Hervé Huitric / Monique Nahas
The visual artist turns to computer programming

Our experiments in Computer Art began in 1970. Starting from small computers, very simple algorithms with abstract pointillistic pictures, we went through figurative 2D drawings and are at this point ' using and developing 3D realistic techniques. This personal itinerary has followed the improvement of our equipment and especially our own relationship to this equipment. Among other things, it always takes us the same time to produce an image, no matter how sophisticated our computer is. In fact, we cannot resist the urge to add new operations as soon as we gained some time. Techniques pile up, we realized that they can be used, even unexpectedly, again and again. For example our old practice of mixture of pictures, pointillism, filtering techniques is still available. Also, our aesthetic desire, despite a tentacle geometry, is seeking for light and colors. We believe that some baroque hides into the algorithms. The artists are there to help finding it.

I. Without Raster: 1970–1974

We began in 1970 with only a small computer and a line printer. Our wish was to work on continuous variations of colors. We started with a pointillistic approach. Since we could not produce directly a complex color, we assume that it should result from the optical addition of primary colored points as in printing. We divided the picture into square blocks, computed the complex color of each block corresponding to a given variation of colors, and then realized it by a statistical distribution of elementary points indicated by a letter. The listing was a collection of letters describing the picture. Afterwards, we painted by hand a little square block around each letter, a very tedious activity.

The percentage of basic colors in each square block was determined by several relations:
– a linear relation giving the brightness of the block by addition of the brightness of each colored point,
– some continuous variations for a subset of colors corresponding to a given repartition of level curves on the pictures.

The images produced by this pointillistic method present a granulated structure due to the distribution of the elementary points: The choice of different random algorithms determine different kind of clusters of points, thus different visual structures. It was what we called the texture, and we used it as an element of the composition.

To summarize our first pictures, they were built around two elements:
– continuous variation of colors defined by a set of level lines,
– a texture associated to the pointillistic realisation.

Other equipments = other realizations: punch cards and Calcomp plotter.

In 1972 we found an IBM 1130 and a punch card. Instead of painting each colored point by hand, we produced, as a result of the program, a series of punch cards to be used as stencils. Each punch card had a code defining its position in the picture and its corresponding color. The rest of the card was used as a stencil for a color. To realize the picture we applied the punched card to the picture support, and used a roller to spread the corresponding color. Since it was faster than painting by hand, we could experiment a little more the texture's variations. But anyway realizing one picture after having produced the punched cards could take one or two weeks.

After that, in 1973, we got a Calcomp plotter and used it to outline stencils for silk-screens, keeping the same ideas for the color programmation.

The plotter was used to print one stencil for each basic color of the serigraphy, and we produced the silk-screens with three basic colors (magenta, cyan and yellow) and with three stencils for each color. The silkscreen was obtained after nine layers of colors. We began to put out multiple products from one program, and to play with permutations of stencils and colors.

II. First Raster: 70 x 56 Pixels, 4096 Colours 1975–1978.

In 1975 we got our first raster. It had a very low resolution, 70 x 56 pixels, but also 4096 colors, 16 levels for the red, the green and the blue. So we abandoned the pointillistic method which has no meaning with such a small number of pixels, and we focused only on color variations.

For technical reasons it was only possible to use the Lisp language and integer number, with a 16 K computer. So our computations were based on straight lines and circles, using the recursive property of Lisp language to combine elementary structures.

We first constructed continuous variations of colors delimited by some simple shapes: rectangle, triangle. Each color, R, V, B had a monotonous relation, increasing or decreasing through the surface. With these basic elements, we experimented various iterations, keeping the structure of each elementary surface. visible.

Then in order to diminish the strict geometry of the pictures, we began to use the iteration of elementary shapes in a non visible way. We tilled a surface by successive triangulations, keeping the same values of colors at each border.

In this way we obtained folding or depth effects where the construction process by triangulation ceases to be noticeable. Simultaneously we introduced a more flexible relation between the colors and the lines of level, increasing the variations of colors. Then we could obtain color peaks on the picture.

Mixtures and Sequences of Pictures

It became possible to develop a series of picture transformations because a picture could be stored and modified easily, as opposed to the previous situation without raster. The mixture of pictures is a very simple but efficient artistic tool. It is probably as common for people working with computer as for traditional artists mixing their colors. The only condition is to be able to store and retrieve the pictures. With a raster we had all these new facilities.

Our first experience was the simplest: combining two pictures by a barycentric function: $f (x, y) = a . x + (1 – a) . y$ with $0 <a< 1$, x, y are the color values of a given pixel i, j in the two initial pictures, and $f (x, y)$ is the value of that pixel in the mixture. Effects of transparency are easily obtained that way, but at that time we were more interested by the consequences of the mixture on the shades. A surprising result of the mixture was that we could always observe an improvement in the perceptible amount of tints.

