

This talk presents two examples of work at our lab used to analyze high resolution image data.



The applications are somewhat different. The first one, exploration of micro-CT scans, is designed to be highly interactive. The second one, centerline extraction from images, is run as a batch job and only the results are to be viewed interactively. But first some potential design goals when dealing with massive data are summarized.



The next few slides will give brief overviews over some important ideas to keep in mind when dealing with massive data.

In general it is not possible to load all the data into the main memory. So out-ofcore storage should be always assumed. Even if they fit into main memory, it would not be a good strategy to load them completely before starting to process. Just because loading takes too long.



First goal: Process data of any size with optimal performance. The general goal is to redesign the algorithms to run with minimal performance loss due to out-of-core data storage. The first step is to understand the data access patterns. Then, when possible, to redesign the algorithm maximizing data access locality and to devise a data storage layout consistent with the access pattern to amortize the cost of I/O operations over several memory access operations.

Algorithms can be analyzed using various models, depicted here is: 1) Parallel disk model \cite{Vitter:ACMCompSurv-01-33-209}. Try do model main disk, main memory, CPU hierarchy. Subject of analysis: Number of block transfers, used disk space, CPU work.

2) Ideal-cache model. More recently, cache-oblivious algorithms \cite{Frigo:FOCS -99} opened a new way of thinking about these problems. The goal there is to optimize algorithms for any kind of memory hierarchy containing caches, without

Process data of any size with optimal performance	
(a) $0 - 1 - 2 - 3 - 4 - 5 - 6 - 7$ (b) $0 - 8 - 16 - 24 - 32 - 40 - 48 - 56 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 1 - 9 - 17 - 25 - 33 - 41 - 49 - 57 - 16 - 17 - 18 - 19 - 20 - 22 - 23 - 2 - 10 - 18 - 62 - 44 - 25 - 56 - 58 - 24 - 25 - 26 - 27 - 28 - 93 - 03 - 1 - 3 - 11 - 19 - 17 - 35 - 43 - 51 - 59 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 99 - 4 - 12 - 20 - 28 - 36 - 64 - 45 - 60 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 47 - 5 - 13 - 21 - 29 - 37 - 45 - 53 - 61 - 48 - 49 - 50 - 51 - 52 - 53 - 54 - 55 - 6 - 14 - 22 - 30 - 38 - 46 - 54 - 62 - 56 - 57 - 58 - 59 - 60 - 61 - 62 - 63 - 7 - 15 - 23 - 31 - 39 - 47 - 55 - 63$	15- 20- 15- 25- 30-25-
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	

A very simple example in which the change of data layout may lead to optimal low number of cache misses for a matrix multiplication is depicted here.

Top: Left row major, Right column major

Bottom: Left blocked, Right bit interleaved

When using a divide and conquer scheme for matrix multiplication, the bit interleaved case is the only one which run optimal on every cache size without tuning any parameter.

Without going into further details, it is worth to keep the basic idea always in mind when dealing with massive data.

If you focus more on the real world, your major goal might be to provide an responsive application to the user. The next slide illustrates this.



Depicted here is a schematic view of an interactive system allowing exploration of large datasets. The user needs to control each step of the visualization pipeline at any time. In general, it cannot be foreseen what the user asks to exactly visualize next. This fact makes it impossible to predict the data access pattern in full depth. Also data management is becoming a challenging task when dealing with huge amounts of data: data consistency, storage capacity, backup facilities and security become problems on their own. Therefore data is typically stored centrally and has to be accessed remotely.

6

The visualization pipeline can then be distributed between server and client in various ways. Depending on the bandwidth and latency of the network and the hardware different scenarios might be optimal.

Some other problems might also influence decisions when building real world software are listed non the next slide.



The original data is the base of all scientific results. Keeping an untainted version available for the future should be a major goal and is requested by scientists. To achieve this, interoperability and language neutral interfaces to the data are important. Real world applications might therefore refuse to use the most optimized data structure if it's not clear how to access them in the future. Using well established standards might be a better choice. Building new standards should be a goal.



The next slides present two applications, which made different decisions about the design focus.

One is oriented towards interactive exploration of the original data.

The other one uses feature extraction in a preprocessing step to generate a geometry which might then be displayed at interactive framerates.



The first application is an interactive system to explor image data of human vertebral bodies.

Full human vertebral bodies are scanned using a micro CT with a voxelsize of about 40µ resulting in an overall dataset size of about 8GB. We expect to acquire about 50 to 70 scans during a bone research project. The overall amount of data which has to be stored is about 500 GB – 1 TB. Remote access to the datasets is possible for all project partners. The medical researchers want to interactively explore the datasets to compare them with results from other imaging devices, mainly with standard CT scans and histological slices. A more detailed example is shown below.



During design of a new software interface to large image data one requirement was to reuse old code already present in our visualization software amira. We stacked some interfaces on top of each other.

