INTERACTIVE ANIMATION OF ALGEBRAIC SURFACES

Dietmar Saupe, Matthias Ruhl

Universität Freiburg, Institut für Informatik
Am Flughafen 17, 79110 Freiburg, Germany
saupe,ruhl@informatik.uni-freiburg.de

ABSTRACT

For the visualization of animated implicit surfaces there exist three principally different approaches: direct raytracing of the implicitly defined object, polygonalization with subsequent standard rendering of the polygonal model, and particle-based methods employing the physically-based modelling paradigm. The three methods are ordered here with respect to decreasing computational complexity as well as decreasing image quality. While the raytracing approach yields the best approximations and renderings we have that the particle-based method delivers a speed that allows near real-time performance. In this paper we consider an implementation of this latter approach based on modelling with differential equations from the work of Witkin and Heckbert (SIGGRAPH'94) which we extended to adaptively take into account surface singularities, clipping of infinitely extending surfaces, and curvature. While these aspects are not truly essential for computer graphics modelling by implicit surfaces they naturally occur in simulations of interest to the mathematical community such as in an animation of the Kummer family of algebraic surfaces.

1. INTRODUCTION

Algebraic surfaces and their deformations have been studied for more than one hundred years. Visualization of these surfaces has always been regarded important and traditionally plaster models were used for this purpose. Recently, of course, computer graphical studies have been carried out. In [3] Hanrahan investigates the raycasting approach to the rendering of algebraic surfaces. The central computational task is twofold. First, one has to efficiently convert the equations of the surface and a given ray into a single equation, i.e., a polynomial in one variable. Secondly, a numerical procedure must be employed to compute the smallest positive root of the polynomial. Hanrahan chose a method developed by Collins and Loos, which is based on Descartes rule of signs. In [5] the approach was tested with other root finding methods but no animation was considered.

An important example of a deformation of algebraic surfaces was proposed by Kummer in the last century. It is given by a parametrized family of fourth-order polynomials in affine coordinates x, y and z. In this deformation a double sphere is transformed into Steiner's roman surface and then into a tetrahedron. Although individual surfaces from this family had been visualized before, only just recently a sequence of (raycast) images illustrating the entire deformation was given by Barth and Endraß in [1]. In our work [4]— shown at the VISMATH Workshop in Berlin in June 1995 — we presented corresponding computer animations. We demonstrated a feasible approach to rendering animations of mathematical objects suitable not only for the computer graphics specialists but also for students and researchers in mathematics without such background. This animation was based on the original method of Hanrahan and supported by the public domain raytracer *rayshade* of Kolb which requires supplying a corresponding program module for the ray surface intersection calculations. See Figures 1 to 3 for PostScript versions of three frames of the animation.

This approach — although suitable for high quality renderings — lacks interactive control which is necessary to arrive at properly chosen camera paths, parameter dynamics et cetera for an animation of some previously unknown surfaces under investigation. In this work we use the sampling method of Witkin and Heckbert from their recent SIGGRAPH'94 paper [8] for algebraic surfaces. The method is particle-based. A simple constraint equation locks a set of particles onto a surface while the particles and the surface are allowed to move. Local repulsion is used to make the points, called *floaters*, spread evenly across the surface. By varying the radius of repulsion adaptively while fissioning and killing particles based on the local density, they can achieve good sampling distributions very rapidly. Their work focussed more on *modelling* of implicit surfaces while we are more interested in a priori defined parameter-dependent surfaces. Moreover, our surfaces have singularities and they are unbounded (or clipped to a finite volume resulting in a boundary curve) in contrast to theirs which are just single connected components without singularities or boundary. To cope with these difficulties we modify the physical model underlying the mechanism of floaters by introducing boundary effects and adapt the radii of repulsion to local surface properties as well as to the particle density. With these modifications we can achieve near real-time sampling and rendering of deforming implicit algebraic surfaces.

2. SAMPLING BY FLOATERS

In this section we briefly summarize the physically-based approach to sampling an implicitly defined surface by floaters as presented by and using the notation of Witkin and Heckbert in [8]. Particle-based approaches for static surfaces have previously been studied by Turk in [6, 7] and by de Figueiredo et al [2].

