

RUSSIAN ACADEMY OF SCIENCES
Institute for Information
Transmission Problems

ANALYTICAL WEB GIS GeoProcessor

USER GUIDE

Moscow

2006

Summary

Web GIS GeoProcessor is intended for interactive representation, analysis and modeling of geographical information in vector and grid-based formats. The system is realized as Java program.

Note: To start the GeoProcessor 1.5 it is necessary, to use browser included a standard option of virtual Java of the machine: jre-1_5_0_06-windows-i586-p-iftw.exe.

Local version of GeoProcessor 1.5 is of importance for developing GIS projects. Installation of JDK1.5.0 or more senior version of Java in this case is required. The local mode supports all functions of the network version and allows to user save the results.

Project leader: Valeri GITIS, Dr.Sci.,

Institute for Information Transmission Problems RAS,
Section of Geoinformation Technologies and Systems
B. Karetnyi Lane 19, 127994 Moscow, GSP-4, Russia,
Phone:+7-495-2995096, Fax: +7-495-2090579, E-mail:
gitis@iitp.ru

CONTENS

1. GIS GeoProcessor 1.5 facilities	4
2. Data preparation	6
2.1.Data types	6
2.2.Data conversion with GPConverter	6
2.3.Point data formats *.PTN and *.PTS	7
2.4.External ASCII format of grid-based data *.FLT	8
2.5. External ASCII format of linear data *.LIN	9
3. Installation	10
4. Operations	14
4.1.Introduction	14
4.2.Cartography analysis	14
4.2.1. <i>Composition of the map</i>	14
4.2.2. <i>Modification of the map setup</i>	16
4.2.3. <i>Reading the grid-based values</i>	16
4.2.4. <i>Visual analysis of cross-section along the user defined profile</i>	17
4.2.5. <i>Creation of the maps of similarity</i>	17
4.2.6. <i>Modification of the color bar for grid-based data</i>	18
4.3. Analytical transformations	19
4.3.1. <i>Calculation of grid-based layers as a function of several grid-based layers</i>	19
4.3.2 <i>Calculating the illumination model</i>	21
4.3.3. <i>Grid-based filtering</i>	21
4.3.4. <i>Point to grid-based layer transformation</i>	22
4.3.5 <i>Linear to grid-based layer transformation</i>	23
4.4. Plausible inference	24
4.4.1 <i>Plausible inference method realized in GeoProcessor</i>	24
4.4.2. <i>Creation of the sample sets</i>	25
4.4.3. <i>Calculation of similarity with the sample sets of precedents</i>	26
4.4.4. <i>Calculation of membership with the sample sets for two classes of precedents</i>	26
4.4.5. <i>Calculation of empirical function of distribution</i>	26

1. GIS GeoProcessor 1.5 facilities

GeoProcessor is a web information technology for on-line cartographic exploration, intelligent modeling, and complex analysis of spatial properties of environment via Internet and Intranet.

GeoProcessor provides an access to GI distributed on web-servers as well as on user's computer.

GeoProcessor is intended for various users with wide range of requirements: from acquainting with the complex of geo-referenced data to solving sophisticated geoinformation problems.

GeoProcessor can be used for decision support in environmental monitoring, natural hazard assessment, natural and man made disaster mitigation, natural resources exploration.

The system provides the following facilities of complex GI analysis:

1. Cartography exploration of grid-based and vector data:

Interactive composition of a map consisted of several vector and grid-based layers, scaling and pan with or without interpolation of grid-based layers, changing the color bar, reading grid-based data in arbitrary points, calculation, visualization and reading the values from a set of grid-based layer cross-sections along the arbitrary profiles, illumination modeling, composition of sample sets in the form of points or/and polygons, creation of the maps of similarity with precedents according to user defined set of grid-based layers.

2. Data transformation:

Generation of a new grid-based layer applying a set of transformations to the linear and point layers, with the help of grid-based data filtering, and using the user defined functions designed of several grid based layers with a set of elementary functions, algebraic and logical operations.

3. Spatial plausible reasoning:

Estimation of empirical distribution function, estimation of similarity function with a set of precedents, estimation of similarity function based on expert knowledge, estimation of membership function for two classes.

Web-GIS GeoProcessor (<http://gis.iitp.ru/projects/geo>) is implemented as a Java 1.5 applet. Applet supports compiling GIS-projects getting the relevant geographical layers from different web-servers and user's computer.

Examples of GeoProcessor 1.5 facilities are shown at Fig.1.

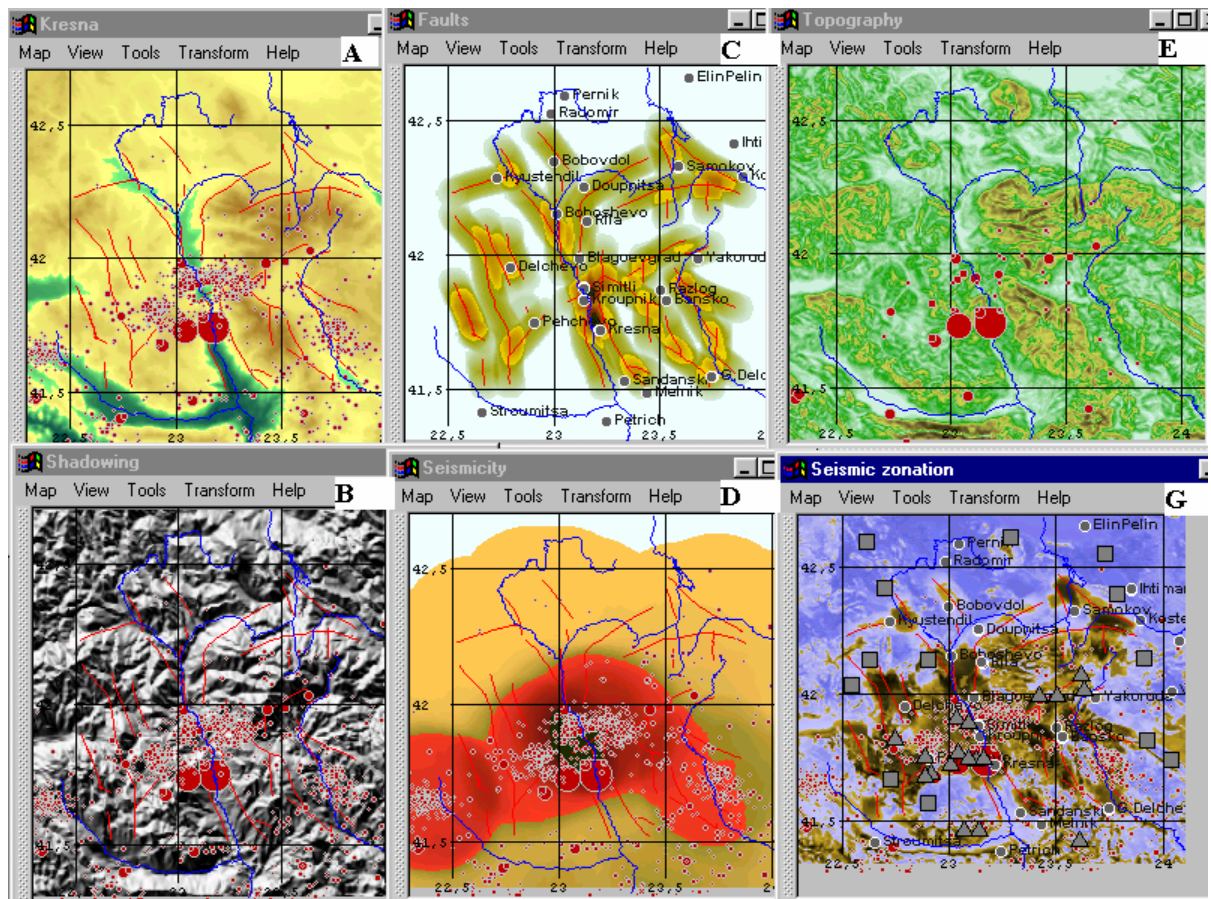


Fig. 1. Data analysis in GIS GeoProcessor for region Kresna (Bulgaria), where strong earthquake with magnitude $m=7.8$ was occurred in 4.04.1904.