Visualization algorithms, access the data through the most abstract layer, the Regular Grid Data interface. Legacy code can be accessed through this interface, too. Massive data have to be available at various resolutions. Right now low resolution previews are generated and stored as additional data. In principle any file format which allows to store a Collection of Arrays could be used to achieve this. We chose HDF5 as our format, which internally provides another abstraction layer. It has a clearly defined interface to the storage system (local hard disk, network accessed data, ...). We use GridFTP as our protocol when accessing data remotely. Internally HDF5 also provide various data layout schemes, e.g. chunked layout.



To keep an application responsive at any time, providing asynchronous access to the data is critical.

A Queue which manages requests and available data might be one way to implement this.

The slide shows some of the most important messages passed between these objects. The Queue uses a background thread to fetch the data, because hdf5 itself only provides a synchronous API, which is in my opinion a major deficiency. The GUI is notified whenever new data is available. It interrupts rendering and user interaction, retrieves the data, builds a new geometry and continues interactive visualization. At any point in time, the user might change the requested data, e.g. by moving a slice, and the application will immediately adapt for this change.



To enable remote access we use GridFTP, integrated into an enhanced version of hdf5. Our modifications allow to send high level block requests via the network instead of low level seek/read calls. Software is available at our website.

GridFTP is used for authorization, for getting the meta data (size of the data, bounding box, textual annotations by the users, ...) and for retrieving blocks of image data. On the client side (right) GridFTP serves as a low level driver integrated into HDF5. The GridFTP server (left) is implemented using HDF5. This allows to access the same HDF5 files either locally or remotely.



For visualization we mainly use volume rendering and slicing. Both of them retrieve data progressively

13

Our volume renderer uses an octree which is populated with high resolution textures only around a focus point, which can be selected by the user. The user can also set memory limits. These textures are filled in the background while the actual available data are already rendered.

In the next slides an actual case where this system was used to investigate the data is presented.



As mentioned before, scientist are interested in comparing image data from different sources. In this case data from a clinical CT scanner and the data acquired using a micro-CT.

14

The location and geometry of the conventional CT slice had been approximately known. Therefore it was easy to select a central slice of the right thickness (top), average it in z direction and visually compare it to a slice from the clinical CT scanner (bottom).

Top left: Select central 4mm slice Top right: 3D view of selected volume Bottom Left: CT scan Bottom Right: micro-CT averaged in z direction



You see a smaller area of the CT slice together with a volume rendering of the micro-CT data. These kind of comparison are used to improve analysis procedures of the convention CT scans.



The goal of the second project is to conduct a morphometric analysis of micro vessels in human brain tissue. Images are acquired using a confocal microscope. To cover a larger area, several overlapping images are used. The images are merged to one large 3D volume.

16

The overall datasize of this image is about 2 GB.

The quality of the image data is such, that low pass filtering followed by a threshold segmentation is sufficient to separate the vessels from the background. This binarized image is processed using a distance ordered thinning algorithm.



To do this, two steps are required:

- Calculating a distance map. We use a Chamfer map.
- Performing a thinning

We modified existing algorithms to allow block wise processing of the dataset.

See C. Fouard, G. Malandain, S. Prohaska, M. Westerhoff, F. Cassot, C. Mazel, D. Asselot, J. P. Marc-Vergnes. Skeletonization by Blocks for Large 3D Datasets: Application to Brain Microcirculation. IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Arlington, Virginia, 2004

Left: Forward scan of a Chamfermap. Scan in video order (top left to right, bottom left to right).

Top correct scan in one block.

Bottom naive block wise scan gives wrong result --> Correct by adding more scans.



At the bottom left a complete slice of human brain tissue is displayed. The area marked by the red rectangular is displayed at the top. Top right: Brain tissue overlaid with the mosaic of confocal microscopy images. Bottom right: Extracted centerlines. The whole lineset contains about 2 million points and about 70 thousand lines.



The line set from the previous slide displayed again together with a zoom view on the left.



I showed one application using feature extraction run as a preprocessing step to allow useful analysis of the data. The other focus was on an interactive system allowing to explore the original data. Especially scientists in the medical and biological field tend to really want to see the original data (often in black and white). They are very used to rely on their own interpretations.

So it is essential to provide these tools in an integrated environment and let the researchers choose what exactly to use.



Some more thoughts on the future:

Image data in the size of 100 GB – 1 TB per scan will be used on a regular basis soon. And we assume that a lot of these scans have to be managed. So if you're starting to build software for this field today, it should scale into this range. Basic data management is an issue then. It would be very useful to have a standardized solution to access these data. I'm not aware of any standardized solution yet. But I think this will be a task for the future.

Acknowledgments

- ZIB Andrei Hutanu – Remote access, GridFTP Ralf Kähler – Hierachical volume rendering Malte Westerhoff – Micro vessels
- Céline Fouard, Grégoire Malandain (INRIA Sophia Antipolis) Skeletonization
- F. Cassot, J. P. Marc-Vergnes, F. Lawers, H. Duvernoy Brain tissue specimens and images
- W. Gowin, G. Beller (Charité, Univ. Med. Berlin), M. Burkhart, B. Koller, A. Laib (Scanco Medical), J. S. Thomsen (Univ. of Aarhus) – Bone specimens and µ-CT
- European Space Agency (ESA), French government Funding
- Indeed Visual Concepts, TGS Amira, http://amira.zib.de