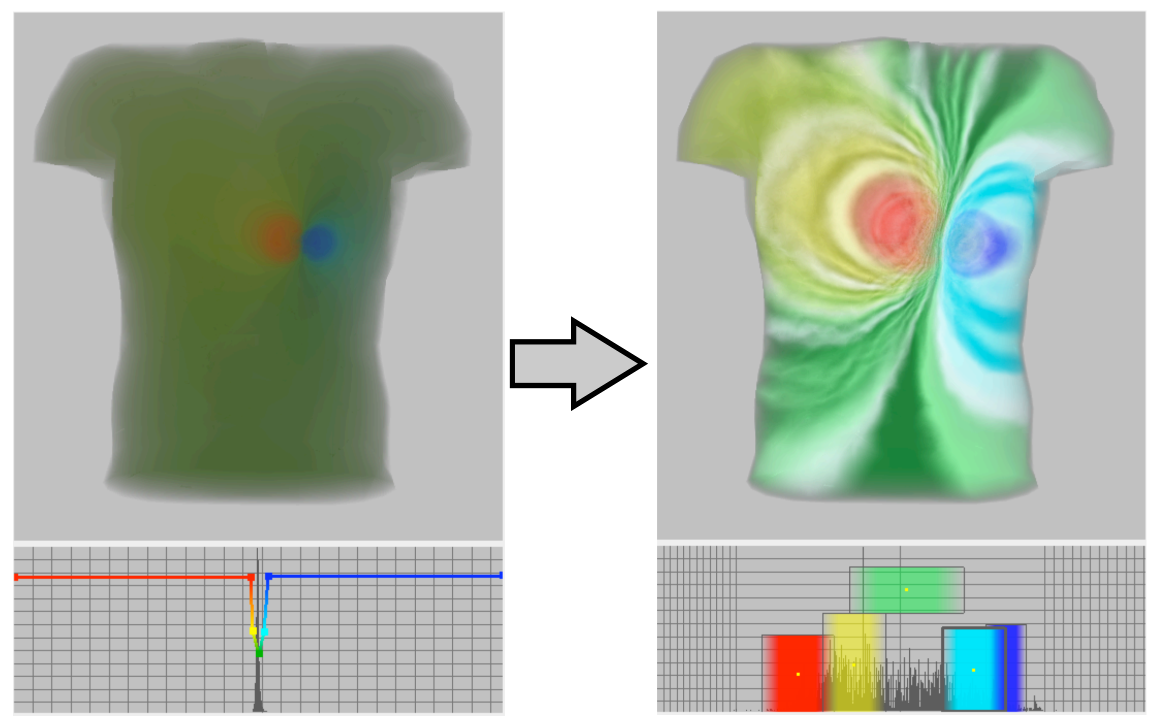
Transfer Function Specification

- Transfer Functions
  - Maps a data value to color and opacity

$$f(x) = \mathbf{R} \rightarrow \mathbf{R}^4, s \rightarrow (r, g, b, \alpha)$$

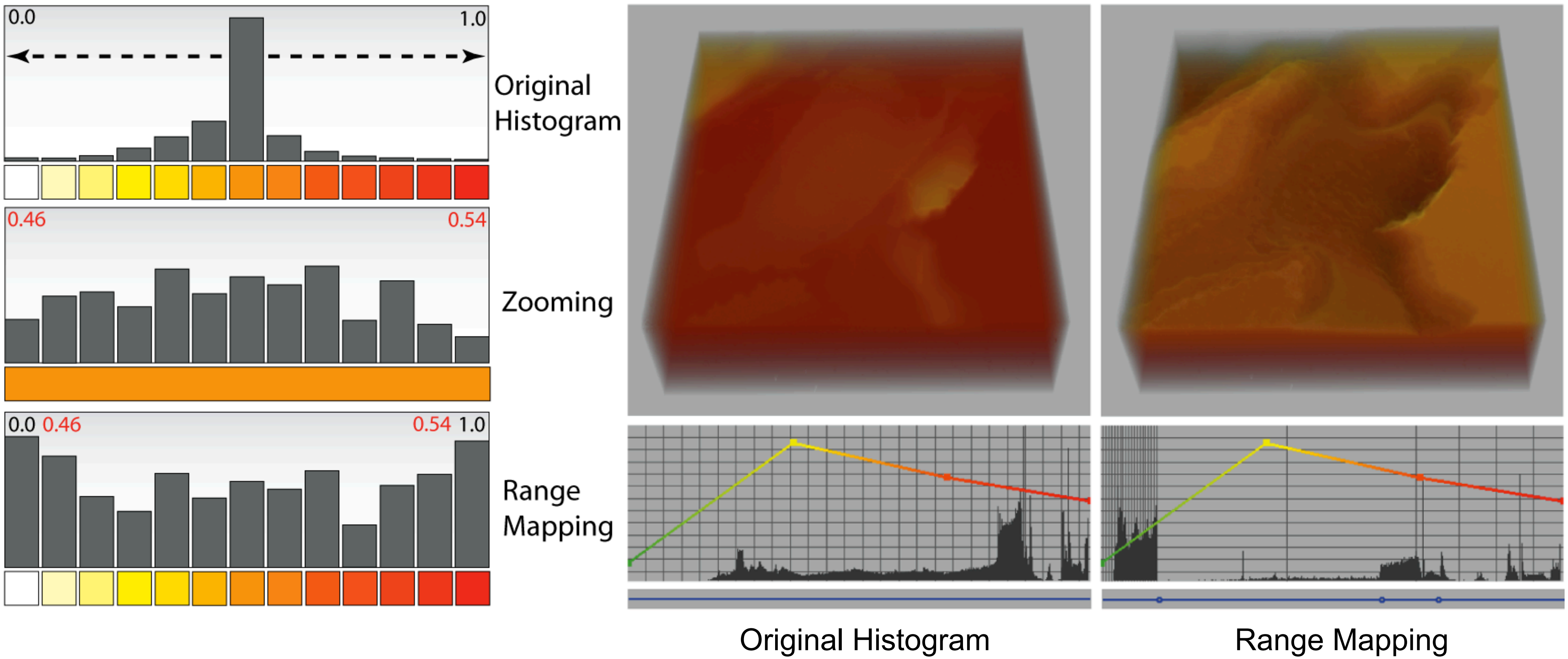
  - Lookup table
    - Fixed number of bins
  - Problems for unstructured grids
    - High-dynamic range data
    - Multiple features
    - Time-varying data





