РОССИЙСКАЯ АКАДЕМИЯ НАУК

**ИНСТИТУТ ПРОБЛЕМ
БЕЗОПАСНОГО РАЗВИТИЯ
АТОМНОЙ ЭНЕРГЕТИКИ**

RUSSIAN ACADEMY OF SCIENCES

**NUCLEAR SAFETY
INSTITUTE**

Препринт ИБРАЭ № IBRAE-2001-05

Preprint IBRAE-2001-05

S. Chernov, S. Zhukov

POSTPLOT3D — VISUALIZATION OF 3-DIMENSIONAL SPATIALLY-DISTRIBUTED DATA

Москва
2001

Moscow
2001

УДК 502.3

Чернов С.Ю., Жуков С.Б. POSTPLOT3D — ВИЗУАЛИЗАЦИЯ ТРЕХМЕРНОЙ ПРОСТРАНСТВЕННО-РАСПРЕДЕЛЕННОЙ ИНФОРМАЦИИ (на англ. яз.). Препринт № IBRAE-2001-05. Москва: Институт проблем безопасного развития атомной энергетики РАН, 2001, 15 с. — Библиогр.: 5 назв.

Аннотация

В работе представлена программа PostPlot3D, написанная специально для визуализации трехмерной пространственно-распределенной информации. Здесь показаны ее возможности, преимущества и ограничения, которые на нее накладываются. В препринте также описаны общие задачи и проблемы визуализации трехмерной пространственно-распределенной информации. Содержится анализ программного обеспечения, которое может быть использовано для решения этой задачи.

Chernov S., Zhukov S. POSTPLO3D — VISUALIZATION OF 3-DIMENSIONAL SPATIALLY-DISTRIBUTED DATA. Preprint IBRAE-2001-05. Moscow: Nuclear Safety Institute RAS, April 2001, 15 p. — Refs.: 5 items.

Abstract

This work presents program PostPlot3D that is written specially for visualization of 3-Dimensional spatially distributed data. There’re shown its opportunities, its advantages and restrictions. This preprint also describes some common tasks and problems of 3D-visualization of spatially-distributed data. It contains analysis of software, which could be used for this task.

PostPlot3D – visualization of 3-Dimensional spatially-distributed data

Chernov S.U., Zhukov S.B.

Nuclear Safety Institute of Russian Academy of Sciences
52, B.Tulskaya st.,Moscow, 113191, Russia,
ph.: (095)955-22-31, e-mail : scher@ibrae.ac.ru, zhukov@ibrae.ac.ru

Contents

About

This work may be treated as an attempt to create interactive visual device for 3-Dimensional visualization in the classical geostatistical program GeoStatOffice.

GeoStatOffice includes great amount of capabilities for working with spatially distributed data. It includes a lot of algorithms, which make possible to treat data in many ways. Next of its advantages is maximally convenient interface. Some results of its work can be represented as an image. In many cases it's enough to have standard 2-Dimensional representation that already exists as a part of GeoStatOffice. User can choose data from his own database and the program gives a visual representation.

The next step should be in 3-Dimensional spatial data visualization. So 3D-representation should inherit main features of the GeoStatOffice: it's interface, it's style of data treating file formats. It should be a logical addition to it, but with some new features, which are necessary for creating new kind of spatial data representation.

This product for 3D-representation was called PostPlot3D and is described in details in Chapter2. Chapter 1 is some kind of viewing over task of 3-Dimensional representation.

Basic concepts of three-dimensional graphics

Interest to data visualization is explained by high informativity of this procedure. Information contained in images is exhibited in maximally concentrated form. And this information is usually more convenient for analysis. Because for its perception user is not obliged to have great amount of special knowledge.

In contrast to photographic, television and so on, systems of machine graphics represent not physical processes and objects themselves, but mathematical models. And produces it on the screen of display. Through entering concrete parameters Machine Graphics System synthesizes image based on common models.

Specifics of models and treating programs in 3D (Three-Dimensional) Machine Graphics is defined by necessarity of delivering perception of spatial depth, spatial form and structure of objects. So there're 3 degrees of evaluation of similarity for produced 3D image: physical, psychophysical and psychological.

Application to Machine Graphics implies that synthesized image in main physical characteristics should resemble *original* (that's called real image).

Physical criterion is set by 3 groups of characteristics: geometric (spatial), brightness (energetical) and timing ones. Accurate resemblance means that some characteristic of image and original must be equivalent.

Psychophysical criterion means that due to restrictions of human visual perception it's possible to set some level of distortion. Then person will not be able to find difference between synthesized image and original because visual perception of them is identical.

Psychological criterion means that main perception of image and original should be identical. That's the main factor of delivering basic idea of real object in spite of serious difference in physical characteristics.

So synthesis of 3D images at the level of psychological resemblance is widely spreaded in Machine Graphics. Recipes for many elements are the same to painting art. That's about algorithms of forming of definite section of object, removing invisible lines from representation, forming perspective. The other group of these methods is setting brightness and color palette for some elements of image, modeling of shadows. But surely Machine Graphics is poorer than painting because of complexity of appropriate algorithms.

In solving tasks like forming 3D-drafts in some projection the task is mainly to have psychological resemblance to imagine the structure of object. The other tasks like some imitations or esthetical view are created based mainly on physical and psychophysical resemblance.

As workstations became more powerful and object-oriented programming gained in popularity, OpenGL (Open Graphics Library) became the defacto standard for 3D graphics development. OpenGL is the computer industry standard API (application program interface) for defining 2-D and 3-D graphic images. Prior to OpenGL, any company developing a graphical application typically had to rewrite the Graphics part of it for each operating system platform and had to be cognizant of the graphics hardware as well. With OpenGL, an application can create the same effects in any operating system using any OpenGL-adhering graphics adapter.

