Implicit Surface Driven by an Atlas of Discoids

Didier Arquès, Sylvain Michelin and Benoît Piranda
Université de Marne la Vallée, 5 Boulevard Descartes, Champs sur Marne
F-77454 Marne la Vallée Cedex 2

Abstract

This paper proposes a new approach for surface modeling. Considering that a surface is described by an atlas of overlapping disk-like patches, this model creates a joining set of implicit surfaces in order to obtain the resulting whole surface. Topological constraints linked to classical models decrease and the geometry of complex surfaces can be easily and naturally described.

Keywords: *Surface modeling, implicit surfaces, atlas of disks.*

1. INTRODUCTION

Computer graphics techniques propose a large choice of solutions for surface modeling.

The most widely used method consists in defining any surface by a polygonal mesh [6]. Even if a classical structure like the winged-edge data structure [14] allows to represent efficiently such a mesh, many drawbacks exist. In a rendering process, visual artifacts due to preponderant directions (the edges of the polygons) appear. Manipulation like patch subdivision becomes complex (see for instance [9] for a complete overview about solid modeling) because topological constraints between neighboring patches have to be maintained. We can also point out that similar topological problems appear in the reconstruction process (using a mesh) of surfaces from a set of points [5].

An interesting alternative consists in using Spline [3] or Nurbs [12] surfaces which model any surface by a collection of piecewise-polynomial patches instead of previous planar ones. More complex surfaces can be modeled but the main drawbacks concern continuity constraints in the junction points [15].

Implicit surfaces give an efficient solution to this continuity problem. This model consists in defining a surface as the solution of an implicit equation. The classical approach consists in constructing a skillet which drives the implicit surface, through for instance the use of blobs [4]. This approach suffers from three serious drawbacks. The first one concerns the impossibility to precisely imagine the resulting surface. The second one concerns the obligation to only describe a surface as the extern contour of a 3D object and the last drawback is that the resulting objects still appear very smooth.

Recently, different interesting works aim at combining the previous approaches. Szeliski and Tonnesen [13] propose to use disks or particles to represent the mesh describing an implicit surface. Oriented particles interact each other according to repulsion and attraction forces to automatically treat modifications of the surface (split, join, extend). Indeed, Witkin and Heckbert [16] use oriented particles to regularly sample an implicit surface.

In this paper, we present a new geometrical model which describes a continuous surface by joining pieces of implicit surfaces which are driven by an atlas of discoids [1]. The surface is just defined by a set of disks that can overlap each other. A merging operator is used to define an implicit surface in each overlapping area. The choice of this merging operator allows to control the continuity of this surface while the natural modeling approach permits to describe easily a large variety of surfaces.

Section 2 describes this new geometrical model, while section 3 discusses the advantages and drawbacks of this new approach. Section 4 gets on implementation considerations in a ray-tracing approach before showing the results in section 5.

2. A NEW GEOMETRICAL MODEL

In this paper, we propose to construct a surface just as a painter draws a sketch in 2D before precisely drawing the final curve as shown in figure 1. By extending this approach in 3D, the surface is reconstructed from a set of disk-like patches. The central point of this approach is that these patches can overlap each other i.e. create one kind of overlapping areas which are used to reconstruct a local implicit function.

So, defining such a surface with a set of disk-like patches imposes few constraints. For example, it avoids completely topological constraints associated to a classical mesh. On the contrary, overlapping areas are welcomed. We just have to place disks in the area we want the surface to exist and we choose the geometry and the size of each disk in order to obtain a "total covering" of the surface. The surface is just described from a set of couples : position and normal of each disk.

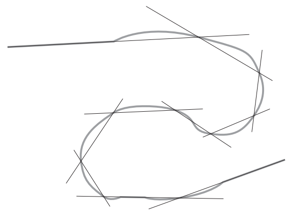

Figure 1 : sketch associated to a 2D function.

