Contour Generators of Evolving Implicit Surfaces

Simon Plantinga *
Gert Vegter

Institute for Mathematics and Computing Science
University of Groningen

ABSTRACT

The contour generator is an important visibility feature of a smooth object seen under parallel projection. It is the curve on the surface which seperates front-facing regions from back-facing regions. The apparent contour is the projection of the contour generator onto a plane perpendicular to the view direction. Both curves play an important role in computer graphics.

Our goal is to obtain fast and robust algorithms that compute the contour generator with a guarantee of topological correctness. To this end, we first study the singularities of the contour generator and the apparent contour, for generic views, and for generic time-dependent projections, e.g. when the surface is rotated or deformed. The singularities indicate when components of the contour generator merge or split as time evolves.

We present an algorithm to compute an initial contour generator, using a dynamic step size. An interval test guarantees the topological correctness. This initial contour generator can then be maintained under a time-dependent projection by examining its singularities.

Categories and Subject Descriptors

I.3.3 [**Computer Graphics**]: Picture/Image Generation—*Line and curve generation*; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Geometric algorithms, languages, and systems*; G.1.0 [**Numerical Analysis**]: General—*Interval arithmetic*

General Terms

Algorithms, Theory

Keywords

Implicit surfaces, contour generators, guaranteed topology, interval arithmetic, singularities, evolving surfaces

*email: {simon,gert}@cs.rug.nl

SMA '03 Seattle, Washington USA

1. INTRODUCTION

An important visibility feature of a smooth object seen under parallel projection along a certain direction is its *contour generator*, also known as outline, or profile. The contour generator is the curve on the surface, separating front-facing regions from back-facing regions. This curve may have singularities if the direction of projection is non-generic. The *apparent contour* is the projection of the contour generator onto a plane perpendicular to the view direction. In many cases, drawing just the visible part of the apparent contour gives a good impression of the shape of the object. In this paper, we will not distinguish between visible and invisible parts of the contour generator. Stated otherwise, we assume the surface is transparent. Generically, the apparent contour is a smooth curve, with some isolated singularities. See Figure 1.

The contour generator and the apparent contour play an important role in computer graphics and computer vision. Rendering a polyhedral model of a smooth surface yields a jaggy outline, unless the triangulation of the surface is finer in a neighbourhood of the contour generator. This observation has led to techniques for view-dependent meshing and view-dependent refinement techniques, cf. [1].

Also, to render a smooth surface it is sufficient to render only the part of the surface with front-facing normals, so the contour generator, being the boundary of the potentially visible part plays a crucial role here. In non-photorealistic rendering [16] one just visualizes the apparent contour, perhaps enhanced by strokes indicating the main curvature directions of the surface. This is also the underlying idea in silhouette rendering of implicit surfaces [6]. In computer vision, techniques have been developed for partial reconstruction of a surface from a sequence of apparent contours corresponding to a discrete set of nearby projection directions. We refer to the book [9] for an overview, and for a good introduction to the mathematics underlying this paper. Other applications use silhouette interpolation [11] from a precomputed set of silhouettes to obtain the silhouette for an arbitrary projection. In computational geometry, rapid silhouette computation of polyhedral models under perspective projection with moving viewpoints has been achieved by applying suitable preprocessing techniques [4].

This paper presents a method for the robust computation of contour generators and apparent contours of implicit surfaces. For an introduction to the use of implicit surfaces for

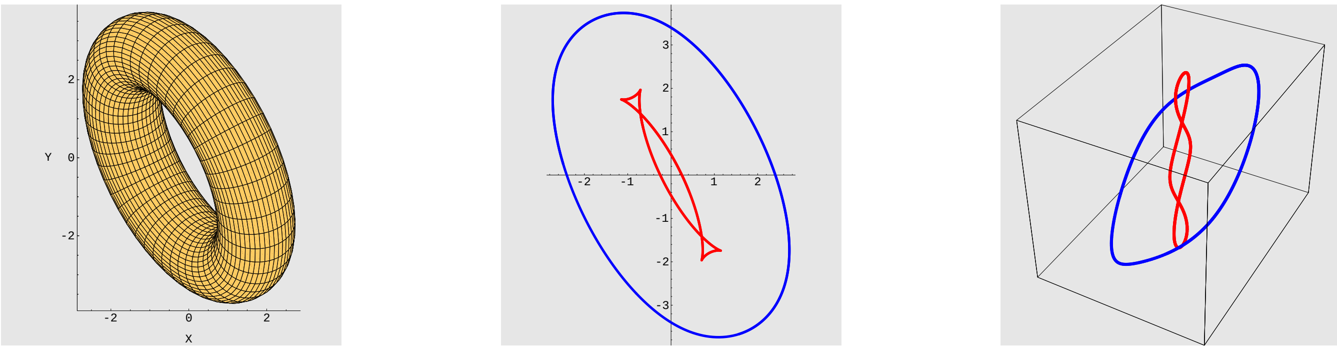

Figure 1: A smooth surface (left), its apparent contour under parallel projection along the z direction (middle), and its contour generator, seen from a different position (right). For generic surfaces (or, generic parallel projections) the contour generator is a smooth, possibly disconnected curve on the surface, whereas the apparent contour may have isolated cusp points.