OpenGL specifies set of “commands” or immediately executed functions. Each command directs a drawing action or causes special effects. A list of these commands can be created for repetitive effects. OpenGL comes with a large number of built-in capabilities requestable through API. These include hidden surface removal, alpha blending (transparency), antialiasing, texture mapping, pixel operations, viewing and modeling transformations, and atmospheric effects (fog, smoke and haze).

And so, 3-D image creation can be viewed as a three-phase process of: tessellation, geometry and rendering. Through tessellation models are created of individual objects using linked points that are made into a number of individual polygons (tiles). Through geometry phase the polygons are transformed in various ways and lighting effects are applied. And finally the transformed images are rendered into objects very fine detail.

Short review on 3D-visualization products.

Products for three-dimensional visualization should be divided on several groups.

First group is consisted of such products as Extreme 3D, 3D Studio Max, Ray Dream Studio and some products from SoftImage (www.softimage.com), AutoDesk (www.autodesk.com), Kinetix (www.ktx.com), LightScape (www.lightscape.com), Maxon (www.maxon.com). This group is unified by great opportunities of 3D-modelling, visualization and even animation. They use to have very convenient interface for 3D-modelling, where practically all-possible operations on creating and transforming 3D-objects are included. And very fine rendering details, and even connections to database. So, they're very popular for creating striking Computer Graphic effects, computer films and so on. Nevertheless they're do not seem so good for researches because of their poor orientation on concrete research subject.

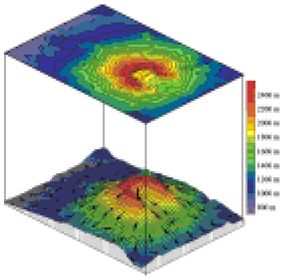
Second group is products for concrete scientific research.

There're a lot of different tasks that can exhibit their best by means of 3D Graphics. So there're specific products, which used to be oriented on specific branch of science: for example modelling car crashes, modelling molecular structure and many, many others. So one should choose his field and to find products, which correspond better to desired concept.

According to our present tasks, we are interested mainly in applications for different kinds of geology. 3D-visualization software, that's oriented on geology and some other kind of geosciences, may be differentiated in turn by smaller groups.

Most of applications are oriented on surface visualization. Here's one of interesting examples:

Golden Software Surfer (http://www.goldensoftware.com/frames/surferframe.htm).

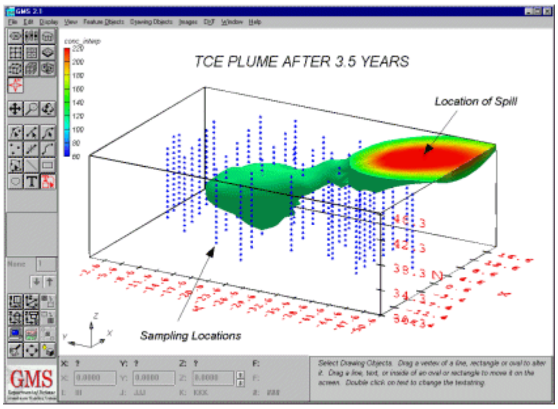


Surfer is a contouring and 3D surface plotting program that has extensive variogram modeling capabilities. Surfer converts data into outstanding contour, wireframe, vector, image, shaded relief, and post maps. Virtually all aspects of maps can be customized to produce exactly the presentation.

Some software is strongly recommended for volumetric visualization of desired structure. Though there're usable for visualization of 2D surface.

(for this software it's added some comments taken from according Web-pages).

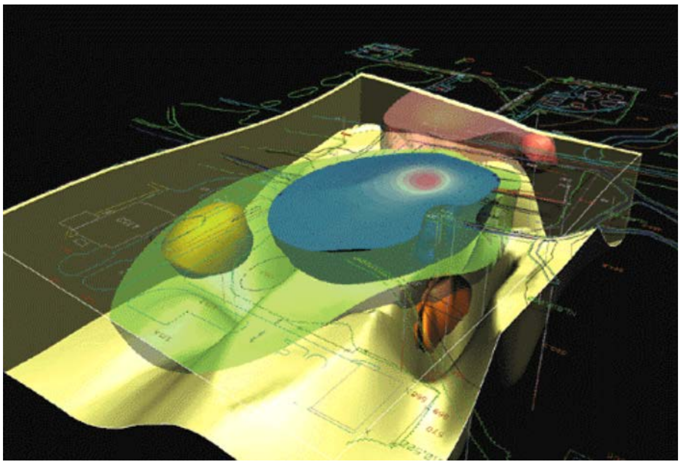
GMS (http://www.rockware.com/catalog/pages/gms.html).



Select Drawing Objects. Drag a vertex of a line, rectangle or oval to alter it. Drag a line, text, or inside of an oval or rectangle to move it on the screen. Double click on text to change the textstring.

GMS is a comprehensive package, which provides tools for every phase of a groundwater simulation including site characterization, model development, post-processing, calibration, and visualization. GMS is the only system, which supports TINs, solids, borehole data, 2D & 3D geostatistics, and both finite element and finite difference models in 2D & 3D.