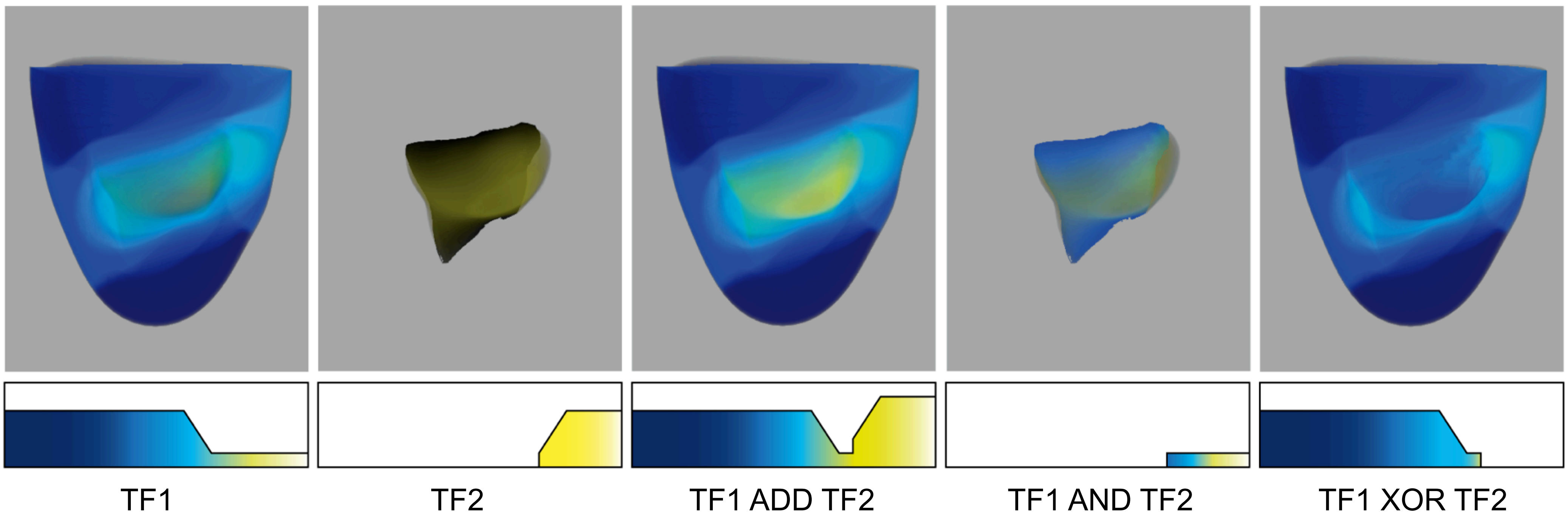
Transfer Function Specification

- Range Mapping for high-dynamic range data



Transfer Function Specification

- Blending for feature finding



ADD

$$C_r(i) = C_1(i) + C_2(i)$$
$$\alpha_r(i) = \alpha_1(i) + \alpha_2(i)$$

AND

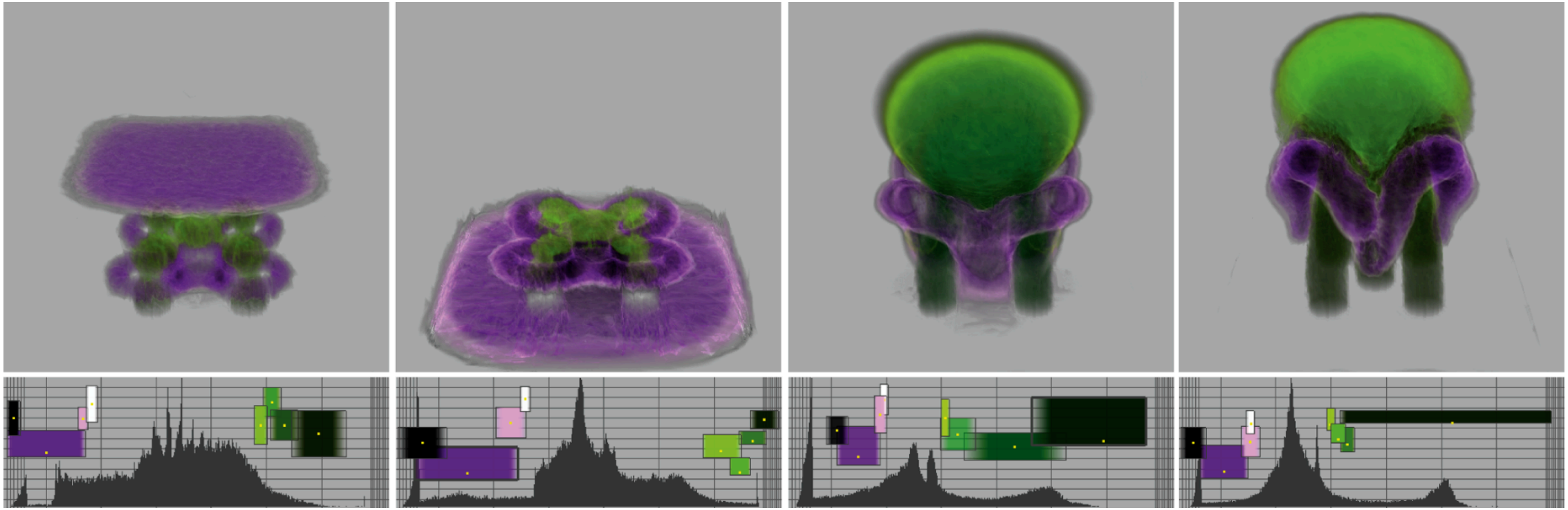