smooth deformable object modeling we refer to [18] and [5].

We first consider generic static views, where both the surface and the direction of projection are static. Then we pass to time-dependent views, where the direction of projection changes with time. We derive conditions that locate the changes in the topology of the contour generator and the apparent contour. It turns out that generically there are three types of events, or bifurcations, leading to such a change in topology. These bifurcations have been studied from a much more advanced mathematical point of view, where they are known under the names lips, beak-to-beak, and swallowtail bifurcation. See also [8]. For a nice non-mathematical description we refer to the beautiful book by Koenderink [15]. Also see Arnol'd [3] and Bruce [7] for a sketch of some of the mathematical details related to singularity theory. Arnol'd [2] contains some of the results of the paper in a complex analytic setting. Our approach is somewhere inbetween the level of Koenderink's book and the sophisticated mathematical approach. We use only elementary tools, like the Inverse and Implicit Function Theorem, and finite order Taylor expansions. These techniques are used to design algorithms, in the same way as the Implicit Function Theorem gives rise to Newton's method.

Most curve tracing algorithms step along the curve using a fixed step size. See for example [6] or [14]. For a good approximation, the user has to choose a step size that is 'small enough' to follow the details of the curve. Some algorithms predict a dynamic step size, based on the local curvature. Both methods cannot guarantee a correct approximation to the curve. Also, these curve tracing algorithms assume there are no singularities. By examining the singularities before tracing the curve we can avoid them in the tracing process. We developed a condition based on interval analysis, that guarantees topological correctness of the traced curve.

In Section 2 we present the framework, and discuss criteria for a point on the contour generator and apparent contour to be regular. Section 3 examines singularities under some time-dependent view, for example when the viewpoint moves or when the surface deforms. In section 4 we explain the transformation to local models. For the implementation interval analysis is used. A brief overview can be found in section 5. The algorithm for computing the contour generator is explained in section 6.

2. CONTOUR GENERATOR AND APPARENT CONTOUR

Contour generators of implicit surfaces

To understand the nature of regular and singular points of the contour generator, and their projections on the apparent contour, we assume S is given as the zero-set of a smooth function $F\colon \mathbb{R}^3 \to \mathbb{R}$, so $S = F^{-1}(0)$. Furthermore, we assume that 0 is a regular value of F, i.e., the gradient ∇F is non-zero at every point of the surface. The gradient vector $\nabla F(p)$ is the normal of the surface at p, i.e., it is normal to the tangent plane of S at p. This tangent plane is denoted by $T_p(S)$. If v is the direction of parallel projection, then the contour generator Γ is the set of points at which the normal to S is perpendicular to the direction of projection, i.e., $p \in \Gamma$ iff the following conditions hold:

$$\begin{aligned} F(p) &= 0 \\ \langle \nabla F(p), v \rangle &= 0. \end{aligned} \tag{1}$$

For convenience, we assume throughout the paper that $v = (0, 0, 1)$. Then the preceding equations reduce to

$$\begin{aligned} F(x, y, z) &= 0 \\ F_z(x, y, z) &= 0. \end{aligned} \tag{2}$$

Here, and in the sequel, we shall occasionally write F_z instead of $\frac{\partial F}{\partial z}(p)$. We also use notation like F_x and F_{zz}, with a similar meaning.

We assume that S is a generic surface, i.e. there are no degenerate singular points on its contour generator. Some functions can yield degenerate contour generators. For example the cylinder $F(x, y, z) = x^2 + y^2 - 1$ has a two-dimensional contour generator for the view direction along the z-axis. Using a small perturbation, we can transform the cylinder into the ellipsoid $F(x, y, z) = x^2 + y^2 + \epsilon z^2 - 1$, having a 1-dimensional contour generator. See [20].

We now derive conditions for the contour generator Γ and the apparent contour γ to be regular at a given point. Recall that a curve is regular at a certain point if it has a non-zero tangent vector at that point. The next result gives conditions in terms of the function defining the surface.

