Modern Methods of Real Time Volume Imaging.

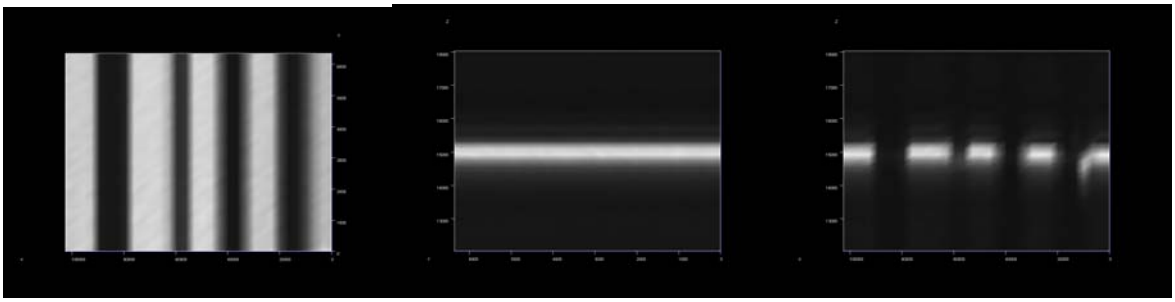
Volkov A.D., Zhdanov M.A., Pozin A.G.
NT–MDT Company, Zelenograd, Moscow, Russia

Introduction

There are two main basic ideas how to visualize 3D datasets: Crossections and Volume Rendering. Until recently only first one was used for real time presentation, excluding Silicon Graphics systems especially designed for similar tasks. The progress in commercial accelerators especially at low cost market encouraged us to develop volume rendering engine for common commercially available videoadapters for IBM PC platform.

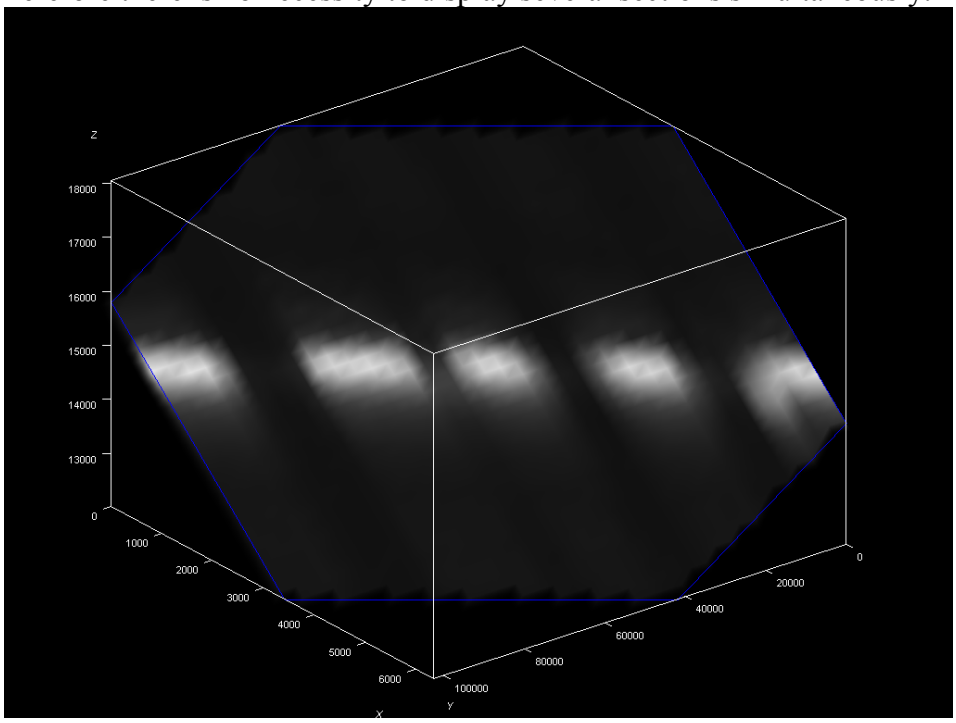
Ortogonal Crossections

The elementary method of 3D data representation is a series of consecutive sections parallel each other along one of 3D axes



