Signal Processing, Firenze 1978

THE SHADING OF TRUE RELIEF DATA

by H. Hügli

Institute of Applied Physics
Swiss Federal Institute of Technology
CH-8093 ZURICH, Switzerland

Abstract

In this paper a comparison of two methods for the generation of interpolated and shaded images is presented which deals with the computing costs for the process and the visual appearance of the image. Practical results are shown using spline interpolation and topographic relief data.

Introduction

The relief data we used consist of the altitude samples of Switzerland in a uniform raster of 250x250 m (fig. 1). By applying a shading algorithm to this data set we get a simulated map showing a lot of informations relative to relief structure, orientation, illumination, shadow, etc. (fig.2). The continuous shading effect of figure 2 is created by the low-pass characteristic of the eye and disappears in the enlarged figure 4. Our aim is the restitution of the shade continuity by interpolation.

It must be mentionned here that the altitude samples were read from a topographic map (by many people in many days) and that we have no knowledge of the relief structure between two samples. Our problem is that of an artificial improvement of the resolution appearance and our judge is the eye.

There are basically two different possibilities to interpolate shaded graphics: 1) interpolation in the object space and 2) interpolation in the image space. The first approach is the natural one in that it is the method we would use if more samples were given. We expect good appearance improvement. The second method is faster and has shown good results in the real time generation of shaded geometrical surfaces [1]. We want now to compare both methods in their ability to satisfy our purposes.

Shading

There are many different rules for shading a surface [2]. In our application the shade value is:

$$s = \cos\theta \qquad (1)$$

where θ is the angle between a normal $\vec{n}$ to the surface and a vector $\vec{\ell}$ to the light source. This rule is simple and has the following physical meaning: s is the luminance of a diffuse white surface lighted from infinite. Calling h(x,y) the height of the relief surface and considering that cos θ is the scalar product of the unity vectors $\vec{n}$ and $\vec{\ell}$, we find:

$$s(x,y) = \frac{-\ell_x \frac{\partial h}{\partial x} - \ell_y \frac{\partial h}{\partial y} + \ell_z}{\sqrt{1 + \left(\frac{\partial h}{\partial x}\right)^2 + \left(\frac{\partial h}{\partial y}\right)^2}} \qquad (2)$$

This transformation from h to s has a derivative nature which comes from the presence of h in its first derivatives. In fact, s is the slope of the surface relative to the light source vector.

Interpolation

Because of their finite spatial extension and their nonnegative values the spline functions are well suited to our interpolation task. A two-dimensional spline function of class m has all its derivatives of order 1,2,...,m-1 continuous everywhere. Andrews and Patterson [2] show that for interpolating an image, at least a spline of class 1 (bilinear or triangle interpolation) is needed. Higher class splines give similar results. In other words our visual system needs an image with continuous derivatives of order zero i.e. with continuous halftone or shade.

Using these results for our purpose, we will expect that:
a) bilinear interpolation is needed in the image space,
b) interpolation in the object space with the bilinear function will lead to shade discontinuities in the image, due to the derivative nature of the shading process, which reduces the order of the interpolation function.
c) interpolation in the object space needs splines of higher class (2 or more).

There are two steps in computing the two-dimensional separable interpolation of the data array [G] of dimension NxN with splines [2]. First the multiplicative coefficients c_{ij} (which then will be used to weight the spline surfaces) are found from [G] and the constant interpolation matrix [A] with

$$[C] = [A^{-1}] \cdot [G] \cdot [A^{-1}] \qquad (3)$$

each matrix having the same size N x N.
The second step consists of weighting the spline surfaces with the coefficient c_{ij} and adding them. This is done by computing the interpolated array $[\hat{G}]$ for each raster cell (i,j) with

$$_K\left[\overset{K}{\hat{G}}\right] = {}_K\left[\overset{M}{S(x)}\right] {}_M\left[\overset{M}{C_{ij}}\right] {}_M\left[\overset{K}{S(y)}\right] \qquad (4)$$

where [S(x)] and [S(y)] are the constant spline matrices and $[C_{ij}]$ is the square submatrix of [C] comprising all the coefficients $c_{k\ell}$ which weight the spline surfaces touching the raster cell. The size of these matrices is given by the linear enlargement factor K and the spline class m (M=m+1).