PROPOSITION 2.1.
1. A point $p \in \Gamma$ is a regular point of the contour generator if and only if

$$F_{zz}(p) \neq 0 \quad or \quad \Delta(p) \neq 0, \tag{3}$$

where $\Delta(p)$ is a Jacobian determinant defined by

$$\Delta(p) = \left.\frac{\partial(F, F_z)}{\partial(x, y)}\right|_p = \begin{vmatrix} F_x(p) & F_y(p) \\ F_{xz}(p) & F_{yz}(p) \end{vmatrix}.$$

2. A point $p \in \gamma$ is a regular point of the apparent contour if and only if

$$F_{zz}(p) \neq 0. \tag{4}$$

PROOF. 1. The condition for p to be regular is

$$\nabla F(p) \wedge \nabla F_z(p) \neq 0.$$

Since $F_z(p) = 0$, a straightforward calculation yields

$$\nabla F(p) \wedge \nabla F_z(p) = F_{zz} \cdot (F_y, -F_x, 0) + \frac{\partial(F, F_z)}{\partial(x, y)} \cdot (0, 0, 1). \tag{5}$$

Here all derivatives are evaluated at p. Since $\nabla F(p) = (F_x(p), F_y(p), 0) \neq 0$, we see that $(F_y, -F_x, 0)$ and $(0, 0, 1)$ are linearly independent vectors. Therefore, the linear combination of these vectors in the right-hand side of (5) is zero iff the corresponding scalar coefficients are zero. The necessity and sufficiency of condition (3) is a straightforward consequence of this observation.

2. Since $\nabla F(p) \neq 0 \in \mathbb{R}^3$, and $F_z(p) = 0$, we see that $(F_x(p), F_y(p)) \neq (0, 0)$. Assuming $F_y(p) \neq 0$, we get

$$\left.\frac{\partial(F, F_z)}{\partial(y, z)}\right|_p = \left.\begin{vmatrix} F_y & F_z \\ F_{yz} & F_{zz} \end{vmatrix}\right|_p = F_y(p) F_{zz}(p) \neq 0. \tag{6}$$

Let $p = (x_0, y_0, z_0)$. Then, the Implicit Function Theorem yields locally defined functions $\eta, \zeta : \mathbb{R} \to \mathbb{R}$, with $\eta(x_0) = y_0$ and $\zeta(x_0) = z_0$, such that (2) holds iff $y = \eta(x)$ and $z = \zeta(x)$. The contour generator is a regular curve parametrized as $x \mapsto (x, \eta(x), \zeta(x))$, locally near p, whereas the apparent contour is a regular curve in the plane parametrized as $x \mapsto (x, \eta(x))$, locally near (x_0, y_0). $\square$

Singular points of contour generators

We apply the preceding result to detect non-degenerate singularities of contour generators of implicit surfaces. This result will be applied later in this section, when we consider contour generators of time-dependent surfaces.

Again, let the regular surface S be the zero set of a C^3-function $F\colon \mathbb{R}^3 \to \mathbb{R}$, for which 0 is a regular value. We consider the contour generator Γ of S under parallel projection along the vector $v = (0, 0, 1)$. The equations for Γ are

$$F = F_z = 0.$$

We consider the contour generator as the zero-set of the function F_z, restricted to S.

COROLLARY 2.2. *Point p is a non-degenerate singular point of Γ iff the following two conditions hold:*

$$F(p) = F_z(p) = F_{zz}(p) = \left.\frac{\partial(F, F_z)}{\partial(x, y)}\right|_p = 0, \tag{7}$$

and $\Sigma(p) \neq 0$, where, for $F_x(p) \neq 0$,

$$\begin{aligned}\Sigma(p) = &- {F_x}^2 {F_{xzz}}^2 {F_y}^2 + 2 {F_x}^3 F_{xzz} F_y F_{yzz} - {F_x}^4 {F_{yzz}}^2 \\ &- 2 {F_x}^3 F_{xyz} F_y F_{zzz} + 2 {F_x}^2 F_{xy} F_{xz} F_y F_{zzz} \\ &+ {F_x}^2 F_{xxz} {F_y}^2 F_{zzz} - F_x F_{xx} F_{xz} {F_y}^2 F_{zzz} \\ &- {F_x}^3 F_{xz} F_{yy} F_{zzz} + {F_x}^4 F_{yyz} F_{zzz},\end{aligned} \tag{8}$$

whereas, for $F_y(p) \neq 0$, we have

$$\begin{aligned}\Sigma(p) = &- {F_{xzz}}^2 {F_y}^4 + 2 F_x F_{xzz} {F_y}^3 F_{yzz} - {F_x}^2 {F_y}^2 {F_{yzz}}^2 \\ &- 2 F_x F_{xyz} {F_y}^3 F_{zzz} + F_{xxz} {F_y}^4 F_{zzz} \\ &+ {F_x}^2 {F_y}^2 F_{yyz} F_{zzz} + 2 F_x F_{xy} {F_y}^2 F_{yz} F_{zzz} \\ &- F_{xx} {F_y}^3 F_{yz} F_{zzz} - {F_x}^2 F_y F_{yy} F_{yz} F_{zzz}.\end{aligned} \tag{9}$$