A: Earth elevation and vector layers of faults, rivers, earthquakes $m>5.0$ after 1900 and $m>1.8$ after 1987.

B: Illumination model.

C: Grid-based model of density of the deep fault.

D: Grid-based model of density of earthquakes.

E: Grid-based model of absolute values of the earth elevation gradient.

F: version of seismo-tectonic zonation map estimated by the following set of features: earth elevation model, absolute values of the earth elevation gradient, density of deep faults, density of earthquakes. Brown colors correspond to high seismic potential zones and blue colors correspond to zones with low seismic potential. Triangles and squares are learning sample sets which are represent zones with high and low seismic levels.

2. Data preparation

2.1. Data types

The GeoProcessor supports operations with grid-based data, lines, marked points and strings. Data are structured as *GIS-projects*. GIS-project is a problem oriented data collection referred to the same spatial area. Each project is placed in a separate directory as a set of files of the data and metadata.

GIS GeoProcessor supports binary and textual ACSII formats. Binary formats *.BIN are used for grid-based data and lines. ASCII formats *.PTS and *.PNT are used for point data. A part of metadata is in the system files *.INI (ASCII-format). Another part of metadata, concerning to grid-based models is saved together with data in data file *.BIN.

The program GPConverter in Java is used to convert data in and off GeoProcessor formats.

2.2. Data conversion with GPConverter

JDK 1.1.8 or elder is necessary to run GPConverter. File RAN.bat is used to run program. There is a dialogue window in GPConverter (fig. 2).

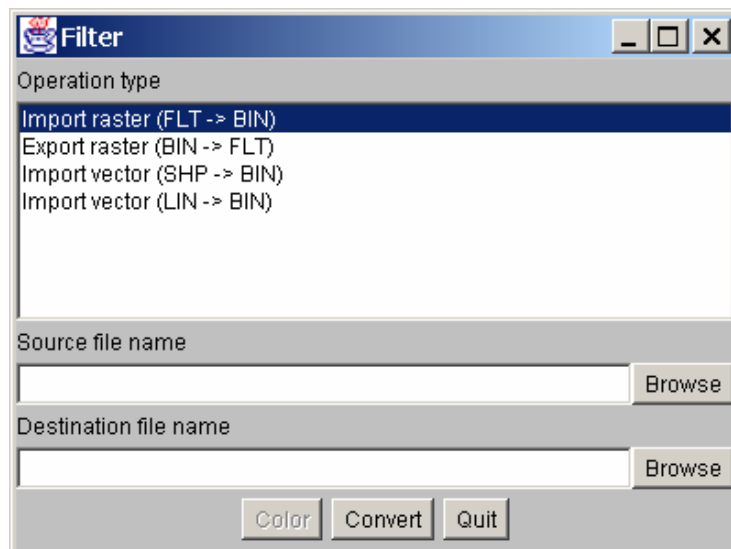


Fig. 2 GPConverter dialog window

To convert the formats you must do the following steps:

1. To select the type of conversions (subwindow “Operation type”). All types of conversions are in the following table:

Import raster (FLT ⇒ BIN)	Convert grid-based data from ASCII format of GEO (filename *.FLT) to binary format of GeoProcessor (filename *.BIN)
Export raster (BIN ⇒ FLT)	Convert grid-based data from binary format of GeoProcessor (filename *.BIN) to ASCII-format of GEO (filename *.FLT)
Import vector (SHP ⇒ BIN)	Convert vector data of ESRI Shapefile format (filename *.SHP) to binary format of GeoProcessor (filename *.BIN).
Import vector (LIN ⇒ BIN)	Convert vector data (lines) from ASCII-format of GEO (filename *.LIN) to binary format of GeoProcessor (filename *.BIN).

2. Select a source filename in subwindow «Browse».
3. Input an output filename in subwindow «Destination file name».

4. Select the color for lines (key “Color”). The option is opened for LIN and SHP input formats.
5. Click “Convert” to run conversion. When the conversion is finished subwindow inform you if «Converted succesfully» or “Wrong file format”

Click «Quit» to close the program.

Configuration files with the names *.INI are created when the files with the names *.FLT, *.SHP, *.LIN formats are converted. Moreover if the converter directory contains “root.ini” file then this file will be modified automatically (see also section 3). If there is no “root.ini” file in the directory, then this file will be created.

2.3. Point data formats *.PTN and *.PTS

Point data are saved in ASCII formats in files with names *.PTN и *.PTS. For presentation of points withb string labels the text line

<name>=<text>, <name.PTN>, < name.PTS >,,

is inserted in the section [vectors] of the root.ini (see also section 3). It is important that <text> does not contain “,”.

For example:

city=regional cities; population>100000, city.ptn, city.pts,,

Configuration file <name>.ini may be missed (see also section 3).

For presentation of point symbols without string the text line

<name>=<text>, < name.PTS >,,

is inserted in the section [vectors] of the root.ini. It is important that <text> does not contain “,”.

Configuration file <name>.ini can be missed (see section 3). It is important that <text> does not contain “,”.

For example:

events-A=all events 1964-1995; N=32541,events-A.pts,,

File of the point symbols (.PTS) without strings is a text file with white-space delimiters:*

1. Symbol longitude X
2. Symbol latitude Y
3. 0
4. Icon:
 - 0 – circle
 - 1 – string
 - 2 – square
 - 3 – triangle
 - 4 – cross
5. 0
6. Color of boundary in RGB code: red color 0-255
7. Color of boundary in RGB code: green color 0-255
8. Color of boundary in RGB code: blue color 0-255
9. 0
10. 1
11. Color of filling in RGB code: red color 0-255
12. Color of filling in RGB code: green color 0-255
13. Color of filling in RGB code: blue color 0-255
14. 0

15. Type of filling:
 - 2 – solid
 - 1 – empty
16. Size: default =1, more size if >1, less size < 1
17. Shift along X in pixels (if 0 then possible to skip)
18. Shift along Y in pixels
19. String without spaces if icon type is text (icon type in item 4 is equal to 1)

For example:

```
131.800 42.600 0 0 0 128 0 0 0 1 64 0 0 0 2 2.097 0
130.900 41.800 0 0 0 128 0 0 0 1 64 0 0 0 2 3.652 0
125.500 53.500 0 0 0 128 0 0 0 1 64 0 0 0 2 2.000 0
122.700 40.700 0 0 0 128 0 0 0 1 64 0 0 0 2 3.242 0
112.200 40.233 0 0 0 128 0 0 0 1 64 0 0 0 2 2.197 0
118.183 39.633 0 0 0 128 0 0 0 1 64 0 0 0 2 3.944 0
117.783 39.450 0 0 0 128 0 0 0 1 64 0 0 0 2 1.907 0
118.650 39.833 0 0 0 128 0 0 0 1 64 0 0 0 2 2.986 0
106.350 40.083 0 0 0 128 0 0 0 1 64 0 0 0 2 2.097 0
117.833 39.400 0 0 0 128 0 0 0 1 64 0 0 0 2 2.863 0
117.800 39.383 0 0 0 128 0 0 0 1 64 0 0 0 2 2.197 0
122.600 40.700 0 0 0 128 0 0 0 1 64 0 0 0 2 2.000 0
119.250 31.450 0 0 0 128 0 0 0 1 64 0 0 0 2 2.000 0
108.117 41.233 0 0 0 128 0 0 0 1 64 0 0 0 2 2.000 0
124.983 40.267 0 0 0 128 0 0 0 1 64 0 0 0 2 1.728 0
```

File of strings attached to points (*.PTN) is a text file with white-space delimiters:

1. Longitude X
2. Latitude Y
3. 0
4. 0
5. Size (default 1)
6. Color in RGB code: red color 0-255
7. Color in RGB code: green color 0-255
8. Color in RGB code: blue color 0-255
9. 0
10. 0
11. 0
12. Shift refer to the anchor point along X in pixels
13. Shift refer to the anchor point along Y in pixels
14. String without spaces if icon type is text

For example:

```
76.917 43.200 0 0 1 0 0 64 0 0 0 3 3 Almaty
71.333 42.867 0 0 1 0 0 64 0 0 0 3 3 Taraz
78.400 45.000 0 0 1 0 0 64 0 0 0 3 3 Taldy-Kurgan
69.600 42.267 0 0 1 0 0 64 0 0 0 3 3 Shimkent
```

2.4. External ASCII format of grid-based data *.FLT

Format *.FLT is the external format of grid based data. It is converted in internal grid-based format with help of GPConverter. File *.FLT consists on 3 parts: *header*, *data*, and *data description*.

