NUMERICAL INTERPOLATION IN FLIGHT SIMULATORS FOR MICROCOMPUTERS USING DIGITAL ELEVATION MODELS

Ricardo Rodrigues Rangel
Instituto de Estudos Avançados - IEAv - CTA
12.231-970 - São José dos Campos - SP
rangel@ieav.cta.br

Luiz Alberto Vieira Dias
Instituto Nacional de Pesquisas Espaciais - INPE
12.227-010 - São José dos Campos - SP
vdias@ltid.inpe.br

KEYWORDS: DEM/DTM K122, Simulation K101, Software K165

ABSTRACT:

This work presents Numerical Interpolation applications for Flight Simulators terrain visualization. The terrain may be synthetic or real, obtained through Digital Elevation Models. Using Computer Graphics techniques, it is possible to generate animations with reasonable realistic results. The Geometric Transformations and Interpolations must be sufficient fast and precise, in order that the Simulator would achieve a good performance.

1. INTRODUCTION

A Flight Simulator has the purpose to simulate the several steps of a real flight in a computational environment. Basically it consists of an animation over a Digital Elevation Model (DEM) , where the model movement simulates the airplane movement.

If the purpose is to simulate in real time, the frames have to be displayed faster than the human vision can perceive, or about 24 frames per second (as in motion pictures).

In Computer Graphics the terrain is represented by regular, or irregular grids, with the grid points representing the terrain altitude. For microcomputers the number of samples have to be kept to a minimum. The sampling of the altitude points may be obtained from maps of by photogrammetry.

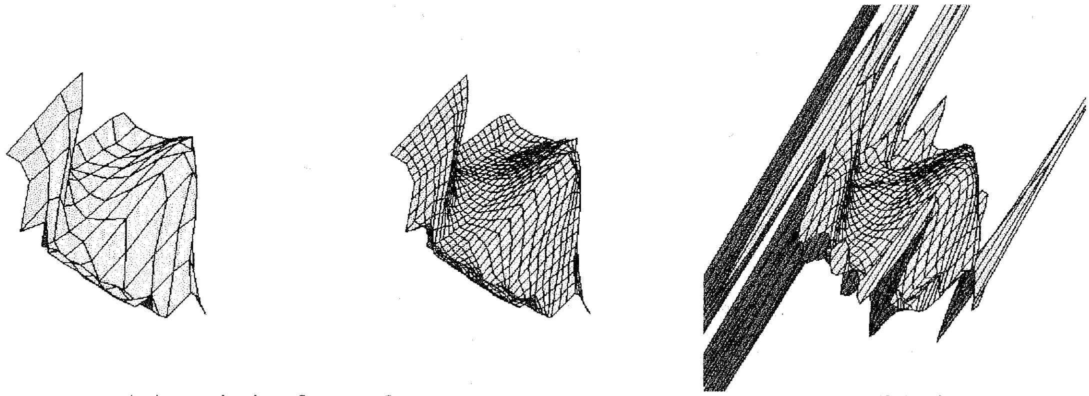
2. NUMERICAL INTERPOLATION

It is important that the numerical model could reproduce the original terrain with certain fidelity. If microcomputers are used, due to memory and speed limitations, the number of samples have to be kept to a minimum, so the interpolators used have to be very fast and reliable. The main purpose is to reconstruct the terrain, with the minimum possible points, and with the maximum fidelity. In this paper it will be used a regular grid.

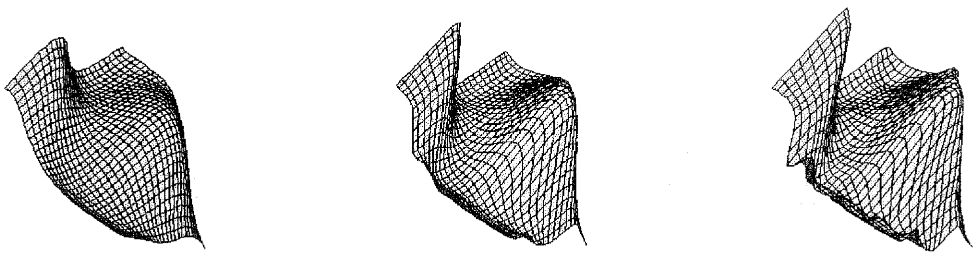
Let's briefly recall some interpolation methods. The linear interpolators are very fast, and, if the number of samples enough, the results are very good. However for insufficient sampling the results do not resemble the reality. In this paper the linear interpolators were used for illumination, shading and rendering of the frames.

