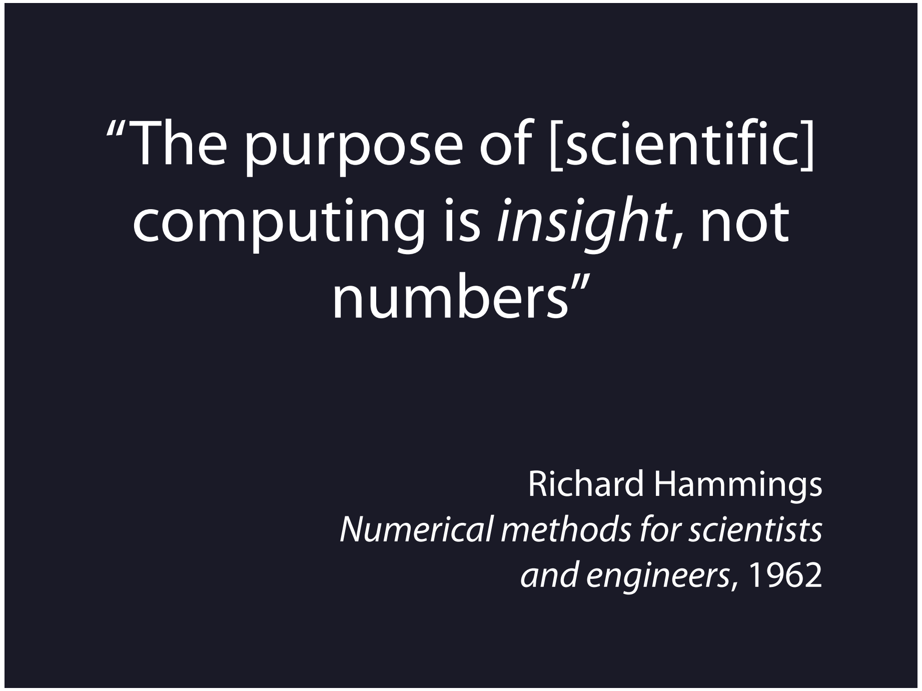
Interactive Visual Data Analysis of Terascale Data: an Achieveable Goal?

Steffen Prohaska

Zuse Institute Berlin

Thanks for the introduction and the opportunity to visit CCT.
Obviously I will not answer this questions but can only present some recent work we did which might help to advance towards this goal. I will conclude with some discussion on what would be needed.

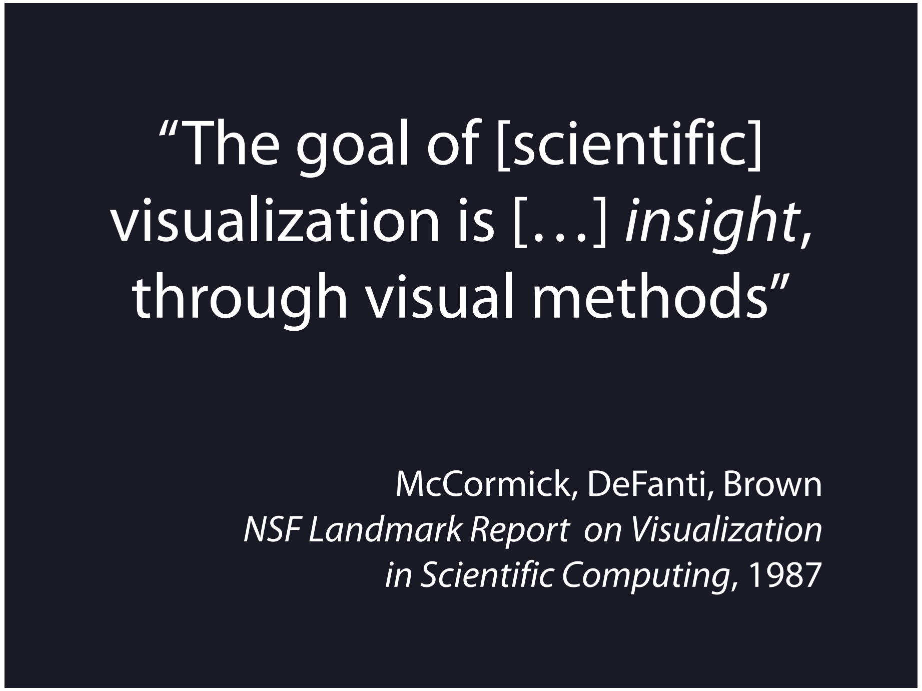
But fist a few words on Visual Data Analysis.
[NEXT SLIDE]



“The purpose of [scientific] computing is *insight*, not numbers”

Richard Hammings
Numerical methods for scientists and engineers, 1962

“The purpose of scientific computing is insight not numbers” – Richard Hammings wrote this often cited statement in his book on Numerical methods for scientists and engineers. He reserved a whole page for this scentence, underlining that it is valuable to remember from time to time that number crunching is not a pupose of its own. **[NEXT SLIDE]** In the same spirit, the 1987 NSF Landmark report defined “the goal of scientific visualization is insight through visual methods.”



“The goal of [scientific] visualization is […] *insight*, through visual methods”

McCormick, DeFanti, Brown
NSF Landmark Report on Visualization
in Scientific Computing, 1987

[cont] In this spirit the 1987 NSF Landmark Report stated “the goal of scientific visualization is insight through visual methods”. So, the main purpose of scientific visualization is not beautiful images but new insight.

[NEXT SLIDE] Visual Data Analysis is perhaps a better term for this.

Visual Data Analysis

[cont] Visual Data Analysis is perhaps a better term for this field.

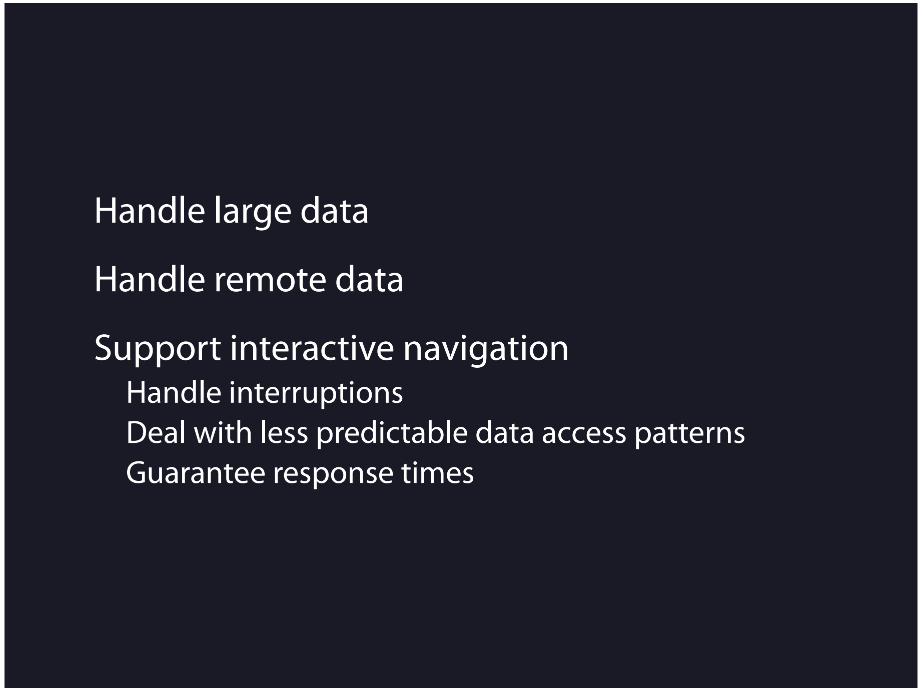
Visual Data Analysis is the topic we’re mainly concerned with at the Department of Visualization at Zuse Institute Berlin.

[NEXT SLIDE] I’d like to start with listing some requirements on visual data analysis when dealing with massive data

Requirements

[cont] I’d like to start with listing some requirements for visual data analysis when dealing with massive data

[NEXT SLIDE] Obviously it has to be able to handle large data



Handle large data

Handle remote data

Support interactive navigation

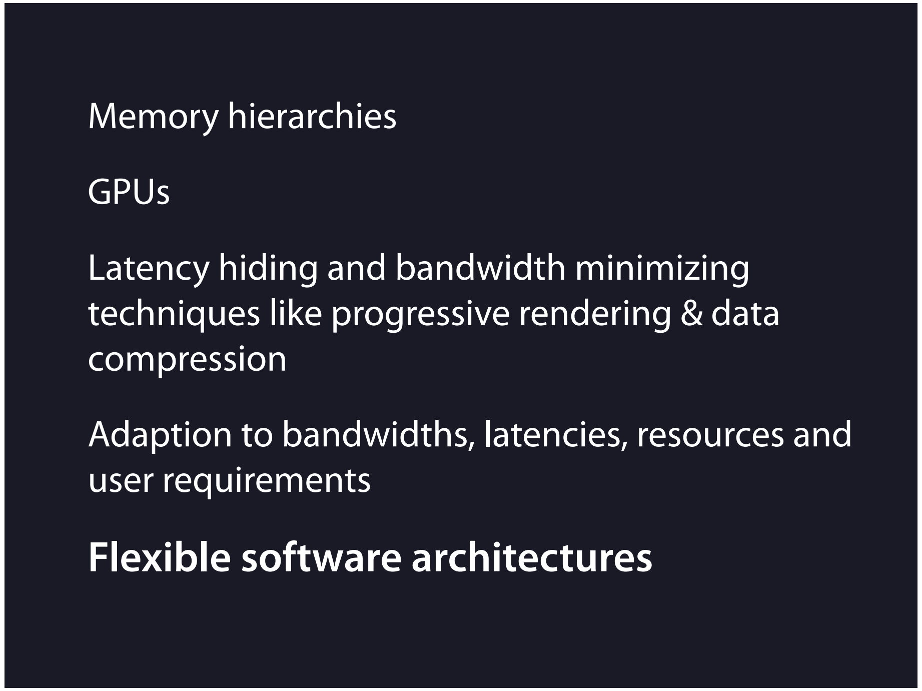
- Handle interruptions
- Deal with less predictable data access patterns
- Guarantee response times

[cont] Obviously it has to be able to handle large data. Because large data are often generated and stored centrally, it has to handle remote data.