We consider parameter-dependent implicit surfaces in three-dimensional Euclidean space, where an animation of the surface occurs by variation of surface parameters, changing the shape of the surface with time. The surface is given by the zeroset of a differentiable function: $F(x, q(t)) = 0$ where $x \in \mathbf{R}^3$ and $q(t) \in \mathbf{R}^m$ denotes a set of m parameters smoothly changing with time t. We assume a set of n particles for the purpose of sampling the surface. Particle i has trajectory $p^i(t)$ (superscripts denote particle indices). Particles moving about on the surface must satisfy the equations

$$F^i(t) = F(p^i(t), q(t)) = 0$$

and

$$\dot{F}^i = F_x^i \cdot \dot{p}^i + F_q^i \cdot \dot{q} = 0,$$

where the dots denote time derivatives and subscripts specify gradients of F with respect to x or q. In order to move particles to the surface and to keep them on the surface in spite of numerical integration errors a feedback term $\dot{F}^i = -\phi F^i$ with a constant $\phi > 0$ is used leading to the basic constraint equation

$$F_x^i \cdot \dot{p}^i + F_q^i \cdot \dot{q} + \phi F^i = 0.$$

For each particle this yields one scalar equation leaving two degrees of freedom. The actual velocities are determined via an optimization. For each particle i a desired velocity P_i will be defined below and the optimization requires to minimize the objective function

$$G = \frac{1}{2} \sum_1^n ||\dot{p}^i - P_i||^2$$

For example, setting all $P_i = 0$ will result in minimizing particle speeds. This constrained optimization can be solved using Lagrange multipliers yielding equations

$$\dot{p}^i = P^i - \frac{F_x^i \cdot P^i}{F_x^i \cdot F_x^i + F_q^i \cdot \dot{q} + \phi F^i} F_x^i.$$

At this point the definition of the desired particle velocities remains to be given. It serves the purpose of quickly producing an acceptable sampling density over the surface and is based on repulsion of particles and the finiteness of the surface. The repulsion forces are based on the concept of the 'energy' of particles and are taken proportional to energy gradients. Witkin and Heckbert define an adaptive scheme by setting the energy of particle i due to particle j as

$$E^{ij} = \alpha \exp\left(-\frac{||r^{ij}||^2}{2(\sigma_i)^2}\right)$$

where $r^{ij} = p^i - p^j$. The parameters $\sigma^i > 0$ can be regarded as some kind of repulsion radius of the particle i. The total energy at particle i is the sum

$$E^i = \sum_{j=1}^n E^{ij} + E^{ji}.$$

Then the desired velocities come out to be

$$P^i = -(\sigma^i)^2 E^i_{p^i} = (\sigma^i)^2 \sum_{j=1}^n \left(\frac{r^{ij}}{(\sigma^i)^2} E^{ij} + \frac{r^{ij}}{(\sigma^j)^2} E^{ji}\right).$$

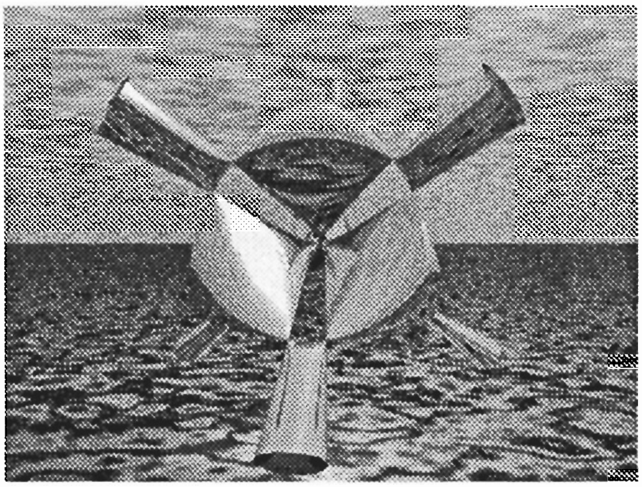

Figure 1: Raytraced Kummer surface with $\nu = 1.02061$.