The Akima interpolator, on the other hand, is also very fast, and does an excellent work in terrain reconstruction. It has partial second derivatives continuity on the patches junctions (Akima, 1974). This interpolator was used in this paper for the reconstruction of the sampled grids.

As an example Figure 2.1 below presents the result of the interpolation, by several methods, of a real terrain (region centered on 41°25'N and 74°59'W, on the Hudson Valley, near West point, VA, USA), with a 11 x 11 points grid with a spacing of 209.46 meters on the E-W direction, and 277.53 meters on the N-S direction. the altitude range in the real terrain is from 0 to 394 meters.



(a) Original surface (b)Linear
Interpolation (c) Lagrange Interpolation



(d) Bézier Interpolation (e) B-Spline
Interpolation (f) Akima Interpolation.

Fig. 2.1 - Terrain reconstruction through Numerical Interpolation.

3. FLIGHT SIMULATION

A flight simulation should generate each DEM frame and, by software, present the frames in sequence with the proper perspective, in order to show to the "pilot" the same terrain he or she would be visualizing in a real flight. The frames and the perspective have to be generated in real time, and in this case by a microcomputer.

Figures 3.1 to 3.6 shows the generation of the DEM for the Flight Simulator, using a regular grid.

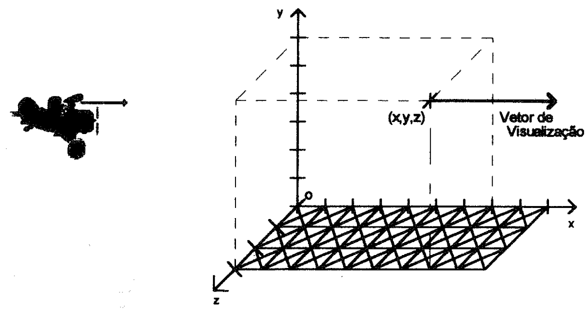


Fig. 3.3 - Visualization for an hypothetical observer.

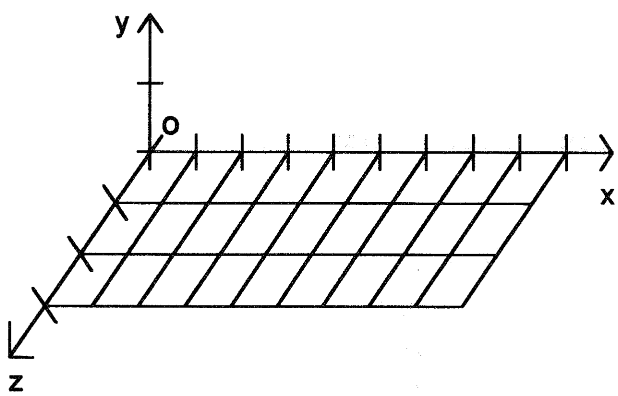


Fig. 3.1 - Regular grid with zero altitude.

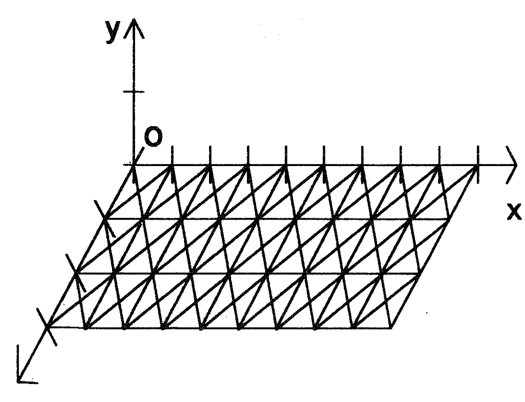


Fig. 3.2 - Regular grid triangulation.