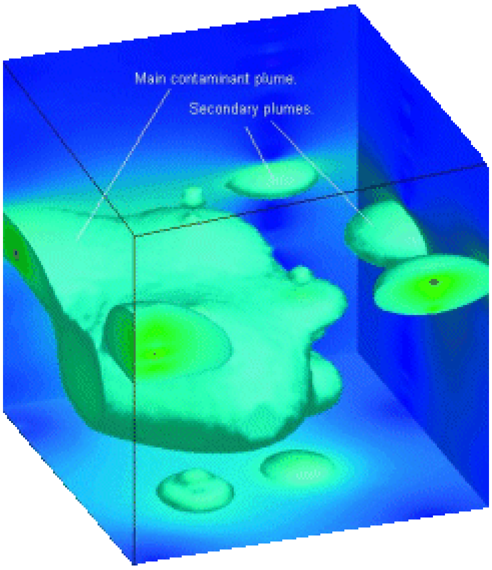
Voxel Analyst (http://www.rockware.com/catalog/pages/voxel.html).



Voxel Analyst is a general-purpose data visualization and analysis tool that helps scientists and engineers understand the relationship between different attributes within 3-dimensional volume data sets. Characteristics within the volume may be described numerically and are not necessarily homogeneous. These can be physical characteristics (such as density, porosity, temperature), chemical characteristics (such as chemical concentrations), or abstract characteristics (such as standard error or probability).

RockWorks (http://www.rockware.com/catalog/pages/rw99/rw_introduction.html).

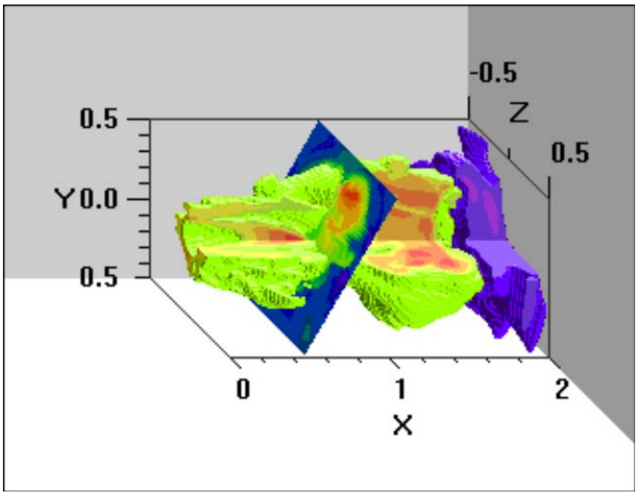
Contaminant Plume Modeling & Visualization ...

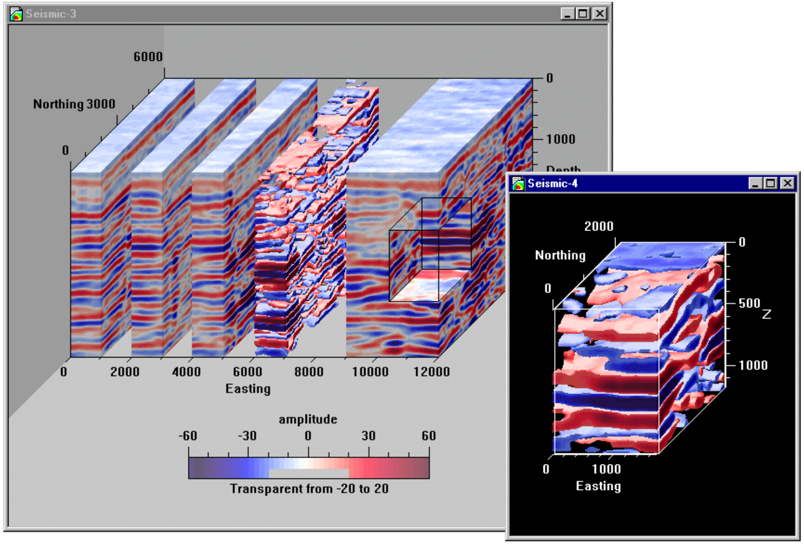


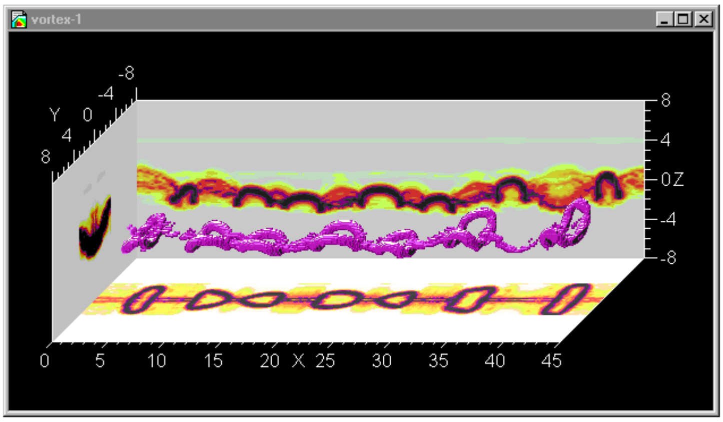
RockWorks includes programs for point mapping, contouring, gridding, solid-modeling, volumetrics, strip logs, cross-sections, fence diagrams, stratigraphic block modeling, univariate statistics, bivariate statistics, trivariate statistics, lineation analysis, planar feature analysis, drawdown modeling, hydrochemistry, coordinate conversions, 3D visualization, data conversions, and much more ...

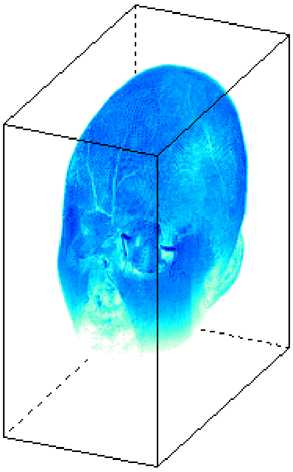
There’s some software that can be used for wide range of tasks in geology: they work with GIS and mapping and volumetric visualization and some more. Interesting applications of this kind are:

SlicerDicer (http://www.rockware.com/catalog/pages/slicerdicer.html).