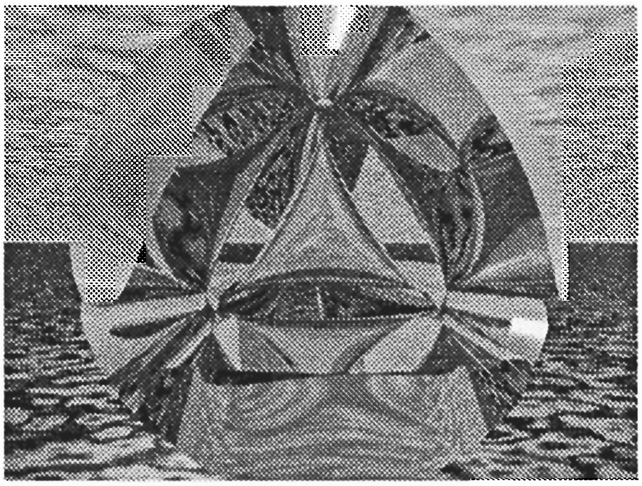

Figure 2: Raytraced Kummer surface with $\nu = 2.442$.

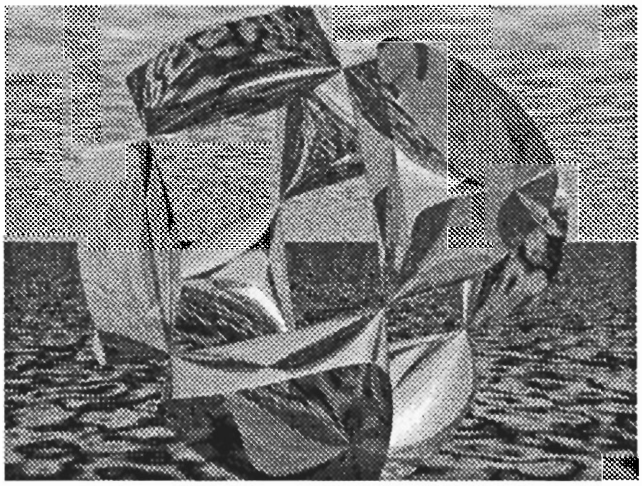

Figure 3: Raytraced Kummer surface with $\nu = 1.2785 \cdot 10^6$.

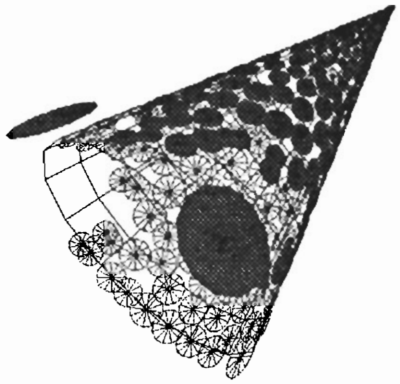

Figure 4: The cone in its initial rendering phase. Two new particle are inserted into the scene.

Figure 5: The maximal number of particles are on the cone surface. An equilibrium point has been reach such that the particle density is higher near he singularity as desired.

Figure 6: Two intersecting plane, a few particles.

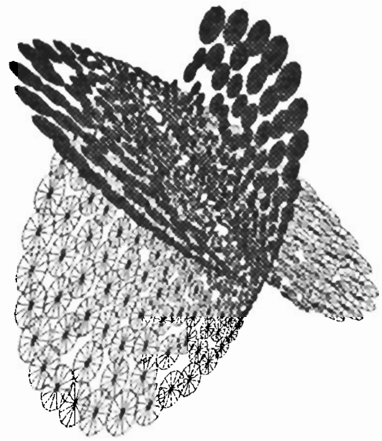

Figure 7: The plane with more particles.

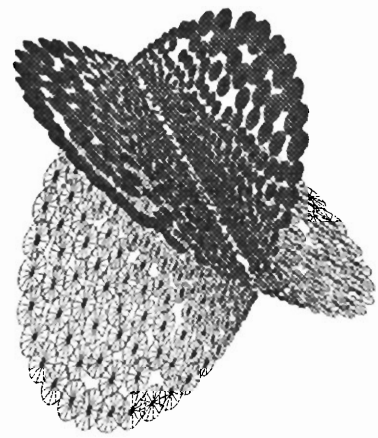

Figure 8: Particles on planes at equilibrium.

