

WOJCIECH PASTERNAK*, ZBIGNIEW LATAŁA**

VRML GENERATOR OF LAND FEATURES 3D REPRESENTATION BASED ON A HYSOMETRIC MAP

GENERATOR KODU VRML TRÓJWYMIAROWEJ REPREZENTACJI UKSZTAŁTOWANIA TERENU NA PODSTAWIE MAPY HIPSOMETRYCZNEJ

Abstract

The paper contains an original program "KARTOGRAF 3D" enabling a 3D model of land features to be created based on the hypsometric map. 3D modelling is carried out by generating a VRML code being a digital representation of the topography. All the project is a Windows application designed for Windows and implemented in language C++ by using the IDE tool of Borland- C++ Builder RAD type [4]. Such environment was chosen because it allows creation of quickly expanded user interfaces and easy assigning of functionalities implemented in language C++, while keeping an attractive interface. The program enables a hypsometric map to be downloaded, viewed as a contour map and as a finally generated 3D-model. In addition, the application allows a texture model and fog effect to be imposed on the generated surface features, thus making the scene more realistic.

Keywords: 3D modelling, hypsometric map, surface, VRML

Streszczenie

W artykule przedstawiony został autorski program „KARTOGRAF 3D” pozwalający na generowanie trójwymiarowego modelu rzeźby terenu na podstawie mapy hipsometrycznej. Modelowanie 3D realizowane jest poprzez generowanie kodu VRML, stanowiącego zapis cyfrowy topografii. Cały projekt jest aplikacją okienkową, napisaną dla systemu Windows i zaimplementowany został w języku C++ z zastosowaniem narzędzia IDE typu RAD firmy Borland – C++ Builder [4]. Celowo zostało wybrane takie środowisko, ponieważ pozwala ono na szybkie tworzenie rozbudowanych interfejsów użytkownika i łatwe podpinanie do nich funkcjonalności zaimplementowanych w języku C++, przy jednoczesnym zachowaniu atrakcyjnego interfejsu. Program pozwala na wczytywanie mapy hipsometrycznej, a następnie na jej przeglądanie w formie mapy poziomicowej oraz jako gotowego, wygenerowanego modelu trójwymiarowego. Dodatkowo aplikacja została tak zaprojektowana, aby można było nakładać na wymodelowany teren tekstury oraz efekt mgły, co w znaczący sposób urealistycznia utworzoną scenę.

Słowa kluczowe: modelowanie 3D, mapa hipsometryczna, powierzchnia, VRML

* Wojciech Pasternak, Msc Eng, U2i Internet Development New York.

** Zbigniew Latała, PhD, Institute of Computer Science, Cracow University of Technology.

1. Introduction

A dynamic development of information technology, in particular huge increase of computer capacity, enables 3D visualization [3]. Cartography is one of the disciplines where such possibilities are used more and more intensively.

Due to the existing applications such as Surfer of Golden Software and Field 3D [2], it is possible to create hypsometric and contour maps based on the data set containing heights of field points. However, none of these programs is able to generate 3D objects from a flat hypsometric map.

The aim of the paper is to develop and implement a Kartograf 3D program generating a 3D model of land features based on a hypsometric map. The model should be saved in the form of a VRML file, thus being a digital representation of object geometry expressed as a code written in the language used for virtual reality modelling.

2. Technologies used for developing the application

The following technologies were chosen to develop the program:

1. VRML (Virtual Reality Modelling Language) – a script language used for creating 3D computer object representations – virtual reality.
2. Borland C++ Builder – RAD type tool enabling windows applications to be built by using C++ algorithms [1].

VRML allows both static and animated objects as well as links to other media such as images, movies and sounds to be created [5].

To view virtual worlds created with the VRML it is necessary to have a Web browser supporting this language. There is a number of expansions called plug-ins to such popular browsers as FireFox and Internet Explorer. An example is "Cortona VRML Client" that was used for the purpose of this paper. The browser of this type has a VRML interpreter that interprets the program code and transfers it to so-called render engine, creating on the display unit an image being the representation of the modelled world. The render engine very often uses the common graphic libraries such as OpenGL or Direct3D, thus enabling smooth object animation.