Header is a line with 8 parameters::

```
<COL> <ROW> <Xbeg> <Ybeg> <Dx> <Dy> <Intr> <Geog>
```


COL	is number of columns.
ROW	is number of rows.
Xbeg	is longitude of the left and upper note of the grid.
Ybeg	is latitude the left and upper note of the grid.
Dx	is distance between adjacent grid columns.
Dy	is distance between adjacent grid rows.
Intr=1	if we want to interpolate the grid values to the display pixels, and Intr=1 if do not interpolate.
Geog=1	if the data of the GIS project are given in geographical coordinates, and Geog=1 if they are in Cartesian.

Data are the values in ASCII with white space or new line delimiters. The 1st value is attached to the grid knot with the coordinates Xbeg, Ybeg, afte that are the values attached to the coordinates (Xbeg+1, Ybeg), (Xbeg+2, Ybeg), ... (Xbeg+COL, Ybeg), (Xbeg+1, Ybeg+1), (Xbeg+2, Ybeg+1) etc. Number of values is equal to COL*ROW. Dummy values are marked by -32767.

Data description is any text comments after the *data* area and starting with blank row delimiter.

2.5. External ASCII format of linear data *.LIN

File consists of the lines presented by the coordinates of joint points of the line. Each line has a *header* and *data*. *Header* is a row: <L>, space delimiter, <n>, (L is a descriptor, n is number of join point in the line). *Data* consist of n rows, each of which contains the coordinates of one join point.

Note: each row in *.LIN format begins without space.

For example: two lines in *.LIN format:

```
L 2
11.2 42.41
11.4 43.5
L 3
20.4 51.0
22.1 51
23.01 52.34
```

3. System installation

The simplest way to install the system is to put the program GeoProcessor and data at the same directory. Example of directory is at Fig. 3

Name	↑Ext	Size	Date	Attr
⚙️[.]		<DIR>	15.05.2006 05:51	—
📁[china-2006]		<DIR>	15.05.2006 05:45	—
📁[GIF]		<DIR>	15.05.2006 05:45	—
📁[images]		<DIR>	15.05.2006 08:54	—
📁[nchi]		<DIR>	15.05.2006 13:08	—
📄China-2006	bat	113	15.05.2006 05:38	-a-
📄china-local	html	25 494	15.05.2006 11:50	-a-
📄help	html	25 623	19.01.2000 14:28	-a-
📄nchi	html	813	15.05.2006 13:18	-a-
📄geoprocessor1.5	jar	351 330	07.02.2006 17:46	-a-

Fig. 3. Directory with the projects and GeoProcessor:

[China-2006] and [nchi] are the directories of the project, [gif] and [images] are the directories of system files, geoprocessor1.5.jar is the program GeoProcessor 1.5, help.html is help file, files*.bat и *.html are the files to run projects.

An example of GIS-project is at the fig. 4.

Name	↑Ext	Size	Date	Attr
⚙️[.]		<DIR>	15.05.2006 05:45	—
📁analys	BIN	2 131	03.07.1999 07:20	-a-
📁C-COAST	bin	189 842	02.05.2006 16:37	-a-
📁china	bin	7 881 992	04.05.2006 18:26	-a-
📁CHINA1\$	bin	2 346 181	16.05.2006 07:28	-a-
📁C-LAKE	bin	41 520	02.05.2006 16:37	-a-
📁C-NATION	bin	122 066	02.05.2006 16:40	-a-
📁C-RIVER	bin	167 979	02.05.2006 16:38	-a-
📁DEFAULT	BIN	1 173	29.06.2001 01:17	-a-
📁F_n	bin	192 994	01.05.2006 19:11	-a-
📁F_q	bin	395 356	01.05.2006 20:39	-a-
📁F_q3	bin	39 569	01.05.2006 21:27	-a-
📁F_q4	bin	411 911	01.05.2006 21:28	-a-
📁F_s	bin	52 526	01.05.2006 21:28	-a-
📁Field	BIN	4 677 496	16.05.2006 07:28	-a-
📄C-COAST	ini	274	02.05.2006 16:37	-a-
📄china\$	INI	827	16.05.2006 07:28	-a-
📄china	ini	302	15.05.2006 11:33	-a-
📄CHINA1\$	ini	299	05.05.2006 15:34	-a-
📄C-LAKE	ini	269	02.05.2006 16:37	-a-
📄C-NATION	ini	278	02.05.2006 16:40	-a-
📄C-RIVER	ini	275	02.05.2006 16:38	-a-
📄F_n	ini	284	01.05.2006 23:36	-a-
📄F_n1	INI	835	16.05.2006 07:28	-a-
📄F_q	ini	272	01.05.2006 23:37	-a-
📄F_q3	ini	276	01.05.2006 23:38	-a-
📄F_q4	ini	276	01.05.2006 23:38	-a-
📄F_s	ini	272	01.05.2006 23:38	-a-
📄root	ini	811	16.05.2006 07:28	-a-
📄MAINCIT1	PTN	798	02.08.2005 15:40	-a-
📄0-6	PTS	6 064 048	04.05.2006 20:17	-a-
📄6-8	PTS	1 676	04.05.2006 20:03	-a-
📄MAINCIT1	PTS	1 062	16.05.2006 05:55	-a-

Fig. 4. Directory of GIS project China-2006

Files analys.bin and deault.bin are the system files. Files *.bin, *.ptn, *.pts are the initial data files.. Files *.ini are configuration files: file root.ini is config-file of the GIS-project. Another files *.ini are config-files of data. These files are generated and modified

automatically by GPCConverter. (It is possible that the project includes the files field.bin., field1.bin ect., which are contained the grid-based layers created by GeoProcessor and saved in local regime).

Let us consider the configuration file magn.ini:

```
[region]
Name=
Projection=0,0
Coords=73.1,43.3,76.0,42.0
Grid=Coordinate grid,true,50,50,50,SansSerif,1,6,73.0,76.0,0.75,43.0,42.0,-0.25
```

```
[order]
MAGN=true
Grid=true
```

```
[vectors]
```

```
[fields]
MAGN=MAGN,MAGN.BIN,null,null
```

```
[samples]
```

```
[areas]
[areas]
[region]
Name=
Projection=0,0
Coords=73.1,43.3,76.0,42.0
Grid=Coordinate grid,true,50,50,50,SansSerif,1,6,73.0,76.0,0.75,43.0,42.0,-0.25
```

```
[order]
MAGN=true
Grid=true
```

```
[vectors]
```

```
[fields]
MAGN=MAGN,MAGN.BIN,null,null
```

```
[samples]
```

```
[areas]
```

The file content is divided by several parts:

[region]
Name= - name of the region
Projection=0,0 - always 0,0
Coords=73.1,43.3,76.0,42.0 - coordinates of the area under analysis: minimal longitude, maximal latitude, minimal latitude, maximal longitude
Grid=Coordinate grid,true,50,50,50,SansSerif,1,6,73.0,76.0,0.75,43.0,42.0,-0.25
- parameters of coordinate grid: name, color, font, minimal and maximal values of longitudes, step along longitudes, maximal and minimal values of latitudes, step along latitudes (sign minus because latitudes are decrease)

[order] - the order of layers in the map, true: screen after loading the data, false - not screen,
MAGN=true
Grid=true

