

Improved Rendering with *Dégradé*

V. Boyer, D. Sobczyk and J.J. Bourdin

G.R.I.S. Laboratoire d'Intelligence Artificielle
 Université Paris 8
 2, rue de la Liberté
 93526 Saint-Denis Cedex
 FRANCE
 boyer, dom, jj@ai.univ-paris8.fr

Abstract

In most paintboxes drawing (and filling) the picture and render the picture are two different steps and tools. Dégradé presents an efficient filling rendering tool: an efficient collection of effect filling. In Dégradé the renderer is included in the drawing process. This method enhance the renderer possibilities and the graphic designer work.

1. Introduction

In raster graphics the filling of a region is merely assigned to produce uniform coloring or very simple effects (such as the filling with a regular pattern or a concentric shading effect). If more spectacular effect is implemented it is included in another tool: the renderer (like Render or Artistic in The Gimp¹). The renderer tool works through the transformation of an original picture^{2, 3, 4}.

For the graphic designer such a schizophrenia is unproductive: the “composition” of the picture strongly depends on the final rendering and the rendering has to be applied to each different object drawn. For example a blurred effect may be used to produce an effect of farness if applied to an object. It produces a very different impression when applied on the whole image.

In *Dégradé* the renderer is included in the filling tool. Therefore whenever an object is drawn it is drawn with its own rendering.

Furthermore if the renderer is applied to the whole image each object is subjected to the same effect. With our model each object is subjected to its own effect and different rendering can be used on the same picture. For example a precise, even photorealistic, object may appear on a blurred background. This enhances significantly the possibilities offered to the graphic designer.

2. The Model of *Dégradé*

For each region \mathcal{R} to fill, the user of *Dégradé* choses one **inner shape function** and one **effect filter**.

- For each pixel P of \mathcal{R} the inner shape function computes an integer value, the *index*.
- For each index the effect filter computes the adequate color. This computation may be achieved interactively if necessary.

This model was inspired by Bourdin's⁵ but is closer to the automatic airbrushing presented by Williams⁶ (using this method, the graphic designer defines separately a region and a color shading, the automatic airbrushing consists to apply the shading to the region).

Using this model, the region can be filled with a regular pattern, a uniform color, a color shading, or many NPR effects: watercolor⁷, line art⁸, silhouette rendering⁹, impressionist effects¹⁰, cubism, pointillism¹¹...

3. The new Collection of Effect Filters

We have implemented a large collection of effect filters. They will be presented from the simplest and more usual to the most sophisticated and original:

- a linear interpolation between two entries of the Color LUT. This allows smooth color shadings. The inner shape function if applied by itself fills the region with isochromatic subsets. Each subset respects th *inner shape* chosen.

Therefore any kind of shading is possible: linear, concentric, elliptic, angular. . . Whatever the object to show the graphic designer is able to design a fine photo-realistic picture of it.

- a bilinear interpolation shading tool. This is an extension of the bilinear shading first presented by Gouraud². Two different directions of interpolation are given by the user. On each direction the user specifies the kind of shading to be used. The bilinear interpolation tool then computes the composition of the two shadings. For example the user may design an horizontal shading on the Hue value (HLS model), to draw the picture of an object of various color, and a vertical shading on the Light value to give an impression of volume on the object. For example in a HSV view the saturation will be used to present a worn-out effect while the value will be used to show the volumes. With this tool it is easy to design a natural impression of clothes.
- a Bezier-like interpolation tool. We even extended these interpolation to the Bezier's curve. Each component of the colors (Hue, Saturation and Value in HSV for example) is computed through a Bezier curve. Therefore for each component two intermediate values are to be given. This tool allows to define most precisely smooth shadings. For example photographs of objects show color ramps continuous but not linear. Figure 1 presents the shadow along the side of a teapot. It is accurately presented by a Bezier curve. An interactive tool helps the graphic designer to interactively build the appropriate Color LUT. He can specify the colors and let compute the interpolation between two colors of the Color LUT. This interpolation could be linear or based on a Bezier's curve. For example let i and j be two index of the Color LUT. Let c_i be a primitive in a color model. We have:

- using a linear interpolation:

$$\forall k \in [i, j], c_k = c_i + (c_j - c_i) \times \frac{k-i}{j-i}$$

- using a Bezier's curve:

$$\forall k \in [i, j], c_k = [\mathcal{B}_k(i, j, l, m)]$$

where \mathcal{B}_k is the point of the Bezier's curve defined by i, j, l and m which are respectively the starting, the ending and the two intermediate points.

