1 Visualization

1.1 Background

Liz && Chris will write something here.

1.2 Progress

1.3 Large-scale Volume Rendering and Manipulation of High-Resolution Data Sets

1.3.1 Data Organization

With the growing size of medical data comes new challenges in visualizing this data on commodity platforms. To this end, we have redesigned the structure of the multiresolution hierarchy generated for UVF files (the native format of ImageVis3D) [?] to allow for vastly increased scalability and flexibility. As we volume render higher resolution data, it has become increasingly apparent that the seminal technique of 'early ray termination' [?] – the idea that the 'front' of a data set forms the most relevant component of a volume rendering – applies to data resolution as well. This means that rendering the portions of the data nearest to the viewer in high resolution is considerably more important than rendering those areas which are far away in high resolution.

However, counter to this notion, prior versions of ImageVis3D's internal format required a constant resolution throughout the volume. We have redesigned this format to circumvent the limitation by using an octree-based file organization. This structure allows one to dynamically choose which resolution is most appropriate. Our next development will be to enable this structure in a more modern renderer, which adjusts to the appropriate level in the multiresolution hierarchy on not just display and data resolution, but on the progress through the volume and the current sample's relevance to the final volume rendering.

1.3.2 Improved Support for 'Flagged' Data Sets

The medical community often generates data sets which consist purely of 'flag' values. In such a data set, the value at each voxel does not describe a measurement in the physical world, but an arbitary function of that measurement. Such data commonly come from the increasingly popular image processing pipelines which are performed on acquired data to accentuate or otherwise identify relevant features. Examples are connected components labelling, or segmentations produced via a tool such CIBC's own Seg3D.

When rendering such data, it is inappropriate to interpolate between neighboring data elements. The nature of a data set of flags does not provide any meaning for partial elements. For example, if one data element is labelled "bone" and a neighboring element is labelled "flesh", the intervening space should not be considered a hybrid bone-flesh 'mixture'. Rather, the volume should be considered to have a hard boundary somewhere between the two elements.

We have implemented a new render mode specifically for these 'flag' data sets, which avoids this inappropriate interpolation. Furthermore, we have implemented a mode to detect encodings of these data sets into wider bit widths, and losslessly remap them to a smaller type. This provides a significant savings to disk space, and also gleans a performance benefit at rendering time.

1.4 Collaborative & Hybrid Visualization

Data management is fast becoming the primary issue in gaining understanding of modern day medical data. As data sizes grow, the issue of getting information in the right place – at the right time – becomes a significant challenge. Existing visualization tools do very little to aid the user in this endeavor. As a result, the modern practitioner spends a considerable amount of time configuring external software to perform mundane tasks like *downloading data from a remote collaborator* or *perform processing on a remote, capable computer*.

Recognizing this trend, we are investigating ways to enable users to work *with* their data, not *on* their data. One foray into this goal has been hybrid volume rendering based on the same rendering engine powering ImageVis3D. A set of data sets lives on a server system (or cluster of systems), which clients can query and access. When a data set is chosen, the user immediately sees a rendering of the data set on their client system. Interactions produce immediate response. Just like in ImageVis3D, responses are immediately visible, but quality is a function of time.

The visualization is produced by a renderer or set of renderers in the cloud. However, instead of shipping a sequence of static images to the client, the data stream is encoded via standard video encoding methods. In addition to making much better use of the available bandwidth and the ability to progressively refine the rendering, this allows much more sophisticated control of the renderer progress. By highlighting regions of interest, users are given full control of how we should allocate network resources, in order to best bring the most relevant data to them fastest. This system is depicted in Figure 1.

The scenario above is a typical cloud-based rendering system organization. However, recognizing the capability of client devices, we make effective use of both visualization-capable devices in such a system. Transparently to the user, a multiresolution representation of the data is being transferred to the client machine while the user is interacting with the data. An intelligent network multiplexer ensures that this happens only during idle periods, so the streaming of the cloud-based renderings is never known to the user. When some data are available, the choice of which renderer to use (local or remote) is made dynamically. A carefully tuned scheduler constantly monitors the network utilization as well as the rendering time required on both systems to inform this choice.