[vectors] - list of the vector layers

[fields] - list of the grid layers

MAGN=MAGN,MAGN.BIN,null,null

[samples] - list of the sample point sets

[areas] - list of the sample area sets

Example: boug_gr.ini (this is map configuration file):

```
[region]
Name=
Projection=0,0
Coords=73.1,43.3,76.0,42.0
Grid=Coordinate grid,true,50,50,50,SansSerif,1,6,72.0,76.0,0.25,43.0,42.0,-0.25
Grid=Coordinate grid,true,50,50,50,SansSerif,1,6,72.0,76.0,0.25,43.0,42.0,-0.25
[order]
boug_gr=false
TOPO$=false
Field=false
MAGN=true
Field1=false
e94-04=true
ELECTR-daily=false
electro-4-5PerMonth=false
gps=false
gps_add=false
Grid=true
[vectors]
e94-04=Bishkek: 2734 events of 1994-2003,e94-03.pts,,
ELECTR-daily=ELECTRO stations daily,ELECTR-daily.ptn,ELECTR-daily.pts,,
electro-4-5PerMonth=ELECTRO stations 4-5 vper month,electro-4-5PerMonth.ptn,electro-4-5PerMonth.pts,,
```

```

gps=gps,gps.ptn,gps.pts,,
gps_add=gps_add,gps_add.ptn,gps_add.pts,,
[fields]
boug_gr=BOUG_GR,BOUG_GR.BIN,null,null
Field=Smoothed BOUG_GR,Field.BIN,Field.BIN,Field.GRA
MAGN=MAGN,MAGN.BIN,null,null
TOPO$=TOPO$,TOPO$.BIN,null,null
Field1=Model for TOPO$,Field1.BIN,Field1.BIN,gray.gra
[samples]
[areas]

```

Sometime file root.ini must be modified by hand. Consider the content of root.ini:

```
[order]
```

```

[fields]
china=chines elevation model,china.BIN,null,null
china$=china$,china1$.BIN,null,null
Field=Gradient of china$,Field.BIN,Field.BIN,Field.GRA

```

```
[areas]
```

```
[samples]
```

```

[vectors]
F_n=F_n,F_n.BIN,,,
F_q=F_q,F_q.BIN,,,
F_q3=F_q3,F_q3.BIN,,,
F_q4=F_q4,F_q4.BIN,,,
F_s=F_s,F_s.BIN,,,
C-COAST=C-COAST,C-COAST.BIN,,,
C-LAKE=C-LAKE,C-LAKE.BIN,,,
C-RIVER=C-RIVER,C-RIVER.BIN,,,
C-NATION=C-NATION,C-NATION.BIN, ,
6-8=31 events 6.0<=m<=8.0 in 1970-2005,6-8.PTS,,,
0-6=114416 events 2<=m<6.0 in 1970-2005,0-6.PTS,,,
CITY=Main cities,MAINCIT1.PTN,MAINCIT1.PTS,,

```

```
[region]
```

```

Name=
Projection=0,0
Coords=72.05,54.9716,131.92340000000002,15.054800000000007
Grid=Coordinate grid,true,50,50,50,SansSerif,1,6,75.0,135.0,10.0,55.0,5.0,-5.0

```

```
[maps]
```

```

F_n1=Faults
china$=North-East China

```

We can see that structure of root.ini is similar to the structure of another *.ini files. However there are two principal differences:

1. All project data must be described in root.ini .
2. There is a section [maps] in the root.ini, in which all multilayer maps are presented.

4. Operations of GIS GeoProcessor

4.1. Introduction

There are three groups of operations in GeoProcessor: *cartography analysis*, *analytical transformations*, *plausible inference*.

1. *Cartography analysis*

1. Map composition.
2. Modification of map setup.
3. Reading the grid based values.
4. visual analysis of cross-sections along the user defined profile.
5. Creation of the maps of similarity.
6. Modification of color bar for grid data.

2. *Analytical transformations*

1. Calculating the grid layer with the help of user defined function of another grid layers with use elementary functions, logical and algebraic operations .
2. Creating the illumination model.
3. Grid based filtering .
4. Transformation of point layer to grid layer.
5. Transformation of linear layer to grid layer.

3. *Plausible inference*

1. Creating the sample sets.
2. Estimation of similarity functions.
3. Estimation of membership functions for two classes.
4. Estimation of empirical distribution function.

4.2. Cartography analysis

4.2.1. *Composition of the map*

The map consists of several grid and vector based layers and coordinate grid of the area under consideration. GeoProcessor can process grid-based data in regular but different grids: with different steps or with shifted grid knots. GeoProcessor automatically makes bilinear interpolation of the grids data to the screen pixels (for cartography representation) or to the grid with minimal steps (for operations with several grid layers). User can add, delete and move any of layers.

After loading GIS project metadata the Window with the list of project maps is open (Fig. 5).

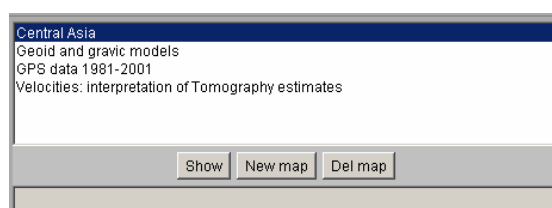


Fig. 5. Window with the list of project maps

Mark the map name by cursor and click the button **Show** to activate the map. Mark the map name and click the button **Del map** to delete a map. Click the button **New map** to create a new multilayer map. After that window **Create new map** appears (Fig 6).

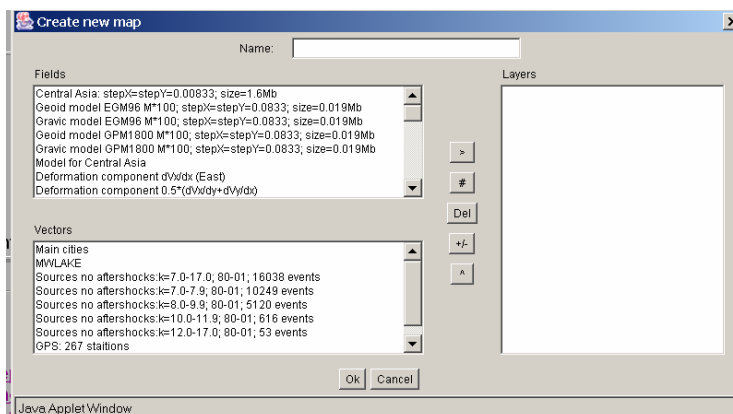


Fig. 6. Window Create new map

Upper panel **Name** is used for writing the name of the map. Each line of the left-up window **Field** is the name of grid based layer. Each line of the left-down window **Vector** is the name of vector based layer. Right window **Layers** represents selected layer. The layers screen in the same order as they are following at the window.

There is a vertical Panel of map composition in the middle of the window (Fig. 7). Five buttons of the panel control the multilayer map composition:

Buttons of the panel	
>	Add layer
#	Add coordinate grid
Del	Delete layer
+/-	screen/nor screen layer
^	Move the layer up

Fig. 7. Panel of map composition

Compiling the map.

1. Input the name of the map in **Name**.
2. Mark layers in **Fields**.
3. Mark layers in **Vectors**.
4. Move marked names of data to **Layers** with help of the button «>».
5. Use the button «^» and «Del» to delete the layers from the list of layer names.
6. Use the button «+/-»: «+» to select active layers, which will be screened after loading the project, and «-» for and nonactive ones.
7. Use the button «#» to screen coordinate grid.
8. Use the button «OK» to save composed map (Fig. 8).

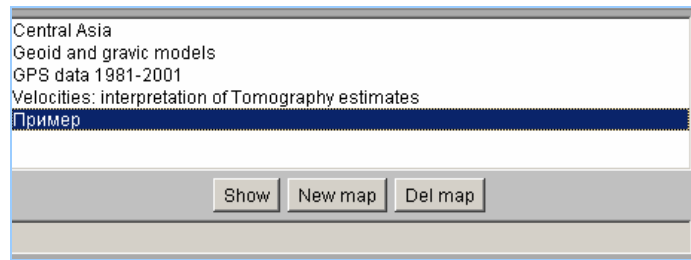


Fig. 8. Window of project maps cwith new map *Пример*.