I also listed interactive navigation. When doing research on new data, ‘Exploration’ is an important tool. Often the interesting features of the data are not known. Therefore, evaluation can not be done as batch processing. Interactive analysis is a promising way to gain new insight. Providing interactive navigation in large data is hard to provide. Some more specific requirements are to handle interruptions and guarantee response times. For rendering, interactive framerates, are required. For navigation, a system typically should be able to generate a response in less than 2 seconds.
[NEXT SLIDE] Topics emerging when working on this problem are.

Topics

[cont] Topics emerging when working on this problem are
[NEXT SLIDE] to Efficiently utilize memory hierarchies. Starting at the CPU on L1, L2, L3 caches, main memory, disks, LAN, WAN are available with various latencies and bandwidths.



Memory hierarchies

GPUs

Latency hiding and bandwidth minimizing techniques like progressive rendering & data compression

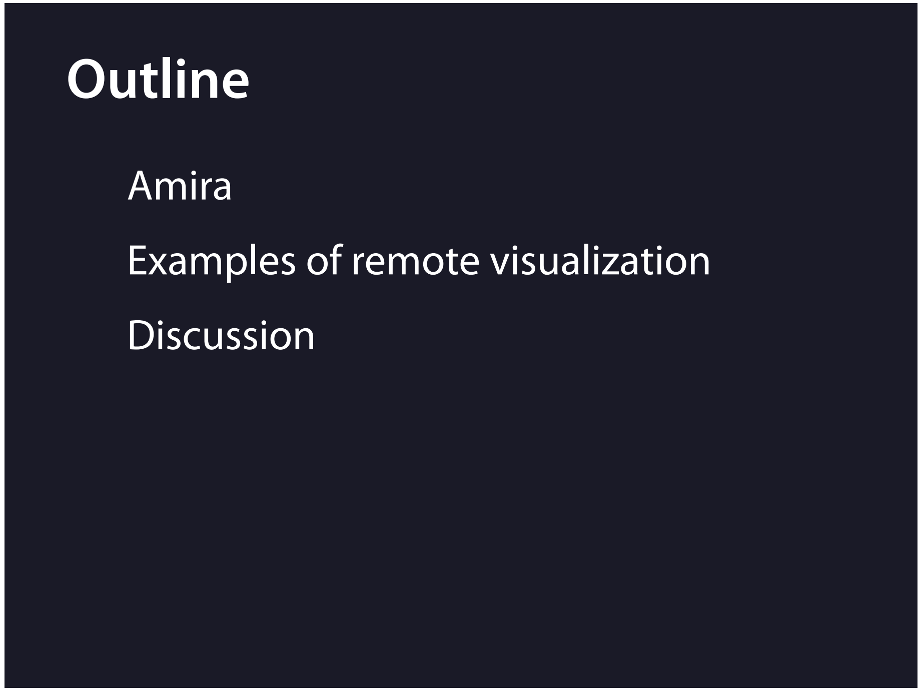
Adaption to bandwidths, latencies, resources and user requirements

Flexible software architectures

[NEXT SLIDE] to efficiently utilize memory hierarchies. Starting at the CPU: L1, L2, L3 caches, main memory, disks and networks are available with various latencies and bandwidths. Data structures and algorithms which perform well in memory hierarchies are required. Graphics processors will provide their own virtual memory management system in the near future and demand similar techniques.

When it comes to remote visualization, dealing with latencies is a major issue. Hiding latency and minimizing bandwidths are important.

Flexible software architectures are required to frame all these techniques. I will mostly focus on this topic in the remainder of my talk. **[NEXT SLIDE]**



Outline

Amira

Examples of remote visualization

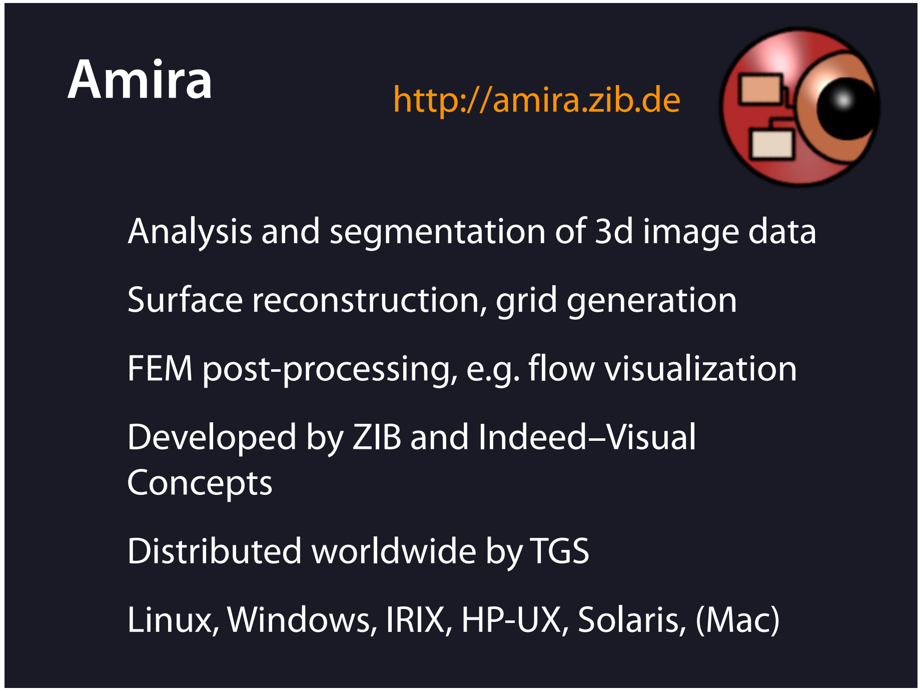
Discussion

[cont] The visualization system developed at Zuse Institute Berlin is Amira. After giving a brief overview of amira, I’ll present some recent work we did on remote visualization followed by a discussion.

[NEXT SLIDE] I’ll start with a brief coverage of Amira

Amira

[cont **]** I’ll start with a brief coverage of Amira
[NEXT SLIDE] Amira is a highly interactive system for visual data analysis



Amira

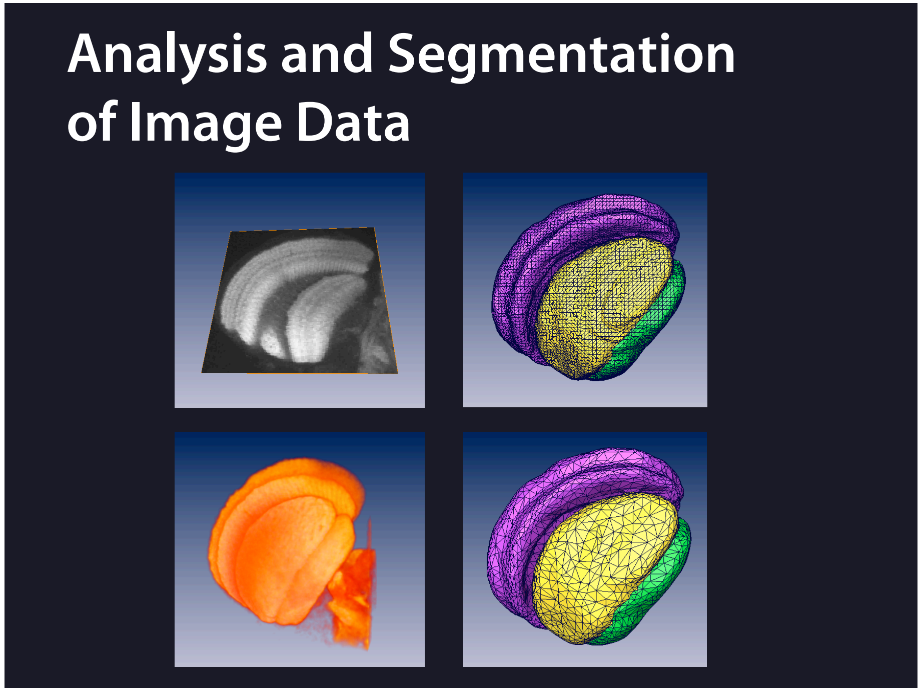
http://amira.zib.de

- Analysis and segmentation of 3d image data
- Surface reconstruction, grid generation
- FEM post-processing, e.g. flow visualization
- Developed by ZIB and Indeed–Visual Concepts
- Distributed worldwide by TGS
- Linux, Windows, IRIX, HP-UX, Solaris, (Mac)

[cont] Amira is a highly interactive system for visual data analysis. It is developed by Zuse Institute Berlin and the spin off company Indeed. A commercial version is distributed worldwide by TGS. All major platforms are supported. A Mac version will be available in the near future.

I'll not be able to go into many details of the available features but I'd like to rush through some of the application areas. You should stop me if you'd like to know more about a specific issue.

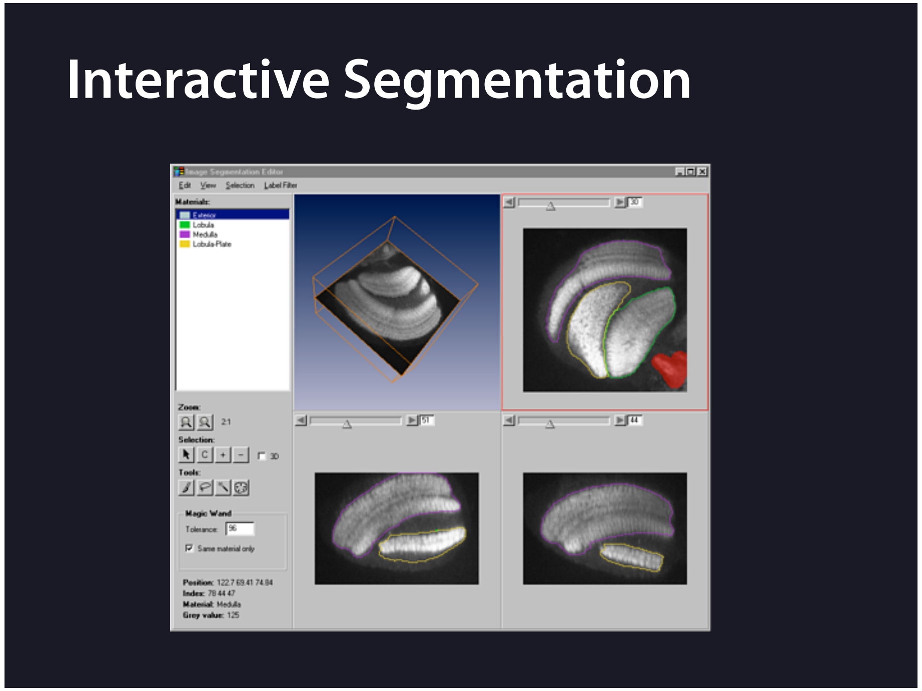
[NEXT SLIDE]



Analysis and Segmentation of Image Data

[cont] Analysis and Segmentation of Image Data is one major application area.
Displayed here is part of a bee brain. Image data were acquired using confocal microscopy. The images on the left show slices and a volume rendering—On the right surfaces of anatomical structures are rendered.

