Large data volume visualization on distributed multiprocessor systems

Peter S. Krinov, Mikhail V. Iakobovski*, Sergey V. Muravyov

Institute for Mathematical Modelling, Russian academy of Sciences
125047, Miusskaya Sq. 4a, Moscow, Russia.

Key words: parallel visualization.

The development of supercomputer centers giving their resources through Internet, has multiplied the number of scientific and applied problems solving by means of numerical experiments. Hence the new class of problems arises. Among them is the problem of visualization of large data volume, which exceeds memory resources of personal computer or a single processor unit. Another problem is the lack of acceptable visualization tools that makes difficulties for carrying out of large-scale three-dimensional simulations of gas dynamic, combustion, microelectronic and some others problems on multiprocessor systems. The volume of grid data, initial data used and simulation results may exceed memory resources of personal computer. That does not allow taking advantages of modern visualization programs and demands development of specific tools.

Computing power of systems now grows very quickly. The number of processors in multiprocessor systems increases and thus one should expect appreciable growth of data volume related the results of calculations. Our work is focused on processing of over hundreds of gigabytes of data. That requires new methods of information processing.

Such data volume cannot be allocated in operative memory of a personal computer nor even on a hard drive. Throughput of any network is limited. These restrictions prevent the use of traditional applications, since transmission through any network of data for analysis and visualization does not give an opportunity of interactive supervision of results.

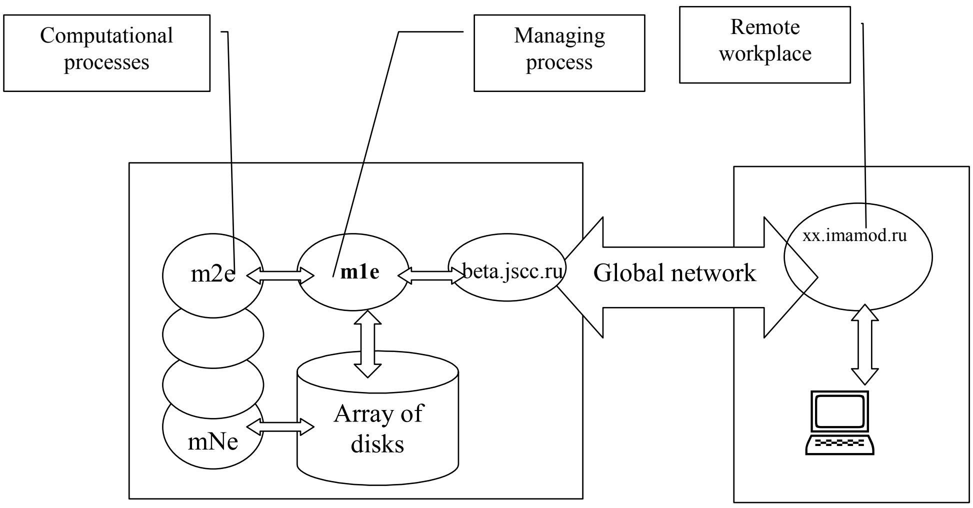

Fig. 1. Visualization system structure

The large data visualization system, described here, is divided into two parts – server and client - Client/Server model (Fig.1). This division allows carrying out the principle part of visualization process on supercomputer and transferring to user's workplace only minimum of information required directly for construction of prepared image. Such approach assumes that the image is finally formed on a user's workplace and it is possible to use modern multimedia hard-

* E-mail: lira@imamod.ru. Phone: +7(095)250-7823. Fax: +7(095)972-0723.

This work has been supported in part by the Russian Fund for Basic Research (Grant No. 02-01-00589).

ware (helmets, stereo glasses, three-dimensional manipulators etc.) for better clearness of visual information.

Method of iso-surfaces is chosen from existing variety of methods for such data visualization. These are the surfaces on which function under consideration (temperature, density, concentration etc.) takes a fixed value. Thus, several surfaces (with function value specified by user) are chosen from three-dimensional scalar field and displayed on the screen.

Within such an approach, some triangular grid determines each iso-surface. The sizes of these grids are in some cases quite comparable to the size of initial three-dimensional grids and cannot be transferred through a computer network (and also processed) for the reasonable time providing an interactive mode. Moreover, they cannot be allocated in the operative memory of a single processor unit. Thus, the central problem under study is the compression of triangulated surfaces presented as unstructured triangular grids. It is necessary both for their fast transfer through a global or local network and for their direct display at user's workplace.

Compression of iso-surface

By triangulated surface is meant a set of points arbitrarily located in three-dimensional space and a set of not crossed triangles given on them so that no more than two triangles bases on one edge connecting two points.