Computation cost

As computation cost (or time) unit we shall use the one for a main operation (MO) such as a multiplication or a division. With this, we find the computation cost of (4) as (M^2K+MK^2). Computing (3) needs basically $2N^3$ multiplications. However, the dominance property of a diagonal band of elements in $[A^{-1}]$ allows us to ignore the further elements. Considering a band of 15 elements reduces the cost to $30N^2$.

In the special case of bilinear interpolation (m=1), [A] is the unit matrix and the cost of (3) is zero. Adding now the costs of both interpolation steps and dividing the total by the amount of interpolated values N^2K^2 leads to the total interpolation costs per pel

$$\Phi_i = \begin{cases} K + \frac{M^2}{K} & \text{for } M=2 \quad (m=1) \\ \frac{30}{K^2} + K + \frac{M^2}{K} & \text{for } M > 2 \quad (m > 1) \end{cases} \tag{5}$$

The computation cost of the shading (2) is largely dependent on the way the square root is executed. The use of an approximation technique [4] reduces its cost to about 10 MO. Expressing again the cost in terms of shading cost per pel gives

$$\Phi_{sh} = \begin{cases} 10 & \text{for object interpolation} \\ \frac{10}{K^2} & \text{for image interpolation} \end{cases} \tag{6}$$

The computation costs for various K are summarized in figure 3.

Results

Results are shown with a 32 x 32 object array interpolated to 512 x 512 pel images (fig.5,6,7) ie. N=32, K=16. Figure 6 is obtained by linear interpolation in the object space and shows the previous expected discontinuities. Although the presence of high frequencies gives a first impression of sharpness, this image is unable to show the true relief.
A visual comparison of figures 5 and 7 shows a similar rendering in the rather flat parts of the relief. However, only the object interpolation technique produces a good rendering of parts with higher spatial frequencies (fig.7). The image interpolation technique fails in this point (fig.5).

A comparison of computation costs shows the merit of the image interpolation technique which lies in an efficiency improvement with respect to the object interpolation by a factor of 256 (=K^2) with a bad shading algorithm and of 7 with the previous mentioned method [4] (fig. 3).

Conclusion

Object and image interpolation were compared and both showed their own merit: computation efficiency for image interpolation and the capability of maximum rendering of relief data for object interpolation. Beyond this, practical results confirmed spline functions of class 2 or more to be fitted to object interpolation.

Aknowledgements

The author wishes to thank Prof.E.Baumann for the support of this work and Dr.T.Celio for his constant encouragement.

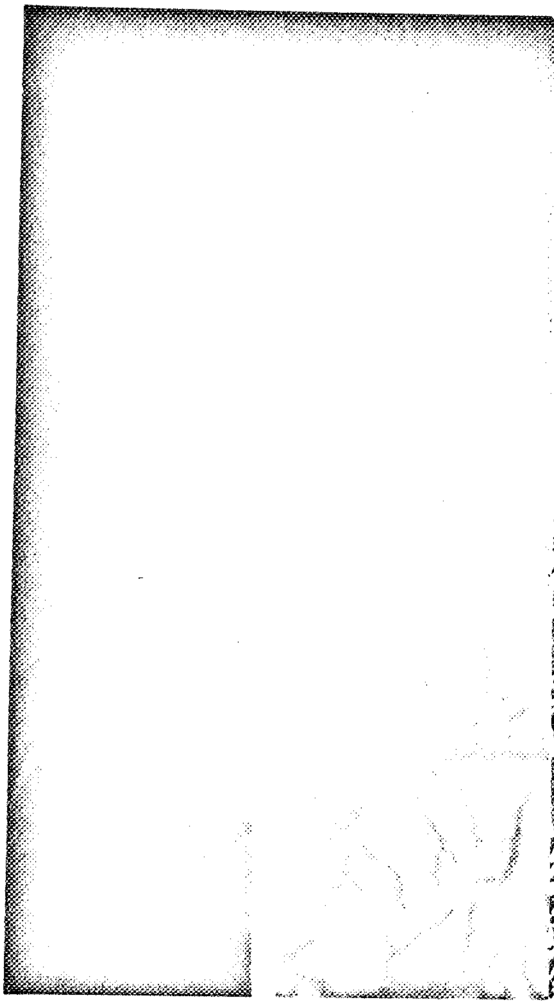
References

[1] H.Gouraud, "Continuous Shading of Curved Surfaces", IEEE Transactions on Computers, Vol. C-19, No. 3, pp. 623-629 (June 1971)