[NEXT SLIDE] such structures can be extracted using semi automatic tools



Interactive Segmentation

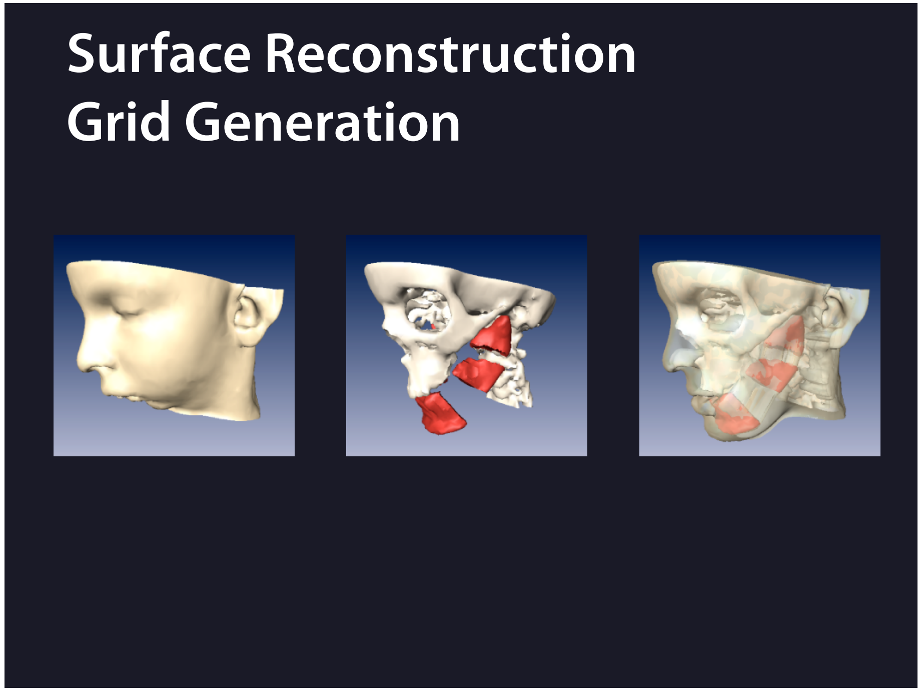
Tolerance: 96

☑ Same material only

Position: 122.7 69.41 74.84
Index: 78 44 47
Material: Medulla
Grey value: 125

[cont] such structures can be extracted using semi automatic tools. Amira is not only a visualization system but also provides many editing tools. This slide shows the GUI of a segmentation editor. It allows to ‘draw’ in the images to mark anatomical structures.

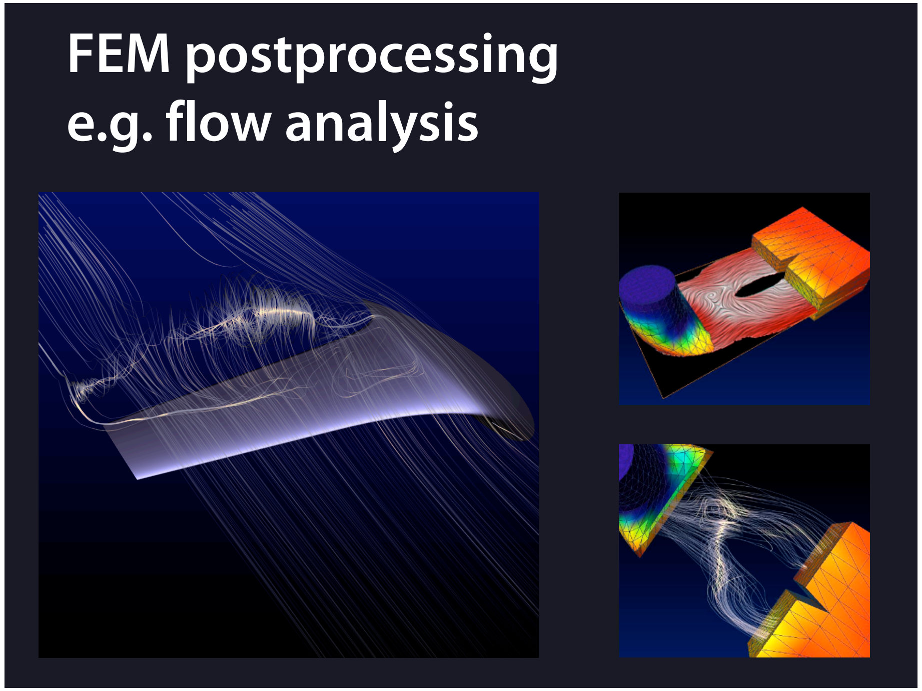
[NEXT SLIDE] In a next step, segmentations are used to generate surfaces and grids for finite element analysis.



Surface Reconstruction
Grid Generation

[cont] In a next step, segmentations are used to generate surfaces and grids for finite element analysis. This slide shows an example of a osteotomy planning system. After segmenting the bone and soft tissue in CT scans, surgeons can plan cuts and movement of bones—displayed in the center. A finite element grid is generated which allows to simulate the soft tissue after a rearrangement of bones—displayed at the right.

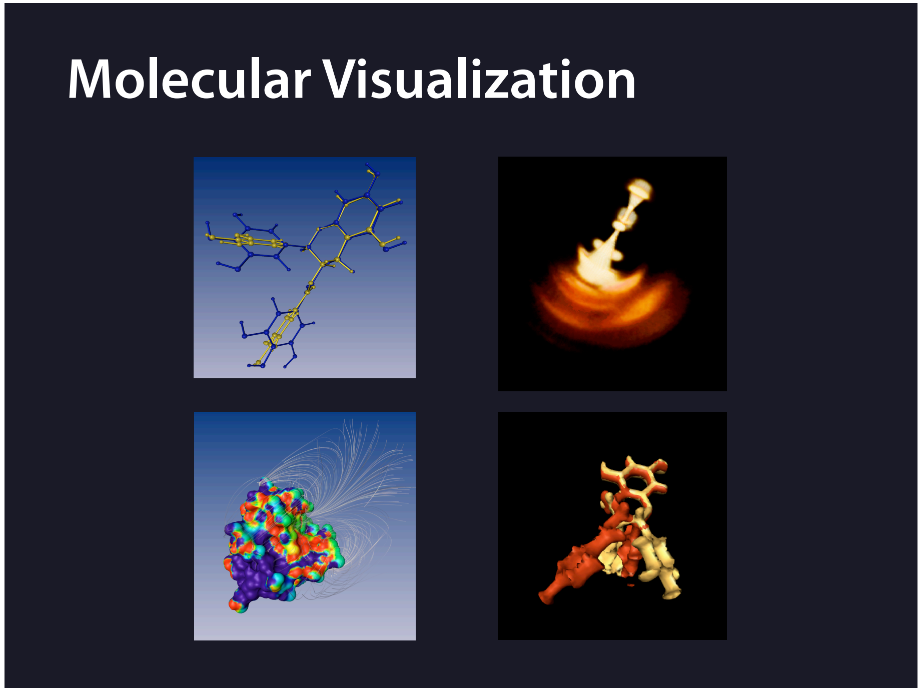
[NEXT SLIDE]



FEM postprocessing
e.g. flow analysis

[cont] Visualization of Finite Element calculations is another application field. Advanced visualization techniques like illuminated stream lines and line integral convolution are available.

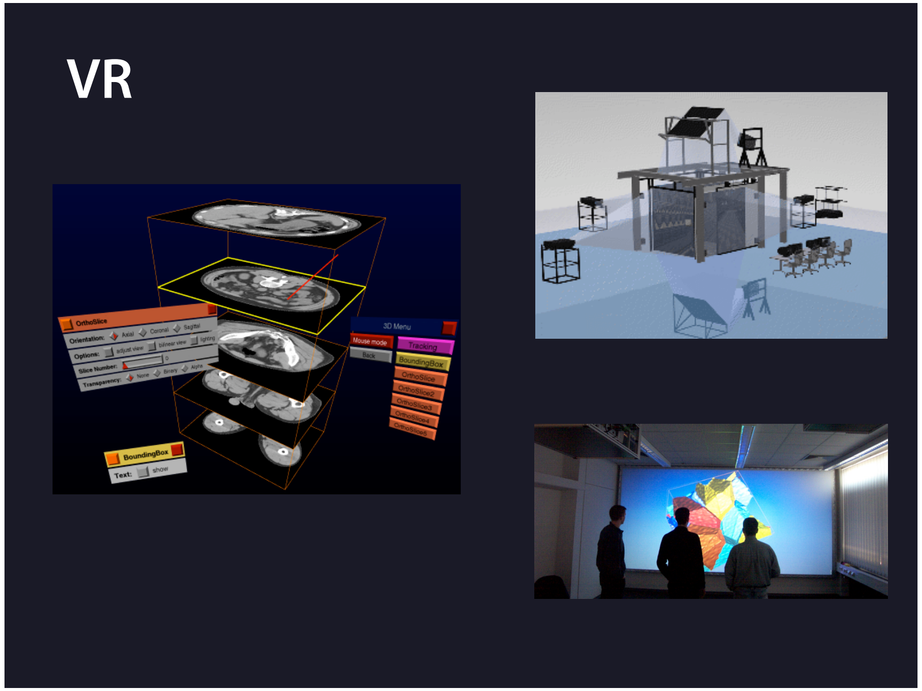
[NEXT SLIDE] A very specific field where a lot of tools are available is molecular visualization



Molecular Visualization

[cont] A very specific field where a lot of tools are available is molecular visualization with a focus on analyzing the dynamic structure of molecules. I don’t go into details.