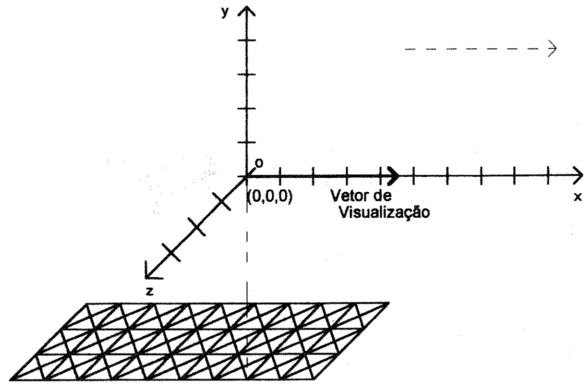


Fig. 3.4 - Visualization vector translation to the origin.

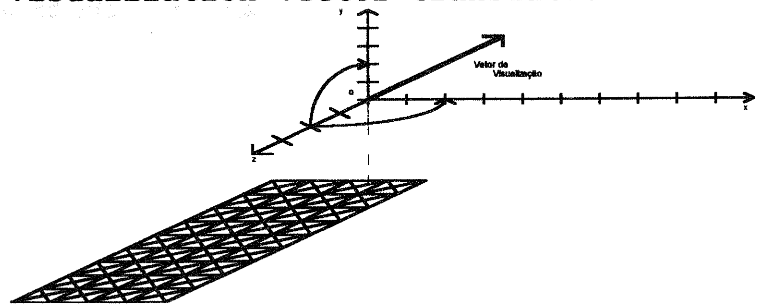


Fig. 3.5 - Visualization vector rotation.

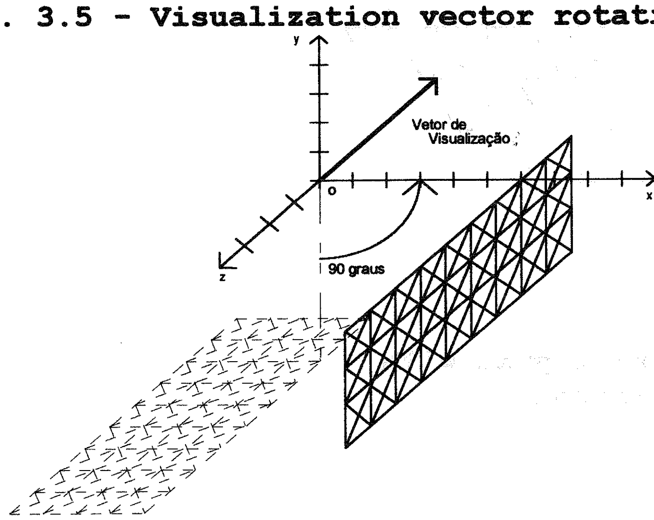


Fig. 3.6 - Rotation over the Z-axis: 90 degrees wing inclination in the hourly sense.

It is necessary now to render the terrain, using Computer Graphic techniques. It must be made a choice between a fast methods and methods that produce more realistic effects.

Figures 3.7 to 3.9 show the sequence to the rendering of the terrain. For hidden lines it was used the Z-buffer algorithm, and for illumination the Gouraud technique. The illumination was computed by the following formula:

$$I_d = C_i \cdot \cos(a) \cdot I_f + I_a,$$

where I_d is the light intensity reflected in each RGB color, C_i is the reflectance coefficient for the material for each RGB color, cos(a) is the cosine of the angle formed by the normal and the incident light, I_f is the intensity of the light source in each RGB color, and I_a is the ambient light. Using Gouraud, the intensities inside each point in the triangles is the linear interpolation of the vertices intensities.