- a gradient effect. As the non-isochromatic region filling builds a map of values it is easy to deduce the gradients of these values. As presented by Litwinowicz¹⁰ the gradients are useful informations. Adapting his model to our graphic tool we have been able to produce impressionist (see figure 4), expressionist or stained glass effects (see the bottom of the figure 5).
- a water color effect. Water color effects such as those presented by Curtis et al.⁷ are easy to produce as a simple filter while coloring the region. This filter is based on a free

spreading of brush strokes. The picture 3 presents such wet-in-wet painting effects.

- a local modification effect. Local modifications of colors can be used to enhance details or obtain fuzzy effects.

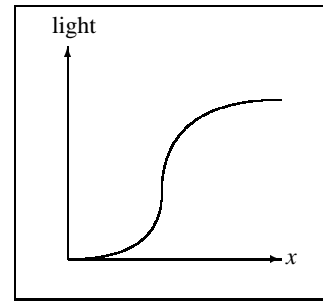


Figure 1: A Bezier-like interpolation

Using chromatic functions, we were able to enhance the color filling possibilities and generate various kinds of effects. For each region one filter is applied. Then the graphic designer can focus on another region, design its form, choose its inner shape and filter and look at it on the CRT device (see the many different effects of figure 5). His work is therefore easier and more accurate. It is even possible to apply different filters to different regions (a cubist monkey on an impressionist tree ?) of the picture.

4. Conclusion

It has been proved that the renderer tool has to be included into the region filling tool: this enhance renderer possibilities and the graphic designer work. The enhancement of a classic model was sufficient to produce NPR pictures. Paint-boxes are therefore proved to be well-suited tools to design any kind of picture. Other kinds of effects like pen-and-ink illustrations, or hand drawing effect are possible.

References

1. P. Mattis and S. Kimball. The gimp v.1.2.1, January 2001. 1
2. Adobe Systems. *Adobe Photoshop 5.0 Classroom in a Book*. Adobe Systems Incorporated, 1998. 1
3. Adobe Systems. *Adobe Illustrator 8.0 Classroom in a Book*. Adobe Systems Incorporated, 1998. 1
4. Deneba. Canvas 7, 2000. 1
5. J.-J. Bourdin and J.-P. Braquelaire. Color Shading in 2D Synthesis. In *Proceedings of Eurographics'90*, pages 41–49,547–548, 1990. 1
6. L. Williams. Shading in Two Dimensions. In *Graphics Interface'91*, pages 143–151, 1991. 1

7. C.J. Cassidy . Curtis, S.E. Anderson, J.E. Seims, K.W. Fleischer, and D.H. Salesin. Computer-generated watercolor. *Proceedings of SIGGRAPH 97*, pages 421–430, August 1997. [1](#), [2](#)
8. G. Elber. Interactive line art rendering of freeformsurfaces. *Computer Graphics Forum (Eurographics'99)*, 18(3):1–12, September 1999. [1](#)
9. J. Buchanan and M. Sousa. The edge buffer: A data structure for easy silhouette rendering. In *NPAR 2000*, pages 39–42. ACM SIGGRAPH, june 2000. [1](#)
10. Peter Litwinowicz. Processing images and video for an impressionist effect. *Proceedings of SIGGRAPH 97*, pages 407–414, August 1997. [1](#), [2](#)
11. Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. *Proceedings of SIGGRAPH 98*, pages 453–460, July 1998. [1](#)
12. H. Gouraud. Continuous Shading of Curved Surfaces. *IEEE Transactions on Computers*, 20(6):623–629, June 1971. [2](#)