[NEXT SLIDE]



VR

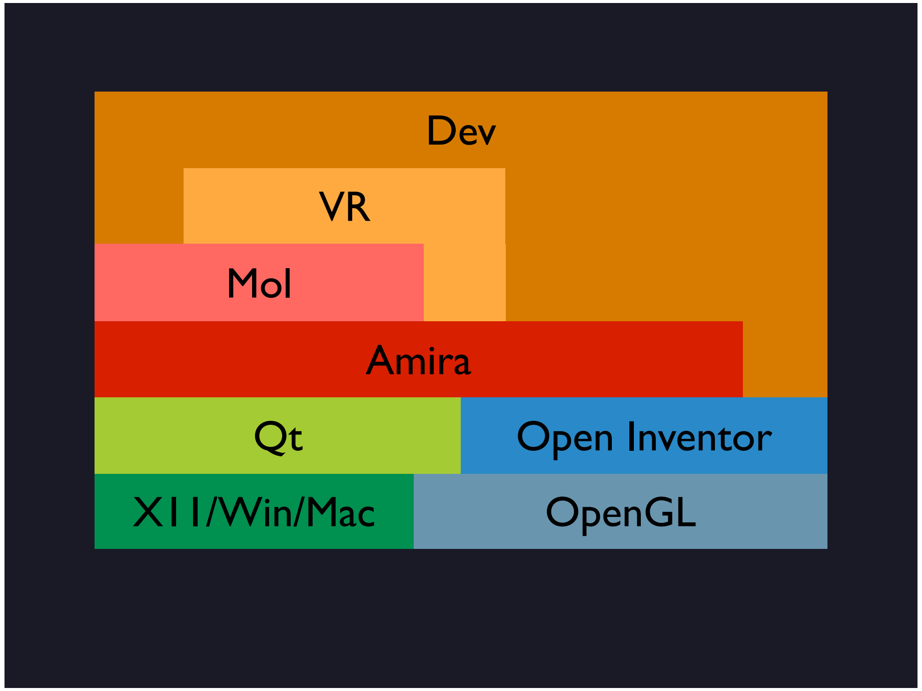
[cont] Amira is able to drive immersive environments like a cave and provides a 3d version of its user interface for basic interaction using tracked devices.

This ends my brief coverage of application fields. —— **[NEXT SLIDE]** I’d continue with the internals of Amira, which will me bring back to the problem of remote visualization.

Software Architecture

[cont] I'd continue with the internals of Amira, which will me finally bring back to the problem of remote visualization.

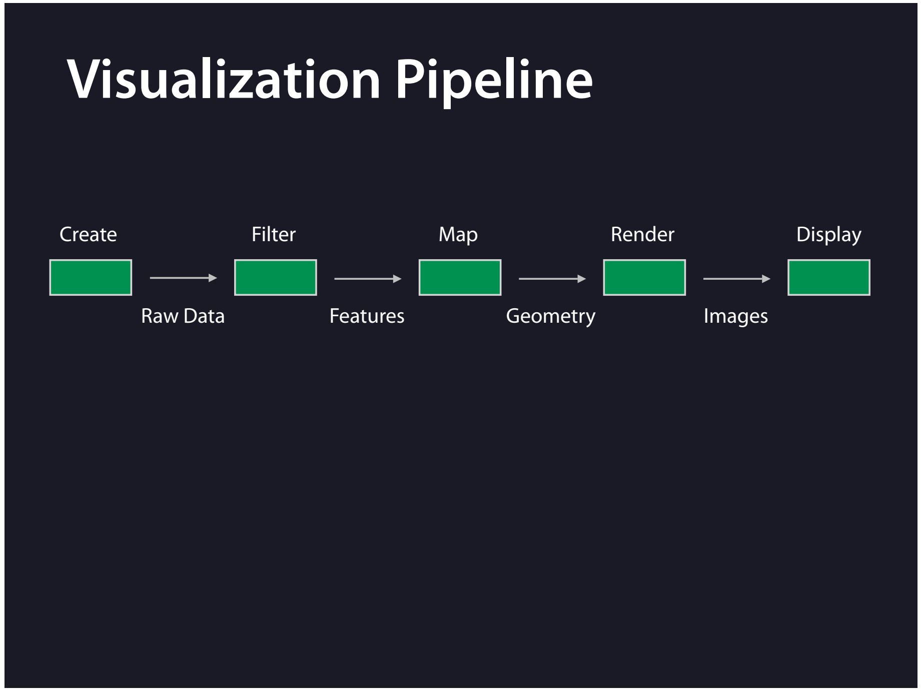
[NEXT SLIDE]



[cont] This diagram shows a high level view of the components of Amira. It is build on top of Open Inventor, which is a C++ scenegraph based on OpenGL. The Grahpical User Interface uses Qt, to abstract from the underlaying operating system.

AmiraMol, the molecular extension, and amiraVR for immersive environments sit on top of amira. A developer version provides access to Amira and OpenInventor.

[NEXT SLIDE] To provide another view on Amira, I'll first briefly explain the conceptual model of the visualization pipeline.



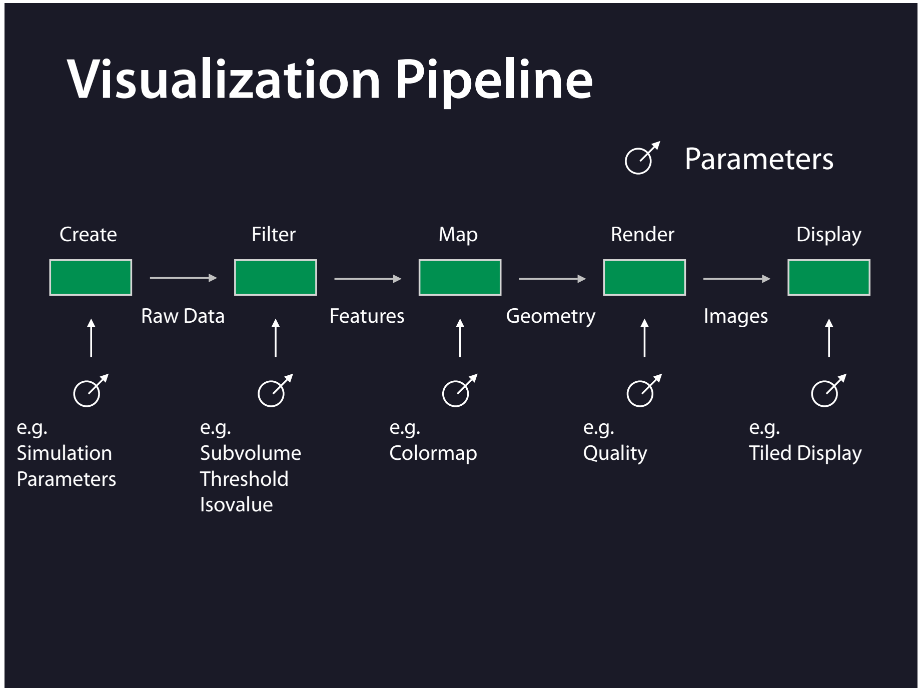
Visualization Pipeline

Create → Raw Data → Filter → Features → Map → Geometry → Render → Images → Display

[cont] To provide another view on Amira, I’ll first briefly explain the conceptual model of the visualization pipeline.

It describes the process of generating visual representations of data. From left to right:
data are created and passed as raw data to a stage which filters them to extract features. These are then mapped to some geometry or texture which is rendered to generate an image which is displayed on a screen.

[NEXT SLIDE] Every stage of the pipeline can be controlled by parameters.



Visualization Pipeline

♂ Parameters

Create → Raw Data → Filter → Features → Map → Geometry → Render → Images → Display

Create: e.g. Simulation Parameters

Filter: e.g. Subvolume Threshold Isovalue

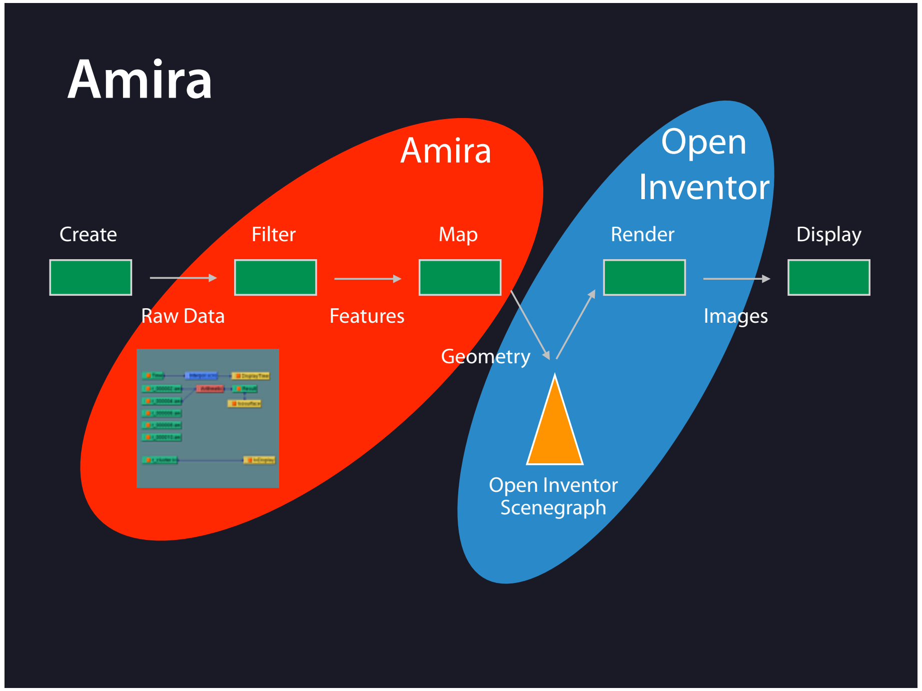
Map: e.g. Colormap

Render: e.g. Quality

Display: e.g. Tiled Display

[cont] Every stage of the pipeline can be controlled by parameters. Examples are shown here. For example displaying an isosurface: A threshold filters voxels intersected by the surface which are than mapped to triangles and rendered.