2.1 Theoretical background : the Atlas System

On a given surface S, the atlas system allows to reconstruct any interest function F, for instance a temperature or a luminance function, from a atlas of discoids. We define an atlas of discoids as a set of N disks-like patches $\{D_i, i=1\ldots N\}$ that cover entirely but approximately the surface S without taking into account overlapping problems. The term of disk-like patch or discoid is used because it looks like a classical patch but with the ability to overlap the others and because its shape admits some properties of a disk. The classical polygonal mesh may be seen as a limit case of this approach where overlapping areas tend towards the empty set.

Only for understanding reasons and notation simplifications, we just consider in this section the case of a planar surface *S*. Any point *M* of *S* is covered by a subset of discoids as shown in figure 2. Then the value of the interest function *F* in *M* is defined by :

$$F(M) = \sum_{i / M \in D_i} \alpha_i(M) F_i(M) \tag{1}$$

where :

- $F_i(M)$ is the local interest function defined for each point of the disk D_i covering *M*;
- α_i is a merging operator, defined and positive on each disk D_i, and which verifies $\sum_{i / M \in D_i} \alpha_i(M) = 1$.

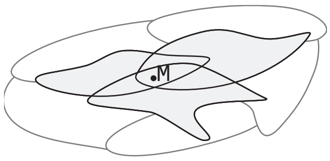

Figure 2 : only gray disks are used to define the interest function in M.

The choice of the geometry of the disk-like patches is relatively free. As shown in figure 3, several shapes of planar discoids may be defined: disk, square, equilateral triangle, star, and more generally we can use many non-planar shapes as for exemple segment of a sphere. In this paper, we only use two kinds of planar discoids: disks and squares.

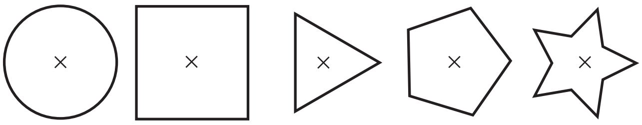

Figure 3: several shapes of discoids.

The choice of the function α_i for each disk D_i is also relatively free. It can be seen as a decomposition of the interest function in a function base with a local geometrical support: the discoid [2]. An interesting method consists in choosing any set of positive functions β_i and to define α_i by:

$$\alpha_i(M) = \frac{\beta_i(M)}{\sum_{j / M \in D_j} \beta_j(M)}.$$

For example, the β_i functions may depend on the distance from *M* to the center of the discoid. By choosing a function β_i which varies continuously from 1 in the center to 0 in the border of each disk, it is easy to verify that the merging operator runs as a smoothing operator only in the overlapping area. On the contrary, we obtain a discontinue function by choosing a β_i function which is not null on the border.

If we consider now the general case, the disks do not exactly cover the surface *S*. We have to associate to the point *M* and for each disk D_i a point M_i. Equation 1 becomes:

$$F(M) = \sum_{i / M \in D_i} \alpha_i(M_i) F_i(M_i) \tag{2}$$

2.2 A new Model

In the previous section, we briefly present how the atlas system reconstructs any interest function from an atlas of discoids placed on a given surface *S*. In this section we suppose that we only know the description of an atlas of discoids and the main problem consists in reconstructing the surface *S*. The main idea developed in this paper proposes to associate an implicit surface to the superposition areas generated by the relative positions and orientations of the discoids.

2.2.1 Superposition areas

In the simple case of planar surfaces, the superposition area between disks classically corresponds to their intersection. In the general case, we have to define what is the superposition area between two or more disks.

We define the superposition area *A* between *N'* disks as the set of points *M* which verify that the orthogonal projection M_i of *M* on each disk D_i exists. In other terms, a superposition area is the intersection between the *N'* infinite cylinders associated to the discoids as shown in figure 4 in the case of two discoids.

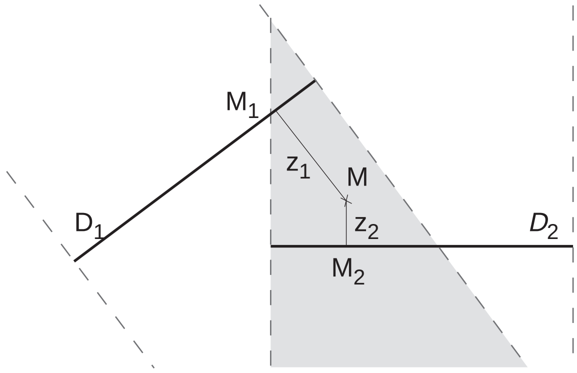

Figure 4 : overlapping area between two discoids.