3. VRML modelling of land features

Nodes are basic elements of a virtual world generated with VRML. When the application "Kartograf 3D" was developed, the node ElevationGrid allowed virtual land features, even highly complicated, to be created.

The node ElevationGrid has the following notation:

```
ElevationGrid {
  eventIn      MFFloat  set_height
  exposedField SFNode  color          NULL
  exposedField SFNode  normal        NULL
  exposedField SFNode  texCoord      NULL
```

```

field      MFFloat  height      []
field      SFBool   ccw          TRUE
field      SFBool   colorPerVertex  TRUE
field      SFFloat  creaseAngle    0
field      SFBool   normalPerVertex  TRUE
field      SFBool   solid          TRUE
field      SFInt32  xDimension     0
field      SFFloat  xSpacing       1.0
field      SFInt32  zDimension     0
field      SFFloat  zSpacing       1.0
}

```

The operation principle of this node is creating a grid spread in the plane defined by axes X and Z . Then, each grid node (grid line intersection) is assigned an appropriate height, i.e. the vector perpendicular to the grid plane.

The grid is defined by setting the fields: $xDimension$ and $zDimension$ as well as $xSpacing$ and $zSpacing$. The former ones determine the number of grid lines, the latter are responsible for line spacing.

In the field height we specify heights for all grid nodes. The first value in this field is the height of point $(0, 0)$ in the XZ co-ordinate system. The next value corresponds to point $x = 1, z = 0$, etc. up the end of the X axis. Then, we analogously define the values for $z = 1, z = 2$, etc. The number of values specified in the field height should be exactly equal to: $xDimension * zDimension$.

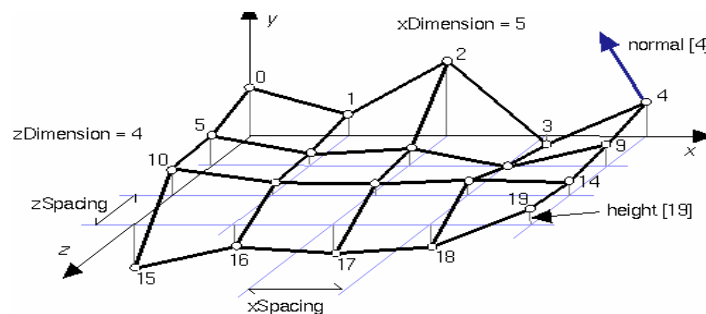


Fig. 1. ElevationGrid node operation principle

Rys. 1. Zasada działania węzła ElevationGridFig

4. The requirements to be met by maps used as program references

The "Kartograf 3D" program has some limitations related to data sets supported. One limitation is necessity to use a hypsometric map without such details as symbolically shown roads, railways, buildings and similar elements. Thus, for proper operation of the application we need a map containing only information on land features, i.e. a hypsometric map in its purest form.

The above limitations result from the fact that the problem is highly complicated. The adaptation of any hypsometric map containing a large number of useless details will be a highly complicated task. Many transformations as well as a powerful image analysis tool, including identification, classification and elimination of specified groups of objects to remove useless data and maintain only those necessary to model land features are required. This could be reached by using artificial neuron grids, but this exceeds the framework of the present paper.

To sum up: a hypsometric map containing only information on land features with each contour assigned to only one shade is required for proper operation of the program.

5. 3D code generation algorithm

To model land features the node ElevationGrid of VRML was used. The model is based on a rectangular grid with assigning each node a specified height.

The main purpose of the program is to superimpose the grid onto the downloaded flat hypsometric map and then to determine the height for each node based on the colour at the specified point and contour spacing.