This pipeline is a concept of the process. It is not necessarily represented as a single pipeline in software.
[NEXT SLIDE] Here is another conceptual diagram of Amira.



Amira

Amira

Open Inventor

Create

Filter

Map

Render

Display

Raw Data

Features

Images

Geometry

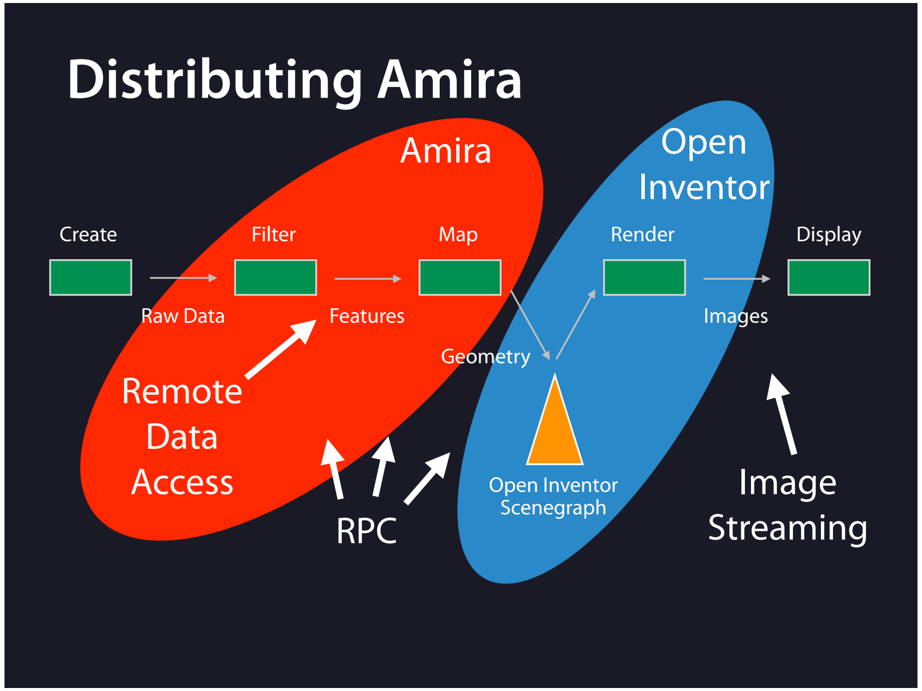
Open Inventor Scenegraph

[cont] Here is another conceptual diagram of Amira.

Data creation, e.g. running simulations, is outside the scope of amira. Amira deals mainly with filtering data and generating geometrical representations which are stored as an OpenInventor Scenegraph. Open Inventor is then used for rendering.

Currently Amira runs as a single process on a single workstation requiring a capable graphics card for rendering.

To provide remote visualization, the pipeline could in principle be cut and distributed at any place. **[NEXT SLIDE]** We did some work on image streaming, remote data access and started to look into remote procedure calls.



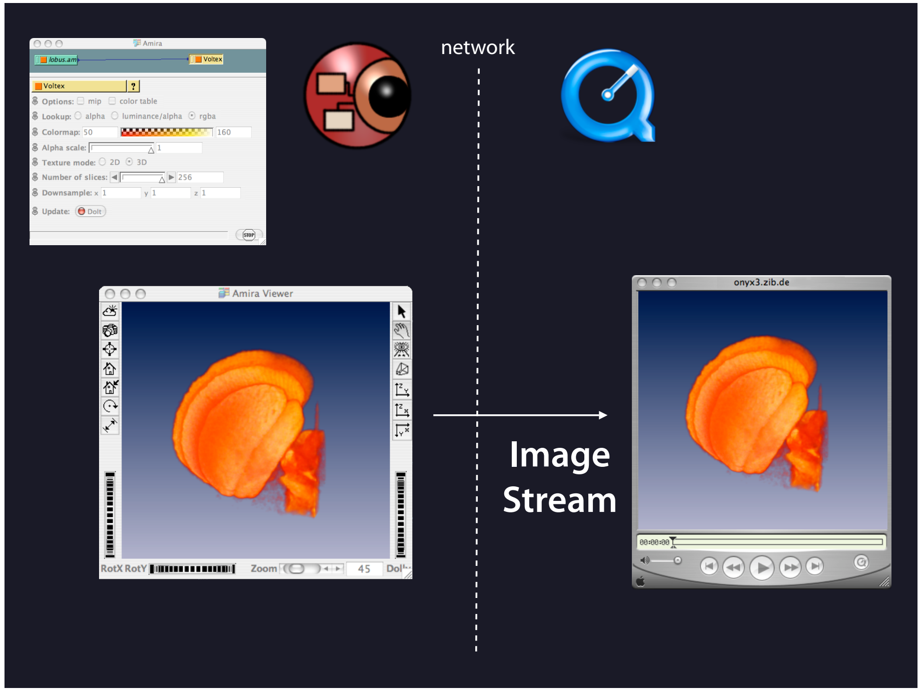
[cont] We did some work on image streaming, remote data access and started to look into remote procedure calls. I'll present these three topics now.

[NEXT SLIDE] Image streaming is a very simple, yet capable approach.

Image Streaming

[cont] Image streaming is a very simple, yet capable approach. Remote desktop solutions like VNC don't even require to modify existing software. VNC is supported out-of-the box by amira.

[NEXT SLIDE] Another solution is to directly stream images from amira which can be viewed using a standard viewer, like e.g. Quicktime or MPlayer.



Amira

lobus.am → Voltex

Voltex

Options: □ mip □ color table

Lookup: ○ alpha ○ luminance/alpha ● rgba

Colormap: 50 160

Alpha scale: 1

Texture mode: ○ 2D ● 3D

Number of slices: 256

Downsample: x 1 y 1 z 1

Update: DoIt

[cont] Another solution is to directly stream images from amira which can be viewed using a standard viewer, like e.g. Quicktime or MPlayer.

Mainly Andrei Hutanu built this solution into Amira.

[NEXT SLIDE] It uses the Real Time Streaming Protocol which provides multicast and efficient codecs.

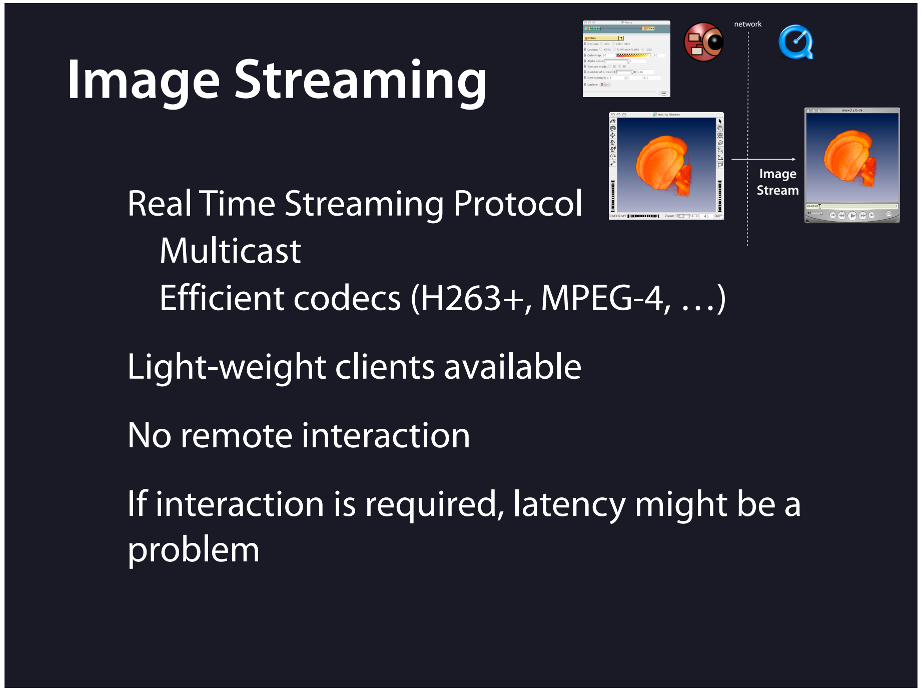


Image Streaming

- Real Time Streaming Protocol
  - Multicast
  - Efficient codecs (H263+, MPEG-4, …)
- Light-weight clients available
- No remote interaction
- If interaction is required, latency might be a problem

[cont] It uses the Real Time Streaming Protocol which provides multicast and efficient codecs. This allows efficient utilization of network resources. Light-weight clients are available and normally already installed. One restriction of this solution is the missing remote interaction. Remote desktop solutions as mentioned before, provide remote interaction but might suffer from network latency problems.