PROOF. Condition (7) reflects the fact that p is a singular point of $F_z|S$, cf. (3), whereas (8) expresses non-degeneracy of this singular point. Condition (8) is obtained by a straightforward expansion[1] of (24) (see appendix A), with $G = F_z$, and $V = X$ as in (25), where $\lambda = \frac{F^0_{xz}}{F^0_x}$, $F^0_z = F^0_{zz} = 0$, and $F^0_{yz} = \frac{F^0_y}{F^0_x} F^0_{xz}$. Condition (9) is derived similarly. $\square$

Generic projections: fold and cusp points

In view of Proposition 2.1, regular points of the apparent contour are projections of points $(x, y, z) \in \mathbb{R}^3$ satisfying

$$F(x, y, z) = F_z(x, y, z) = 0, \quad \text{and} \quad F_{zz}(x, y, z) \neq 0.$$

This being a system of two equations in three unknowns, we expect that the regular points of the apparent contour form a one-dimensional subset of the plane. Furthermore, the singular points of the apparent contour are projections of points satisfying an additional equation, viz. $F_{zz}(x, y, z) = 0$, and are therefore expected to be isolated. This is true for *generic* surfaces. To make this more precise, we consider the set of functions $F : \mathbb{R}^3 \to \mathbb{R}$, satisfying

$$(\, F(x, y, z), F_z(x, y, z), F_{zz}(x, y, z), \Delta(x, y, z) \,) \neq (0, 0, 0, 0), \tag{10}$$

and

$$(\, F(x, y, z), F_z(x, y, z), F_{zz}(x, y, z), F_{zzz}(x, y, z) \,) \neq (0, 0, 0, 0). \tag{11}$$

If F satisfies (10), then Proposition 2.1 tells us that the contour generator Γ of $S = F^{-1}(0)$ under parallel projection along v is a regular curve. Moreover, for a point $(x, y, z) \in \Gamma$ there are two cases:

1. $F_{zz}(x, y, z) \neq 0$; in this case the point projects to a regular point (x, y) of the apparent contour γ. Such a point is called a *fold point* of the contour generator.

[1] using a computer algebra system

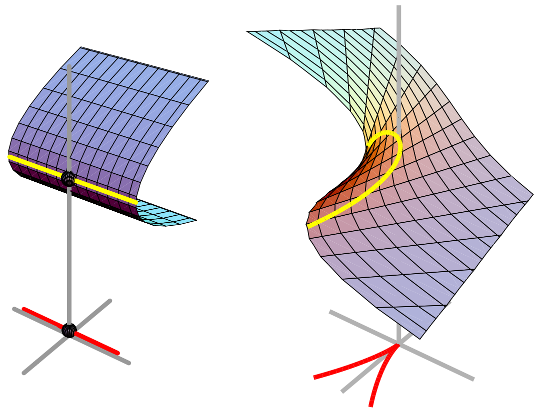

Figure 2: (a) A local model of the surface at a fold point is $x + z^2 = 0$. Both the contour generator Γ, and the visible contour γ, are regular at the fold point, and its projection onto the image plane, respectively. (b) A local model at a cusp point is $x+yz+z^3 = 0$. Here the contour generator is regular, but the apparent contour has a regular cusp.

This terminology is justified by the *local model* of the surface near a fold point, viz.

$$x + z^2 = 0. \tag{12}$$

See also Figure 2a. Here the contour generator is the y-axis in three space, so the apparent contour is the y-axis in the image plane.

2. $F_{zz}(x, y, z) = 0$; in this case the point projects to a singular point (x, y) of γ. Such a point is called a *cusp point* of the contour generator if, in addition to (10), condition (11) is satisfied, i.e., if both $\Delta(x, y, z) \neq 0$ and $F_{zzz}(x, y, z) \neq 0$. In this case the surface has the following local model near the cusp point:

$$G(x, y, z) = x + yz + z^3 = 0. \tag{13}$$

See also Figure 2b. The local model G is sufficiently simple to allow for an explicit computation of its contour generator and apparent contour: the former is parametrized by $z \mapsto (2z^3, -3z^2, z)$, the latter is a *regular cusp* parametrized by $z \mapsto (2z^3, -3z^2)$.