The following two algorithms for iso-surface compression are considered further:

Compression of zones of unambiguity.

Compression by consecutive excluding of units.

Compression of zones of unambiguity

We shall consider the construction algorithm for triangulated surface ***G***, approximating iso-surface ***F***, built in such a manner that the necessity of surface ***G*** topology storaging is almost completely excluded.

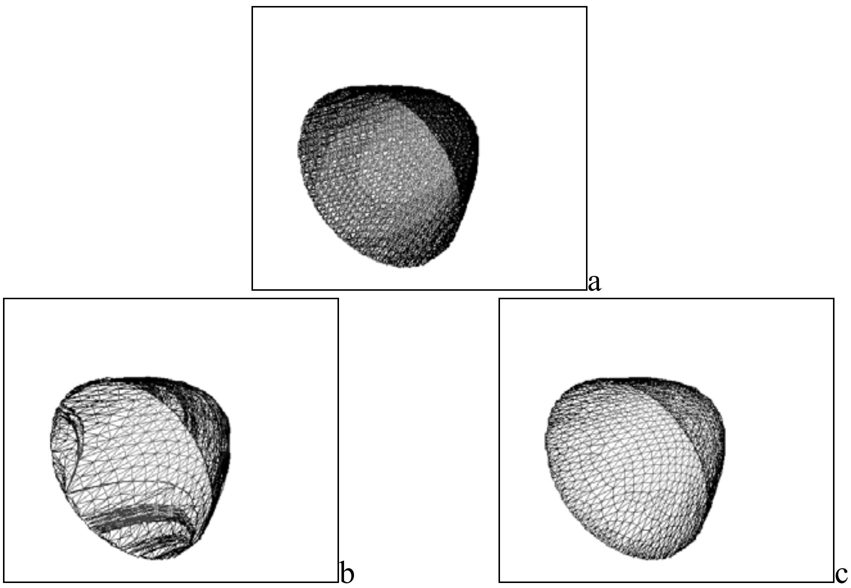

Fig. 2. Stages of iso-surface compression.

Let us consider compression of iso-surface ***F***, unequivocally projected on a plane (***x,y***) (fig. 2a). In that case all points forming iso-surface can be divided into inner (which neighbours form a simple loop) and outer. Deleting all inner and a part of outer points we shall get the set of basis points of approximating surface ***G***. Now we shall construct a triangulation of inner area, consecutively adding the inner vertices of triangulation into the area one by one. While defining coordinates of the sequential points we shall use only the information on coordinates (***x,y,z***) of basis points and coordinates (***x,y,z***) of already added points, where ***z*** is the coordinate of intersection of a normal to a plane (***x,y***) with iso-surface ***F(x,y)***. Thus, for the restoration of surface ***G*** it is necessary to transfer to the client only the data (***x,y,z***) on basis points and value (*z*) in the added points. There is no necessity to transfer coordinates (***x,y***) of added points and topology of connections between them as they can be completely defined in process of restoration of a surface, just having repeated the actions executed in the course of its compression.

An example of the surface received by such method is given in fig 2b. Simple algorithm of consecutive addition of points used for surface construction looks as follows:

1. Choosing of the longest edge (a,c) in the already constructed part of surface G (fig. 3);
2. Addition of a point (e) in middle of the chosen edge (a,c);
3. Connection of the added point (e) by edges with opposite tops of triangles (b),(d), basing on an edge (a,c);
4. Determination of a level z of iso-surface in a point (e).

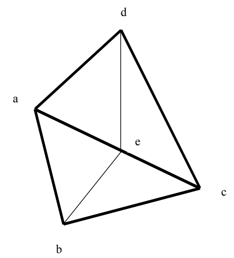

Fig. 3.Steps of compression of iso-surface

It is obvious that the given algorithm allows restoration of iso-surface on the client part of the system. However, the simplicity of the algorithm results in the fact that too many points close to each other are added in the narrow lengthy triangles constructed on reference points. Use of more advanced algorithms allows to receive quite acceptable results by insignificant increase of time of formation of surface ***G***. Good results are achieved by addition of a point in the center of gravity of triangle of the maximum area with the subsequent updating a triangulation according to criterion of Delone.

The stated approach allows obtaining a dual benefit at once. At first, decrease considerably number of points (generally it is enough to have several hundreds of points to specify the form of a surface). Second, the volume of the transmitted data reduces as there is no need to transfer the information on topology of surface ***G***.