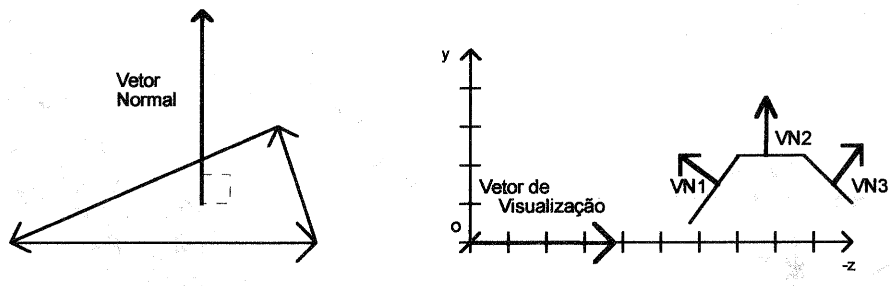


Fig. 3.7 - Vector Normal to the plane.

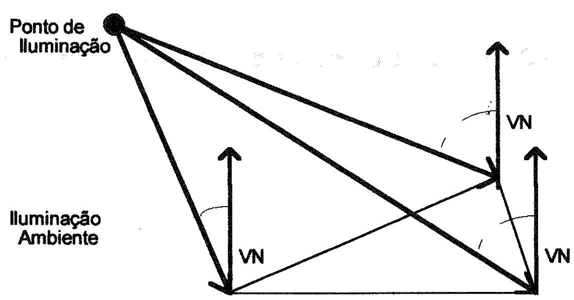


Fig. 3.8 - Vectors between a illumination point and triangle vertices.

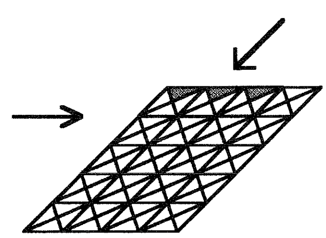


Fig. 3.9 - Painting sequence with the Z-Buffer algorithm de o Z-Buffer.

Next, the conic perspective projection has to be performed. Figure 3.10 presents the steps for the calculation of a conical perspective projection, with the reduction from three to two dimensions.

The last step is the painting of each polygon, through gray levels. It is possible to use more sophisticated methods as fractals, or an airplane or satellite image, however due to the need for speed in the computer calculations in this work the gray levels were painted according to their light intensity, from 0 (black) to 255 (white) in each RGB color.

Figure 3.11 presents a sequence of frames computed according to the method described above. An example of a five minute flight is available, running on a IBM-PC, 486/66, 8 Mbytes of RAM, under MS-DOS 6.0 and WINDOWS 3.11.