2.2.2 Definition of the implicit surface in an overlapping area

In the following, we call *Δ* the subset of discoids associated to an overlapping area *A*.

Definition : In the overlapping area *A*, the surface *S* is the set of points *M* which verifies the implicit relation:

$$Z(M) = \sum_{i / D_i \in \Delta} \beta_i(M_i) z_i(M) = 0 \tag{3}$$

where z_i are the algebraic distances between *M* and their orthogonal projections M_i on each discoid D_i as shown in figure 4. The function $Z(M)$ represents an algebraic average distance between *M* and its projections M_i in every discoid of *Δ* and equation 3 an iso-surface.

Another important data is necessary to representing and rendering algorithms: the normal vector $\vec{N}$ to the surface in *M*. The normal to the surface *S* at the point *M* is classically defined by the gradient of the function *Z* evaluated at *M*:

$$\vec{N}(M) = \left(\frac{\partial Z(M)}{\partial x}, \frac{\partial Z(M)}{\partial y}, \frac{\partial Z(M)}{\partial z} \right)$$

3. ADVANTAGES AND DISCUSSION

3.1 General considerations

The main advantage of this approach is to be able to define any surface S as a collection of local implicit surfaces and to manage the continuity of the resulting surface only by the choice of the functions β_i.

As explained before, the surface can be constructed just as a painter draws a sketch in 2D by placing the discoids without any complex topological constraints. Adding a new discoid to improve locally the quality of the resulting surface only modifies the list of discoids in the atlas and locally the overlapping areas. Contrary to Spline description where junction problems impose the positions of several control points, the continuity of the resulting surface is just driven in a second step while choosing the β_i function as explained in 3.2.

Interesting properties can also be pointed out. For example, if Δ contains a single disk D_h, the previous relation is verified only if M belongs to D_h:

$$\sum_{i / D_i \in \Delta} \beta_i(M_i) z_i(M) = 0 \;\Rightarrow\; \beta_h(M_h) z_h(M) = 0$$

$$\Rightarrow z_h(M) = 0 \;\Rightarrow\; M = M_h \in D_h$$

3.2 Continuity of the surface and choice of the β_i functions

The choice of the β_i function affects both the shape of the surface in the superposition areas and junctions between these areas. It allows to control the continuity of the generated surface.

Defining any discoid D_i by its center C_i and its radius r_i, we choose a set of functions β_i which depends on the distance d_i between M_i and C_i. Choosing a function $\beta_i(d_i)$ which tends to zero when d_i tends to r_i implies a C^0 continuity of the surface when passing from an overlapping area to another.

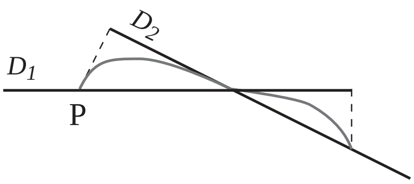

Figure 5: C^0 continuity of the surface.

For example in figure 5 the choice of the following C^0 function:

$$\beta_i^1(d) = 1 - \frac{d^2}{r_i^2}$$

produces a C^0 continuity in the border of the superposition area (in P).

More generally, for obvious reasons due to the definition of the implicit surface, the C^n continuity on the border of superposition areas depends on the fact that the nth derivative of β_i exists and tends towards zero when d tends to r_i. For example, the functions β_i^1, β_i^2 and β_i^3 plotted in figure 6 are defined in order to generate surfaces that admit respectively C^0, C^1 and C^∞ continuity.