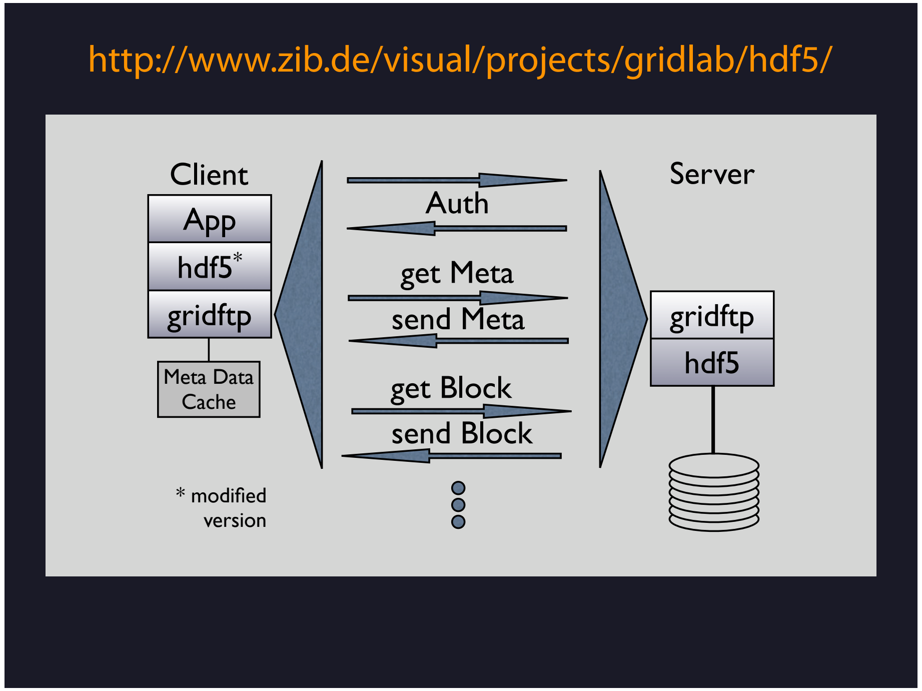
[NEXT SLIDE] In the next approach filtered data from the remote side is transfered to the local client which then generated geometry and renders images.

Remote Data Access Using HDF5

[cont] In the next approach filtered data from the remote side is transfered to the local client which then generated geometry and renders images.

We did some work on using HDF5 for the remote access. HDF5 has a virtual file driver layer which allows to implement custom I/O. When using this for remote access we hit some latency problems which we solved to some extent by modifying the HDF5 internals.

[NEXT SLIDE] The overall architecture of our solution is displayed here.

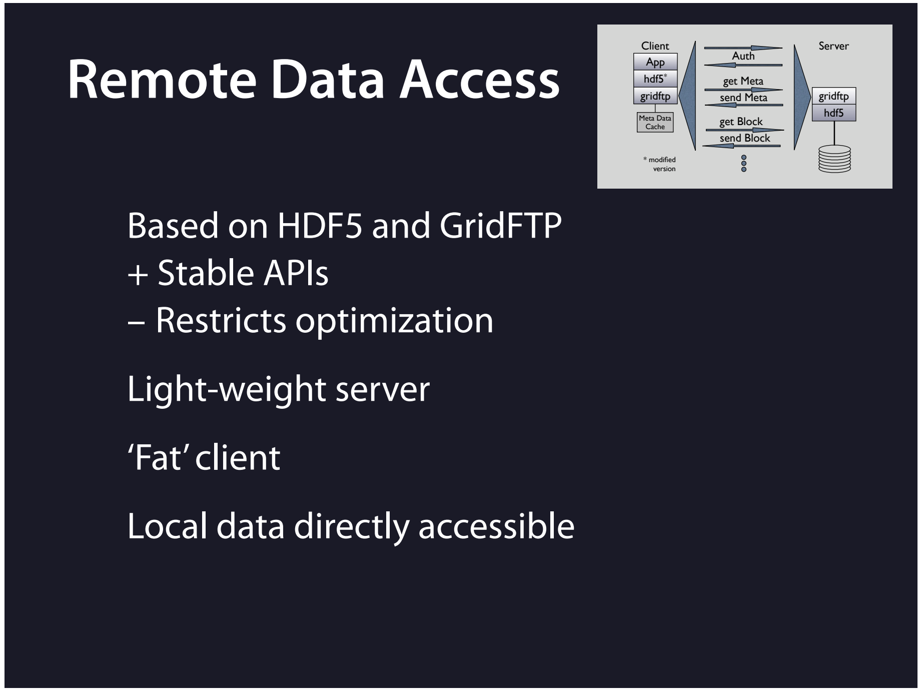


http://www.zib.de/visual/projects/gridlab/hdf5/

[cont] The overall architecture of our solution is displayed here. On the client side, the application uses HDF5 with modified internals. HDF5 provides an interface to access subblocks of data sets. We also generate lower resolution versions of the data in a preprocessing step which are used for fast previews. GridFTP is used to implement the network I/O. It communicates with a GridFTP server which uses HDF5 to access the data file. From the application’s view the interface to the data is the same for local or remote access.

To optimize a remote file open we send all HDF5 meta data in one block after authentication and store them in a meta data cache locally. Later, only sub block requests are sent to the remote side.

[NEXT SLIDE] This solution used a light weight server and a ‘fat’ client.



Remote Data Access

Client
App
hdf5*
gridftp
Meta Data Cache
* modified version
Auth
get Meta
send Meta
get Block
send Block
Server
gridftp
hdf5

Based on HDF5 and GridFTP
+ Stable APIs
– Restricts optimization

Light-weight server

'Fat' client

Local data directly accessible

[cont] This solution used a light weight server and a 'fat' client. A complete amira is running locally. Local data is thus directly available.

Interactivity is achieved by local graphics hardware. Depending on the size of the region of interest and the required resolution of the data, data transfer might be a bottleneck.

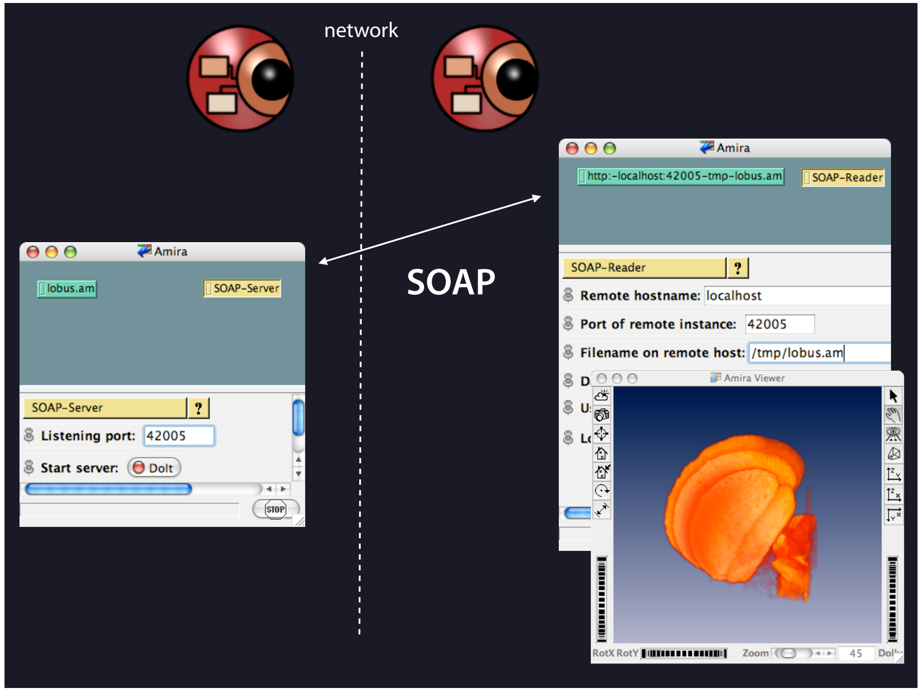
These two very different solutions, image streaming and remote access, already show that a flexible distribution of the visualization system could be useful to provide optimal solutions to specific problems.

[NEXT SLIDE] Currently we are looking into Remote Procedure Calls to implement this idea.

RPC

[cont] Currently we are looking into Remote Procedure Calls to implement this idea.

[NEXT SLIDE] We use SOAP for the remote calls.



SOAP-Server

Listening port: 42005

Start server: DoIt

SOAP-Reader

Remote hostname: localhost

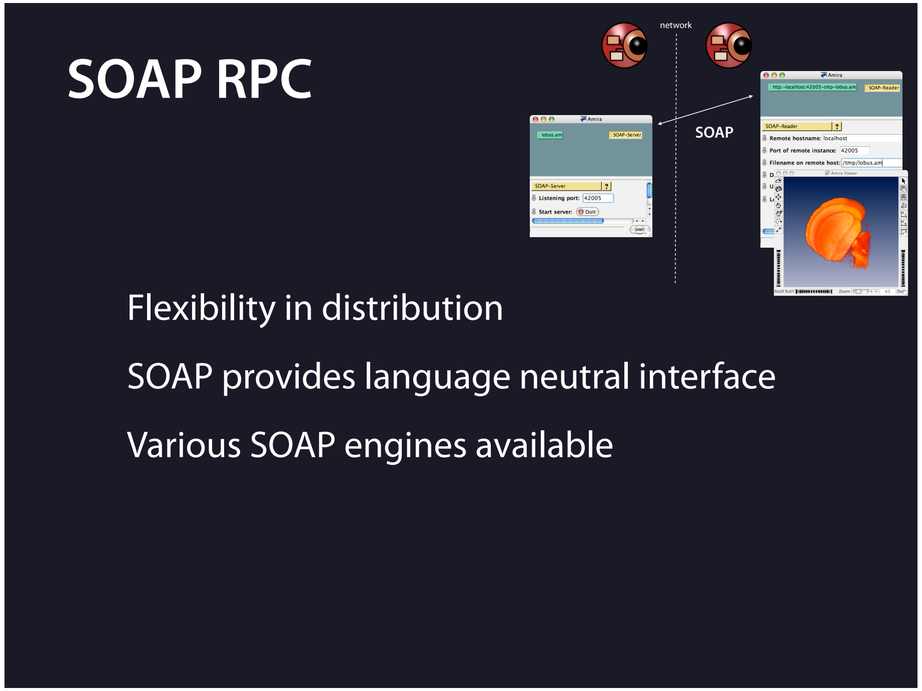
Port of remote instance: 42005

Filename on remote host: /tmp/lobus.am

[cont] We use SOAP for the remote calls. It is a xml based standardized Protocol. We run amira on both sided. But in principle any client or server could be used.

Our first examples continue the work on remote data access. An Amira object representing volumetric image data is accessed via SOAP. Geometry generation and rendering runs on the client machine.

[NEXT SLIDE] The architecture would in principle support any distribution of the application.