4.2.2. Modification of the map setup

Mark the map and click button *Show* to start the project. Main control window appears (fig. 9).

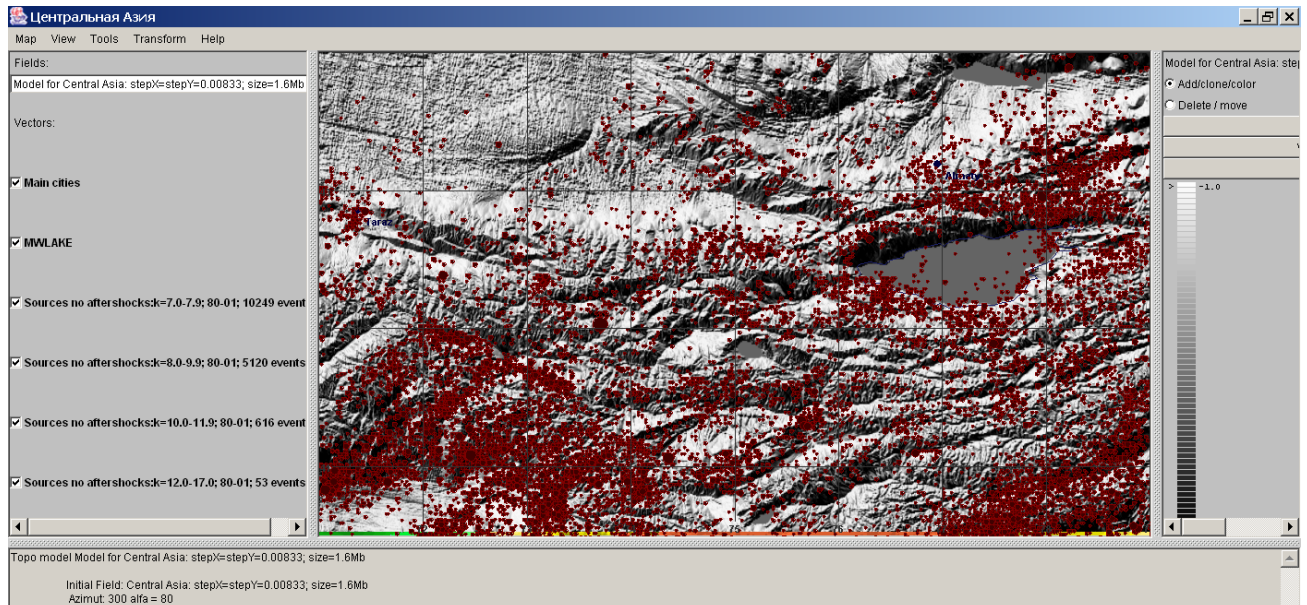


Рис. 9. Main control window

You can modify scale, coordinates of the considered region and coordinate grid.

Enter option *Map > Compose* to change a map (see also section 4.2.1).

Enter option *View* and click *Zoom In*, *Zoom Out* or *Zoom to Fit* to zoom a map. In *Zoom In* map image increase without interpolation of grid layer. Use option *View > Refresh* to interpolate grid layer.

Use option *View > Region* to modify boundaries of the region.

Use option *View > Grid* to modify coordinate grid.

4.2.3. Reading the grid based vales

Use option *Tools > Values* to read the values of all grid layers. Enter the option *Tools > Values* and click right mouse button at the point on study. The values appear in window *Values in a Point* (fig. 10).

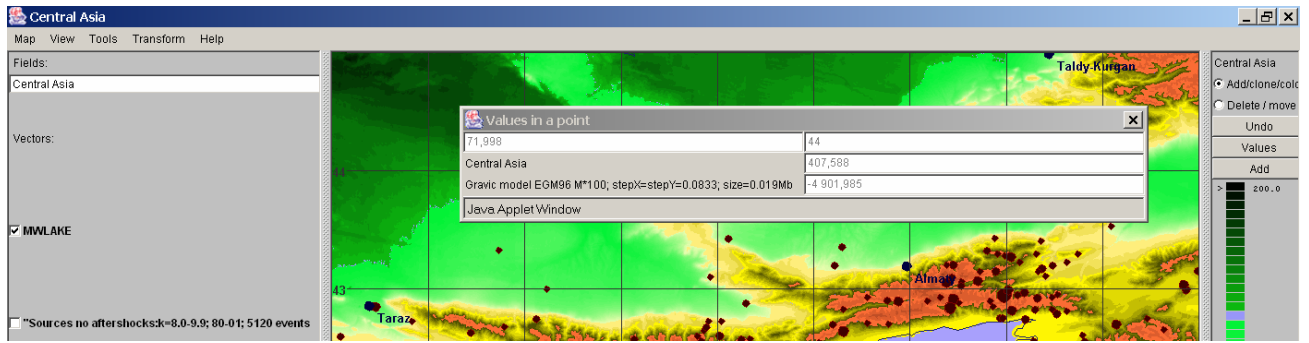


Рис. 10. Window *Values in a Point*.

4.2.4. Visual analysis of cross-sections along the user defined profile

Use option *Tools > Cross-section* to calculate cross-section of several grid layers. Enter the option, put cursor in the map and click left mouse button. After that put cursor at the 1st point of the profile which you are going to consider, push left mouse button and drag cursor to the next point of the profile, release the button, push the button again and drag it to the next point etc. The last point of the profile enter using double click of left mouse button. After this window *Cross-section* appears (fig. 11). Each graphic is normalized according to the window high. If cursor moved along the upper part of window *Cross-section*, then the table with coordinates and grid values along the profile is presented in the window underframe. At the same time a cross is moving along the profile to show the point where grid values are reading.

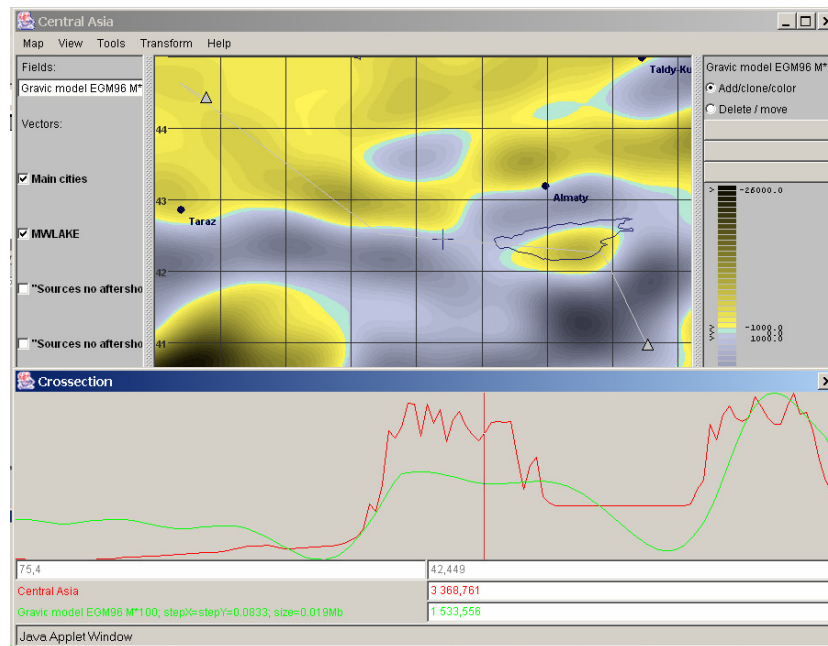


Fig 11. Cross-section of two grid layers.

