Relief shape inheritance
and graphical editor for the landscape design

Egor A. Yusov, Vadim E. Turlapov
Nizhny Novgorod State University after N. I. Lobachevsky,
Nizhny Novgorod, Russia
yusov_egor@mail.ru, vadim.turlapov@cs.vmk.unn.ru

Abstract

In the context of this work a number of rules for landscape design editor development are proposed. These rules are aimed at the support of the reciprocal influence of the relief and objects located on its surface. An experimental graphical editor of dynamic 3D scenes has been created on the basis of relief shape inheritance principle (inheritance of z coordinate of any object). Various inheritance rules (including object binding to the specific point, continuous and discrete object dependence on the relief) as well as efficient landscape modification methods are implemented in the editor. The program also realizes texture mapping, objects import from 3D Studio MAX, objects movement along the defined trajectories, water surface modeling. A landscape reconstruction algorithm according to the coloration of geographical map has been developed. A new image filtration algorithm is described.

Keywords: landscape design, 3D-graphics, graphical editor, relief inheritance, landscape reconstruction

1. INTRODUCTION

Modeling of terrain with objects located on its surface is one of important problems of computer graphics. Quite often this modeling must result in creation of panoramic fields of view, which can be used in landscape and architectural design or in video simulators and computer games.

High-quality modeling of such dynamic scenes requires realization of two conflicting demands [1]. On the one hand virtual model must be enough detailed. On the other hand it must work in real-time, i.e. it must support a high speed of visualization (not less then 25 frames per second) during flying around scene.

There are many problems in this field of computer graphics. The most significant are:

- landscape reconstruction and its texturing using different data sources;

- scenery reconstruction according to maps and other sources;

- relief shape inheritance;

- simplification of terrain geometric model;

- active environment modeling based on the landscape and scenery;

There are a lot of works devoted to the terrain model simplification. But the problem of interconnection between landscape and objects is not considered. For example, in [5] simulating of autonomous vehicles on planet-sized terrain is described, but stationary objects like buildings, roads, etc. are not

taken into account by the authors. This paper is mainly concerned with the problem of relief shape inheritance, landscape reconstruction and active environment modeling.

2. PROBLEM OF RELIEF SHAPE INHERITANCE

In order to solve landscape development task it is necessary to design 3D editor, which allows the user not only to restore 3D scene of a real area by the flat map with minimal expenses, but also to easily continue scene modeling (landscape modification, objects locating etc.).

Nowadays the problem of landscape reconstruction according to topographical plan is solved by many geoinformation systems, but 3D modeling of situation is usually absent. 3D Studio MAX [2] is the system which provides the user with capabilities, which are the most close to the demands of applied task. It includes Terrain module, which can restore 3D landscape by the aggregate of isolines closed contours. After landscape has been reconstructed, the user can proceed with farther modeling of the situation. However the user has to solve by hand all tasks, concerning congruence between landscape and objects, which are placed on its surface. If user decides to change relief, then he or she will have to manually edit all objects situated on it. If many objects have been already positioned on the landscape surface, it would be very inconveniently to update all their locations and forms. This problem becomes more difficult if objects are complex and multi-layer in respect to geometry and texture or they are built by the means of artificial intelligence [3], [4].
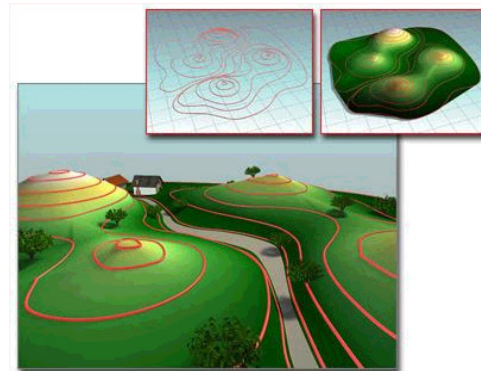

Figure 1: Relief reconstruction by the facilities of Terrain module.