SOAP RPC

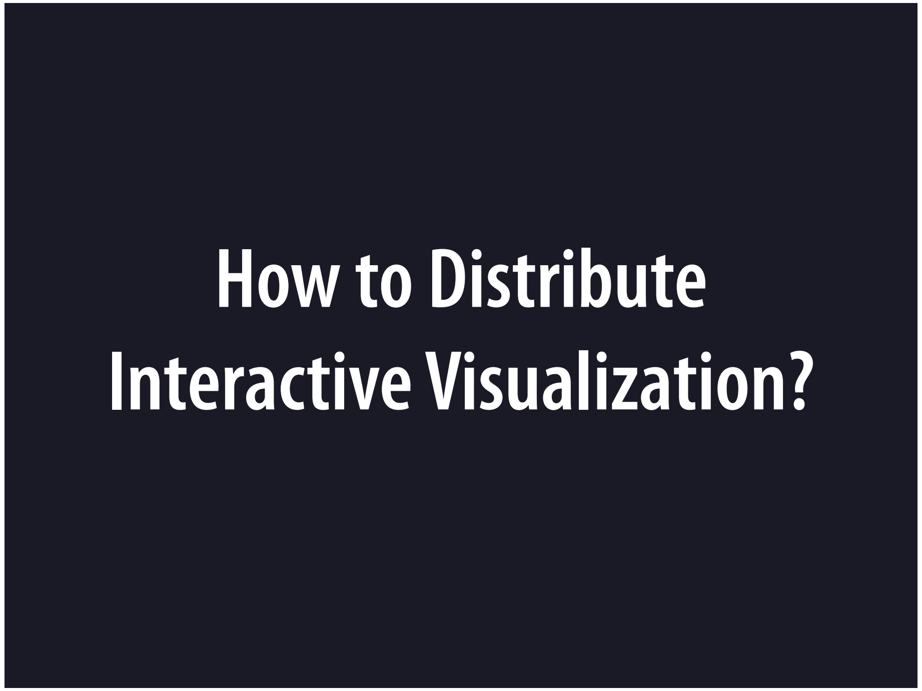
SOAP-Server
Listening port: 42005
Start server: DoIt

SOAP-Reader
Remote hostname: localhost
Port of remote instance: 42005
Filename on remote host: /tmp/lobus.am

- Flexibility in distribution
- SOAP provides language neutral interface
- Various SOAP engines available

[cont] The architecture would in principle support any distribution of the application. This flexibility could be used to provide optimal solutions for specific problems. SOAP provides a language neutral interface and various SOAP engines are available. Interoperability with other systems could thus be achieved.

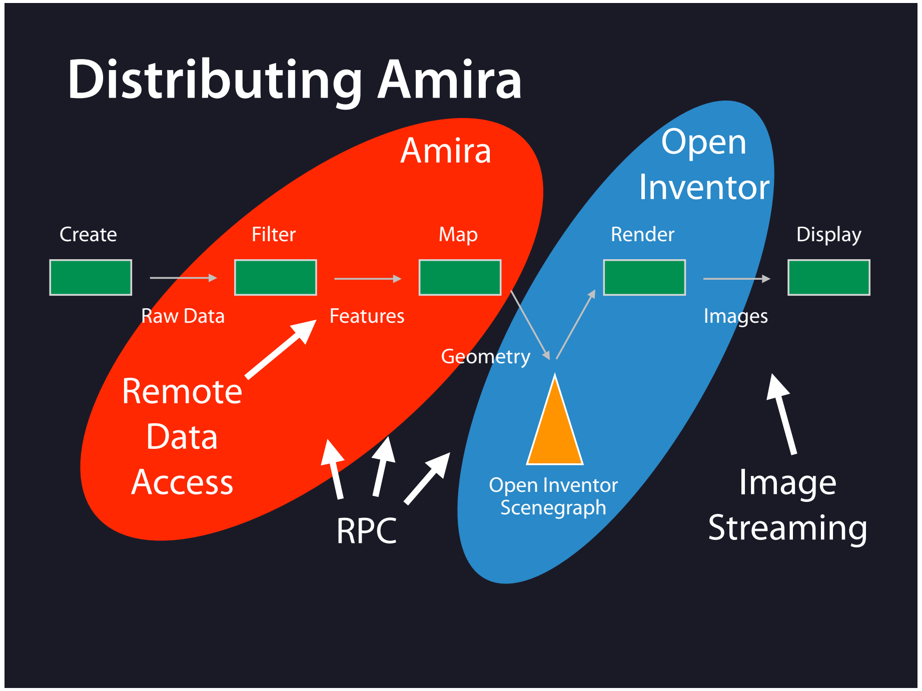
[NEXT SLIDE] This brings us back to the earlier question: “How to Distribute Interactive Visualization?”



How to Distribute Interactive Visualization?

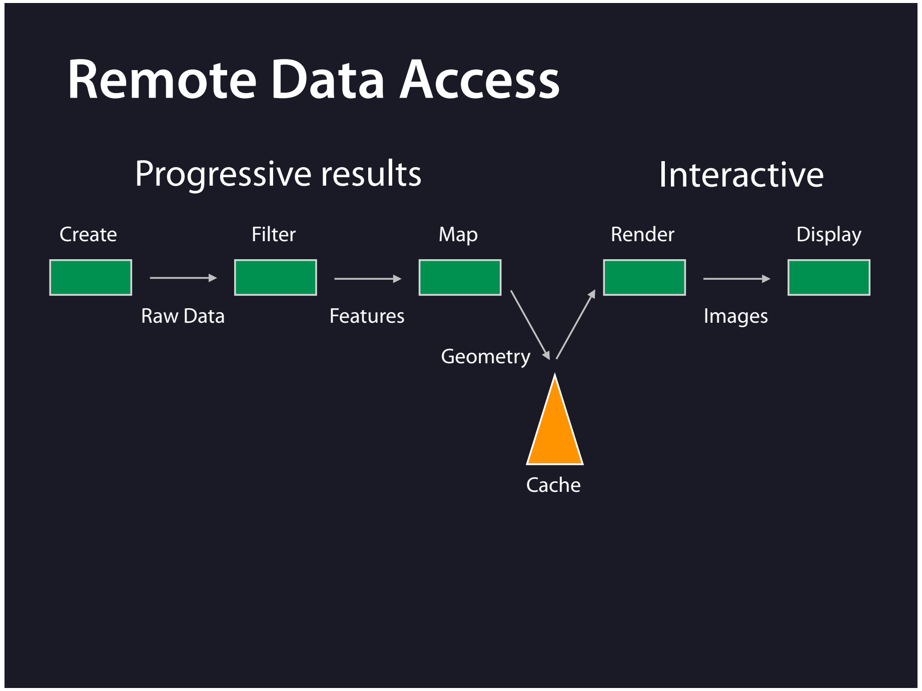
[cont] This brings us back to the earlier question: “How to Distribute Interactive Visualization?”

[NEXT SLIDE] I briefly presented some specific solutions and first work towards a more general software architecture.



[cont **]** I briefly presented some specific solutions and first work towards a more general software architecture. Image Streaming and Remote Data Access are specific solutions.

[NEXT SLIDE] More abstractly, our remote data access solution can be described as a system which splits into a local interactive part and a part progressively retrieving results.

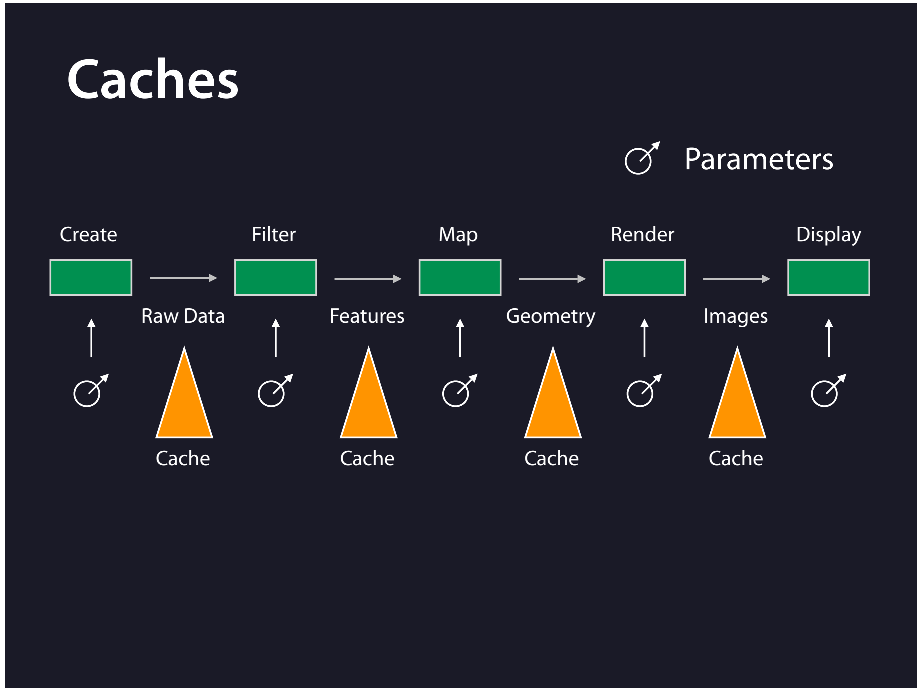


Remote Data Access

[cont **]** More abstractly, our remote data access can be described as a system which splits into a local interactive part and a part progressively retrieving results. In between sits the OpenInventor scenegraph which serves as a geometry cache.

Rhis structure allowed us to deploy an interactive system. We think it is very suitable to fulfill this requirement.

[NEXT SLIDE] A general analysis could analyze systems based on a pipeline with caches—and try to provide interactive systems in this setting.