In order to move particles into sparse regions quickly the repulsion radii can be controlled adaptively. This is achieved by driving all of the energies to a global desired energy level. In effect, this yields another set of simple differential equations for the radii σ^i. To achieve a uniform sampling density particles with large repulsion radii are required to fission and likewise, particles with small repulsion radii can be eliminated. Witkin and Heckbert suggest to fission a particle, if it is near equilibrium (small speed relative to its radius), and either its radius is large ($\sigma^i > \sigma_{max}$) or it carries high energy and its radius is above a given nominal radius $\hat{\sigma}$. The radius of the two new particles are set to $\sigma^i/\sqrt{2}$ and their desired velocities are taken as random directions scaled appropriately. A particle is suggested to die, if it is near equilibrium, its repulsion radius is small, and if a randomized test succeeds (in order to prevent 'mass suicide' in overcrowded regions).

We remark that our exposition of the method necessarily is very brief; implementation details and motivations can be found in the original article [8]. Moreover, the method also allows the determination of the parameters of the surface via control points (useful for modelling with implicit surfaces), the theory of which is based on the same differential equations approach as that of the floaters described above.

Figure 9: Kummer surface at $\nu = 1.5$ with a few particles not yet at equilibrium.

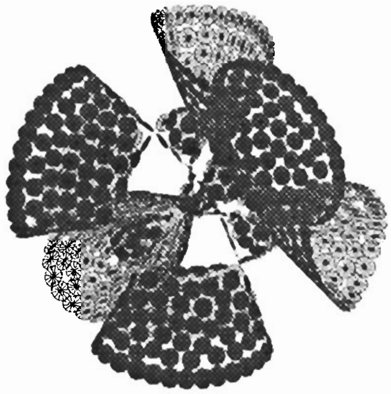

Figure 10: Kummer surface at $\nu = 1.5$ with more particles and at equilibrium.

3. EXTENSIONS AND RESULTS

As demonstrated in [8] the method works fine for compact smooth surfaces without singularities. Selfintersections and cusp points of the surface, however, will produce the artifact of nonuniform sampling since particles distributed near such singularities locally repel each other stronger than particles in a regular surface patch with the same sampling density. To overcome this problem we propose to use the surface gradient as a practical measure for closeness of a singularity. For example, we have tried to adapt the desired particle repulsion radius $\hat{\sigma}$ and the global fissioning radius σ_{max} to the individual particles by setting $\hat{\sigma}^i = \mu^i \hat{\sigma}$, where the factor μ^i is derived from local surface gradient:

$$\mu^i = \begin{cases} 0.1 & \text{if } ||F_x||/\nabla_0 < 0.1 \\ ||F_x||/\nabla_0 & \text{if } ||F_x||/\nabla_0 \in [0.1, 1.0] \\ 1.0 & \text{if } ||F_x||/\nabla_0 > 1.0 \end{cases} .$$

and the constant parameter ∇_0 is hand-tuned (equal to 1 in our first experiments). The fissioning parameter is adapted as $\sigma^i_{max} = \max(d/2, 1.5\hat{\sigma}^i)$, where d denotes the surface diameter.

In order to allow for unbounded surfaces, where the original sampling method would not work, we have introduced half spaces and spheres as clipping volumes. Instead of extending the physical model to incorporate forces that would keep the particles constrained within the clipping volumes we found that a simpler approach yields the same result. We suggest to monitor the particle trajectories and in the case that a particle is moved outside the clipping volume we simply project it back onto the boundary of the clipping volume. In the case of a half space this is an orthogonal projection onto the hyperplane defining the volume, and in the case of the sphere the projection occurs in the direction towards the sphere center.

We have tested the method using two very simple implicit surfaces with singularities and more complex algebraic surfaces of degree four, the Kummer family:

1. A cone with a single cusp point clipped at two planes, one of which contains the cusp. No parameter animation was employed ($q(t) \equiv constant$).
2. The union of two intersecting planes, clipped at a sphere. The planes intersect orthogonally and the intersection line moves back and forth by changing the parameters $q(t)$ appropriately.
3. The parametrized family of Kummer surfaces as used in the raytraced animation. The equation is

$$(x^2 + y^2 + z^2 - \nu^2)^2 - \frac{3\nu^2 - 1}{3 - \nu^2} pqrs = 0$$

where $\nu \in \mathbf{R}$ is the parameter and

$$\begin{aligned} p &= 1 - z - x\sqrt{2}, \\ q &= 1 - z + x\sqrt{2}, \\ r &= 1 + z + y\sqrt{2}, \\ s &= 1 + z - y\sqrt{2}. \end{aligned}$$

The surface is unbounded for $\nu^2 \geq 1$ and clipped at a sphere whose radius is chosen so that the displayed surface contains all of the singular points (except the singularity at infinity).