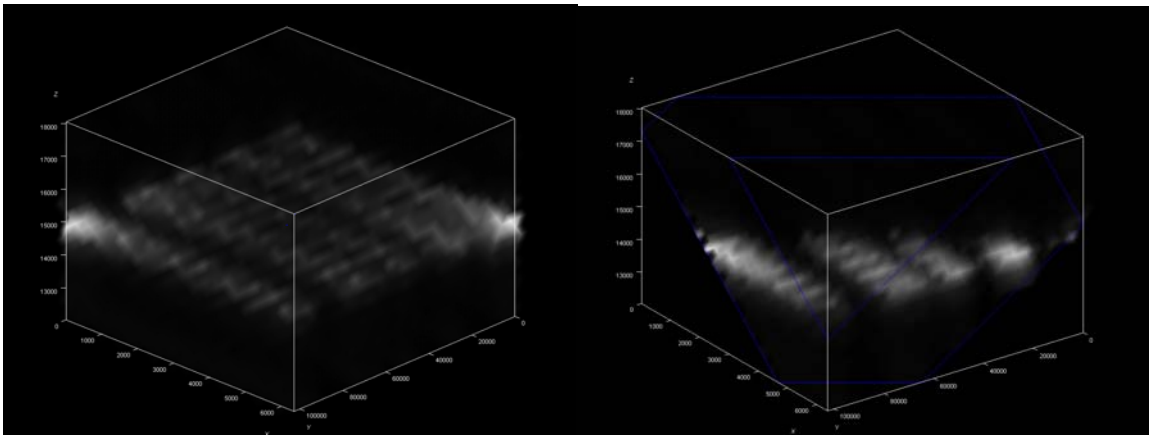
Arbitrary Crossections

Development of this idea are the sections at any distance under arbitrary corner. These sections can be calculated e.g. through linear interpolation. Modern computers easily calculate sections fast enougth to allow realtime operating of viewpoint by mouse or keyboard .(approximately up to 512*512) . Therefore there is no necessity to display several sections simultaneously.



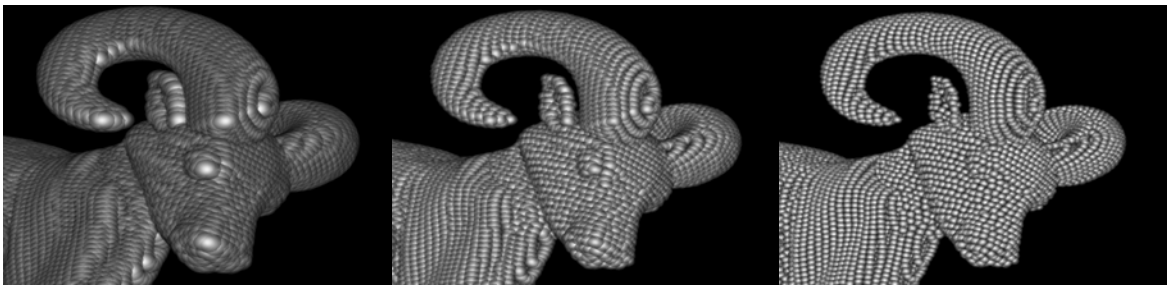
Superposition

More evident 3D representation of the data is the superposition. The superposition can be carried out by different methods, such as addition (the extended focus image), retaining the maximum or minimum along the line (the autofocus image), etc. In our program this is addition with consecutive averaging of all points projections on the plane selected (the extended focus mode). It is possible to set two parallel planes to display only data situated between them. Such an image is already similar to 3D object. Unfortunately, the realtime formation of such images is limited, performance time of this procedure is function of amount of points and grows under the cubic law for proportional growth of point number per dimension. Restriction - approximately 2 000 000 points.



Sparse fields of objects

For sparse fields of points (or non-intersecting spheres) quite good results are achieved by displaying them as individual objects in three-dimensional space. The advantage of this method is that every object can demonstrate individual size and form along with a size depending on distance and a radial transparency distribution, all - by means of videoaccelerator hardware. Only limitation – for correct visualisation the form of the object should be spherical. Restriction ~ 30000 points for real time operation.