Intuitively speaking, a local model of the surface near a point is a 'simple' expression of the defining equation in suitably chosen local coordinates. Usually, as in the cases of fold and cusp points, a local model is a low degree polynomial, which can be easily analyzed in the sense that the contour generator and the apparent contour are easily determined.

So far we have only considered parallel projection. The standard perspective transformation [13], which moves the viewpoint to ∞, reduces perspective projections to parallel projections. By deforming the surface using this transformation, perspective projections can be computed by using parallel projection on the transformed implicit function.

3. EVOLVING CONTOURS

As we have seen, generic surfaces satisfy conditions (10) and (11), since violation of one of these conditions would correspond to the existence of a solution of four equations in three unknowns. However, *evolving surfaces* depend on an additional variable, t say. Time dependency is expressed by considering implicitly defined surfaces

$$S_t = \{(x, y, z) \in \mathbb{R}^3 \mid F(x, y, z, t) = 0\},$$

where $F : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}$ is a smooth function of the space variables (x, y, z) and time t. Generically we expect that exactly one of the conditions (10) and (11) will be violated at isolated values of (x, y, z, t). For definiteness, we assume $(0, 0, 0, 0)$ is such a value.

Violation of (10) corresponds to a singularity of the contour generator. In this case the implicit surfaces, defined by $F(x, y, z, 0) = 0$ and $F_z(x, y, z, 0) = 0$, are tangent at $(x, y, z) = (0, 0, 0)$, but the tangency is non-degenerate. Stated otherwise, the function $G : \mathbb{R}^3 \to \mathbb{R}$, defined by $G(x, y, z) = F_z(x, y, z)$, *restricted* to the surface S_0, has a non-degenerate singularity at $(0, 0, 0)$.

Generically, there are two types of *bifurcations*, corresponding to different scenarios for changes in topology of the contour generator. The *beak-to-beak* bifurcation corresponds to the *merging* or *splitting* of connected components of the contour generator. Under some additional generic conditions (inequalities), a local model for this phenomenon is the surface, defined by

$$G(x, y, z, t) = x + (-y^2 + t)z + z^3, \tag{14}$$

Here the contour generator is defined by $x = 2z^3, -y^2 + 3z^2 = -t$. See also Figure 3.

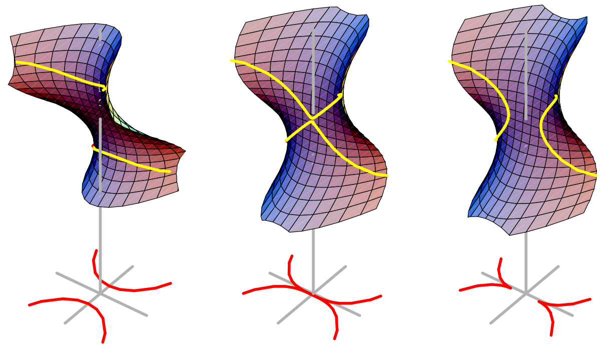

Figure 3: The beak-to-beak bifurcation. With respect to the local model (14) the bifurcation corresponds to $t < 0$ (left), $t = 0$ (middle), and $t > 0$ (right).

Putting $G^t(x, y, z) = G(x, y, z, t)$, we check that G^0 satisfies (7) at $p = (0, 0, 0)$, and that $|\Sigma(p)| = -4$. (In fact, $G^0_x = 1$, so all higher order derivatives of G^0_x vanish identically, so only the last term in the right hand side of (8) is not identically equal to zero.) Therefore, $G^0_z|S$ has a non-degenerate singular point of saddle type at p. According to the Morse lemma (see [10] or [17]), the level set of $G^0_z|S$ through p consists of two regular curves, intersecting transversally at p, which concurs with Figure 3 (middle).

A second scenario due to the violation of (11) is the *lips* bifurcation, corresponding to the *birth* or *death* of connected components of the contour generator. Again, under some additional generic conditions a local model for this phenomenon is the surface, defined by

$$G(x, y, z, t) = x + (y^2 + t)z + z^3, \qquad (15)$$

Here the contour generator is defined by $x = 2z^3, y^2 + 3z^2 = -t$. In particular, for $t > 0$ the surface S_t has no connected component of the contour generator near $(0, 0, 0)$, for $t = 0$, the point $(0, 0, 0)$ is isolated on the contour generator, and for $t < 0$ there is a small connected component growing out of this isolated point as t decreases beyond 0. See also Figure 4.