Figures 4 to 12, taken from a video sequence taped directly from the computer monitor, display some of the results. The renderings show the floaters as shaded disks with normal vectors obtained from the gradients of the surface's defining function F. The radii of the disks are taken proportional to the particles' repulsion radii. Particles viewed from the 'back side' are shown as wireframes. The program starts by randomly inserting particles into the scene (up to some maximum number) which quickly settle on the surface, then spread over the surface and start fissioning yielding an overall acceptable density.

4. CONCLUSION

The extensions to the floater mechanisms of Witkin and Heckbert introduced in this paper are shown to adapt the interactive animation capabilities to surfaces that are clipped and have singularities. Although not yet tested it is clear that one may also use surface curvature as one of the criteria that determines the local particle density. The interactive animated rendering of algebraic

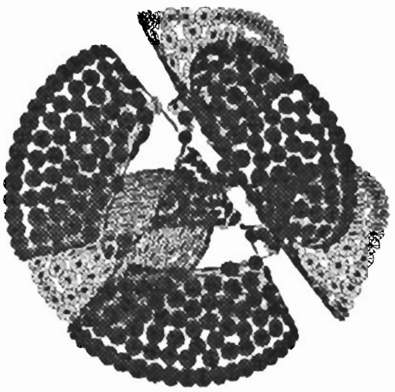

Figure 11: Kummer surface at $\nu = 2$.

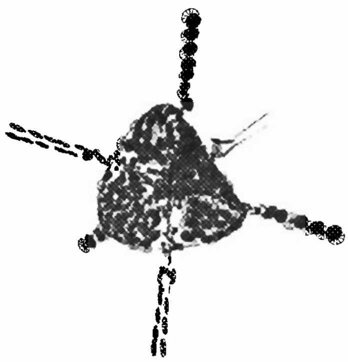

Figure 12: Kummer surface at $\nu = 1.01$. Note that all of the 8 thin cylinder-like parts of the surface have been correctly covered by the automatic sampling scheme.

surfaces as shown here aids the researcher in finding desired parameters, clipping volumes and and view parameters for a given surface for offline rendering at higher quality including animations.

The work presented here is only at its very beginnings. Alternative rules for fissioning and particle death as well as optimization of computational efficiency remain an issue. In some cases we observed that the particle population did not arrive at an equilibrium: near singularities, where the nominal repulsion radius is small, new particles were born, and older ones were pushed outwards, where they grew in size and overcrowded a region so that they finally died. Thus, the question on what conditions generally can ensure an attractive stable equilibrium remains to be answered.

5. REFERENCES

[1] W. P. Barth, S. Endraß, *Deformation, a series of computer pictures,* Internal Report, Mathematisches Institut, Universität Erlangen, 1995.

[2] L. H. de Figueiredo, J. de Miranda Gomes, D. Terzopoulos, L. Velho, *Physically-based methods for polygonalization of implicit surfaces,* Graphics Interface '92 (1992) 250–257.

[3] P. Hanrahan, *Ray tracing algebraic surfaces,* In P. Tanner, (ed.), *Computer Graphics (SIGGRAPH'83 Proc.),* **17**, 3, 83–90, (July 1983).

[4] D. Saupe, M. Ruhl, *Animation of Algebraic Surfaces,* in Proc. VisMath, Berlin, June 1995, to be published.

[5] O. Stelzner, *Visualisierung algebraischer Flächen mit Raytracing-Verfahren,* Diploma Thesis, Fachbereich Mathematik, Universität Bremen, 1990.

[6] G. Turk, *Generating textures on arbitrary surfaces using reaction-diffusion,* Computer Graphics (SIGGRAPH'91 Proc.) 25,4 (1991) 289–298.

[7] G. Turk, *Re-tiling polygonal surfaces,* Computer Graphics (SIGGRAPH'92 Proc.) 26,2 (1992) 55–64.

[8] A. Witkin, P. Heckbert, *Using particles to sample and control implicit surfaces,* Computer Graphics (SIGGRAPH '94 Proc.) (1994) 269–277, .