Thus if you work in 3D studio, you should firstly completely design your landscape and only then start locating objects on it. Otherwise you can be confronted with great difficulties if need to modify relief arises.

Scene texturing in 3D Studio almost completely excludes flying around the scene in real time.

3. MAIN THESES

In order to create facilities, allowing easily landscape reconstruction according to a flat plan (map), it is proposed to introduce into the geoinformation systems and 3D editors relations of relief shape inheritance by the objects. The inheritance rules must conform to the real object relations, including the following:

1) object binding to the specific point of the landscape surface or another object with inheritance of its position by z coordinate;

2) reciprocal influence of the landscape and objects, located on it (for example, landscape leveling by the object's bottom side or object's bottom side correction by the relief surface);

3) continuous or discrete object dependence on the shape of relief on which it is situated (continuous, discrete, continuous with alignment by direction, etc.).

3.1 Types of dependence

As it has been said earlier, one of the fundamental methods of landscape and objects interconnection is specifying for object one of three possible types of dependence. Now we describe these types more detail.

3.1.1 Continuous dependence

In the process of landscape design a situation, when some object (for example, pipe of heating system, wall, road and so on) should completely repeat the shape of relief, on which it is located, can occur. The characteristic feature of these objects is that their form is directly defined by the terrain surface under them.
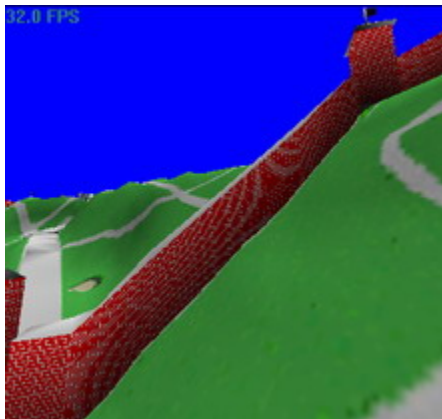

Figure 2: Kremlin wall continuously following the slope on which it is built

If relief under the object with continuous dependence is modified, then this object must automatically change its shape so that it conforms to the new landscape state. In contrast to systems like 3D Studio MAX the user should not be required to perform any additional actions to achieve this conformity.

3.1.2 Discrete dependence

Along with the continuous dependence, when object flow over the terrain, on which it is placed, another typical situations can be met. In these situations an object must be divided into some parts according to some rules, and each part must watch landscape height under it irrespective of the other parts. For example, a wall, built on the slope, can be divided into sections and go down like a big stairs (Figure 3).

If landscape under the object with this type of conformity is modified then every object's discrete part must automatically move in order to correspond to the new terrain state.

Figure 3: Kremlin wall discretely following the slope on which it is built

3.1.3 Continuous dependence with generatrix horizontality retention

One of the most widespread city objects are roads. They should follow the terrain surface, on which they are built, just as objects with uninterrupted dependence. But there is one essential difference: every transverse cut of road top side should always be horizontally. Important factor which has an impact on the object shape in this case is its centre line or generatrix. It is centre line that allows correct determining transverse cut of the side. 3D editor must support this type of dependence.

Figure 4: The road for which a continuous dependence with alignment by object direction is specified

3.2 Animation

High-grade design of real scenes can not be achieved without objects animation modeling. During landscape designing objects motion is closely related with landscape shape along which it is performed. 3D editor must support various types of movement of real objects (cars, airplanes, helicopters etc.): moving along terrain surface, flying over the landscape with repetition of its contour or flying on constant height, etc. Objects trajectories must be automatically updated every time when landscape changes. And it is also the graphical editor's responsibility.

4. TERRAIN RECONSTRUCTION ACCORDING TO THE GEOGRAPHICAL MAP

There are many ways to obtain terrain shape description. If landscape design is being carried out for a real territory, the source information for it can be obtained from a topographic plan, which contains the image of situation and scenery. Flat plan describes landscape surface by means of isolines and elevations Geoinformation systems usually use flat topographical plans as base source for the reconstruction. However, flat plans are not used to describe large regions. Geographical maps are usually utilized for this purpose.