This system enable scientific *inquiry* to occur when scientific *hypothesis* occurs. When a user wonders, "I wonder if the new scans also exhibit a characteristic vessel enlargement leading up to tumor areas," they can immediately load up 'the new scans' and see, without having to worry (or even think about) whether those scans came from their lab or a colleagues lab on the other coast – or even another continent. Contrasted with modern mechanisms for data visualization, which may require a user to contact a colleague, wait for a response which details the location of the data, and data transfer – this can provide a real improvement to the modern practice of visualization and analysis.

1.4.1 Medical Visualization on Mobile Platforms

In last years report, we reported on the use of a mobile tool for programming deep brain stimulation parameters using mobile visualization platforms. The tool brought down the standard of care for this task from 4 ± 1.4 hours to 1.7 ± 0.8 *minutes*. We are happy to report that this result is now published work [**?**].

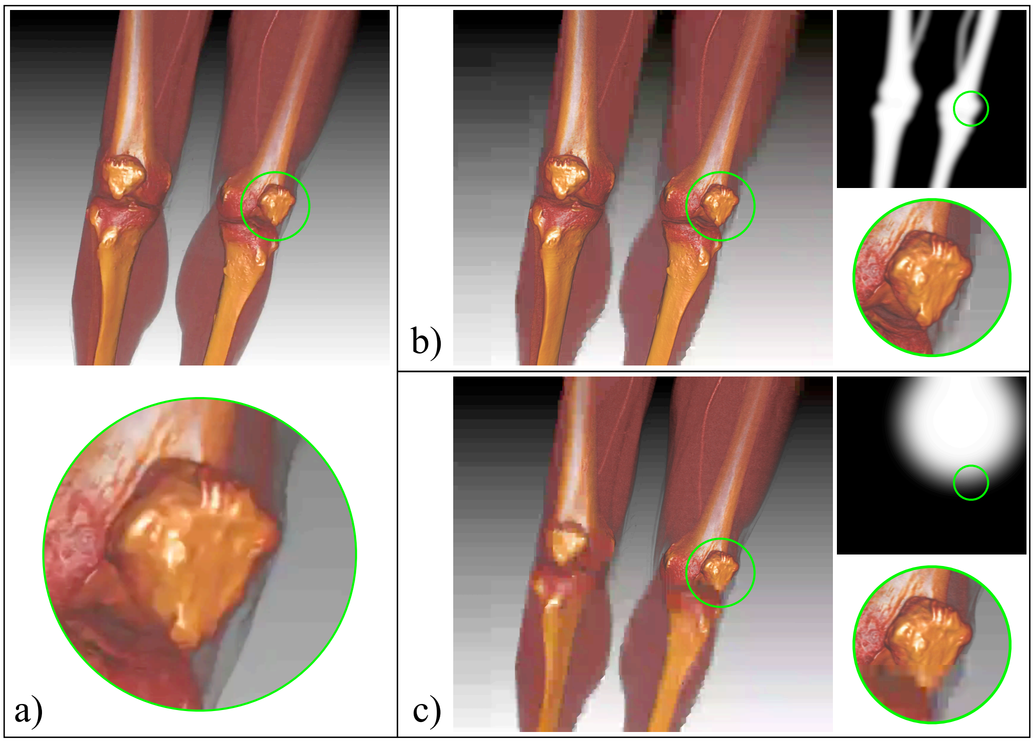

Figure 1: Progressive progress of cloud-based rendering in our ImageVis3D-powered rendering system. In (a) the video encoder was configured with default parameters, providing uniform quality across the image space. In (b), the user has provided a mask which explicitly highlights the bone and major vascular structure. As such, these regions appear in high resolution, whereas surrounding regions appear with noticeably less quality; notice the additional texture apparent on the bones, as compared to (a). In (c), the user has highlighted a specific region of the visualization, and so the system has adapted to provide higher quality in those areas. The inlet highlights the transition region from low-to-high quality. All scenarios would eventually display all data at native quality. Low quality areas allow much more efficient transfer of the data until that state is reached.

1.5 Visualization on Unstructured/Curvilinear Grids

We have not made strides in this area during this cycle.

1.6 Parallel Architectures

We have not made strides in this area during this cycle.

1.7 Provenance

James will write something here.

References

someone will fix my references.