$$\beta_i^2(d) = \left(1 - \frac{d^2}{r_i^2}\right)^2$$

$$\beta_i^3(d, a) = \exp\left(\frac{a \times d^2}{d^2 - r^2}\right)$$

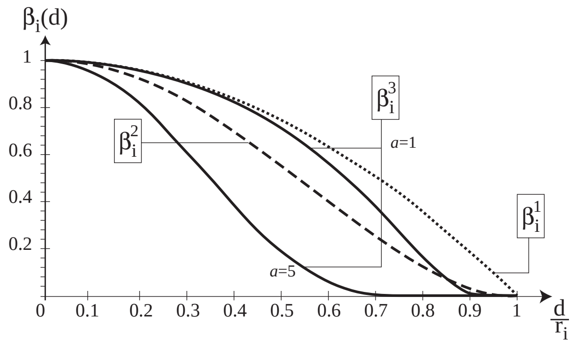

Figure 6: four different β_i functions.

In the third case, $\beta_i^3(d, a)$ uses a parameter a that gives a complete class of functions admitting the same continuity properties.

Figure 7: reconstruction of surface using several β_i functions.

The figure 7 shows four images representing the reconstruction of a surface described by the same atlas of disks but using different β_i functions respectively β_i^1 (a), β_i^2 (b), β_i^3 (c) with $a=1$ and β_i^3 with $a=5$ (d). In the first case (7a), the use of a C^0 β_i function produces some emboss effects. In the image 7b, the limits of the bumps are smoother. The image 7c shows a very flat surface due to the C^∞ continuity of β_i^3 while the last image (7d) shows the result of a surface generated with β_i^3 function with a=5. The shape of this function generates fast crossing from an area to another that produce interesting round edge effects between discoids.

4. VISUALIZATION CONSIDERATIONS

In this section, we develop several implementation details necessary to visualize a surface described by an atlas of discoids. Implicit surfaces are classically visualized using ray-tracing algorithms [7]. We develop in this section equations and algorithms which are used in order to answer the two following questions:

- how to compute the first intersection point M between a ray and the surface;
- and how to obtain the normal vector $\vec{N}(M)$ of S at M.

Then, we propose an algorithm that integrates these previous results and we give some details about texture mapping and optimization techniques.

4.1 A ray/surface intersection

Let $\{D_i, i = 1 \ldots N\}$ an atlas of discoids describing a surface S and a ray $[O, \vec{u})$. We just consider in the following the case of an atlas of disks. Similar approaches (and equations) can easily be developed with other shapes of discoids. Following notations of figure 8 that described a simplified geometrical configuration which contains only one disk, the coordinates of the intersection point M between S and the ray is obtained by solving the following system:

$$\begin{cases} \overrightarrow{OM} = k\vec{u} \\ \overrightarrow{OM} = \overrightarrow{OC_i} + x_i \overrightarrow{X_i} + y_i \overrightarrow{Y_i} + z_i \overrightarrow{Z_i} \\ \sum\limits_{i / D_i \in \Delta} \beta_i(M_i) z_i(M) = 0 \end{cases} \tag{4}$$

where k is the abscissa of P along the ray.

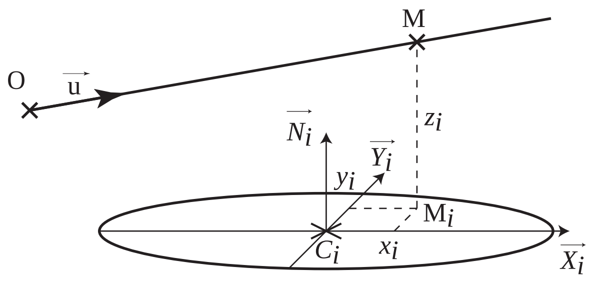

Figure 8: coordinates of *M* in D_i system.

Substituting the two first equations in the third one (by developing $z_i(M)$) gives a polynomial in k. The complexity of the resolution depends on the choice of the function β_i. For example, using β_i^1, it leads to the search of the root of a polynomial of degree 3, that can be analytically solved. In general cases, classical numerical techniques are needed.

