

The Real-time Interactive Visible Human Navigator

Sebastian Gerlach, Roger D. Hersch

Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Sebastian.Gerlach@epfl.ch

Introduction

Several viewing applications for the Visible Human dataset are currently available. Some applications provide access to annotated slices, either online or from a CD-ROM. Other applications provide services for extracting slices. The first Web application to allow extraction of slices perpendicular to the main axes was the NPAC Visible Human viewer applet (www.npac.syr.edu/projects/vishuman/VisibleHuman.html). The more recent Visible Human Slice and Surface Server from EPFL (<http://visiblehuman.epfl.ch>) provides access to arbitrarily oriented and positioned slices and surfaces.

A new highly optimized application, the Real-time Visible Human Navigator, provides the possibility to extract arbitrarily oriented and positioned slices from the Visible Human dataset in real-time on a standard PC. A continuous sequence of slices is displayed in real-time in response to navigation commands from the user. They reflect current position and orientation variations within the 3D Visible Human volume. The volume can be annotated with a set of markers to facilitate the localization of particular structures. The application runs on a standard PC with any of the 32-bit Windows operating systems (Windows 95, 98, NT4, 2000).

To realize the Real-time Interactive Navigator, we had to meet the following challenges:

1. Provide sufficient throughput from the storage subsystem to have enough data available to extract the required slices. The media type is a CD-ROM, a DVD-ROM or a hard disk.
2. To extract and display the slices rapidly, using both multiresolution and trilinear interpolation techniques.

User interaction

The navigator provides two views of the dataset: the slice as defined by the current position and orientation, and a 3D miniature model of the Visible Human. The miniature model is cut by the current slice in order to indicate the current

position and orientation to facilitate navigation. The position and orientation parameters are also provided numerically for later reference.



Figure 1. Real-time Visible Human navigator interface

The navigator provides real-time interactive navigation capability with standard input devices, such as a mouse or a joystick. The navigation in a 3D dataset requires 3 degrees of freedom for translation, and 3 additional degrees of freedom for rotation around the main axes of the current slice's local coordinate system. A seventh degree of freedom is added for freely zooming in and out of the slice. Since the available axes on a standard mouse or joystick are not sufficient, device buttons and keyboard keys must be used as well.

Dataset annotation

The user can annotate the 3D volume by using markers, which are fat 3D polyline segments. These markers can be freely placed on the current slice with the mouse, and navigation is permitted between the positioning of successive points of a marker. The polyline need not be continuous; it can be split at any point. Each marker has an associated name, color, and diameter. The markers are extruded as cylinders, and the display shows their intersection with the current slice. The user can select which markers should be displayed.

It is possible to jump to the starting point of any given marker to locate marked structures quickly. Markers can be saved and loaded from disk in simple ASCII text files, thus allowing their coordinates to be reused in other applications.

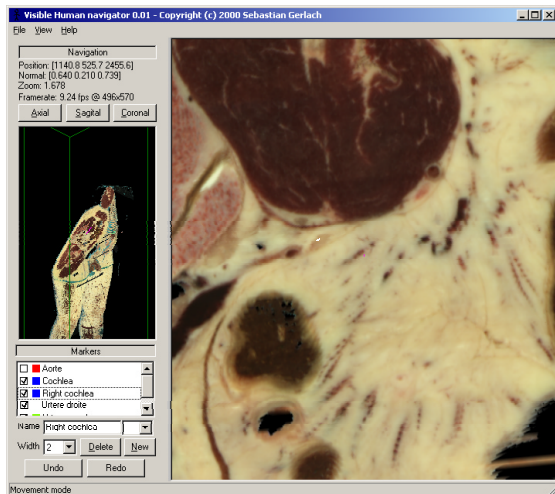
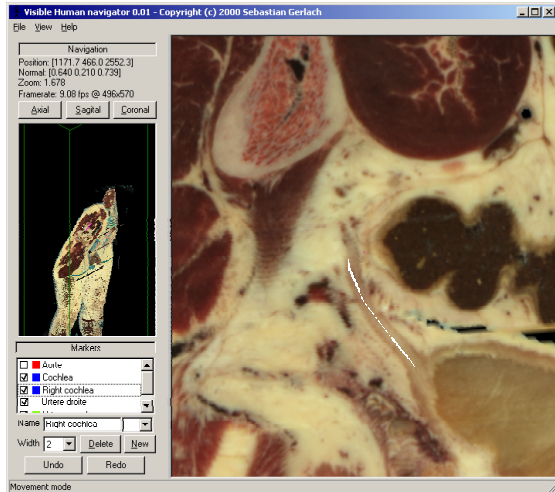


Figure 2. Marker showing the ureter as it enters the bladder (top) and marker showing the ureter on a slice slightly above the previous one (bottom)

Data storage

The data storage system is inspired by the PS2 storage system developed at the EPFL [Messerli99], although the file formats and access mechanisms have been modified and adapted for this application. The Visible Human dataset is stored as small subvolumes called extents, each subvolume containing 32x32x16 24-bit RGB voxels. Extents are striped over one or more files, which can reside on one or more disks.

In addition to the standard dataset, zoomed down versions of the Visible Human are also stored in the files, at reduction factors of 2, 4, 8, 16, and 32. The lowest resolution is stored first, and the full resolution dataset is stored last. We assume that the data is stored sequentially on the media. If a disk has a limitation on how large a file can be due to its file system or operating system, the file is split into several subfiles.