The ad hog algorithm is developed for approximation of the surfaces, which cannot be projected unequivocally on any plane. This algorithm performs splitting of any triangulated surface into connected fragments, so that each one can be unequivocally projected on some plane (generally not parallel to any of coordinate planes)

The presented result can be considerably improved by reduction of number of the points determining the border of iso-surface and the borders between zones of unambiguity. The advantage of the algorithm in question lies in the simplicity of the analysis and in elimination of necessity to store the full topology of approximation surface. It is necessary to keep only the parameters of projecting planes, the borders defining unequivocally projecting zones, distances from the projecting plane designed for each zone to its points, and a sequence of numbers of triangles in which addition of points was made while construction of the surface on the server. That allows using any given algorithm for construction of surface on server.

Compression by consecutive excluding of units

The second algorithm does not suppose preliminary splitting of connected surface in parts. The main idea of the algorithm consists in removing of some points from initial surface so that the triangulation constructed from residuary points, approximates the initial surface with required accuracy.

Points of the initial data are consistently looked over, thus, for each point is checked whether it is possible to remove it without lessening required accuracy of approximation. In case of removal the point, adjacent vertices should be connected to each other so that the formed polygon will be filled by triangles. The removed point is recorded in the list associated to one of these triangles, the nearest to the point. The appropriate lists are required to supervise size of deviation of formed surface from initial. The possible maximum deviation is beforehand determined by some value d. This value depends on the desirable accuracy of approximation of the initial data by the obtained one, and is taken equal to the product L and e, where L is the linear size of displayed data and e is a factor of desirable accuracy (value 0.01 gives quite allowable accuracy). For improvement of the approximation quality removal of point occurs iteratively. In doing so the points are removed at each step uniformly in regular intervals over the whole surface, therefore the removal of the points neighboring to the point already removed on the current step is temporary relayed. Thus, some points are left out of account and processed later on. The possibility of their removal is examined in subsequent revision of points list. On termination of the next step of analysis (if any point was removed) the step is repeated. Thus, during each revision of points list some of them are removed together with the triangles containing these points. At the same time some new triangles are added instead. The points assigned to the earlier removed triangles are preserved. Now they are assigned to the nearest new triangles, constructed in the area of the built polygon. Thus, some number (may be null) of removed points (located closer to this triangle than to any other) is stored in the list concerning to each triangle.

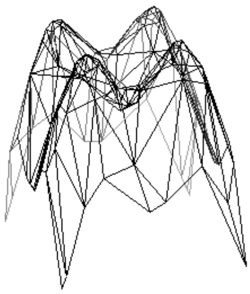

Fig. 4. Compression with accuracy 2%

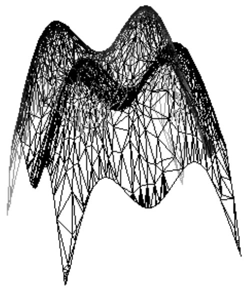

Fig. 5. Compression with accuracy 0,1%

The advantages of algorithm are the high quality of approximation from the point of view of the human perception of the constructed surface (fig. 5), and the possibility of compression with desired accuracy of any triangulated non-self-crossing and generally disconnected surfaces. The given algorithm shows the property of "locality", i.e. during processing of each point it is necessary to analyze only the information about a little fragment of surface (directly adjacent to the point). That makes it possible to speak about the high level of inner parallelism and expect the appropriate algorithm (based on domain decomposition method) for the multiprocessor systems to have high efficiency.

In the conclusion it should be noted that the suggested surface compression algorithms possess the high computational efficiency especially for processing results of the computing experiments carried out on multiprocessor systems. This is because the suggested algorithms allow effective paralleling and are suitable for processing grid data which volume is comparable with the whole operative memory of computer system.

References

1. http://visual.imamod.ru/
2. P.S Krinov, S.V. Polyakov, M.V. Iakobovski. Visualisation in distributed computational systems of results of three-dimensional computations. // 4 international conference of mathematical modeling: "Stankin", 2001., v.2, p.126-133
3. D.E. Karasev, M.V. Iakobovski. Visualisation of gas-dynamics currents on multiprocessing systems in the distributed computer networks. Fundamental physical and mathematical problems and modelling of tehniko-technological systems: the collection of proceedings. M.: "Stankin", 2000, p.160-168.
4. M.V.Iakobovski, D.E.Karasev, P.S.Krinov, S.V.Polyakov. Visualisation of grand challenge data on Distributed systems. Kluwer Academic, Plenum Publishers. 2001, p.71-78
5. http://www.gzip.org/zlib/, http://www.info-zip.org/pub/infozip/zlib/