Knowing the coordinates (x_i, y_i, z_i) of M, the normal to the surface is in the direction of the gradient of the surface S defined by the function Z evaluated at the point M,

$$\vec{N}(M) = \left(\frac{\partial Z(M)}{\partial x}, \frac{\partial Z(M)}{\partial y}, \frac{\partial Z(M)}{\partial z} \right)$$

The x coordinate of $\vec{N}(M)$ vector components is expressed by:

$$\frac{\partial Z(M)}{\partial x} = \sum_{i / M_i \in \Delta} \left(\frac{\partial \beta_i(M_i)}{\partial x} z_i(M) + \frac{\partial z_i(M)}{\partial x} \beta_i(M_i) \right)$$

Similarly expressions can be obtained for the other components $\frac{\partial Z(M)}{\partial y}$ and $\frac{\partial Z(M)}{\partial z}$.

The choice of β_i functions easy to derive implies that this equation generally admits an analytical expression.

4.2 Algorithm and implementation

We propose in this section an algorithm allowing to calculate the first intersection point between a ray and the surface associated to an atlas of discoids. Considering that the model defines an element of the surface for each superposition area, the ray $[O, \vec{u})$ is subdivided into a list of segments associated to each superposition area crossed by the ray. The algorithm runs into two steps:

- first it searches all the intersections abscissa between the ray and the cylinders associated to the disks and memorize them sorted out into increasing order.
- then it solves the equation 4 for each pair of consecutive abscissa until it finds a solution.

In the software we develop, the numerical resolution of the equation 4 is obtained using a classical false-position algorithm [10]. This method is well adapted to solve our equation because the function $Z(M)$ is monotonous for each interval, i.e. for each superposition area, using previous β_i functions. This program applies a classical optimization in order to reduce ray-surface intersection complexity. Each atlas is geometrically shared into a regular 3D grid of voxels (volume element). First we associate to a voxel V, the list of discoids that admit an overlapping area that crosses V. Then, the intersection calculation process is only applied to the list of discoids associated to the voxels crossed by the ray.

4.3 Coloring or texturing surfaces

Images presented in this paper are rendered using the Phong illumination model [11] and some surfaces are textured. We present below how to apply these classical illumination models in the implicit surfaces driven by the disks, and particularly how to associate a color to a point of the implicit surface defined in a superposition area.

For each intersected point M we compute the radiometric coefficients (diffuse and specular reflectance) of the Phong's illumination model by applying equation 1 where the local interest function is the radiometric coefficients supposed to be constant over each discoid.

Texture mapping is treated by applying the same equation using the position in the texture map as interest function. We associate a position P_i in the texture map to each point M_i of the disk. Then we deduce the final position P of M in the texture map by: $P(M) = \sum_{i / D_i \in \Delta} \alpha_i(M) P_i$

This method allows to deform the texture in order to regularly follow the geometrical deformation of the surface. Figure 10 shows an example of mapping of a wood texture (figure 9) in a cylinder approximated by some large disks. As the reconstructed surface is very embossed the texture appears deformed.

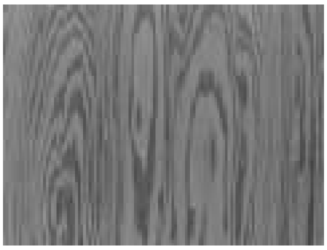

Figure 9 : a wood texture

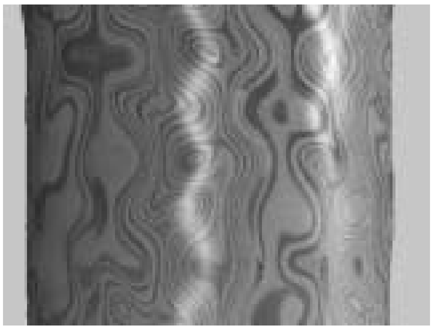

Figure 10 : an example of texture mapping with the previous wood texture.

5. RESULTS

The first image shown in figure 11 presents a scene entirely described with discoids.