Geographical map keeps information about landscape shape in the coloration of the map. Each map has their own height scale, which sets up a correspondence between colors presented in the map and real heights of the region.

A special algorithm which allows restoring landscape by the image of the original geographical map has been developed.

This algorithm has three stages: (1) population of the color levels table, which sets up correspondence between colors and heights of the area; (2) image filtration and (3) reconstruction itself.

4.1 Image filtration

4.1.1 Purpose

Geographical maps usually contain much subsidiary information such as various names, parallels, meridians etc. Since algorithm being proposed performs terrain reconstruction based on the coloration of the map, the colors of this subsidiary information can not be used to calculate height of appropriate point of the region. Thus it should be removed from the map. This is what the filtration is aimed at. Two algorithms have been implemented: median filtration and filtration by mean square deviation.

4.1.2 Description of filtration by mean square deviation algorithm

For each pixel of the image the following actions are performed:

1. An average color in the square area, in the centre of which a current pixel is located, is calculated. Algorithm has parameter a, which defines the size of this area (fig. 5).
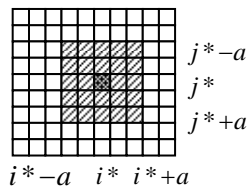

$j*-a$
$j*$
$j*+a$

$i*-a \quad i* \quad i*+a$

Figure 5: A square area of the image in which an average color is calculated

The following formulas are used to calculate red, green and blue components of the average color:

$$\overline{R} = \frac{\sum\limits_{(i,j)\in D} R_{i_j}}{m}, \ \overline{G} = \frac{\sum\limits_{(i,j)\in D} G_{i_j}}{m}, \ \overline{B} = \frac{\sum\limits_{(i,j)\in D} B_{i_j}}{m}$$

Where R_{i_j}, G_{i_j} and B_{i_j} are red, green and blue constituents of (i,j) pixel. $D = \{(i,j): i_{\min} \leq i \leq i_{\max}, j_{\min} \leq j \leq j_{\max}\}$ - is an averaging area; $i_{\min}, i_{\max}, j_{\min}, j_{\max}$ can be obtained using following expressions:

$i_{\min} = \max(0, i*-a)$, $i_{\max} = \min(W_{\text{Picture}} -1, i*+a)$

$j_{\min} = \max(0, j*-a)$, $j_{\max} = \min(H_{\text{Picture}} -1, j*+a)$

Here W_{Picture} and H_{Picture} are image width and height correspondingly.

$m = |D| = (i_{\max} - i_{\min} +1)(j_{\max} - j_{\min} +1)$ is a power of set D.

2. On the second step a dispersion of colors in the averaging area is calculated by the following formula:

$$\overline{\rho^2} = \frac{\sum\limits_{(i,j)\in D}\left((R_{i_j} - \overline{R})^2 + (G_{i_j} - \overline{G})^2 + (B_{i_j} - \overline{B})^2\right)}{m}$$

And square deviation of central pixel from average color is calculated:

$$\rho*^2 = (R_{i*_{j*}} - \overline{R})^2 + (G_{i*_{j*}} - \overline{G})^2 + (B_{i*_{j*}} - \overline{B})^2$$

3. If after that the following inequation

$$\rho*^2 > \lambda^2 \overline{\rho^2}$$

is true, the color of central pixel is improved with corrected average color. In the last expression $\lambda \geq 1$ is a rejection coefficient. It is a second parameter of the algorithm.

4. If centre pixel has to be updated, the corrected average color is computed. Its components can be obtained by the following formulas:

$$\overline{R}_{\text{Corrected}} = \frac{\sum\limits_{(i,j)\in \widetilde{D}} R_{i_j}}{\widetilde{m}}, \overline{G}_{\text{Corrected}} = \frac{\sum\limits_{(i,j)\in \widetilde{D}} G_{i_j}}{\widetilde{m}}, \overline{B}_{\text{Corrected}} = \frac{\sum\limits_{(i,j)\in \widetilde{D}} B_{i_j}}{\widetilde{m}}$$

where $\widetilde{D} = \{(i, j) \in D : \rho_{i_j}^2 \leq \lambda^2 \overline{\rho^2}\}$ is corrected averaging area; $\widetilde{m} = |\widetilde{D}|$ - is a number of pixels in the set $\widetilde{D}$.