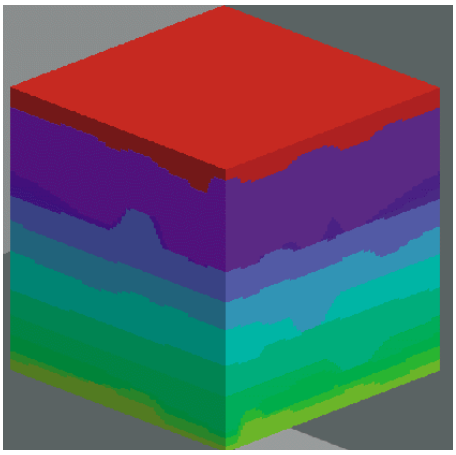




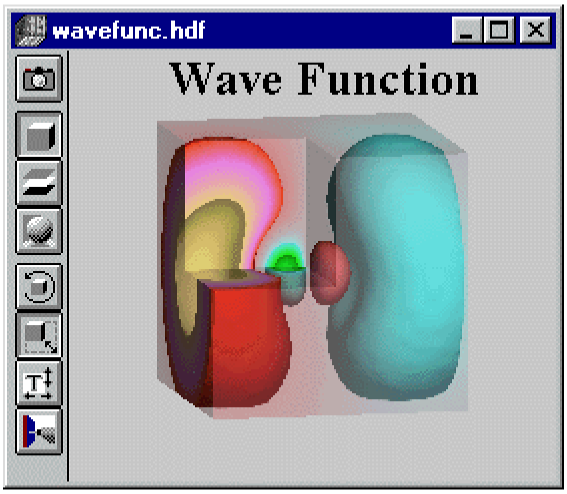


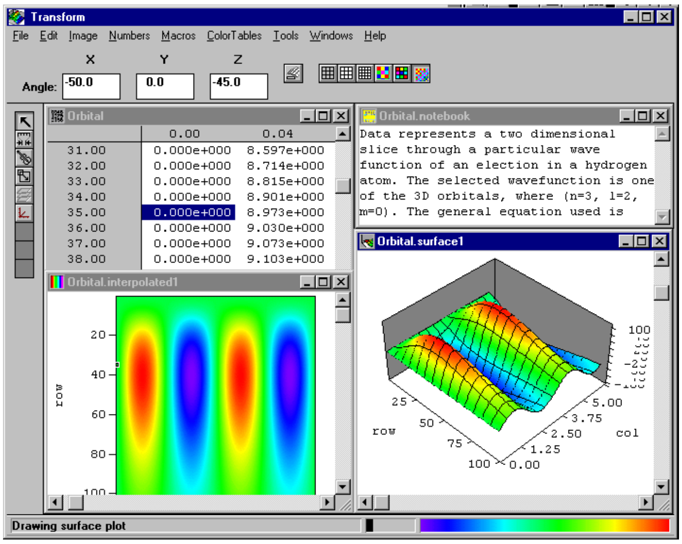
(This is demonstration of rotating capabilities also).

NOESYS (http://www.rockware.com/catalog/pages/noesysapps.html).



Some visualization of layers

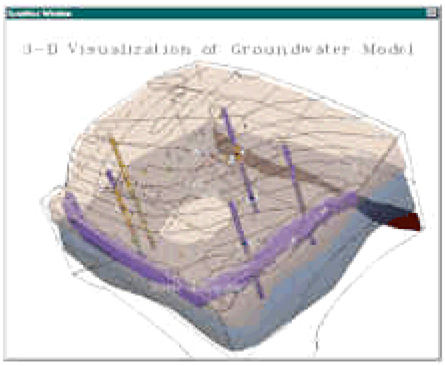


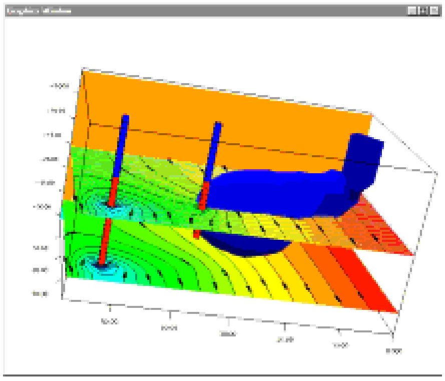


	0.00	0.04
31.00	0.000e+000	8.597e+000
32.00	0.000e+000	8.714e+000
33.00	0.000e+000	8.815e+000
34.00	0.000e+000	8.901e+000
35.00	0.000e+000	8.973e+000
36.00	0.000e+000	9.030e+000
37.00	0.000e+000	9.073e+000
38.00	0.000e+000	9.103e+000

```
Data represents a two dimensional
slice through a particular wave
function of an election in a hydrogen
atom. The selected wavefunction is one
of the 3D orbitals, where (n=3, l=2,
m=0). The general equation used is
```

Interesting software for groundwater visualization: Visual Groundwater (http://www.scisoftware.com/products/visual_groundwater_details/visual_groundwater_details.html).





One more animation.

Last group may be unified by concept of relatively easy creating of applications oriented on our needs. This group used to have status of programming language or its extension.

So the choice used to be by concrete needs. For specialists in C++ it's quite possible to keep it. Java programmers will surely like Java extension Java3D, though it requires some additional installation.

For Internet-oriented needs it's possible to use either Java3D or VRML. VRML means Virtual Reality Markup Language and like HTML frequently allows setting of object on scene by one "tag". Withal there is a project for VRML and Java interaction.

For other tasks there exist specially oriented Software Developer Kits. For example, for programming of Virtual Reality games one can choose something like SDK or GDK from SoftImage etc.