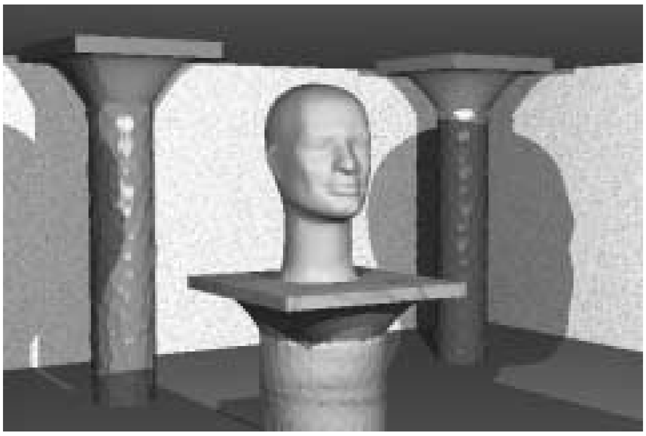

Figure 11: scene of the bust.

Disks are used in the interiors of the surfaces, and squares are placed in the border of some surfaces in order to obtain sharp angles. The columns in this image are constructed in three parts, the foot is a cylinder, the top is a block and in the intermediary area, disks are placed in order to regularly pass from a geometry to the other. The disks are placed with some random variations around a regular grid in order to obtain an impression of irregularity. The bust is obtained from a large database replacing 30000 small triangular patches by an equivalent set of covering disks.

The following figure (figure 12) presents two images of a detail of a face computed with the same atlas of discoids (about 4000 disks) but with two different β_i functions. The reconstruction method allows us to pass from a coarse and rough aspect to a smoother one without modifying the database.

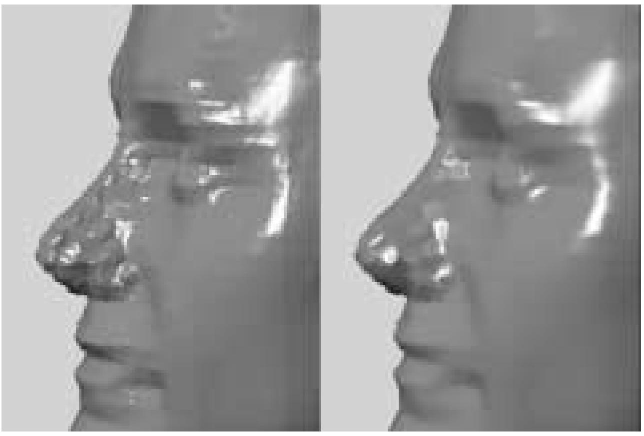

Figure 12 : detail of a face

5.1 Calculation time

The program, written in C++ language under Linux, runs on a bi-PII 450 MHz computer and uses multi-processes to share the calculation of the points of the images in the two processors. The calculation of the image of the figure 11 representing the large scene of the bust (about 50000 discoids and two light sources) takes about 10 minutes for a high resolution (1200x800 pixels). For a smaller scene as the face of the figure 12, the rendering takes less than a minute for an image composed by 600x900 pixels.

5.2 Modelisation tools

Some software has been developed to automatically modelize surfaces using atlas of discoids. An algorithm using oriented particles system that admit attraction and repulsion properties is for example used to regularly place disks over a predefined implicit surface like the sphere presented in figure 7.

Another software (cf. figure 13) generates an atlas of disks from a polygon mesh as the bust of figure 11. This tool allows to correct the atlas by moving and turning manually the discoids.