$\widetilde{m}$ is always grater than 0, because of the following reasons: $\rho*^2 > \lambda^2 \overline{\rho^2}$, $\lambda \geq 1$, consequently $\rho*^2 > \overline{\rho^2}$. Central pixel is included into the set D and $\rho*^2 > \overline{\rho^2}$, hence there must be at least one pixel in the D, which has a lesser deviation from average color than average deviation. Since $\widetilde{m} \neq 0$, corrected average color will always be accurately computed. Excluding pixels, which have greater deviation from the average color than $\lambda\overline{\rho}$ allows reducing their influence on the improved color.

Filtration algorithm has two versions: continuous and discrete. In the first case filtration is performed as it has been described above.

Discrete filtration takes into account the fact that only some number of predefined colors are allowed for the map. These colors populate color level table. In case of discrete filtration before some pixel will be improved a search in the color levels table of the most similar to the corrected average color is preformed. And central pixel is set to this obtained color.

4.1.3 Median filtration algorithm description

A second filtration algorithm implemented in the editor is median filtration. The algorithm does the following:

1. For each pixel of the image a sample of its surroundings of size 3x3, 5x5 or 7x7 is performed, just like for filtration by mean square deviation.

2. Pixels of the sample are arranged by its intensities, which can be calculated using the following expression:

I=0.3R+0.59G+0.11B

3. Color of the central pixel is improved with the color, corresponding to the middle value of intensity in the sorted list.

Algorithm also has two modifications: continuous and discrete. In the second case color of central pixel is improved with the color form the table, which is the most similar to the median color.

4.1.4 Algorithms comparison

Figure 6 shows initial image of the map, figure 7 shows results of filtration by mean square deviation algorithm and figure 8 shows results of median filtration.
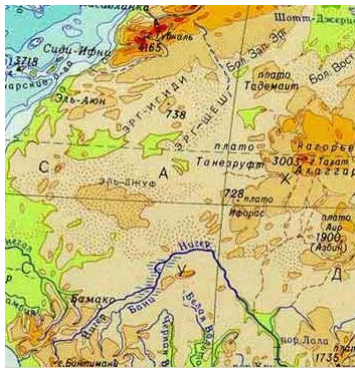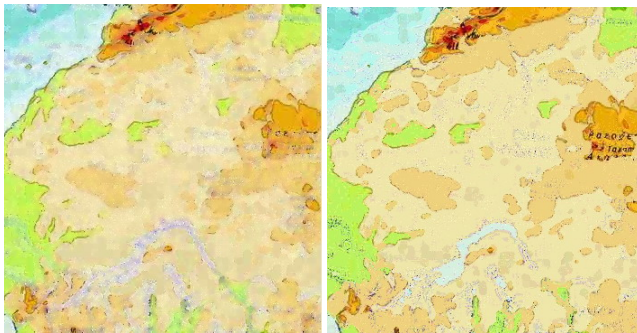

Figure 6: Initial image of the map

Figure 7: Continuous (on the left) and discrete (on the right) mean square filtration results
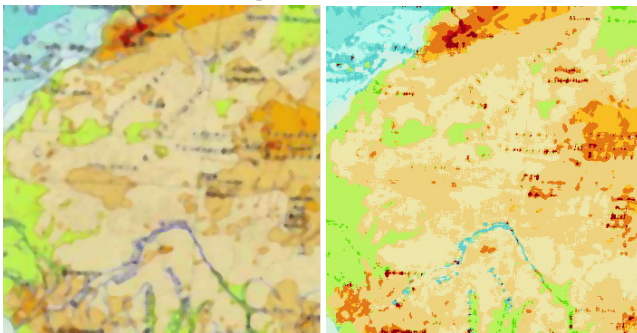

Figure 8: Continuous (on the left) and discrete (on the right) median filtration results

Figures reveal that the best result is obtained when discrete filtration by mean square deviation algorithm is used.