Specifics of PostPlot3D

So here's description of PostPlot3D itself.

1.Tasks of PostPlot3D

So, the first criterion for PostPlot3D was to test it over some typical tasks on which it could be oriented on and over some typical extremes.

Particularly, first task was to check it as a variant of checking probes of ground.

Each probe can be represented by sphere or box in the node. Like in GeoStatOffice the difference in characteristics of each probe is defined by color of chosen mark. (Figure 1).

From database one can choose coordinates of the node and corresponding values of its characteristics. If the value is less than some bound value – the mark is absent.

As additional step there's some generalization made – that means removing some negative extremes. For example, if the desired visualization should be done for small steps between nodes – the frequency of grid lines should be reduced to avoid blurring of picture, but size of mark should not be less than it's possible to visualize.

These procedures are precisely similar to GeoStatOffice 2D-representation.

But surely 3-Dimensional task requires some more parameters. So, one task was to check how effective could be such approach – how many nodes could be treated in such way because size of data is greater. It is checked that volume of memory in 64Mb is enough for treating with hundreds of thousands of nodes – so it seems to be not bad.

Next task is that setting grid is commonly depend on:

1) Steps of the grid along each axis;
2) Angles of rotation of grid around each axis;
3) Starting point.
4) Some ideas for making visualization more convenient: rotation by left mouse button, translation by right mouse button, zooming by central mouse button or by buttons on the representation itself.

All these ideas are already set in PostPlot3D. Now intensive work goes over activating real capabilities of PostPlot3D. That includes, drawing arbitrary objects, drawing them one inside another (using transparency), drawing arbitrary sections of objects. These features should make PostPlot3D much more efficient.

2. Structure and Realization

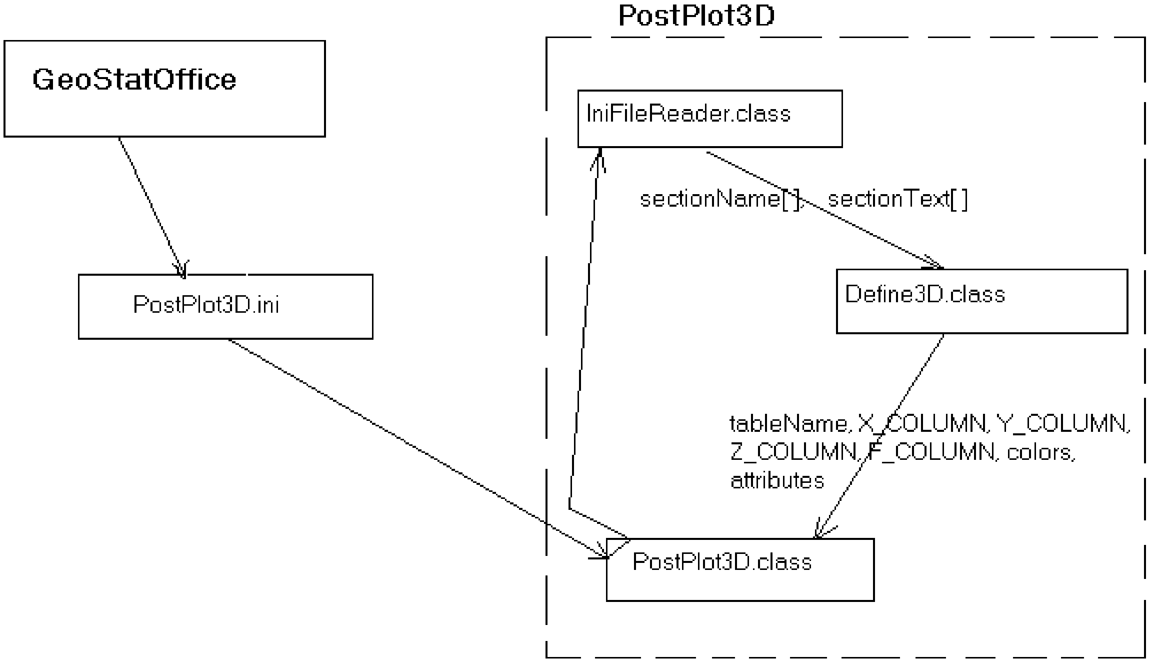
The structure of PostPlot3D is hardly defined by GeoStatOffice's standards. So, most of parameters, are taken from GeoStatOffice's dialogues and textfields.

Then these parameters are saved in file *.ini and 3D-visualisation program is activated using this ini-file. Unlike GeoStatOffice, which is written on C++, PostPlot3D is written on Java. So GeoStatOffice's interface controls only till it activates PostPlot3D with ini-file. All further controls are part of PostPlot3D.

In dialogs of GeoStatOffice itself one can choose which table he wants to work with. Then which column of the table he wants to set as x-coord, y-coord, z-coord and value at this node. Then one can choose desired color palette, levels for characteristics and some additional attributes like color of the grid, steps along each axis, type and size of mark.

In runtime all changes in PostPlot3D can be made only by means of PostPlot3D. So one can rotate whole image with mouse or set grid parameters by this dialog. These parameters are set in following way: rotation in radians, translation in relative units where center of the screen is assigned to 0,0,0 and border is defined as 1 divided by scale (normal scale equals 1, default for this program is 0.4).