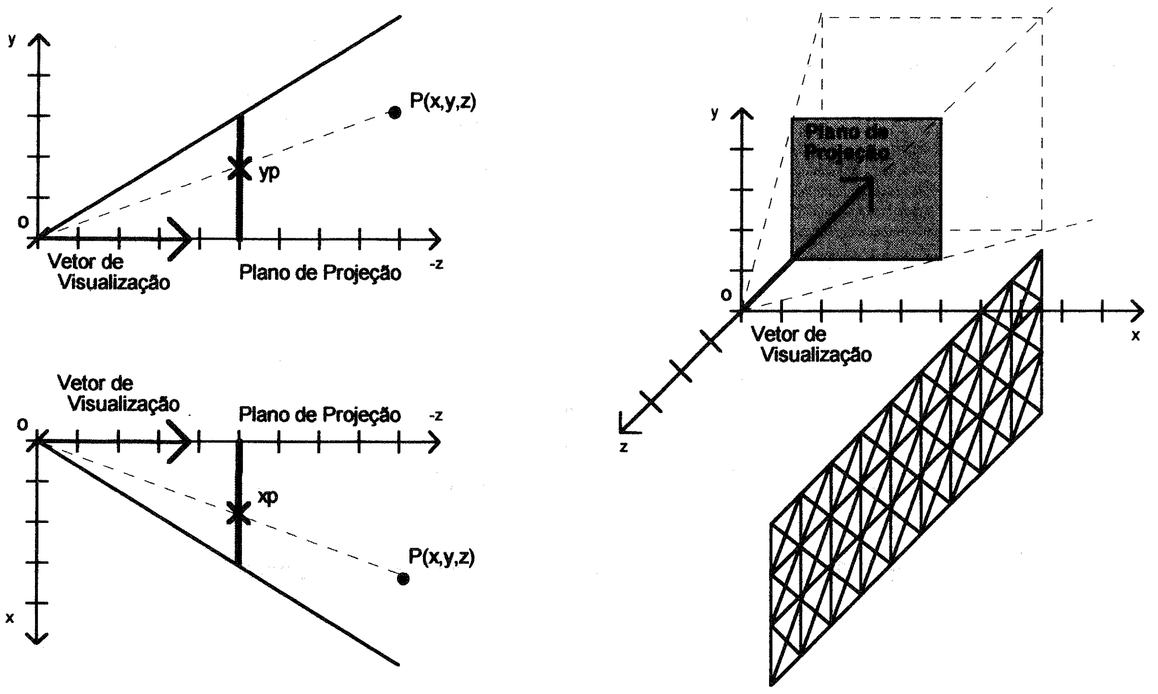


Fig. 3.10 - Conic perspective projection.