4.2 Terrain reconstruction

After geographical map image has been cleared from the additional information, a terrain reconstruction itself can be carried out. In this stage the filtered image is uniformly stretched on the terrain frame and for each node a height is calculated. In order to make terrain more smooth nodes heights are computed based on the square image area, in the centre of which a pixel corresponding to the node is situated (fig. 9). A height of each color in this area is calculated and average height is assigned to the node.

Dimensions of this area depend on the ratio of map image and terrain frame sizes.
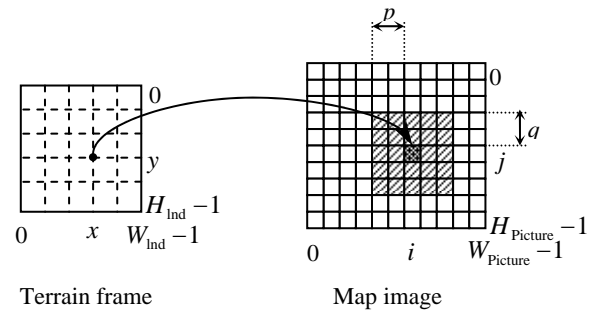

Terrain frame   Map image

Figure 9: Frame node and corresponding smooth area of the image

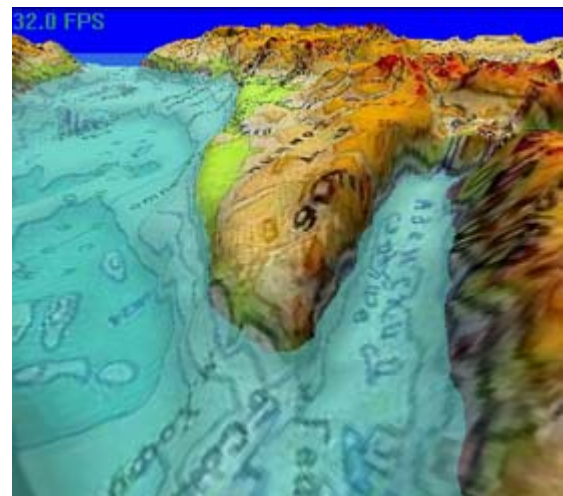Figure 10 shows snapshot of Africa terrain, reconstructed by the original geographical map.

Figure 10: A landscape reconstructed by the geographical map of Africa

5. DESCRIPTION OF THE DEVELOPED EDITOR

In order to investigate effectiveness of suggested inheritance facilities an experimental version of 3D editor, which uses in practice all theses listed above, has been developed. Experimental editor realizes different inheritance rules (for instance, object binding to the specific point, all kinds of dependence). A number of special relief editing methods (which include different types of surface deformation – interpolation, shifting etc.) are also granted by the program.

5.1 Random landscape generation

The editor defines landscape as a height matrix. One possible way to obtain this matrix and to simulate unknown area is to generate it using random number generator. A special land producing algorithm, based on the Peril's noises, has been developed. Selecting different values of generation parameters allows producing various landscapes.
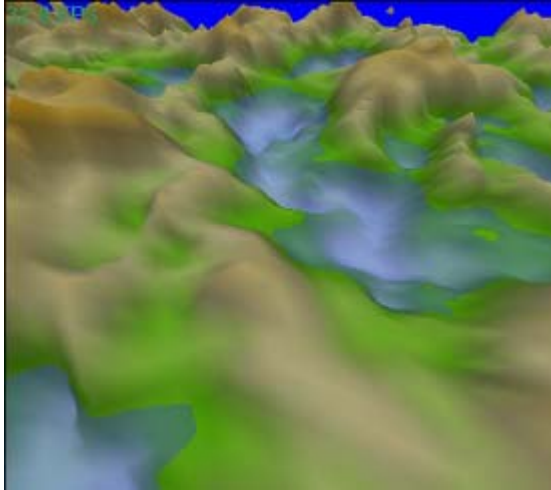

Figure 11: The landscape obtained using random number generator

A good method of improving landscape quality, which used in the program, is to color it depending on the height. This means, that low regions of surface have blue color (they imitate water); green (color of grass) is assigned to the higher regions; it is followed by the brown (mountains) and the highest regions have white color (they modeling snow on the tops of mountains).