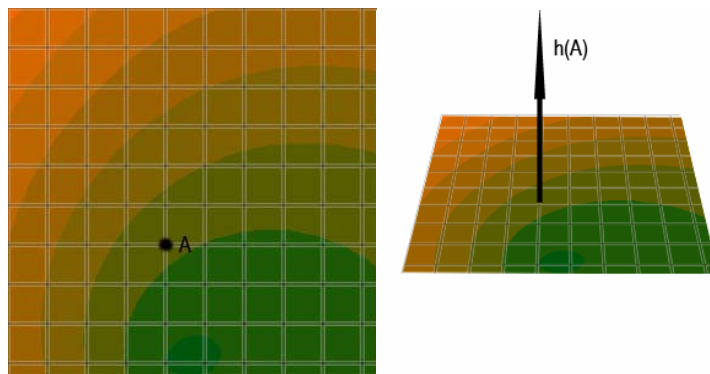


Fig. 2. Grid-based model of land features

Rys. 2. Model terenu oparty na siatce

The determined heights are entered into the field height of the node ElevationGrid, thus creating a 3D-model of land features.

The height of each field point "A" represented on the map as a grid node depends on the contour colour at this point and the distance from the adjacent contours. In hypsometric maps each colour has the maximum height assigned that can be reached by any point located in the contour of the specified colour. The points lying on the same contour can take the values from the range $[h_{i-1}, h_i]$, where: h_i – the maximum height for "i th" contour on which the point lies, h_{i-1} – maximum height of "i-1-th" contour representing the previous (one order lower) height interval.

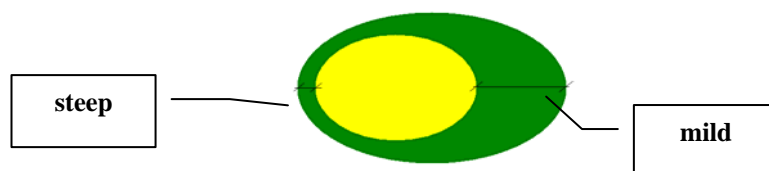


Fig. 3. The principle of slope steepness presentation on hypsometric maps
 Rys. 3. Zasada przedstawiania nachylenia stoku na mapie hipsometrycznej

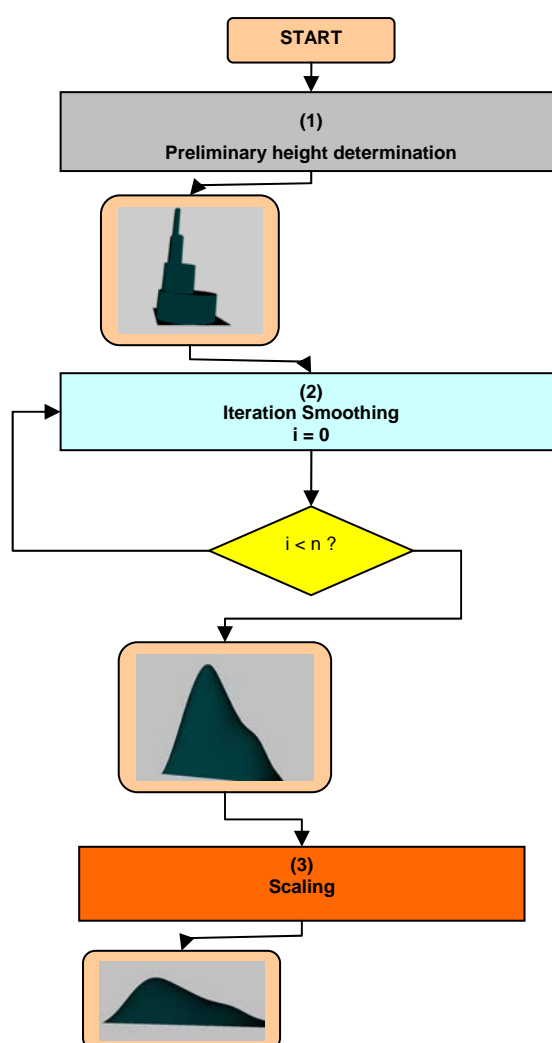


Fig. 4. Algorithm operation principle (described in the text)
 Rys. 4. Zasada działania algorytmu (opis w tekście)

The general rule for hypsometric maps is that the wider contour in the specified area, the smaller the land gradient. In other words, "densely packed" contours represent very steep slopes (large height changes in the horizontal length), while "loosely packed" contours correspond to gentle slopes (small height changes in the horizontal length).