$$C_r(i) = Max(C_1(i), C_2(i))$$
$$\alpha_r(i) = Min(\alpha_1(i), \alpha_2(i))$$

XOR

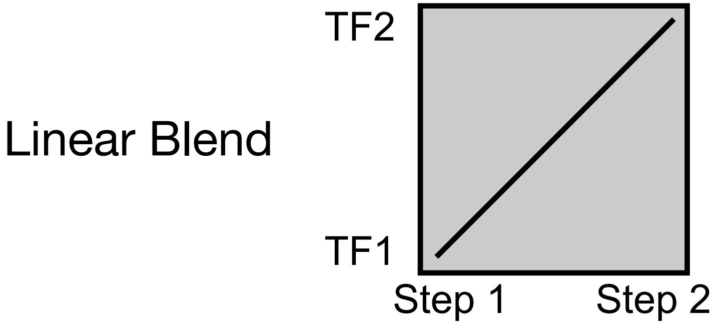
$$C_r(i) = (C_1(i) \wedge \overline{C_2(i)}) \vee (\overline{C_1(i)} \wedge C_2(i))$$
$$\alpha_r(i) = (\alpha_1(i) \wedge \overline{\alpha_2(i)}) \vee (\overline{\alpha_1(i)} \wedge \alpha_2(i))$$