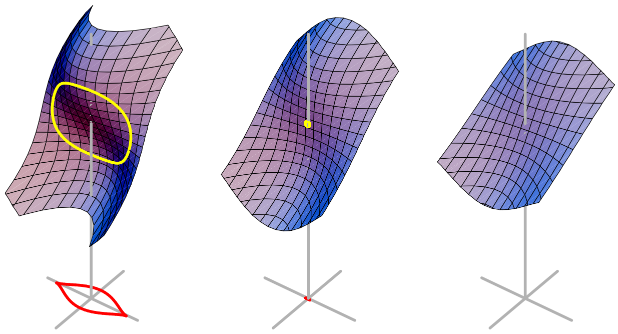

Figure 4: The lips bifurcation. Left: $t < 0$. Middle: $t = 0$. Right: $t > 0$.

As for the beak-to-beak bifurcation, we show that $G^0|S$ has a non-degenerate singular point at $(0, 0, 0)$, which in this case is an extremum.

Violation of (11) involves the occurrence of a higher order singularity of the apparent contour. Note, however, that in this situation the contour generator is still regular at the point (x, y, z), cf. Proposition 2.1. Imposing some additional generic conditions a local model for this type of bifurcation is

$$G(x, y, z, t) = x + yz + tz^2 + z^4 = 0. \qquad (16)$$

Here the apparent contour is parametrized as $z \mapsto (tz^2 + z^4, -2tz - 4z^3)$. See also Figure 5.

4. TRANSFORMATIONS AND NORMAL FORMS

In Section 2 we presented local models of various types of regular and singular points on contour generators and apparent contours, both for generic static surfaces, and for surfaces evolving generically in time. These local models are low degree polynomials, which are easy to analyze, and which yet capture the *qualitative behavior* of the contour generator and the apparent contour in a neighborhood of the point of interest. In this section we explain more precisely what we mean by capturing local behavior.

Consider two regular implicit surfaces $S = F^{-1}(0)$ and $T =$

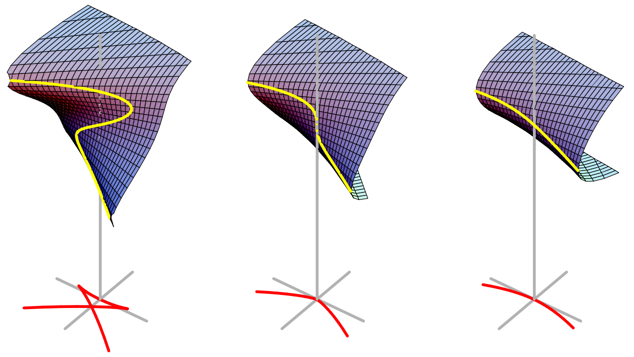

Figure 5: The swallowtail bifurcation. Left: $t < 0$. Middle: $t = 0$. Right: $t > 0$.

$G^{-1}(0)$. An invertible smooth map $\Phi \colon \mathbb{R}^3 \to \mathbb{R}^3$ for which

$$F \circ \Phi = G \qquad (17)$$

maps T to S. In fact, we consider Φ to be defined only locally near some point of T, but we will not express this in our notation. The map Φ need not map the contour generator of T onto that of S, however. To enforce this, we require that Φ maps vertical lines onto vertical lines, i.e., Φ should be of the form

$$\Phi(x, y, z) = (h(x, y), H(x, y, z)), \qquad (18)$$

where $h \colon \mathbb{R}^2 \to \mathbb{R}^2$ and $H \colon \mathbb{R}^3 \to \mathbb{R}$ are smooth maps. The map h is even invertible, since Φ is invertible. To allow ourselves even more flexibility in the derivation of local models, we relax condition (17) by requiring the existence of a non-zero function $\varphi \colon \mathbb{R}^3 \to \mathbb{R}$ such that

$$F(\Phi(x, y, z)) = \varphi(x, y, z)G(x, y, z). \qquad (19)$$

DEFINITION 4.1. *Let $S = F^{-1}(0)$ and $T = G^{-1}(0)$ be regular surfaces, near $p = (0, 0, 0) \in \mathbb{R}^3$. An admissible local transformation from T to S, locally near p, is a pair (Φ, φ), where $\varphi \colon \mathbb{R}^3 \to \mathbb{R}$ is non-zero at p, and $\Phi \colon \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^3$ is locally invertible near p, and of the form (18), such that (19) holds. We also say that Φ brings F in the normal form G.*

