Interactive 3D Visualization of Full Spatial Resolution Visible Human Datasets Using Out-of-Core and Cache Management Strategies

K. Palaniappan, J. Fraser, J. Hoberock
Dept. of Computer Engineering and Computer Science
Univ. of Missouri-Columbia
Columbia, MO 65211
573-884-9266
palani@cecs.missouri.edu

Submitted to Fourth Visible Human Project Conference, Keystone, Colorado, Oct. 17-19, 2002

A new hierarchical volumetric data structure, compression techniques and disk file format is proposed to enable interactive visualization of the full resolution Visible Human Datasets. Broader access to the VHD is facilitated by developing customized visualization tools that can be specialized to provide rapid access to large biomedical datasets. The volumetric data is decomposed into 3D tiles (ie. bricks or slabs) that are compressed using lossy or lossless techniques. An orthogonal plane of 3D tiles is moved from disk to a first level memory cache. 2D tile components are extracted dynamically as needed to construct user requested views. The specific transverse, coronal or sagittal slices are reconstructed dynamically from the set of 2D tiles in a second level memory cache. The reconstructed slice data is sent to texture memory for fast rendering. The data within a tile is organized as a Peano scan that further enables optimized block data transfers from disk to memory. Texture mapping is used to support smooth roaming and zooming across resolutions.

Application level control of resources such as memory, CPU and disk bypassing the operating system process manager enables better optimization and scalability to large datasets for well specified visualization tasks. A number of tradeoffs were investigated to achieve interactive performance. Large 3D tiles are preferred for fast throughput from disk. However, large tiles require large amounts of core memory, only a fraction of which is needed for immediate display and more time to extract individual slices. Changing the orthogonal plane for visualization necessitates discarding greater amounts of data with large 3D tiles. The size of the 3D tiles used ranged from 32x32x32 to 128x128x128 and significantly affects performance. It would be more convenient to store the data as 2D tiles on disk for orthogonal plane visualization. However, this would require three times as much disk space, one for each projection plane, which is expensive in terms of storage requirements. A novel priority distance calculation and flushing mechanism is used to replace tiles in cache as they age or as the tiles become distant from the current viewpoint.

Compression of 3D tiles achieves disk storage savings and also network bandwidth requirements at the expense of greater computational demands. Both lossless and lossy compression algorithms were tested to evaluate performance. Lossless techniques include run length encoding, zip, bz2 and png. Lossy techniques include JPEG and JPEG 2000. JPEG 2000 achieves high compression ratios with high peak signal to noise ratios. The compressed code stream can be further processed to add resiliency against network transmission errors and bandwidth congestion. Tiles can be prioritized or locked in memory if desired for certain applications.

Previous approaches to visualizing extremely large 3D volumetric data relied on hardware-based methods for rendering, reprocessing the data or both. The new software tool, Kolam achieves nearly all of the features of clip-textures much more efficiently in terms of texture memory usage without maintaining or building a complete hierarchical clip-map. Kolam uses a quad-bundle data structure that consists of a local set of compressed tiles from adjacent levels, and is extremely lightweight in terms of memory usage. The quad-bundle distance calculation is a novel approach for out-of-core visualization and can be used to establish a priority queue to manage the worker threads in a multithreaded implementation. Kolam can be enhanced for high pixel density output (>20Mpixels) for supporting display wall capabilities with multiple displays, for collaborative visualization.

Multithreaded reading, overlapped with tile decoding, tile assembly and rendering to the display significantly improves performance and reduces latencies from large data transfers. Additionally, tile prefetching using multiple read threads is implemented to minimize latencies during interactive visualization and maintain constant refresh rates in the graphics pipeline. Prefetching can be done on a tile basis. Toroidal texture maps provide an incremental method using spatial coherency within a quad-bundle for updating the display that is extremely fast. Innovative features of 3DKolam include interactive overlay display, texture maps for smooth (fractional) zoom, toroidal texture maps for fast display updates, color index image display, multiple color maps (one per layer), interactive brightness and contrast enhancement, grayscale and color histogram enhancement (histogram based on center tile, visible tiles, or top-most layer of pyramid), and image blending/multi-image display compositing using the alpha channel. Additional utilities include multithread performance monitoring, cache occupancy, age of resident tiles in cache, image pyramid display navigation glyph, etc.

Like the visible human datasets current multimodal imaging, functional MRI, and volume imaging are generating extremely large volumetric datasets. In order to facilitate correlative studies between datasets, allow intercomparison between datasets at different resolutions, utilize novel diagnostic techniques such as virtual endoscopy or combine anatomical surface annotation with volume data will require new approaches to visualization, file storage layout and data structures.