[2] W.M.Newman and R.F.Sproull, "Principles of Interactive Computer Graphics", McGraw-Hill, 1973

[3] H.C.Andrews and C.L.Patterson, "Digital Interpolation of Discrete Images", IEEE Transactions on Computers, Vol. C-25, No. 2, pp. 196-202,(Feb. 1976)

[4] H.Blaser und F.Braun, "Schnelle digitale Amplitudenbildung von Quadraturpaaren", AGEN-Mitteilungen, Nr. 17, Dez. 1974



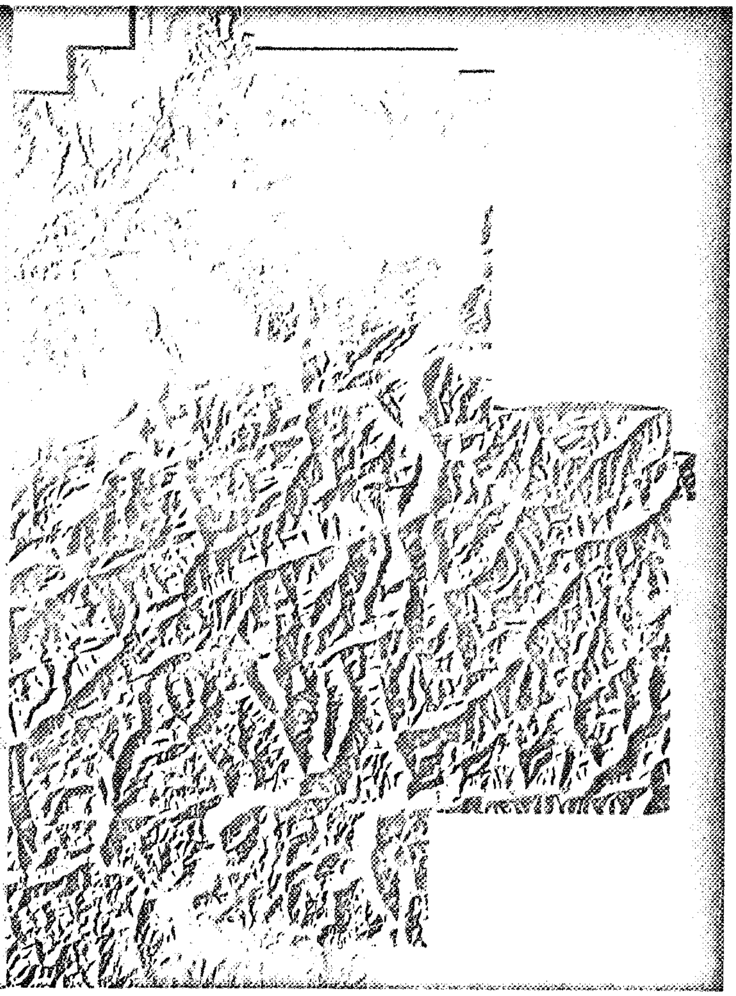
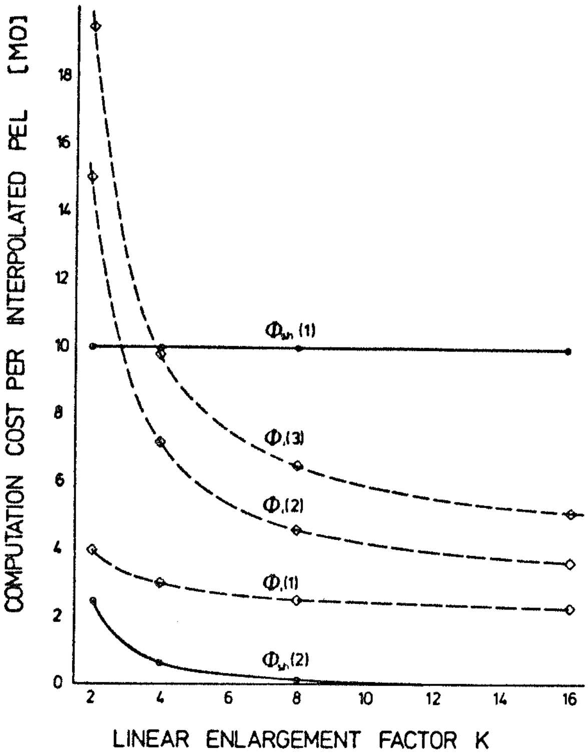