Figure 2: *Original mandrill*

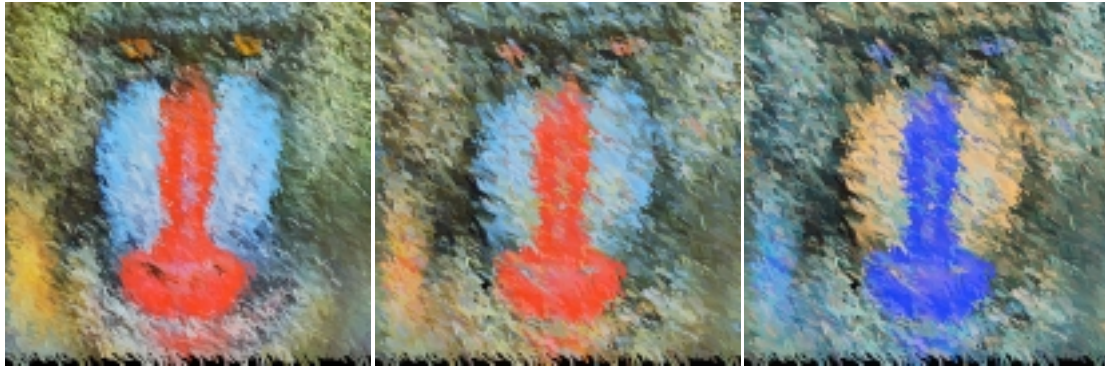


Figure 3: *Painterly rendering with color variation*



Figure 4: *Cubist, Impressionist, pointillist effects*

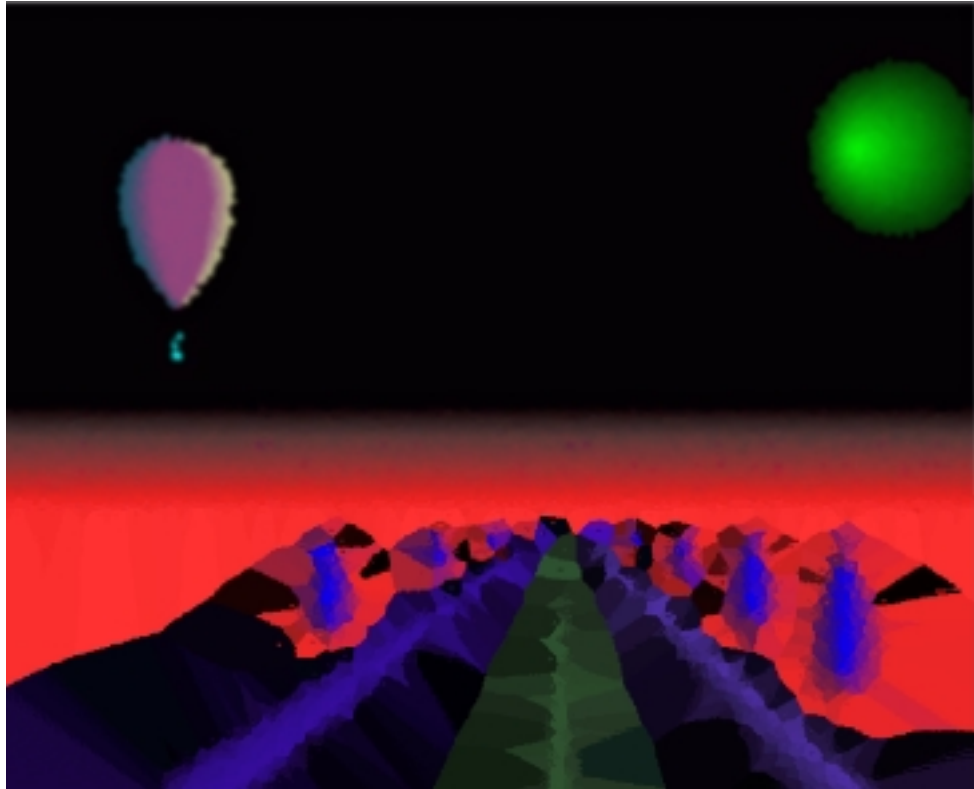


Figure 5: *Turn the page*