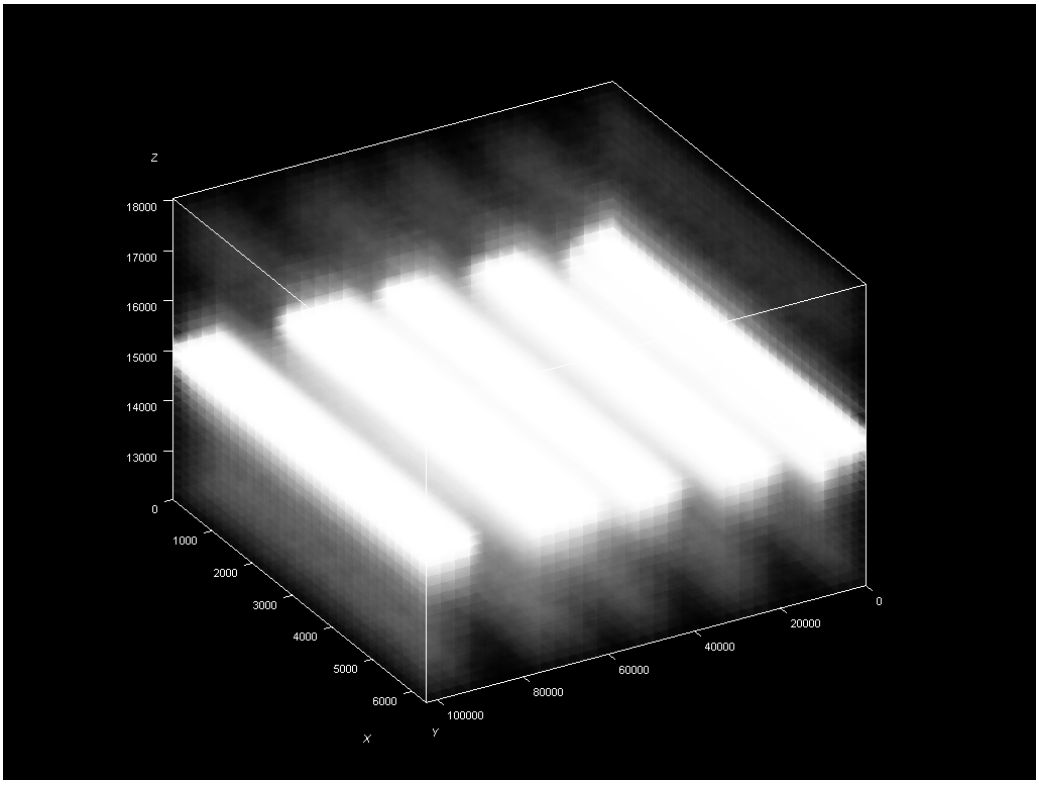


Triangle Superposition

Modern videoaccelerators are strongly optimized for the triangle rendering. Thus, with the help of hardware smoothing and color blending, the time to draw a point is comparable with the time of triangle rendering with smoothing of colour between vertices and blending colors with already rendered image. Therefore we have developed the 3D engine, displaying data not as points, but as rectangles (each of them consist of two triangles) with vertices coinciding in the points of qubic grid. Adjacent points form the rectangles in planes xy, yz, xz . Colour and transparency smoothing between tops of triangles create the realistic image, in which distant objects are overlapped by proximal ones. The rendering speed of this method does not concede to the superposition discussed above. It is necessary to tell, that would be possible in the superposition to make dependence of the contribution of a point in the final image from distance up to the image and to receive more realistic picture, close to this method, but as far as it would substantially increase rendering time there is no sense to use it.

Triangle Quasi-Superposition

Taking into account, that the temporary expenses practically are proportional to the number of triangles rendered, and the texturing of a triangle takes place without substantial expenses of time, it is possible to unite these triangles into large planes, and to generate single common texture for this layer of the data. This method gives the significant acceleration ~ 8 times. Thus 16 000 000 points can be rendered in realtime (the time is proportional to the sum of dimensions of dataset). The only problem here is that Z order of small triangles is damaged because all the plane is rendered at once, but if the set of parellel planes most perpendicular to the direction of a sight is the last one rendered, the distortions are neglible because contribution of most ortogonal planes is most significant for the resulting image. (Other plane sets provide smoothing of regional defects on the edges of the object). This method is the basic one for our program. Program also allows to change palettes, background, transparency, range of the data, point of view and so on.



Conclusion

Now we plan to increase framerate for realtime 3 times at the expense of rendering not 3 sets of sections, but only most appropriate one. It will allow to achieve up to 50 000 000 points real time presentation. If the quality of the image will substantially suffer, it will be possible to introduce other sets of diagonal sections, (the videomemory of modern videocards is large enougth) and show only the most appropriate set at every moment. To visualize large data sets (for example 1024x1024x1024) it is possible to use interpolation to some smaller number of points for preview and after to render with better quality using all data volume available for the chosen viewpoint, palette, etc. As is planned to make more realistic image (as with the help interpolation) in a case, when distance between points on different coordinates is unequal; to represent up to three functions in one three-dimensional grid simultaneously through different RGB color channels, make available to zoom the chosen site and observe a stereoimage through stereoglasses provided with some popular models of videoaccelerators.

1. G.J.Brakenhoff, H.T.M. Van der Voort and J.L.Oud. Three-demensional Image Representation in Confocal Microscopy.

2. Андрей Воробьев. The review of a videoaccelerator Nvidia GeForce 2 GTS.
http://ixbt.stack.net/koi/video/geforce2-gts.html.
3. Microsoft DirectX 7 SDK Manual.