The original complete Visible Human Male dataset cannot fit on a single sided, single layer DVD (4.3 GB where 1 GB = 2^{30} bytes). The size of the dataset has been reduced by removing all the extents that are fully outside the useful portion of the Visible Human (i.e. all extents situated in the blue gel). The resulting complete cropped dataset takes 4.13 GB and is split into 5 subfiles on the DVD.

When rendering the slices, file access is done in two parts: a synchronous read, and an asynchronous read. The synchronous read loads the data that is required to start rendering, usually the lowest resolution dataset extents. For the asynchronous read, a list is generated containing all the extents needed to extract the slice at successive levels of detail. This list is sorted to minimize disk head movements. Since the lower resolutions are located first on disk, these will be loaded first, generating a low quality image. As higher resolutions are loaded, the image quality gradually increases. If the extraction parameters change before the list is fully loaded (e.g. when the user changes the current position or orientation), the current asynchronous load list is cancelled, and a new one is generated. The asynchronous reads are implemented using a thread pool, with one thread per disk.

An extent cache mechanism is also implemented. A static cache base contains the lowest resolution dataset at all times. A fixed size cache managed with a LRU replacement policy is used for all other extents. The size of the cache is based on the amount of physical memory available in the computer. The size of the cache determines the quality of the final extracted slices since all extents required for slice display have to be available in memory. If the slice requires too many extents for full resolution display (e.g. when the viewing window is very large), lower resolution extents are used.

Slice extraction

The slice extraction algorithm must be able to produce an arbitrarily oriented and positioned slice from the dataset. This operation is decomposed into two parts:

1. The extents at all resolutions that are intersected by the slice are determined and inserted into the lists for synchronous or asynchronous disk accesses.
2. The slice is rendered using the highest resolution extents available at rendering time.

The slice to be rendered is defined by four vectors as illustrated in figure 3. The *normal* vector is not required and can easily be calculated from the *up* and *right* vectors.

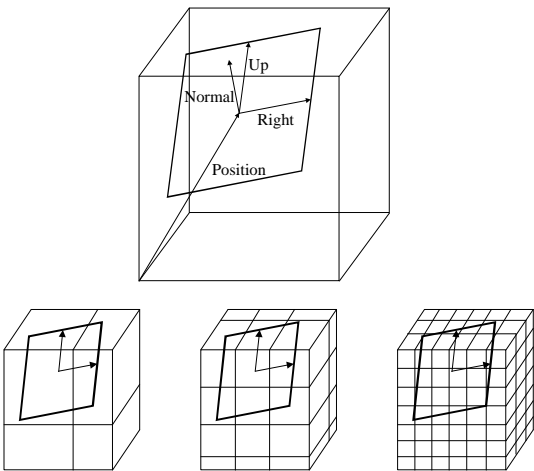


Figure 3. Definition of a slice in 3d space using 4 vectors (top) and decomposition of volume into extents of various resolutions (bottom)

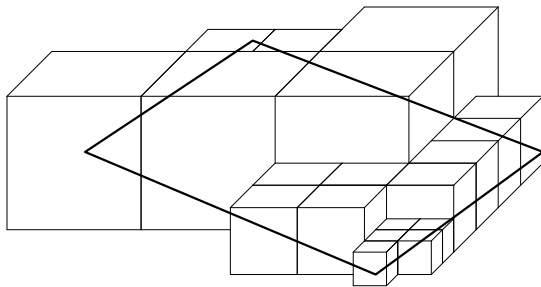


Figure 4. Intersection of a slice with extents of various resolutions

The rendering of the slice is done using an incremental fixed-point algorithm. The rendering starts at the top-left corner of the slice, and the 3D coordinates of the corresponding point in the dataset are evaluated. This coordinate is then

incremented using the right and up vectors of the current viewing parameters. The extent containing the corresponding voxel from the full resolution dataset is then retrieved from the cache. If it is not loaded yet, the different resolution datasets are traversed from the highest to the lowest resolution until a voxel is found. A voxel will always be found since the lowest resolution always resides in the static extent cache.

The final color displayed on the screen depends on the interpolation algorithm chosen by the user. Nearest-neighbor, linear interpolation along the Z-axis and trilinear interpolation are available. If one of the interpolation modes is active, more voxels need to be accessed from the extent cache. Trilinear interpolation is optimized in MMX assembly language.

Perspectives and conclusion

The Real-time Interactive Navigator opens new possibilities for the extraction of slices and animated slice sequences from the Visible Human dataset. It is possible to interactively follow complex twisted anatomic structures and navigate through an organ in any given direction. Thanks to the real-time interaction, it is easy to position slices at odd angles across structures in order to find the optimal observation direction.

The possibility to annotate the dataset provides a valuable tool for teaching anatomy. By navigating interactively through the dataset, students can mark specific structures. The markers can later be sent to the teacher, who can verify their correctness.

References

- [Messerli99] V. Messerli, O. Figueiredo, B. Gennart, R.D. Hersch, Parallelizing I/O intensive Image Access and Processing Applications, IEEE Concurrency, Vol. 7, No. 2, April-June 1999, 28-37