Fig. 4 explains why the height at the specified point depends both on the colour and contour spacing.

The algorithm implemented in the application "Kartograf 3D" determines heights based on the relationship presented above. It consists of the following three parts:

1. **Preliminary height determination** – For each grid node superimposed on the map, the colour and corresponding maximum allowable height is read based on the appropriate contour. As a result of this operation the "stepped model" is created – the land is modelled with a large number of steps, there are no real smooth transitions between land roughness.
2. **Iteration (smoothing)** – For each grid node the heights of its eight "neighbours" determined as described under 1 are read, and then the point height is calculated as the arithmetic mean of all eight neighbours.

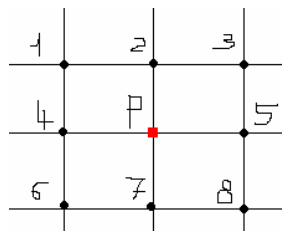


Fig. 5. Neighbourhood of point P

Rys. 5. Otoczenie punktu P

$$h(P) = \frac{h(1) + h(2) + \dots + h(8)}{8} \quad (1)$$

This operation is carried out n times, where n corresponds to the degree of land smoothing (the larger number of iterations, the "steps" become more smoothed). After the first iteration only the height of grid nodes lying in the close vicinity of the contours (at the border between two contours) are changed. Next iterations propagate the height changes across the entire grid.

As narrow, densely packed contours are located close to each other, the corresponding grid nodes are more quickly subjected to height changes (after smaller number of iteration). This is why the height change after performing n – iterations is here higher than that of loose contours and largely spaced grid nodes representing the wide contours that require a greater number of iterations for height change propagation. It follows from the above that the wide contours are modelled as smooth transitions (small height change on the horizontal length), while narrow dense contours as large slopes (large height change on the horizontal length).

The result of algorithm iterative part is a model of land features containing moderate, naturally-looking transitions between surface irregularities. However, this model has an improper vertical scale.

3. **Vertical scaling** – In this part of the algorithm the model obtained under point 2 is re-scaled to become more natural. The necessity of vertical (height) scaling results from the fact that the real heights were used in algorithm parts 1 and 2 (data specified on the hypsometric map). Since the flat map has a scale when is transformed into the 3D form. Thus it is necessary to rescale the new dimension (height) so that it will correspond to the map scale.

The scaling is carried out according to the following formula

$$h'(A) = \alpha \frac{h(A)}{k} \text{ and: } \alpha = \frac{h_{\max}}{H_{\max}} \quad (2)$$

where:

- $h'(A)$ – height at point A after scaling,
- $h(A)$ – height at point A before scaling,
- h_{\max} – maximum height of the highest contour on the hypsometric map,
- H_{\max} – maximum height obtained for a grid node after steps (1) and (2),
- k – scaling coefficient.



Fig. 6. An example of a modelled mountain with the texture superimposed

Rys. 6. Przykład wymodelowanej góry z nałożoną teksturą

6. Summary

1. The program Kartograf 3D can be successfully used as:
 - background of architectural presentation and visualization,
 - generator of virtual "worlds" when developing computer games and media,

- didactic aid for reading the hypsometric map,
 - widely understood computer art.
2. In the future the authors would like to redesign the algorithm to take into account a larger number of adjacent points, thus enabling the higher degree of smoothing of the modelled land features.

References

- [1] Cashman M. et al., *C++ Builder Vademecum profesjonalisty*, 2000.
- [2] <http://field.hypermart.net/>.
- [3] Latała Z., *Zastosowanie komputerowej analizy obrazu ultrasonograficznego do badania serca*, Wiomości Stereologiczne, Kraków 2002.
- [4] Shtern V., *Core C++ Inżynieria Oprogramowania*, Helion, Gliwice 2004.
- [5] Vacca J.R., *VRML Clearly Explained, 2nd Ed.* – AP PROFESSIONAL, 1997.