4.2.5. Creation of the maps of similarity

We can consider a point of a map as a point in geographical space and as a point in the multidimensional space of thematic properties presented by grid-based layers (multidimensional space of features). Similarity function is calculated in space of the features which is normalized by the feature standard deviations. It is clear, that in normalized feature space standard deviations of features x_i are equal to 1. Let us $\rho(\mathbf{x}, \mathbf{x}^{(n)})$ is metrics, R is a constant, $\{\mathbf{x}^{(n)}\}$ are region point selected as precedents to compare them with all another points of the region with a set of features. Similarity function is given by:

$$S(\mathbf{x}, \{\mathbf{x}^{(n)}\}) = 1 - \frac{\min_{\langle n \rangle} \rho(\mathbf{x}, \mathbf{x}^{(n)})}{R} \quad \text{for } \min_{\langle n \rangle} \rho(\mathbf{x}, \mathbf{x}^{(n)}) \leq R.$$

GeoProcessor can use the metric l_2 or c . As well known a distance in l_2 is given by $\rho(\mathbf{x}, \mathbf{x}^{(n)}) = (\sum_{i=1}^l (x_i - x_i^{(n)})^2)^{1/2}$, and in metric c a distance is given by $\rho(\mathbf{x}, \mathbf{x}^{(n)}) = \max_{\langle i \rangle} |x_i - x_i^{(n)}|$.

Option **Tools > Analysis** is used to calculate similarity function. After entering the option you must select parameters in window **Similarity Analysis**: metrics l_2 or c , constant R , and a set of features. Use **OK** to close the window. Use double left mouse button click to mark a point selected as precedent. This point is marked by blue triangle. Spatial distribution of similarity is presented in gray color bar. The more values of the similarity function correspond more dark colors. Use the second left mouse button double click to mark the next precedent point. Use right mouse button click to take similarity map off the screen. To exit the operation you must click of flag in option **Tools > Analysis**.

4.2.6. Modification of color bar for grid data.

Color bar and its control buttons are located in the right part of the main control window (Fig. 12). The name of the active grid layer is at the upper part of the window. The control buttons and color bar are bellow.

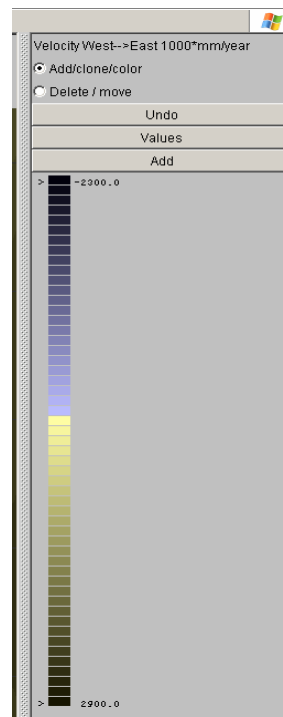


Fig. 12 Color bar

Color bar is divided on intervals. The value near the color rectangular gives upper level of the interval. The first color of the bar refer to the grid layer values within the interval between minimal and given in color bar value (this value is -2300 at fig 12). Last color of the bar refer to the interval between penultimate and maximal value (this value is 2900 at fig 12). The values of intervals illuminated automatically when cursor moving along the color bar. To modify color put cursor beside the color rectangular and click left mouse button. All colors will be change after that. To restore the color bar click button *Undo*. Use double left mouse button click to call window *Choose color*. Select color, color depth and click botton *Choose this color*. For cancellation click botton *Return to this color*. Enter option **View > Refresh** to change filling of grid layer according to a new color bar. Use the buttons **Add/clone/color** and **Delete/move** to select regime of color bar modification.

Add/clone/color regime.

The button **Values** is used to install the values of a new color scale. After clicking it the dialog window **Input new values** is appeared (fig. 13). You must enter parameters of the 1st subinterval of the scale in the panels: **Min** (minimum of subinterval), **Max** (maximum of subinterval), **Delta** (scale step in subinterval). After that click the button **Add**, which call the window **Input new values**, and input values **Min**, **Max** и **Delta** for the next subinterval etc.

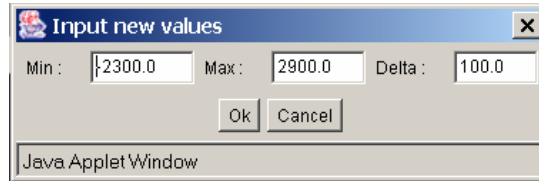


Fig. 13. Window Input new values

Delete/move regime.

Colors may change fluently inside subintervals and stepwise on the scale boundary In this regime you can change scale boundaries. Use button **Undo** to reset scale.

4.3. Analytical transformations

4.3.1. Calculating the grid-based layer as a function of several grid-based layers

GIS GeoProcessor 1.5 supports calculating the grid layer with the help of user defined function of another grid layers with use elementary functions, logical and algebraic operations

Their list of GeoProcessor elementary functions is given bellow:

Elementary functions	
ABS()	Absolute value of grid layer data (grid layer descriptor is in the parenthesis)
ATAN()	Arctangent of grid layer data (grid layer descriptor is in the parenthesis)
BLK()	$BLK(x)=1$, if x is defined, $BLK(x)=0$ otherwise (grid layer descriptor is in the parenthesis)
COS()	Cosine of grid layer data (grid layer descriptor is in the parenthesis)
DAY(year, month, day)	Number of days from 1970.01.01
EXP()	Exponent of grid layer data (grid layer descriptor is in the parenthesis)
IF(a, b, c)	If logical expression a is true, then the result is b , c in otherwise. E.g., expression $IF(\$1>0, \$1-\$2, -32767)$ means, that if the values of the grid layer $\$1>0$, then a new grid layer values are equal to differences between the values of the grid layers $\$1$ and $\$2$, else the result is -32767 (dummy value)
INTR(x, x₁, y₁, [,x₂, y₂,...])	Values of the layer defined as peace-wise function which is given by the notes (x_i, y_i) . E.g, expression $INTR(\$1,20,-30,37,99,41,-50)$:

	Grid layer values \$1 are the arguments, if the value <20 then the result value is -30, if values from 20 till 37 then output values increase linearly from -30 till 99, if values from 37 till 41 then output values decrease linearly from 41 till -50, if values >41 then the result value is -50.
LN()	Natural logarithm of grid layer data (grid layer descriptor is in the parenthesis)
LG()	Decimal logarithm of grid layer data (grid layer descriptor is in the parenthesis)
MAX(,)	Maximum of grid layer values (grid layer descriptors are in the parenthesis)
MIN(,)	Minimum of grid layer data (grid layer descriptors are in the parenthesis)
RAND()	Result is the value of input grid layer plus random noise (grid layer descriptor is in the parenthesis)
ROUND()	Round-up of grid layer data (grid layer descriptor is in the parenthesis)
SIN()	Sinus of grid layer data (grid layer descriptor is in the parenthesis)
SQRT()	Square root of grid layer data (grid layer descriptor is in the parenthesis)

To calculate a new grid layer enter option *Transform > Field combination*. Window *Input Function* is appeared (fig.14). Row *Name* is for the name of output lauer. Use *Functions*, *Operators* and *Fields* to define the function. Click button *OK* to calculate the result.

Note: if layers are given in different coordinate grid then they are interpolated automatically to the more details grid. Bilinear interpolation is used for this aim.

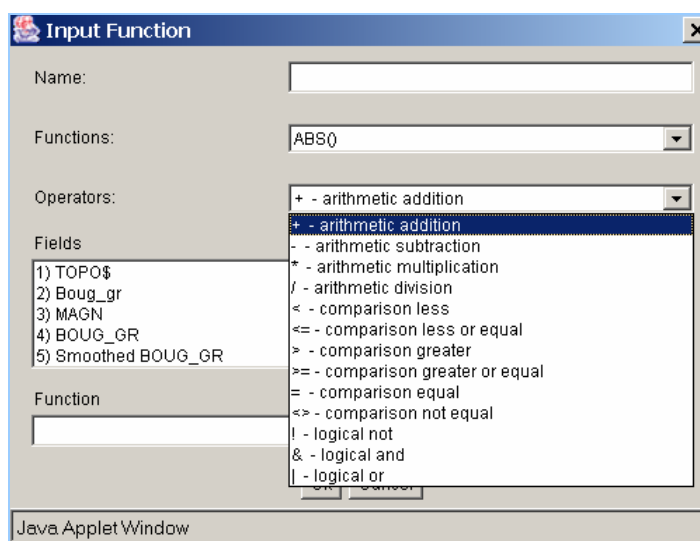


Рис.14. Window *Input Function*

4.3.2. Creating the illumination model

Use option *Transform > Topo model* to create the illumination model of the active layer. Input parameters *azimuth*, *angle* (angle with horizon) and *factor* (scale factor for the grid based values). Click *<Preview>* to look the result in cartography window, click *<OK>* to save the result, click *<Cancel>* to exit without saving.