Caches

♂ Parameters

Create

Raw Data

Filter

Features

Map

Geometry

Render

Images

Display

Cache

Cache

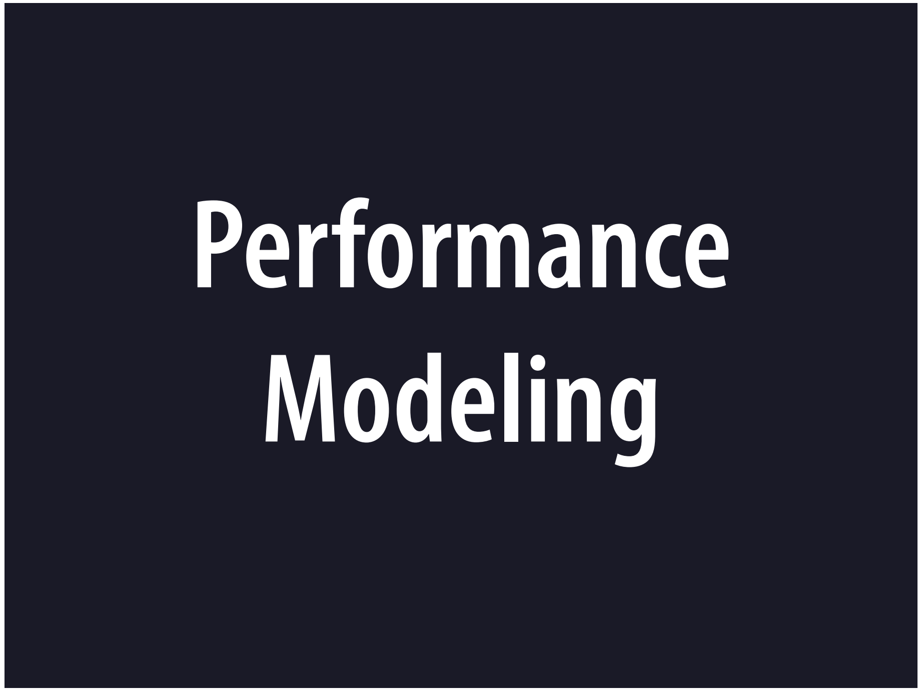
Cache

Cache

[cont] A general analysis could analyze systems based on a pipeline with caches—and try to provide interactive systems in this setting.

We did not do any theoretical work on this but think it could be a fruitful field to look into.

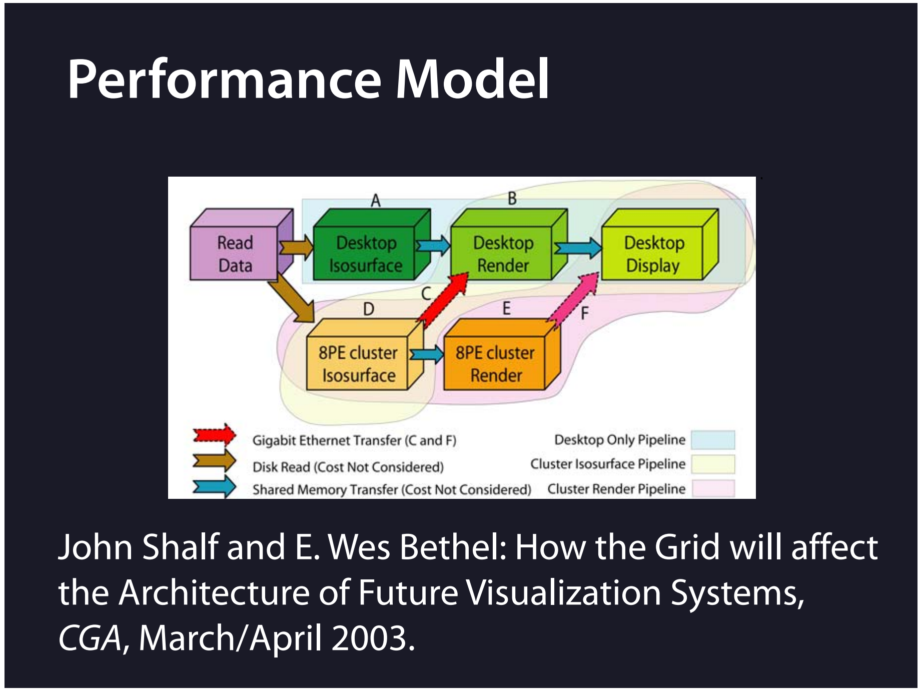
[NEXT SLIDE] Some basic work has been done on a general analysis of this problem.



Performance Modeling

[cont] Some basic work has been done on a general analysis of this problem.

[NEXT SLIDE] I’d like to show a simple illustrative example by John Shalf and Wes Bethel.



Performance Model

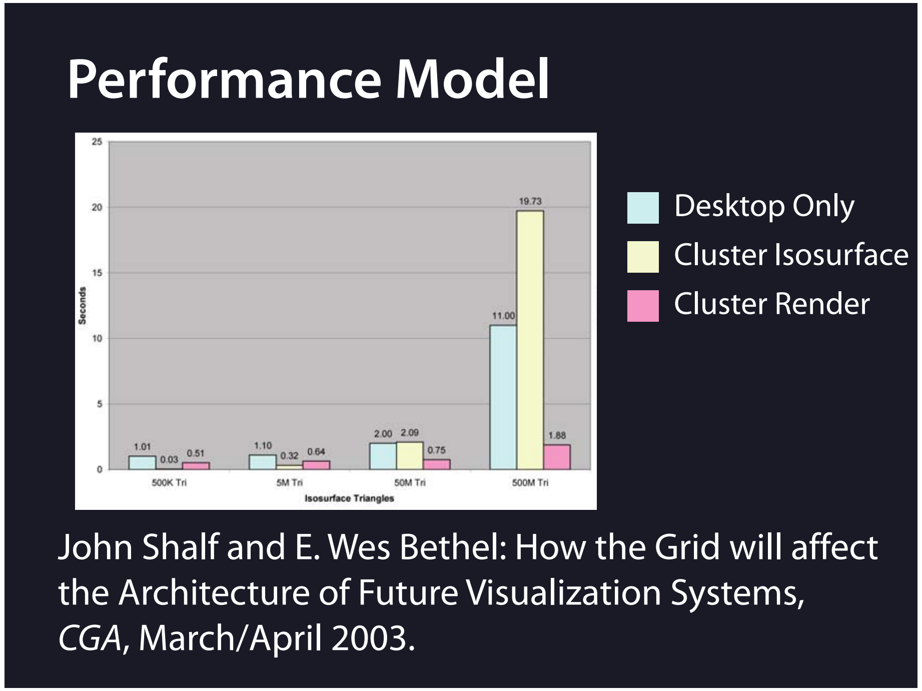
John Shalf and E. Wes Bethel: How the Grid will affect the Architecture of Future Visualization Systems, *CGA*, March/April 2003.

[cont] I’d like to show a simple illustrative example by John Shalf and Wes Bethel. They analyzed how a system for remote Isosurface rendering performs if the Isovalue is varied.

They use either:
- remote data access (Pipeline A - B)
- remote isosurface extraction and geometry transfer to the local machine (Pipeline D-C-B)
- or remote rendering with image streaming (Pipeline D-E-F)

Depending on the threshold parameter—one single parameter !!!—the perfomances of the 3 pipelines are quite different.

[NEXT SLIDE] For small triangle counts, extracting the isosurface remotely and transferring it to the local machine is most efficient (5M Tri). For higher triangle



Performance Model

John Shalf and E. Wes Bethel: How the Grid will affect the Architecture of Future Visualization Systems, *CGA*, March/April 2003.

[cont] For small triangle counts, extracting the isosurface remotely and transferring it to the local machine is most efficient (5M Tri). For higher triangle counts, transferring the rendered images is more efficient.

The measured time is to generate a single image by processing the full pipeline completely. This simple example already illustrates the dramatic effects which should be considered when thinking about an automatic distribution of a visualization pipeline, e.g. in a grid environment.

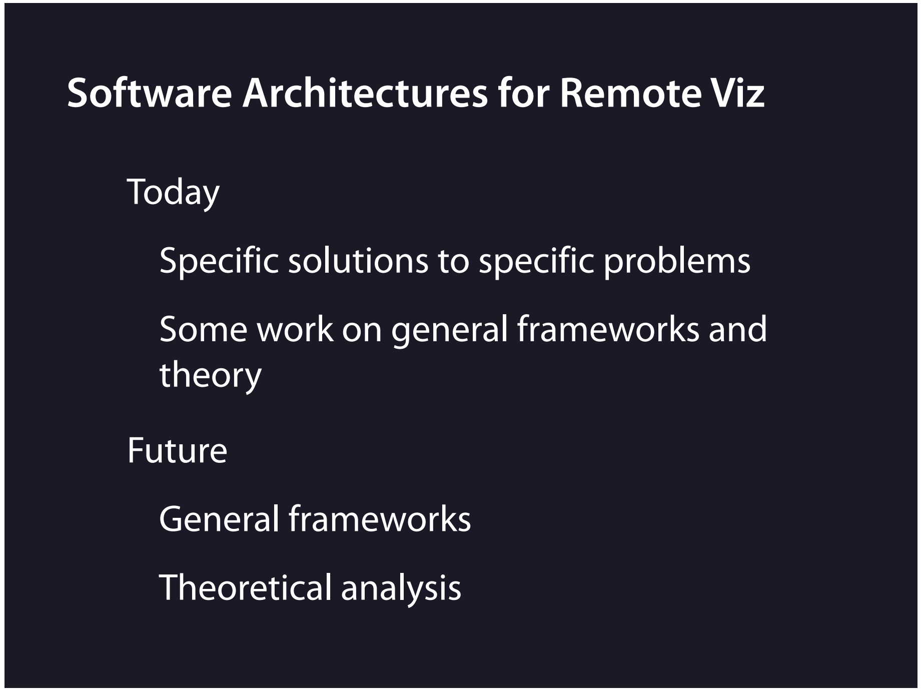
[NEXT SLIDE] To Summarize

Shalf_IEEECGA-VisViewpointsMay2003-LBNL-51723.pdf

Summary

[cont **]** To Summarize

[NEXT SLIDE]



Software Architectures for Remote Viz

- Today
  - Specific solutions to specific problems
  - Some work on general frameworks and theory
- Future
  - General frameworks
  - Theoretical analysis

[cont] I presented some work on software architectures for remote visualization. Specific solutions to specific problems are available today. Some work on general frameworks and theory is done.

But a lot more work is ahead, to provide general purpose interactive visual data analysis on large data sets. General frameworks will be required. But perhaps more important, a theoretical analysis on how to built them is needed.

That’s all I have, Thank you.

[redacted]