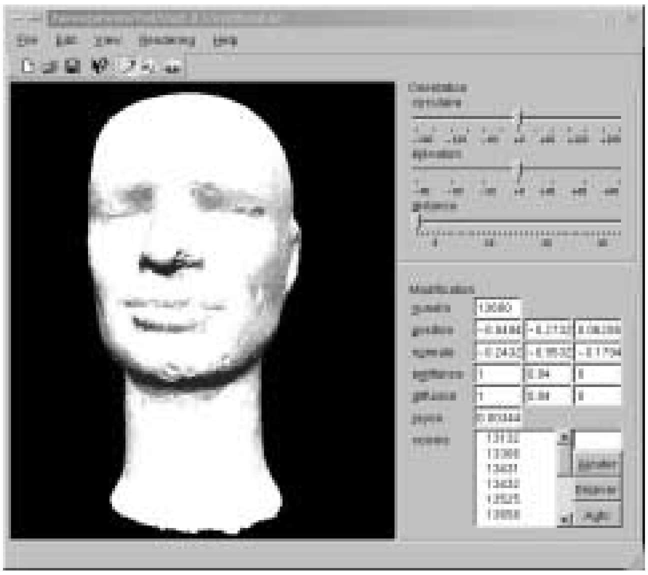

Figure 13: a modeling tool

6. CONCLUSIONS

In this paper we propose a new approach for surface modeling. This new method represents any surface by a joining set of implicit surfaces which can be easily and naturally described by an atlas of discoids. This new approach combines the advantages of both Spline surfaces that locally join pieces of small surfaces, and implicit surfaces that allow to model surfaces admitting a high level of complexity. As a consequence, this new approach reduces topological constraints by simply describing any object by an atlas of discoids. It also allows to reconstruct through the use of a merging operator a large choice of surfaces without changing the geometry of the atlas.

REFERENCES

[1] D. Arques, S. Michelin and B. Piranda, The Atlas System: a new model for Complex Surfaces}, in revision to Computer Graphics Forum, 1999.

[2] D. Arques, S. Michelin and B. Piranda, Overlapping radiosity: using a new function base with local disk support, WSCG'2000, vol. 3, 2000, pp. 236-243.

[3] R.H. Bartels and C.J. Beatty, A technique for the direct manipulation of spline curves, Graphics Interface'89, 1989, pp. 33-39.

[4] J.F. Blinn, A generalization of algebraic surface drawing, ACM Transactions on Graphics, vol. 1(3), 1982, pp. 235–256.

[5] J.D. Boissonnat, Geometric structure for three dimensional shape representation, ACM Transactions on Graphics, vol. 3(4), 1984, pp. 266-286.

[6] J. Foley, A. Van Dam, S. Feiner and J. Hugues, Computer graphics: principles and practice, 2nd edition, Addison Wesley, 1990.

[7] A. Glassner, An introduction to ray tracing, Academic Press, 1989.

[8] P. Hanrahan, Ray Tracing Algebraic Surfaces, Computer Graphics, vol. 17(3), 1983, pp. 83–90.

[9] M. Mäntylä, An introduction to solid modeling, Computer Science Press, 1988.

[10] S. Nakamura, Applied Numerical Methods with Software, Prentice-hall, 1991.

[11] Phong B., Illumination for Computed Generated Pictures, Communication of the ACM, vol. 18(6), 1975, pp. 311-317.

[12] A. Rockwood, K. Heaton and T. Davis, Real-Time Rendering of Trimmed Surfaces, SIGGRAPH '89 Proceedings, vol. 23(3), 1989, pp. 107-116.

[13] R. Szelisky and D. Tonnesen, Surface Modelling with Oriented Particle Systems, Computer Graphics, vol. 26(2), 1992, pp. 185-194.

[14] F.X. Sillion and C. Puech, Radiosity and global illumination, Morgan Kaufmann publisher, 1994.

[15] A. Watt, 3D Computer Graphics, 2nd edition, Addison Wesley, 1993

[16] A.P. Witkin and P.S. Heckbert, Using Particles to Sample and Control Implicit Surfaces, SIGGRAPH '94 Proceedings, 1994, pp. 269–277.

About the authors

Didier Arquès is currently professor in the University of Marne la Vallée and head manager of the image synthesis team since 1994. His research interest in computer graphics concerns modeling, rendering models, and photo-realistic simulation. He also develops researches related to combinatorial maps and bio-informatics.
E-mail: **arques@univ-mlv.fr**

Sylvain Michelin is Maître de Conférences in the University of Marne la Vallée since 1996. His main interest concerns high level rendering techniques, animation, participating medium and image processing. He also has in charge the industrial relationship of the team.
E-mail: **michelin@univ-mlv.fr**

Benoît Piranda is Maître de Conférences in the University of Marne la Vallée. His main research interest concerns rendering techniques and surface modeling. In the same time, he coordinates all software developments of the team.
E-mail: **piranda@univ-mlv.fr**