4.3.3. Grid based filtering

Use option *Transform > Filter* for grid based filtering. Dialog window *Setup filter* is shown at fig. 15. To run the filtering select the layer to be transformed in the row *field*, input radius of moving window in row *R*, select filter: *mean, median, RMS, local anomalies, gradient module, gradient azimuth, maximum, minimum, maximum-minimum*, input the name of output layer in row *Name*. Click *<Preview>* to look the result in cartography window, click *<OK>* to save result, click *<Cancel>* to exit without saving.

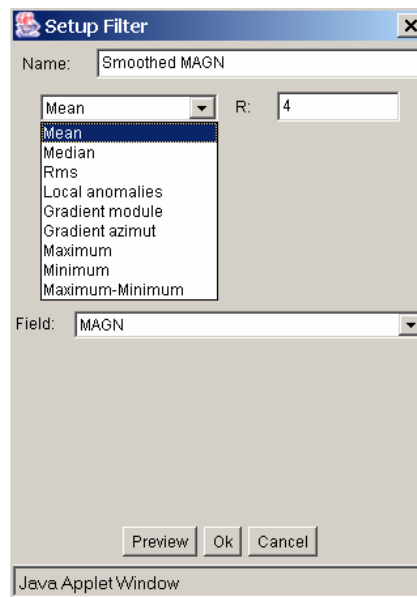


Рис. 15. Window *Setup Filter*

Let us consider filtering operations.

Mean is smoothing. Output value y_j at a grid point j is equal to average of the input grid values x_i within moving square window with size $2R$ in km.

$$y_j = \frac{\sum_i x_i \Delta_i}{\sum_i \Delta_i},$$

where summation is made for all grid cells which are crossed with moving window, Δ_i is a size of a part of the cell (with the grid knot i), which is covered by j moving window, x_i – value in grid knot i .

Median is median smoothing. Output value y_j at a grid point j is equal to median of the input grid values x_i within moving square window with size $2R$ in km.

RMS is calculating the grid layer of standard deviations. Output value RMS_j at a grid point j is equal to standard deviation of the input grid values x_i within moving square window with size $2R$.

$$RMS_j = \frac{\sum_i (x_i - y_j)^2 \Delta_i}{\sum_i \Delta_i}$$

Local anomalies is trend removing. Output value **Local anomalies_j** at a grid point *j* is equal to standard deviation of the input grid values *x_i* within moving square window with size 2*R*.

Gradient module is calculating the absolute value of the gradient.

$$\text{Gradient module}(i) = (A^2 + B^2)^{1/2}$$

where

	<i>i4</i>	
<i>i1</i>	<i>I</i>	<i>i2</i>
	<i>i3</i>	

$$A = (x_{i2} - x_{i1}) / (i2 - i1), \quad B = (x_{i4} - x_{i3}) / (i4 - i3),$$

(*i2 - i1*) and (*i4 - i3*) are distances between grid knots (*i2* and *i1*) and (*i4* and *i3*) in km.

Gradient azimuth is calculating the azimuth of the gradient:

$$\text{Azimut}(i) = A/B, \text{ where}$$

where the notations are given before.

Maximum is selection of maximal value within moving window. Output value **Maximum_j** at a grid point *j* is equal to maximal value of the input grid values *x_i* within moving square window with size 2*R*.

Minimum is selection of minimal value within moving window. Output value **Minimum_j** at a grid point *j* is equal to minimal value of the input grid values *x_i* within moving square window with size 2*R*.

Maximum - Minimum: is calculating difference between **Maximum** and **Minimum**. Output value (**Maximum - Minimum**)_{*j*} at a grid point *j* is equal to (maximal value - minimal value) of the input grid values *x_i* within moving square window with size 2*R*.

4.3.4. Point to grid-based layer transformation.

Use option **Transform > Event filter** to transform point data to grid layer. Dialog window **Setup Event filter** is shown at fig 16. To run point-grid transformation select the point layer to be transformed in the row **vector**, input radius of moving window in row **R**, select filter: **Density, Distance, Closeness, Influence**, input the name of output layer in row **Name**. Click **<Preview>** to look the result in cartography window, click **<OK>** to save the result, click **<Cancel>** to exit without saving

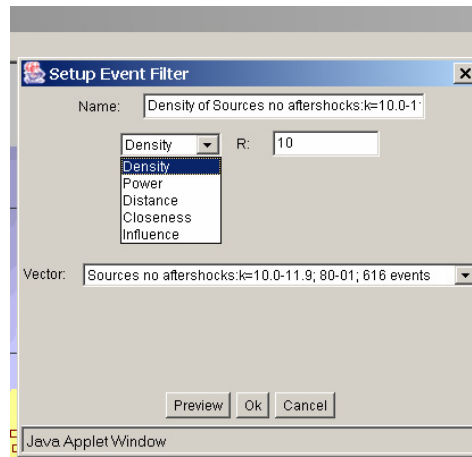


Fig. 16. Window *Setup Event Filter*.

Consider the list of operations.

Density : calculation of grid layer of point density within circle moving window with radius R in km. Density of points in the knot j is equal to ratio of number of points within the moving window j to $3.14 * R^2$.

Distance : calculation of grid layer of Euclid distances ρ_{in} between the grid knot i and nearness point n within circle window with radius R km. If there are no points within the window the distance is equal to R .

Closeness : calculation of grid layer of closeness which is equal $S_i = 1 - \frac{\rho_{in}}{R}$ if $\rho_{in} < R$, and $S_i = 0$ if $\rho_{in} \leq R$.

Influence: calculation of grid layer of influence, which is equal to $influence_i = \sum_n (1 - \rho_{in}/R)$.

4.3.5. Linear to grid-based layer transformation

Use option *Transform > Line filter* to transform linear layer to grid layer. Dialog window *Setup Line Filter* is presented at fig 17. To run line-grid transformation select the line layer to be transformed in the row *Vector* , input radius of moving window in row R , select filter: *Distance, Influence, Nearness, Number, Sum length*, input the name of output layer in row *Name*. Click *<Preview>* to look the result in cartography window, click *<OK>* to save the result, click *<Cancel>* to exit without saving.

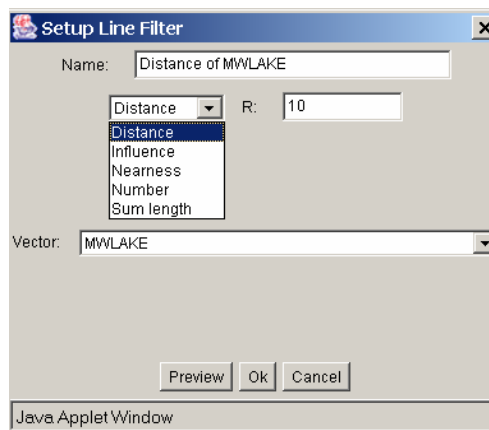


Fig. 17. Window *Setup Line Filter*

Consider the list of operations.

Distance : calculation of grid layer of Euclid distances ρ_{in} between the grid knot i and nearness line n within the circle window with radius R km. If there are no points within the window the distance is equal to R (km).

Closeness : calculation of grid layer of closeness to the lines which is equal $S_i = 1 - \frac{\rho_{in}}{R}$ if $\rho_{in} < R$, and $S_i = 0$ if $\rho_{in} \leq R$.

Influence: calculation of grid layer of influence, which is equal to $influence_i = \sum_n (1 - \rho_{in}/R)$.

Number – calculation of grid layer of density of lines, which is equal the ration of the number of lines crossed the moving window to $3.14 * R^2$.

Sum Length : calculation of grid layer of total length of lines within the mowing circle window with radius R .

4.4. Plausible inference

4.4.1. Plausible inference methods of GIS GeoProcessor

Enter the following notations: R^I is a space of normalized features with root mean square $\sigma_i \equiv 1$, $\rho(\mathbf{x}, \mathbf{x}^{(n)})$ is a metric on R^I , C is a constant.

