Curvilinear color shading with DÉGRADÉ

V. Boyer, J.-J. Bourdin
Université Paris 8,
email: boyer@ai.univ-paris8.fr

Most pictures are drawn with smooth gradations of colors and include color shaded regions. To produce color shading the graphic designer may use many different methods: Partitioning [1], DDA algorithm [2], Automatic-airbrushing [3], Bi-linear interpolation [4]. In this paper we focus on an extension of Gouraud's method, the curvilinear-linear interpolation. This method is efficient and permits to design very quickly the general shape of a shading associated to a region (or an object in animation).

1 Notations

Hereafter, a pixel \( P(x_p, y_p, c_p) \) is an elementary element of the raster device and is given by its co-ordinates \( x_p, y_p \) and its color \( c_p \), this color may be an entry in a Color Look-Up-Table or a triple associated to a color model, \([P, Q] \) is the discrete segment between the pixels \( P \) and \( Q \), a contour \( C = \{P_1, P_2, ..., P_n\} \) is a set of \( n \) pixels \((n \in \mathbb{N}^+)\) 4-connected pixels, a region \( \mathcal{R} \) is a set of 8-connected pixels and is color shaded if filled with various colors without any noticeable color separation.

2 Model

A region \( \mathcal{R} \) will be defined by two edges \( \mathcal{E}_1 \) and \( \mathcal{E}_2 \), these edges respect:

\[
\forall P \in \mathcal{R}, \exists (P_1, P_2) \in \mathcal{E}_1 \times \mathcal{E}_2 / P \in [P_1, P_2]
\]

\[
\forall (P_1, P_2) \in \mathcal{E}_1 \times \mathcal{E}_2, \forall P \in [P_1, P_2], P \in \mathcal{R}
\]

If this decomposition is not possible the region will be partitioned into smaller parts. The edges are given by the user, drawn for example, the color of their extremities are also given. \( \mathcal{E}_1 \) is a set of \( n_1 \) pixels \((\mathcal{E}_1 = \{P_1, ..., P_{n_1}\})\) and \( \mathcal{E}_2 \) is a set of \( n_2 \) pixels \((\mathcal{E}_2 = \{P'_{1}, ..., P'_{n_2}\})\).

The new algorithm consists in two steps:

- to compute the color of each pixel \( P_i \) (and \( P'_{j} \)) of the edges \( \mathcal{E}_1 \) (and \( \mathcal{E}_2 \)),
- for pairs of pixels \((P_i, P'_j)\) of the edges to draw the color shaded line from \( P_i \) to \( P'_j \).

Three main problems have been solved and will be presented in the following sections, they are:

- The adequate non linear interpolation to obtain the colors of each \( P_i \) (and \( P'_j \)).
- The association of pairs of pixels \( P_i \) and \( P'_j \).
- The computation of \([P_i, P'_j] \).

3 Curvilinear interpolation

Let \( P(x_p, y_p, c_p) \) be a pixel of the edge and \( A(x_a, y_a, c_a) \) and \( B(x_b, y_b, c_b) \) be the extremities of the edge. In Gouraud's algorithm [4] the color of \( P \) is given by an interpolation between \( A \) and \( B \):

\[
c_p = \frac{d_c(P, B) \ast c_a + d_c(P, A) \ast c_b}{d_c(A, B)}
\]

where \( d_c \) denotes the difference of ordinates.

In our method the \( d_c \) function is a curvilinear length. For example if the edge is a circle of radius \( \rho \), for \( P \) and \( Q \) two points of the circle, \((\rho, \theta_p) \) and \((\rho, \theta_q) \) being their polar coordinates, the circular distance \( d_c(P, Q) \) is:

\[
d_c(P, Q) = \rho |\theta_q - \theta_p|
\]

When the edge is a straight line, the curvilinear length is computed as in Gouraud's method.

4 Corresponding pixels

If \( n_1 < n_2 \) (number of pixels of \( \mathcal{E}_1 \) and \( \mathcal{E}_2 \)):

\[
\forall P'_j \in \mathcal{E}_2, \exists ! P_i \in \mathcal{E}_1 /
\]

\[
\approx i \approx j \frac{n_1}{n_2}
\]

This property allows the computing of the associated pairs. If \( n_2 > n_1 \) a symmetrical property is used. As this is a linear interpolation, a fast discrete line computation algorithm [5] is used.

5 The 3D line computation

Between each pair of pixels \((P_i, P'_j)\) the algorithm computes and draws a shaded line. The color can be given as a simple integer value (an entry in a Color LUT or a gray scale) or a triple in a color mode. In most shading process the extremities may have one or two common values. For example in the sphere (cf Fig.1) every line drawn is of constant Hue and Saturation. Therefore DÉGRADÉ automatically adapts to compute a 2D, 3D, 4D or 5D discrete line with the fastest known algorithm [5]. An efficient technique [6], based on projections is applied to the 4D or 5D lines.

6 Utilization of DÉGRADÉ

The new technique is very fast. Figure 1 presents an image produced with DÉGRADÉ.
The picture illustrates different kinds of color shading. The edges of the balloon are two bezier curves. The edges of the sphere are the point of illumination and a circle. The edges of the road are two lines. The trees are composed by two color shading defined by a bezier curve and a line. For example the definition of the road consists in a few clicks: the limits of the region, the infinite point and three points at the bottom of the picture. Then DEGRADÉ computes the road. This example illustrates how easy it is to use the DEGRADÉ color shading tool. The method is very intuitive and the shape of shading easily predicted.

7 Conclusion

A new method to produce color shading was presented. The DEGRADÉ color shading tool permits to produce easily different kind of effects. Moreover it is easy to produce any shape of color shading desired. This technique could be implemented easily in a graphic library.

References