Fig. 1 Image of the altitude data of Switzerland. This digital array has 1000 x 1500 pels.

Fig. 2 Computer generated shadow map. The light source position is that of sun on March 21 at 8 am.



$\Phi_{sh}(1)$ shading cost for interpolation in object space

$\Phi_{sh}(2)$ shading cost for interpolation in image space

Φ_i (1) interpolation cost with bilinear function (m=1)

Φ_i (2) interpolation cost with quadric spline (m=2)

Φ_i (2) interpolation cost with cubic spline (m=3)

Fig. 3 Diagram showing computation costs to interpolate and shade a picture

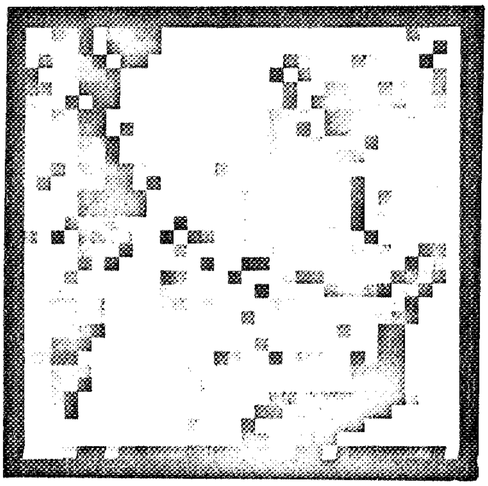


Fig.4

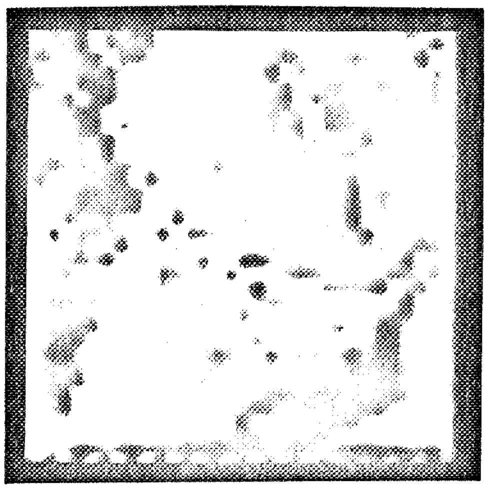


Fig.5

Fig. 4 Enlarged part of fig. 2. The 32 x 32 relief samples shown cover an 8 x 8 km domain.
Replication interpolation in the image space (K=16,m=0)

Fig. 5 Bilinear interpolation in the image space (K=16,m=1)

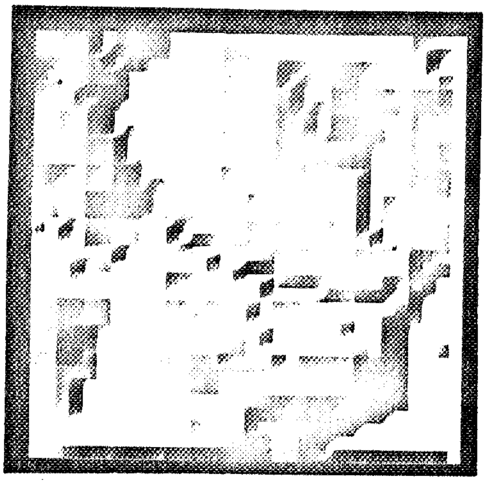


Fig.6

Fig.7

Fig. 6 Bilinear interpolation in the object space (k=16,m=1)

Fig. 7 Cubic B-spline interpolation in the object space (K=16,m=3)