5.2 Additional capabilities

The following features have been also implemented:

– texture mapping on the landscape and objects;

– fogging;

– modeling of different types of scene illumination and characteristics of landscape surface;

–object importing from 3D Studio MAX;

– water surface modeling;

– a library of standard objects.

Solutions, which are assumed as a basis of the editor, allow the user to fly around textured scene, being designed, in real time even on a computer without 3D accelerator. A high speed of animation is achieved on the computer with 3D accelerator. Direct 3D is used as base graphical program software.
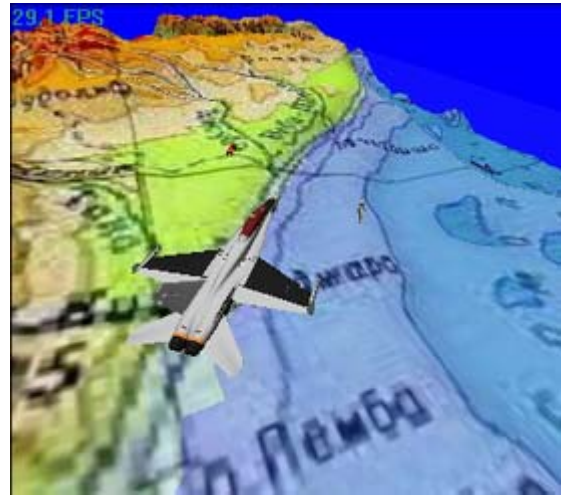

Figure 12: Using of models imported from 3D Studio MAX during scene design

6. CONCLUSION

As a whole experimental version of the graphical landscape editor confirms efficiency of objects location inheritance by z coordinate facilities. Inheritance instrument and means of landscape reconstruction according to the flat geographical map can be easily introduced into the instrumental geoinformation systems with object data structure.

7. REFERENCES

[1] Mikhailuk M.V. *Computer graphics for video simulator // Proc. of the Conf. of Computer Graph. and Visual. - GraphiCon'2002, N. Novgorod, 16-21 sep. 2002.*

[2] Murdock K.L.. *3ds max 5. Bible. – Wiley Publishing, Inc., 2002. – 1136p.*

[3] Dimitri Plemenos D., Miaoulis G., Vassilas N. *Machine Learning for a General Purpose Declarative Scene Modeller // Proc. of The Conf. of Computer Graph. and Visual. - GraphiCon'2002, 16-21 sep. 2002, p.23-30*

[4] Sosnov A., Huot S., Mace P., Hegron G. *Rapid Incremental Architectural Modeling from Imprecise Perspective Sketches and Geometric Constraints // Proc. of the Conf. of Computer Graph. and Visual. - GraphiCon'2002, N. Novgorod, 16-21 sep. 2002, p.42-49*

[5] Selcuk Sumengen, Selim Balcisoy *Real-time simulating of Autonomous Vehicles on Planet-Sized Continuous LOD Terrains // Conference Proceedings ISBN 80-903100-7-9 WSCG'2005, January 31 – February 4, 2005, Plzen, Czech Republic.*

About the author

Yusov Egor - student of Nizhny Novgorod State University after N.I. Lobachevsky, Department of Computational Mathematics and Cybernetics. E-mail: yusov_egor@mail.ru

Turlapov Vadim - professor of Nizhny Novgorod State University after N.I. Lobachevsky, Department of Computational Mathematics and Cybernetics. E-mail: turve@pent.sci-nnov.ru