If the surfaces S and T depend smoothly on k parameters, i.e., they are defined by functions $F \colon \mathbb{R}^3 \times \mathbb{R}^k \to \mathbb{R}$ and $G \colon \mathbb{R}^3 \times \mathbb{R}^k \to \mathbb{R}$, respectively, then we require that the parameters are not mixed with the (x, y, z)-coordinates, i.e., we require that (19) is replaced with

$$F(\Phi(x, y, z, \mu)) = \varphi(x, y, z, \mu)\, G(x, y, z, \mu),$$

where $\Phi \colon \mathbb{R}^3 \times \mathbb{R}^k \to \mathbb{R}^3 \times \mathbb{R}^k$ is of the form

$$\Phi(x, y, z, \mu) = (h(x, y, \mu), H(x, y, z, \mu), \psi(\mu)).$$

Then Φ^μ, defined by $\Phi^\mu(x, y, z) = \Phi(x, y, z, \mu)$, maps T^μ to $S^{\psi(\mu)}$, and preserves contour generators. Furthermore, the map $h^\mu \colon \mathbb{R}^2 \to \mathbb{R}^2$, defined by $h^\mu(x, y) = h(x, y, \mu)$, maps the apparent contour of T^μ onto that of $S^{\psi(\mu)}$.

PROPOSITION 4.2. *If Φ is an admissible local transformation from T to S, locally near a point p on the contour generator of T, where Φ is of the form (18), then*

1. *Φ maps T to S, locally near $p \in S$;*

2. *Φ maps the contour generator of T to the contour generator of S, locally near p;*

3. *h maps the apparent contour of T onto the apparent contour of S, locally near the projection $\pi(p) \in \mathbb{R}^2$.*

PROOF. From (18) and (19) it is easy to derive

$$\begin{aligned} F(\Phi(p)) &= \psi(p)\, G(p), \\ F_z(\Phi(p))\, H_z(p) &= \psi_z(p)\, G(p) + \psi(p)\, G_z(p). \end{aligned}$$

Since $\psi(p) \neq 0$, and $H_z(p) \neq 0$, we conclude that $G(p) = G_z(p) = 0$ iff $F(\Phi(p)) = F_z(\Phi(p)) = 0$. $\square$

Example: local model at a truncated cusp point

We now illustrate the use of admissible transformations by deriving a local model for the class of implicit surfaces defined as the zero set of a function of the form:

$$F(x, y, z) = a(x, y) + b(x, y)z + c(x, y)z^2 + z^3, \qquad (20)$$

with $a(0, 0) = b(0, 0) = c(0, 0) = 0$, and

$$\left.\frac{\partial(a, b)}{\partial(x, y)}\right|_0 \neq 0. \qquad (21)$$

Note that the local model $x + y\,z + z^3 = 0$, derived in Section 2 for a cusp point, belongs to this class. Our goal is to show that the latter is indeed a local model for all surfaces of the type (20). Since $F(0) = F_z(0) = F_{zz}(0) = 0$, and

$$\left.\frac{\partial(F, F_z)}{\partial(x, y)}\right|_0 = \left.\frac{\partial(a, b)}{\partial(x, y)}\right|_0 \neq 0,$$

we see that $0 \in \mathbb{R}^3$ is a regular point of the contour generator of S, whereas $(0, 0)$ is a singular point of the apparent contour, cf. Proposition 2.1

As a first step towards a normal form of the implicit surface, we apply the *Tschirnhausen transformation* $z \mapsto z - \frac{1}{3}c(x, y)$ to transform the quadratic term (in z) in F away. More precisely,

$$\begin{aligned} F(x, y, z - \tfrac{1}{3}\, c(x, y)) &= a(x, y) - \tfrac{1}{3}\, b(x, y)\, c(x, y) + \tfrac{2}{27}\, c(x, y)^3 \\ &\quad + (b(x, y) - \tfrac{1}{3}\, c^2(x, y))\, z + z^3 \\ &= G(\overline{\varphi}(x, y), z), \end{aligned}$$

where $G(x, y, z) = x + yz + z^3$, and

$$\begin{aligned} \overline{\varphi}(x, y) = (\, & a(x, y) - \tfrac{1}{3}\, b(x, y)\, c(x, y) + \tfrac{2}{27}\, c(x, y)^3, \\ & b(x, y) - \tfrac{1}{3}\, c(x, y)^2\,). \end{aligned}$$

It is not hard to check that the Jacobian determinant of $\overline{\varphi}$ at $0 \in \mathbb{R}^2$ is equal to

$$\left.\frac{\partial(a, b)}{\partial(x, y)}\right|_0,$$

so $\overline{\varphi}$ is a local diffeomorphism near $0 \in \mathbb{R}^2$. Let φ be its inverse, then, putting

$$\Phi(x, y, z) = (\varphi(x, y), z - \tfrac{1}{3}\, c(\varphi(x, y)),$$

we get

$$F \circ \Phi(x, y, z) = G(x, y, z).$$

In other words, the *admissible* transformation Φ brings F into the normal form G. In particular, it maps the surface $T = G^{-1}(0)$ and its contour generator onto $S = F^{-1}(0)$, and φ maps the apparent contour of T onto the apparent contour of S.