By varying the coefficient a from 1 to 0 in the previous relation, we obtained a sequence of pictures, going continuously from the first to the second picture. Reciprocally, any picture could be identified as an element of the series: for each picture we could determine a family tree which was made with light and dark filtering of this picture. As many kind of mixtures are possible as you can imagine different formulas to do it, all you need is a function applying two colors on a third one. For example we used the formula $f (x, y) = V x . y$ which increases the proportion of black in the mixture, or the formula !x–y! or !15–x–y! using the complementary color. Since we had only 16 values for each basic color, each continuous function from [0,151 x [0.15] to [0,15] could give a continuous mixture, thus keeping the continuity of the color variations.

Pictures Transformations

At that time, the development of digital music was very impressive compared to our simple experiments. (And the 30 developments were too far from us.) we were wondering if the powerful techniques of the musicians generating the sounds, could be of some use for us. So we began to look at the pictures in term of frequencies (spatial frequencies instead of temporal ones). Using a Fourier development in trigonometrical functions of i and j, it was easy to build smooth variations of colors. Borrowing the ideas of a musician, Chowning, we introduced a modulation of frequencies and could observe a large diversity in the corresponding variations of colors following the parameters of the modulation and some effects of vibration of colors.

In a reverse sense, a given picture could be developed in a Fourier series, exactly as a given sound has a precise content in harmonics. Acting on the Fourier coefficients was another way of transforming a picture. The only problem was to compute the Fourier transformation with integer numbers and in Lisp. We probably realized the longest FFT transformation which could be imagined. It was then natural to be interested in the possibilities of techniques coming from digital image processing. We used Fourier or Walsh expansions to realize filters of high or low frequencies. The transformation of histogram was an easier technique that we found very interesting. By egalisation of histogram, we reduced the number of Color levels present in an image in a way that accentuates the forms.

None of these techniques broke the continuity of the variation of colors. We used to combine them in various ways.

2 Dimensional Drawings 1979–1981

In 1979 we got a new equipment: a LSI 11, with 24K memory and a raster of 380 x 255 pixels with the same number of 4096 colors. With these new programming facilities (real numbers and Fortran language), we first produced some other continuous variations of colors, moving the brightness through the picture and computing the three colors R, V, B of each pixel by a linear relation: R+V+B=L (with L=brightness) as in our first pictures or silkscreens. For the first time, it was easy to draw a curve, without all the previous constraints working on a very small computer with only integers values. So we began to introduce more complex curves, and instead of playing with straight lines and circles, we brought in figurative elements in our pictures. Two draw 2-dimensional shapes, we chose to use parametrical curves instead of analytical curves, because it was a convenient way to escape from a rigorous geometry. The particular use of B-Spline curves was not only a practical choice but also an aesthetical one. These curves have inherent continuous properties producing a smooth aspect. They also have some useful locality properties. To draw a B-Spline curve or surface, you only have to give the x y coordinates of some points, and the complete curve is determined by these points, called control points, without going exactly through them. If you move one of these points, only a corresponding part of the curve will change, so you can modify your drawing locally. Surfaces are constructed in an analogous way by a dense network of curves. The way of filling the surface becomes a new parameter of density: on some pictures (a hand for example) we chose to make a visible distribution of curves on the surface which produces a kind of net. It is also possible to fill partially the surface, computing only some points and producing only a distribution of colored points.

The coloration of these surfaces was first a simple extension of our previous computations of the colors. We used the same parametrical approach for spreading the colors on the surface as for the computation of the geometry. Each control point was associated with a value of brightness. Thus during the computation of the surface, a brightness was computed for each point, and the set of colors R, V, B was computed as before, using the brightness and the variations of two colors in order to determine the third one.

We have applied this procedure to reconstruct images from digitized images, poorly' defined by eight levels of grey and 256 x 256 pixels. The control points were given by a grid on the picture, and the brightness attached to each control point was the corresponding value of grey of the digitized picture. Thus the computation of a B-Spline surface with these control values automatically produces a smooth interpolation of the brightness. Extending the idea of a non geometrical control value, it was possible to add many non geometrical parameters to each control points, and as a result, we obtained a smooth variation of these parameters on the surface. For example, a set of colors can be given with each control point, afterwards the colors were directly computed for each point of the surface, and continuously diffused.

Now all our previous experiences with abstract pictures could be repeated in that case. In particular the effects due to mixture of pictures were still interesting, producing a new amount of shades and hues. All the enhancement techniques could be used again, facilitated by an easier programmation. We began to add some other "post-treatment", in particular a pointillistic treatment which was possible because of the larger number of pixels. To obtain a pointillistic effect is very simple: add in any way a random perturbation to the computed values of the colors. Of course there exists as many possibilities to do that as you can imagine. On many pictures we used the simplest way, adding a random of given variation separately to the three colors. Another possibility is to give a random variation to the brightness only, keeping the computed hues and saturation or we could give a random perturbation to the hue and saturation, keeping the same brightness. It is also possible to change the random distribution through the picture, some regions could be more pointillistic than some others, with a smooth transition.