So the structure of interaction between GeoStatOffice and PostPlot3D is shown on the following figure:

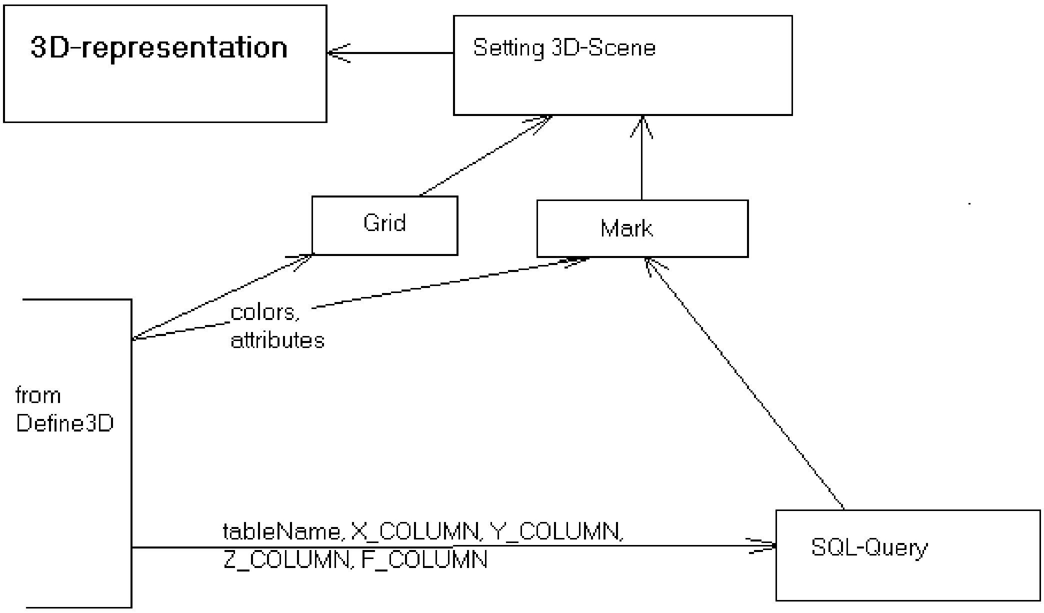

PostPlot3D
GeoStatOffice
IniFileReader.class
sectionName[], sectionText[ ]
Define3D.class
PostPlot3D.ini
tableName, X_COLUMN, Y_COLUMN, Z_COLUMN, F_COLUMN, colors, attributes
PostPlot3D.class

The idea is that PostPlot3D opens ini-file that's recommended and prepared for him by GeoStatOffice with IniFileReader. IniFileReader returns array of section's names and section's texts so that each section text corresponds to section name.

Then Define3D gets this information and defines what table from database contains information to represent. It can be defined which column should be taken as column for X-,Y- and Z- value and function for this node. If user wants to set his scale of colors then it would be read from this ini-file. If user want to change color of the grid, to switch mark from sphere to box, to set size of mark and to set initial parameters for grid (steps along each axis, rotation angles of grid and initial position) then it would be read from the same ini-file also.

(IniFileReader and Define3D are part of PostPlot3D concept).

And few words about 3D-representing part of PostPlot3D itself.



To create SQL-Query information about database is sent from Define3D. This query will return the data on which it's defined if mark exists in this node and what value is assigned to this node. Information about all grid-attributes is sent to module that's responsible for grid image creation. And color scale is sent directly to mark-creating module.

So, all this information is sent to module, which creates Scene where information on where to place each element and how to transform it is worked on. And finally it all is rendered to the image that's expected to be desired 3D-representation.

This structure seems to be basic enough to preserve it through all consequent changes and additions to PostPlot3D.

Acknowledgments

The work was supported in part by INTAS grants 97-31726 and 99-00099.

Literature:

1.http://java.sun.com/products/java-media/j3d/collateral.

2.http://www.whatis.com

3.Ivanov V.P., Batrakov A.S. 3-Dimensional Computer Graphics.

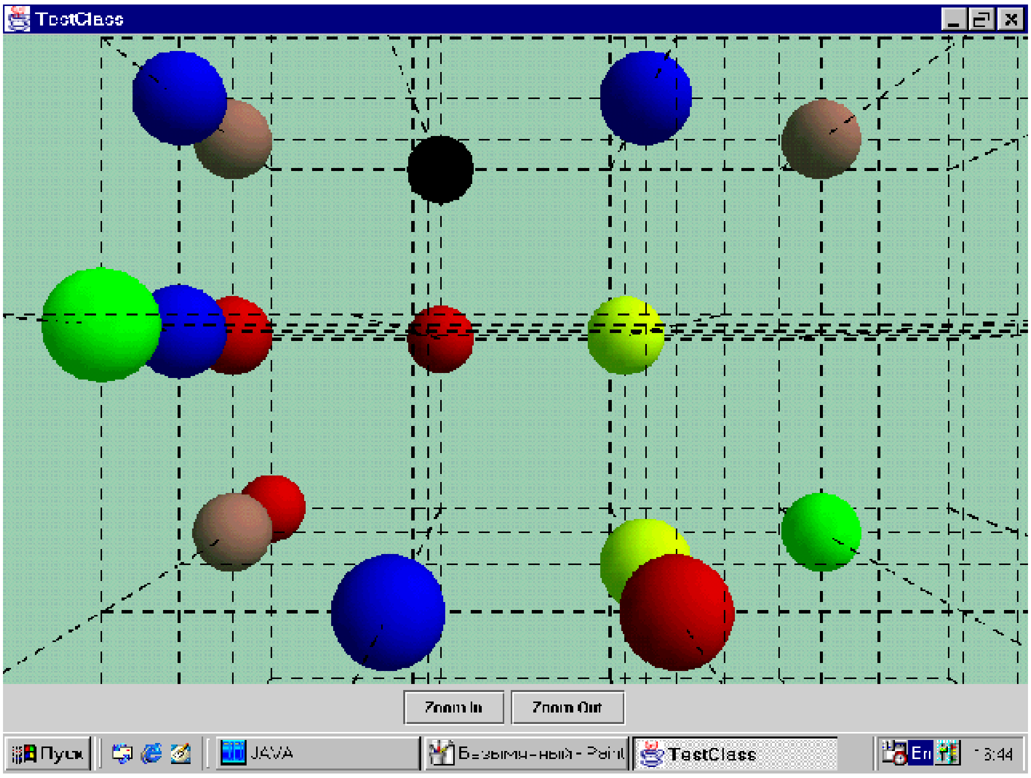
4. M. Kanevski, S. Chernov, V. Demyanov. 3Plot Software: Advanced Spatial Data Plot. Preprint IBRAE 97-02, Moscow, 1997, 32 p.