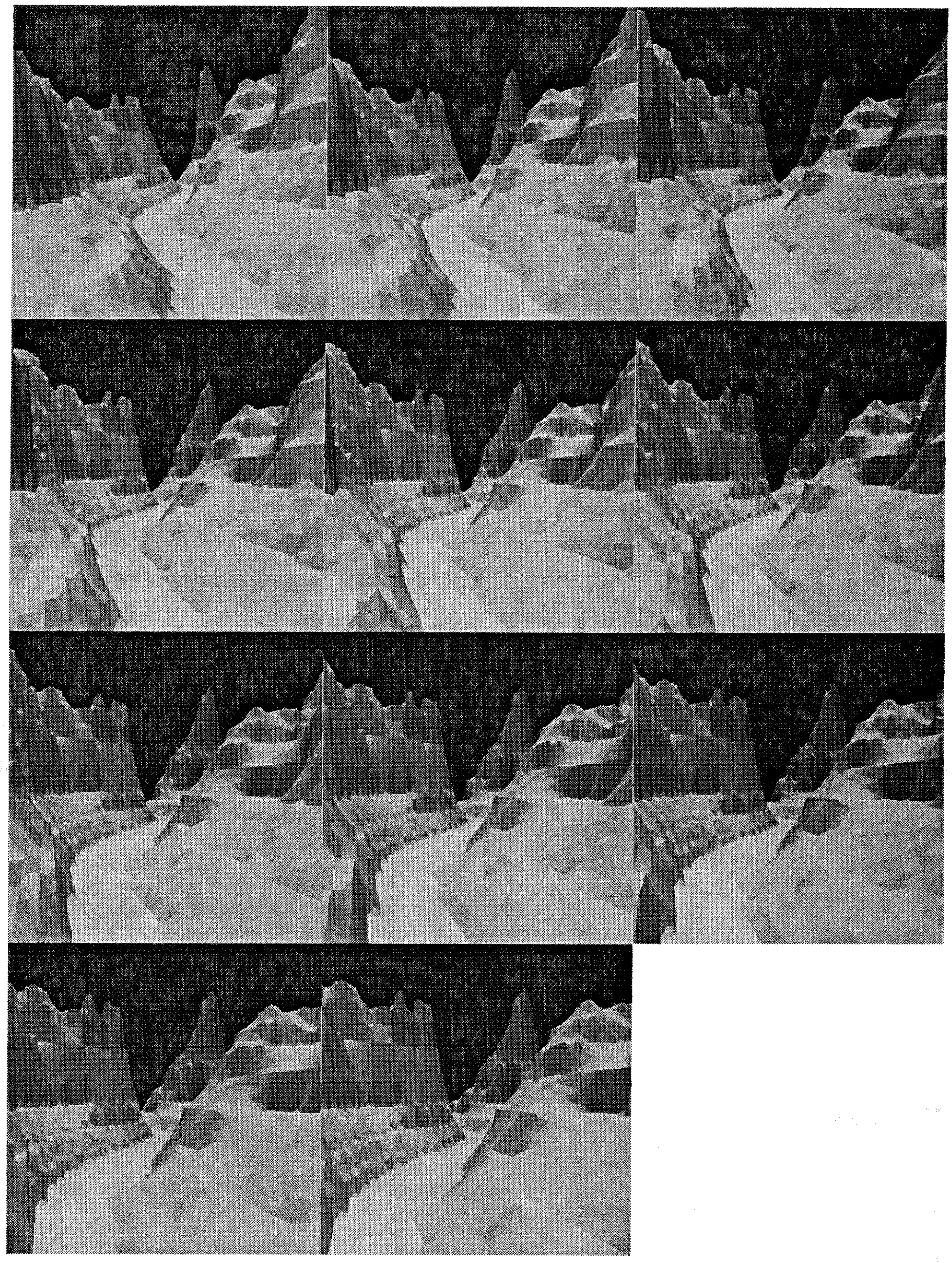


Fig. 3.11 - Sequence of frames for a flight on the simulator, based on the real terrain described above, using a gray level painting.

CONCLUSIONS

In this work it was shown how to develop a flight simulator, based on a terrain, using Digital Elevation Models. It is necessary to decide between a fast display and a more realistic display of the results. In this case, the proper choice of interpolators linear for shading and Akima for grid refining) optimized the results. Thus this simple simulator can be run on a typical microcomputer (IBM-PC, 486/66, 8 Mbytes RAM)

The realistic quality of the generated frames can be improved, if more sophisticated illumination and texture methods are employed, but care should be taken to avoid a loss in computer speed. It is possible to include vegetation, buildings, and other features. In future works this can be done.

While improvements can be added, it is believed that the goal of having a simple flight simulator, based on a real terrain, running on a standard microcomputer was achieved. This software can also be used to the development of better flight simulators systems.

5.BIBLIOGRAPHY

Ahuja, D.V. **An algorithm for generating spline like curves.** IBM Systems Journal, Vol. 17, n° 3, pp 206-217, 1968.

Akima, H. **A new method of interpolation and smooth curve fitting based on local procedures.** Journal of the Association for Computing Machinery, Vol. 17, n° 4, pp. 589-602, oct. 1970.

Akima, H. **Algorithm 433, Interpolation and smooth curve fitting based on local procedures.** Communications of Association for Computing Machinery, Vol. 15, n° 10, pp. 914-918, 1972.

Akima, H. **A method of bivariate interpolation and smooth surface fitting based on local procedures.** Communications of Association for Computing Machinery, Vol. 17, n° 1, pp. 18-20, Jan. 1974.

Akima, H. **Algorithm 474. Bivariate interpolation and smooth surface fitting based on local procedure.** Communications of Association for Computing Machinery, Vol. 17, n° 1, pp. 26-31, Jan. 1974.

Beatty, J.C.; Barsky, B.A. **Introduction to the Use of Splines in Freeform Curve and Surface Design.** SIGGRAPH 86 COURSE, University of Waterloo, Ontario, Canada, 1986. 330 p.

Bézier, P.E. **Emploi des machines à commande numérique.** Masson et Cie., Paris, 1970 [traduzido por A.Robin Forrest as **Numerical Control - Mathematics and Aplications,** John Wiley & Sons, London, 1972].

Bézier, P.E. **Mathematical and Practical Possibilities of UNISURF, Computer Aided Geometric Design.** Academic Press, New York, pp 127-152, 1974.

Foley, J.D.; et al. **Computer graphics: principles and practice.** 2.ed. USA, Addison-Wesley Publishing Company, Nov. 1992, 1160 p. English.

Mortenson, M.E. **Geometric modeling.** John Wiley & Sons, New Yory, New York, 1985.

Nery, C.; Rangel, R.R.; Dias, L.A.V. **Digital Elevation Models Interpolated by Linear, Akima and Beta-Spline Methods: A Comparison.** International Archives Of Photogrammetry and Remote Sensing, Vol. 30, Part. 4, pp. 416-420, 1994.

Norton, P.; Yao, P.L. **Borland C++ programing for windows.** New York, New York, Bantam Computer Books, 1992, 746 p. English.