1. Similarity on a precedent set

Let $\{\mathbf{x}^{(n)}\} \subset R^I$, $n = 1, \dots, N$, is a set of precedents, then the similarity function is

$$S(\mathbf{x}, \{\mathbf{x}^{(n)}\}) = 1 - \frac{\min_{\langle n \rangle} \rho(\mathbf{x}, \mathbf{x}^{(n)})}{C} \quad \text{for } \min_{\langle n \rangle} \rho(\mathbf{x}, \mathbf{x}^{(n)}) \leq C. \quad (4.1)$$

2. Similarity on expert knowledge (under developing)

Let $g^{(n)} = (a_i^{(n)}, b_i^{(n)})$, $i = 1, \dots, I$, is an I -dimensional interval in R^I that represents expert knowledge and, $\{g^{(n)}\}$, $n = 1, \dots, N$ is a set of intervals, then the similarity function is

$$S(\mathbf{x}, \{g^{(n)}\}) = 1 - \frac{\min_{\langle n \rangle} \rho(\mathbf{x}, g^{(n)})}{C} \quad \text{for } \min_{\langle n \rangle} \rho(\mathbf{x}, g^{(n)}) \leq C,$$

where $\rho(\mathbf{x}, g^{(n)})$ is a distance from \mathbf{x} to $g^{(n)}$.

3. Membership function for two classes

Let $\{\mathbf{x}^{(q,n)}\}$ and $\{\mathbf{x}^{(\bar{q},m)}\}$, $n = 1, \dots, N$, $m = 1, \dots, M$, are two sets of precedents in R^I , then the membership function is

$$f(\mathbf{x}, \{\mathbf{x}^{(q,n)}\}, \{\mathbf{x}^{(\bar{q},m)}\}) = -\frac{\rho(\mathbf{x}, \mathbf{x}^{(q,n)}) - \rho(\mathbf{x}, \mathbf{x}^{(\bar{q},m)})}{\rho(\mathbf{x}^{(q,u)}, \mathbf{x}^{(\bar{q},v)})} \quad \text{for } \min_{\langle n \rangle} \rho(\mathbf{x}, \mathbf{x}^{(n)}) \leq C, \quad (4.2)$$

where $\mathbf{x}^{(q,u)} = \arg \min_{\langle n \rangle} \rho(\mathbf{x}, \mathbf{x}^{(q,n)})$, $\mathbf{x}^{(\bar{q},v)} = \arg \min_{\langle m \rangle} \rho(\mathbf{x}, \mathbf{x}^{(\bar{q},m)})$.

4. Nonparametric regression function on a set of precedents (under developing)

Let $\{\mathbf{x}^{(n)}, \hat{f}^{(n)}\}$, $n = 1, \dots, N$, is a set of precedents, then the regression function is,

$$f(\mathbf{x}, \mathbf{x}^{(n)}, y^{(n)}) = \frac{\sum_{n=1}^N h(\mathbf{x} - \mathbf{x}^{(n)}) \hat{f}^{(n)}}{\sum_{n=1}^N h(\mathbf{x} - \mathbf{x}^{(n)})},$$

where $h(\mathbf{x} - \mathbf{x}^{(n)}) = 1 - \frac{\rho(\mathbf{x}, \mathbf{x}^{(n)})}{C}$ for $\rho(\mathbf{x}, \mathbf{x}^{(n)}) < C$ and $h(\mathbf{x} - \mathbf{x}^{(n)}) = 0$ for $\rho(\mathbf{x}, \mathbf{x}^{(n)}) \geq C$.

5. Certainty function for classification on monotone feature space

Enter a condition of monotonicity $f(\mathbf{x})$ on R^l . Let $\mathbf{x} = (x_1, \dots, x_l) \in R^l$, $\mathbf{x}' = (x'_1, \dots, x'_l) \in R^l$ and $x_i = x'_i$ for all $i \neq j$. Then $f(\mathbf{x})$ is non-decreasing function by the feature j if $f(\mathbf{x}) \geq f(\mathbf{x}')$ for $x_j > x'_j$ and non-increasing function if $f(\mathbf{x}) \leq f(\mathbf{x}')$ for $x_j > x'_j$.

Let $f(\mathbf{x})$ is an unknown classification rule for two classes for which the monotonicity condition on the feature space is given by an expert, $\{\mathbf{x}^{(n)}\} \subset R^l, n = 1, \dots, N$, is a set of precedents, then the certainty function is

$$\Psi_f(\mathbf{x}, \{\mathbf{x}^{(n)}\}) = \frac{v_f(\mathbf{x}, \{\mathbf{x}^{(n)}\})}{N}, \quad (4.3)$$

where $v_f(\mathbf{x}, \{\mathbf{x}^{(n)}\})$ is number of precedents for which monotonicity condition of $f(\mathbf{x})$ on the feature space is confirmed at the point \mathbf{x} .

4.4.2. Creating the sample sets

Sample sets are used in plausible inference operations. A sample set is created by user. Possible to create two types of samples: point and polygons. Point sample set consists of user defined points. Polygon sample set consists of the all grid points located inside the polygons.

Use option *Map > New Sample* to create a point sample set, and *Map > New Areas* to create a area sample set. A set icons are used for crating the sample sets (fig. 18).









Buttons of the panels for editing the samples sets		
		Icon to mark sample set 1
	Icons for the types of the point sets	Icon to mark sample set 2
		Icon to mark sample set 3
		<i>Enter polygon</i>
	Icons to control creating area sets	<i>Close polygon</i>
		<i>Delete the last point of polygon</i>
		<i>Select the object</i>
	Control operations	<i>Delete selected object</i>

Fig.18. Buttons of the panels for editing samples sets.

4.4.3. Calculation of similarity function with a set of precedents

Similarity function is defined in (4.1).

To estimate similarity function we need a sample set (see section 4.4.2).

Use option *Map > Compose* to select grid layers which will be used for similarity function.

Make the following steps:

1. Run option *Transform > Similarity Field*.
2. Insert the name of similarity layer in row *Name* of the window *Similarity Field Creation*.
3. Select the metrics in row *Metric* (*C* or *L2*), which will be used in similarity function.
4. Insert radius *R* (usually from 0.5 till 3) in row *R (Sigm)*.
5. Select the name of sample set in subwindow *Samples*. Use button ">" to copy this name to subwindow *Similarity to*.
6. Click button *OK*.

Grid layer with similarity values appear at screen in gray scale. Grid layer with similarity values appear at screen in gray scale. More dark colors correspond to more values of similarity distribution function.

4.4.4. Calculation of membership functions with sample sets of two classes of precedents

Membership function is defined in (4.2).

To estimate a membership function we need two sample sets (see section 4.4.2).

Use option *Map > Compose* to select grid layers which will be used for a membership function.

Make the following steps:

1. Run option *Transform > Membership Field*.
2. Insert the name of membership layer in row *Name* of the window *Membership Field Creation*.
3. Select the metrics in row *Metric* (*C* or *L2*), which will be used in membership function.
4. Insert radius *R* (usually from 0.5 till 3) in row *R (Sigm)*.
5. Select the name of sample sets in subwindow *Samples*. Use button ">" to copy the names of sample sets to subwindows *Class A and Class Bo*.
6. Click button *OK*.

Grid layer with similarity values appear at screen in gray scale. More dark colors correspond to class A.

4.4.5. Calculation of empirical function of distribution

Empirical distribution function is defined in (4.3).

To estimate s empirical distribution we need a sample set (see section 4.4.2).

Use option *Map > Compose* to select grid layers which will be used for similarity function.

Make the following steps:

1. Run option *Transform > Monotonicity Field*.

2. Insert the name of output layer in row *Name* of the window *Monotonicity Field Creation*.
3. Select the name of sample set in subwindow *Samples*. Use button “>” to copy this name to subwindow *Monotonicity to*.
4. Click button *OK*.
5. Grid layer with similarity values appear at screen in gray scale. More dark colors correspond to more values of empirical distribution function.