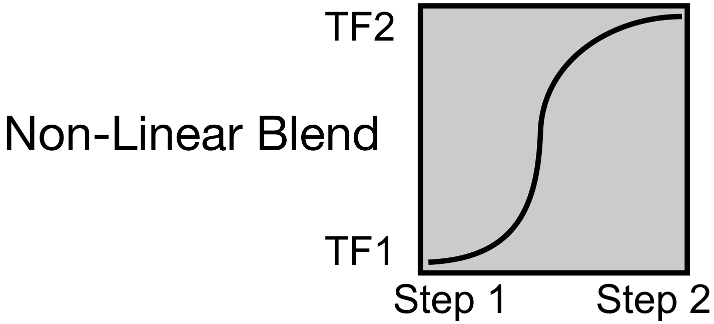
Transfer Function Specification

- Keyframing for time-varying data

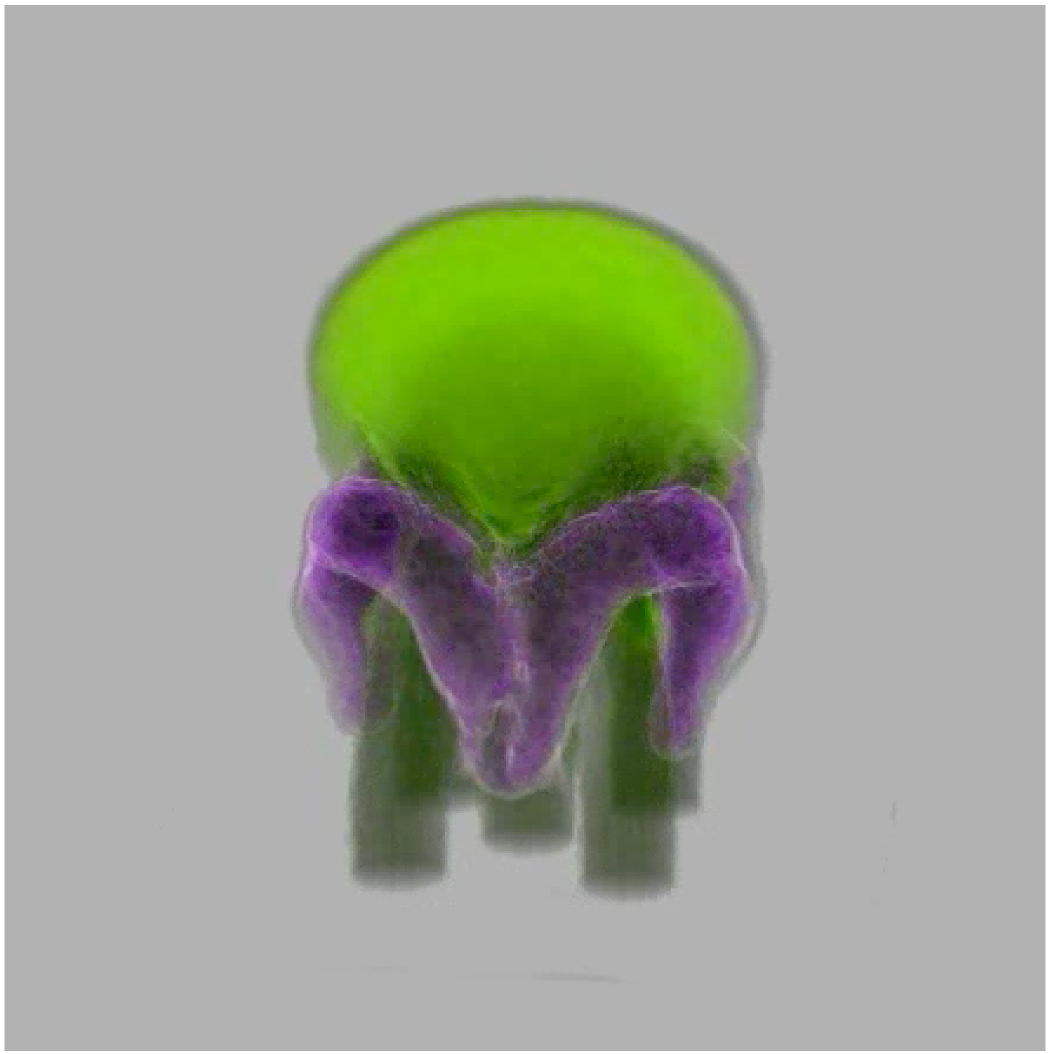


Time Step 12 Time Step 15 Time Step 32 Time Step 35

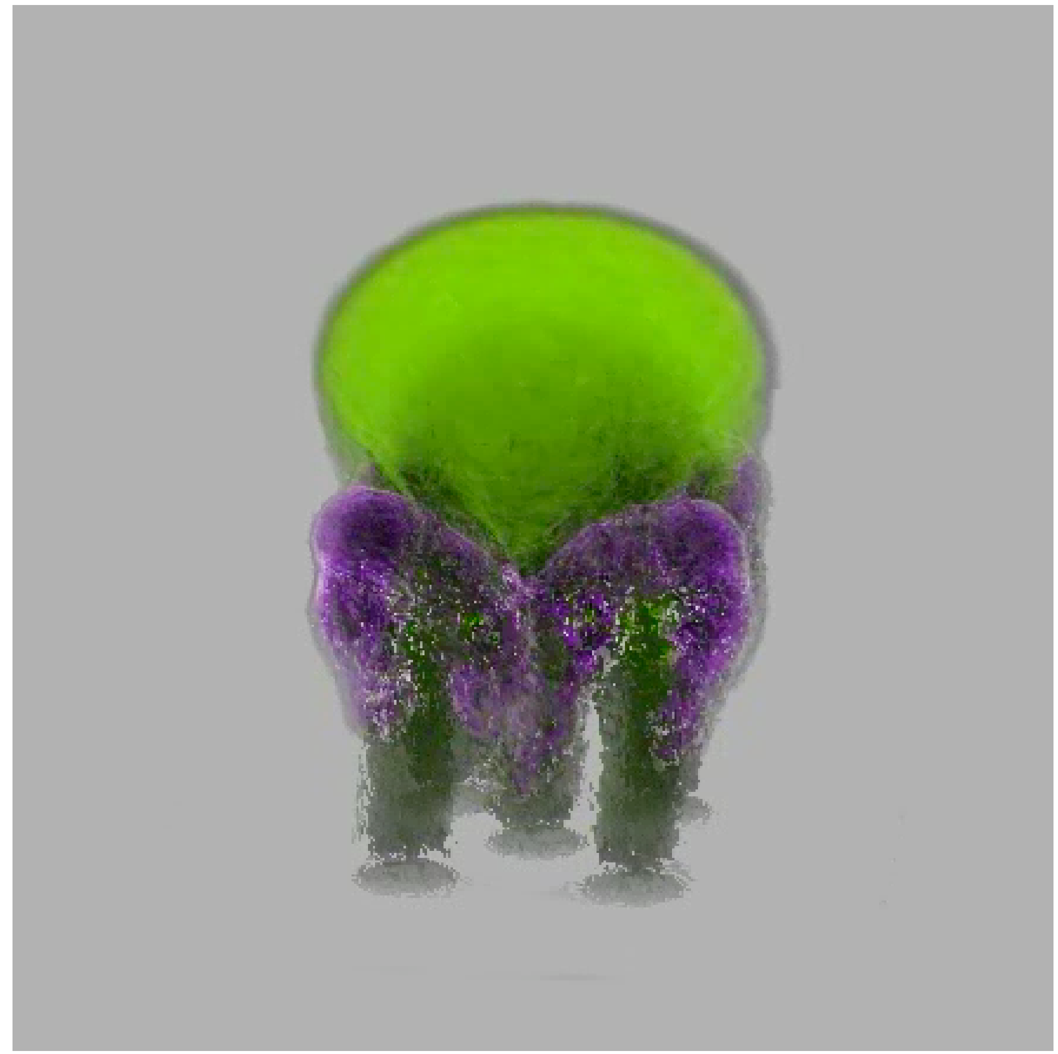




Results



Full Quality



20% LOD

1 Million Tetrahedra
40 Time Steps
4 Transfer Functions

Summary

This dissertation introduced new algorithms and frameworks that efficiently use available hardware for improving the state-of-the-art in volume rendering for large, dynamic unstructured volumes

Contributions:

- Image-space acceleration for volume rendering
- Object-space acceleration for volume rendering
- Progressive volume rendering for large data
- Time-varying volume rendering for dynamic data
- Transfer function specification for large, dynamic data

Future Work

- Bricking strategies for unstructured meshes
- Stencil-routed k-buffer for volume rendering
- Dynamic geometry/topology
- Higher order elements
- Source code release of complete tool

Acknowledgments

- Advisor:
  - Clàudio Silva
- Committee:
  - Clàudio Silva, Peter Shirley, Mike Kirby, Valerio Pascucci, João Comba
- Contributors:
  - Erik Anderson, Fabio Bernardon, Louis Bavoil, João Comba, Linh Ha, Valerio Pascucci, Carlos Scheidegger, John Schreiner, Clàudio Silva
- Other:
  - SCI, SoC, VGC, VisTrails groups
  - Karen, Deb, and other administrative staff
  - Faculty and Staff in SCI and SoC
- Funding:
  - Department of Energy, National Science Foundation, U of U Graduate Research Fellowship
- Family:
  - Kristy Callahan, Lowell and Janice Callahan, Helen and Verlon Duncan

Thank You!

Questions?