Again all these treatments will be used in the following, on the 3D pictures.

IV. Modelization of 3D realistic shapes

Since 1980 we have extended our computation of B-Spline surfaces to 3 dimensions. The technical details are given in the following section.

A 3D B-Spline surface is modelled from a network of control points, and the modeling task consists of finding the coordinates x y, z of these control points. This is more difficult than in 2D. In 2D drawing, control points can be easily selected by placing the drawing on a graphic tablet. The locality of B-Spline is well adapted to interactive drawing. Modifying a control point will only change a local region of the curve. With some practical experience drawing using control points is rather easy.

In 3D, the modeling is still a problem which has to be solved by various appropriate techniques, both theoretical and experimental. If certain shapes, such as water, mountains, hills or grounds can be easily approximate by some mathematical functions, the situation is of course different for a body or a face. Following the possible equipment, we used a number of different "ad hoc" methods.

One archaic but possible method consists in hand-drawing two views of the model, a front and a profile view, and to measure the coordinates of the selected points in these views.. In 1981, we made a head in that way. We had to spend several days manipulating manually the control points to achieve a satisfactory result, and got several interesting monsters during that time …

Algorithms can help the manipulation. For B-Spline surface, the so-called "OSLO algorithm" proved to be a very precious tool. By allowing to introduce new control points without changing the surface, it provides a supplementary means for modifying objects locally in the regions where they are richer in details, because the neighborhood moving with a control point becomes smaller.

In 1982 to model a dinosaur, we started with a wood skeleton. Still by hand, we measured the coordinates of points along the spinal column, and on various plane cross sections, with a corresponding rotation angle for each slice (only one was necessary). By positioning the slices appropriately with respect to the spine, we could compute the three coordinates of the chosen control points. The spine/slices combination had the advantage of being rapidly adaptable for a different position: this only requires changing the spinal column position and the orientation of slices, keeping information obtained in the internal system of reference of each slice.

Afterwards we used different 30 models in plaster, clay or other material, and covered it by one or several grids of curves. Then we measured the points of the grid either manually, either automatically. It remains to find the control points corresponding to these measurements. For example, using an automatic system developed by the car company Renault, we got the set of control points of corresponding Bezier surfaces. In case of patches made with only 16 points, Bezier and bicubic B-Spline are the same surface. We used the Bezier collection of control points to produce a face corresponding to a given plaster model.

We can also directly use the selected points as control points. The difference is not always noticeable. In any case, after you got your collection of control points, you still have a certain amount of work to do. Some examples of the difficulties which can occur are the following: if the object is divided into several B-Spline patches, how ensure the continuity between them? Even if the patches are correctly joined, a discontinuity of their tangency will produce a discontinuity of the brightness at the borders, a very disagreeable result. A solution is to work on the control points until the different patches will be continuously linked. A lot of algorithms and programmation work are helpful in order to achieve that work.

Another connected problem is the modeling of tree structures: since B-Spline surfaces are not interpolating surfaces, how can we make sure that a branch will be properly attached to a trunk, and how can we blend the attach? The OSLO algorithm is again very useful here. As a consequence of this algorithm, it is possible to cut a B-Spline surface into two B-Spline surfaces, and to cut it at different places. The two resulting surfaces are perfectly smoothly linked, and if they are displayed, they produce the same shape than the initial surface. We used this property to concatenate and blend two surfaces. For example we add a "branch" surface to a "trunk" surface in the following way: we cut the trunk in two at a chosen spot. The control points of the "branch" are added to those of the chosen part of the "trunk", thus constituting a same surface. Then we display that surface together with the second part of the trunk. Processing recursively, we produced some trees in that way, starting from two surfaces: one for the trunk and one for a branch. We used the same ideas to attach the legs to a dinosaur body: the body was one B-Spline surface and so are the four legs. To concatenate the first foreleg, we cut the body into two parts, then we juxtaposed the control points of the foreleg with those of the first part, and we displayed the corresponding surface as a whole. The process continues by cutting the rest of the body, juxtaposing the new leg and displaying it. Finally, the last part, the tail, was displayed alone.

Improving the methodology of data acquisition is of fundamental importance. Certainly we benefit from the possible accumulation of data, which can always be used in different work, but we are far from an easy convenient procedure to acquire the data, even with the help of programming systems of assistance. Consider for example, the construction of a movie, and think of how many different objects are needed to keep an audience from being bored after a few minutes. Look at any photograph or any picture at the TV, and consider how many objects should be modeled in order to approach their simulation …