5. M. Kanevsky, S. Chernov, V. Demyanov, N. Koptelova, E. Savelieva, V. Timonin. Geostat Office: Innovative Software Tool for Spatial Data Analysis. Internal report 97-III, 1997.

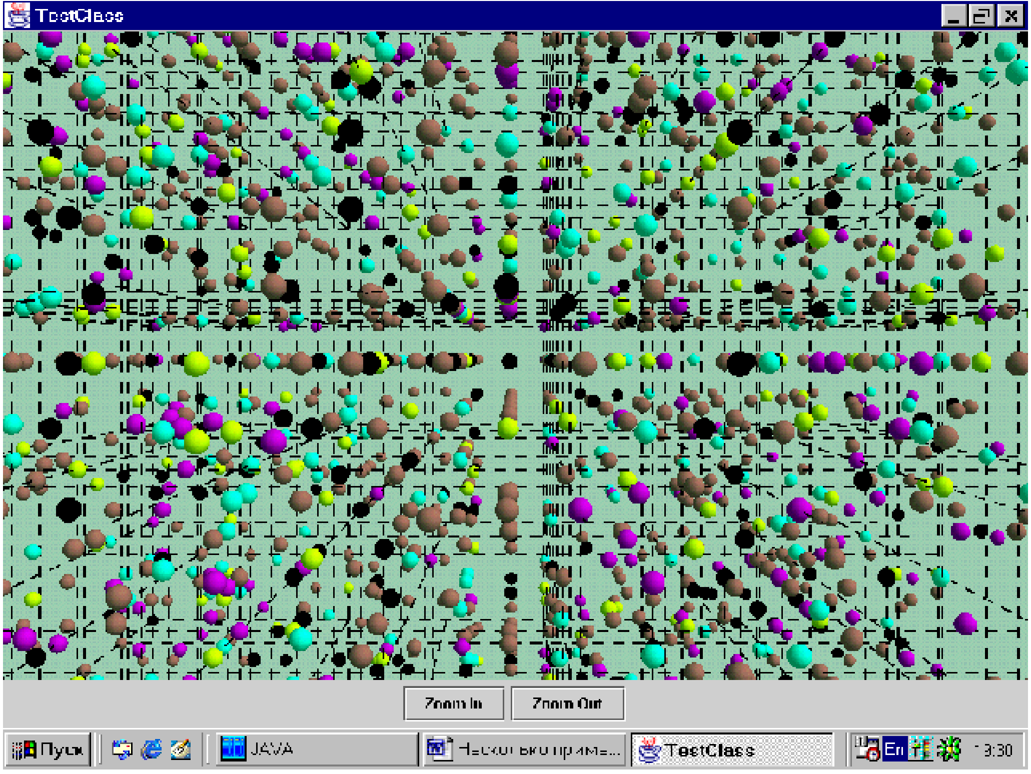
Appendix

On this figures some comparative results of this program's work are represented.

1. few objects for visualization:

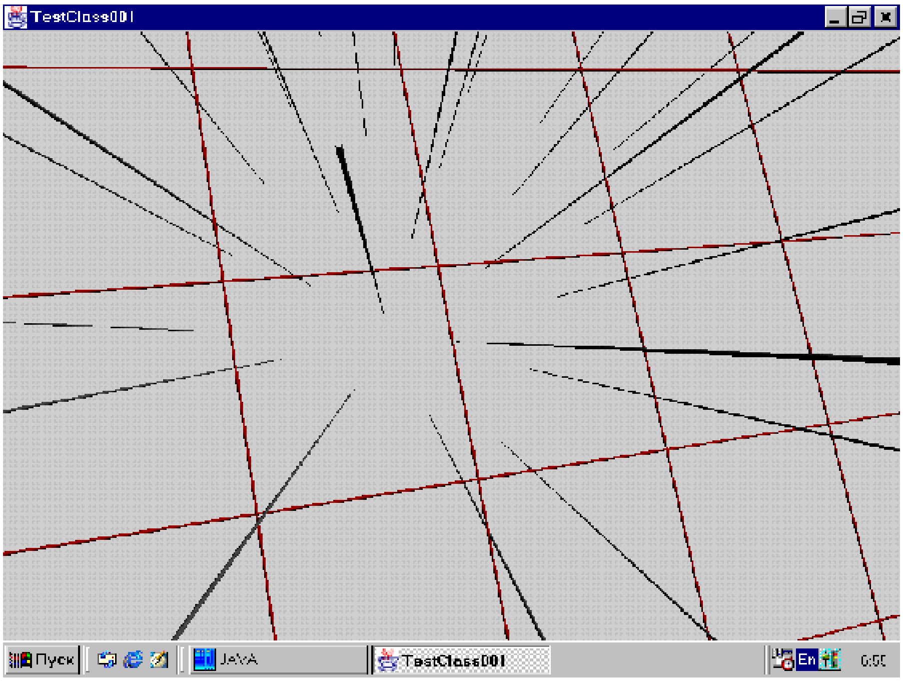


2. A lot of objects



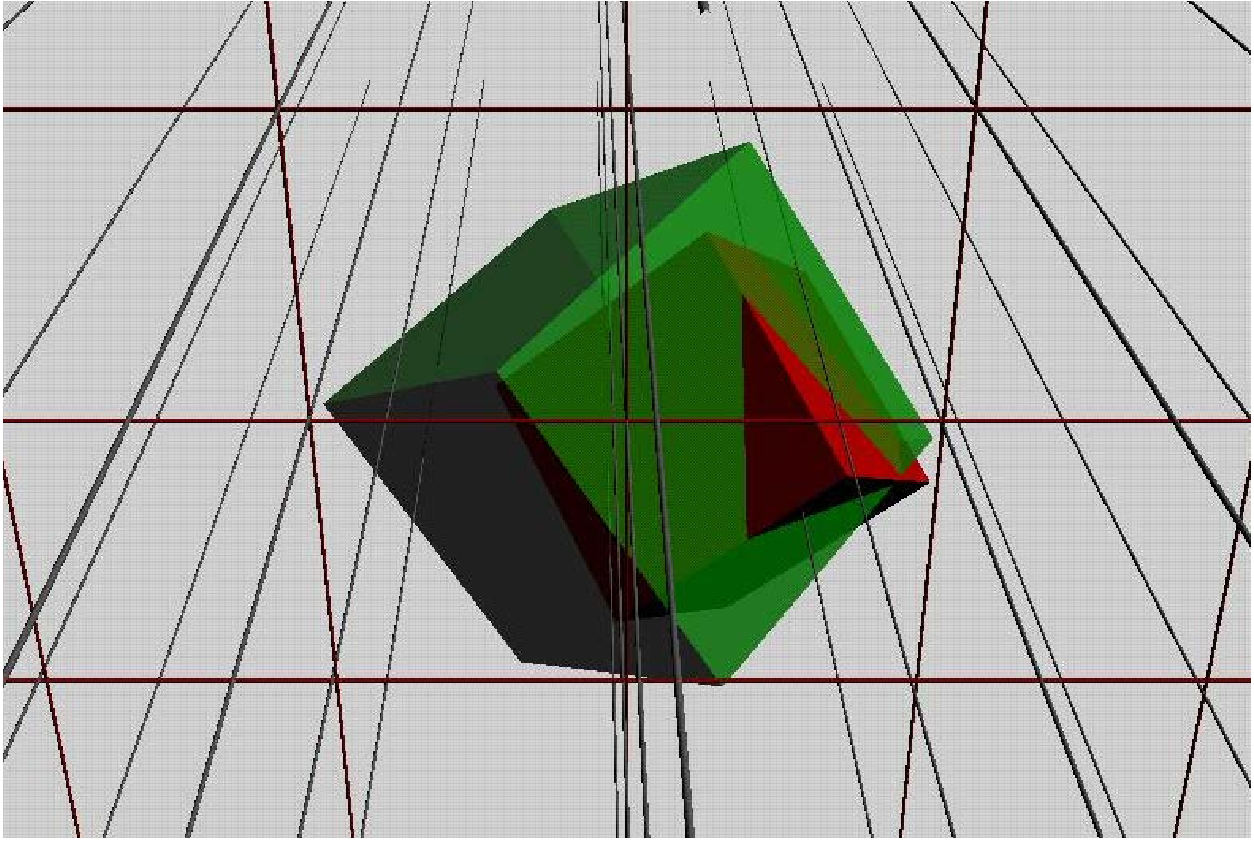
Second figure is already demonstrates some consequences of generalization when frequency of drawn grid lines is getting low, because of necessarity to reduce it for the sake of convenience of visualization. Limits of machine power exceed the sensible bound of this visualization to be normally distinguished, but this degree seems to be good enough.

As you’ve seen on previous pictures, grid was set as usual lines of one pixel fatness. Next stage was to make perspective view for grid lines: to make easy differentiation of how distant is every shown node of the grid.



Principally for high-resolution monitor this update seems to be quite convenient.

Next idea is setting transparency of objects to set effect of object inside another object.



Well, this idea seems to be worth of being worked on.

And here’s a variant of content for ini-file, which is used for delivering data to PostPlot3D:

```
[3DPostPlot]
;ini for 3D-Graphics
TABLE=3DFUN
XXID=XGRID
YYID=YGRID
ZZID=ZGRID
FFID=RAND01
LEVELS=SetLVS
COLORS=CSet_Rainbow
ATTRIBUTES=New_Attrib

[SetLVS]
LEVEL000=0.5
LEVEL001=0.55
LEVEL002=0.6
LEVEL003=0.65
LEVEL004=0.7
LEVEL005=0.75
LEVEL006=0.8
LEVEL007=0.85
;LEVEL000=0.8
;LEVEL000=0.9

[CSet_RainBow]
; RainBow color set
COLOR000=255,255,255
COLOR001=0,  0,160
COLOR002=0,  0,255
COLOR003=128,128,255
COLOR004=0,192,255
COLOR005=0,255,  0
COLOR006=192,255,  0
COLOR007=255,255,  0
COLOR008=255,192,  0
COLOR009=255,128,  0
COLOR010=255,  0,  0
COLOR011=255,  0,128
COLOR012=255,  0,255
COLOR013=255,192,255
COLOR014=224,192,255
COLOR015=192,255,255
COLOR016=255,255,255

[New_Attrib]
;attributes of net
GridColor=
GridSteps=hx:hy:hz:0.2,0.1,0.1
MarkType=Sphere
MarkSize=1,0.5
```