5. INTERVAL ANALYSIS

One way to prevent rounding errors due to finite precision numbers is to use interval arithmetic. Instead of numbers, intervals containing the exact solution are computed. An *inclusion function* $\Box f$ for a function $f\colon \mathbb{R}^m \to \mathbb{R}^n$ computes for each m-dimensional interval I (i.e. an m-box) an n-dimensional interval $\Box f(I)$ such that

$$x \in I \quad \Rightarrow \quad f(x) \in \Box f(I)$$

An inclusion function is *convergent* if

$$\mathrm{width}(I) \to 0 \quad \Rightarrow \quad \mathrm{width}(\Box f(I)) \to 0$$

where the width of an interval is the largest width of I.

For example if $f\colon \mathbb{R} \to \mathbb{R}$ is the square function $f(x) = x^2$, then a convergent inclusion function is

$$\Box f([a, b]) = \begin{cases} [\min(a^2, b^2), \max(a^2, b^2)], & a \cdot b \geq 0 \\ [0, \max(a^2, b^2)], & a \cdot b < 0 \end{cases}$$

Inclusion functions exist for the basic operators and functions. To compute an inclusion function it is often sufficient to replace the standard number type (e.g. double) by an interval type.

We assume there are convergent inclusion functions for our implicit function F and its derivatives, and will denote them by F (and similiar for the derivatives). From the context it will be clear when the inclusion function is meant.

Interval arithmetic can be implemented using demand-driven precision. For the interval bounds, ordinary doubles (with conservative rounding) can be used for fast computation. In the rare case that the interval becomes too small for the precision of a double, a multi-precision number type can be used.

Interval Newton Method

For precision small intervals around the required value are used. Another use of interval arithmetic is to compute function values over larger intervals. If for an implicit surface $F = 0$ and a box I we have $0 \notin \Box F(I)$, we can be certain that I contains no part of the surface. This observation can be extended to the Interval Newton Method, that finds all roots of a function $f\colon \mathbb{R}^n \to \mathbb{R}^n$ in a box I.

The first part of the algorithm recursively subdivides the box, discarding parts of space containing no roots. If the boxes are small enough a Newton method refines the solutions and guarantees that all roots are found. Solving

$$f(x) + J(I)(z - x) = 0,$$

where x is the centre of I, J is the Jacobian matrix of f and $J(I)$ is the interval matrix of J over the interval I, results in an interval Y containing all roots z of f. This interval can be used to refine I. Also, if $Y \subset I$ there is a unique root of f in I. See [12] for the mathematical details. A more practical introduction can be found in [19] or [20].

6. TRACING THE CONTOUR GENERATOR

Our goal is to approximate the contour generator by a piecewise linear curve. This initial approximation can then be maintained under some time-dependent view. To this end the singularities of the contour generator for an evolving view or surface can be precomputed using interval analysis. Since the topology doesn't change between these singularities, the initial contour generator can be updated continuously, until we reach a time where a singularity arises. The local model at this singularity indicates how the topology has to be updated. See Section 4. For details we refer to the full version of this paper.

Note that for a singularity of the contour generator of a time-dependent surface, we have

$$\begin{pmatrix} F(x,y,z,t) \\ F_z(x,y,z,t) \\ F_{zz}(x,y,z,t) \\ \Delta(x,y,z,t) \end{pmatrix} = 0.$$

These singularities can therefore be considered as the zeroes of a function from $\mathbb{R}^4$ to $\mathbb{R}^4$. Using the Interval Newton Method, we can find all t for which a singularity occurs.

For the initial contour generator we can assume there are no singularities. The construction consists of two steps.

Firstly, for each component we have to find an initial point to start the tracing process. Interval analysis enables us to find points on all components of the contour generator. See below for details. These (regular) points serve as starting points for the tracing process.

Secondly, we trace the component by stepping along the contour generator. For each starting point we trace the component, by moving from a point p^i to the next point p^{i+1}. We take a small step in the direction of the tangent to the contour generator at p^i. Then, we move the resulting point back to the contour generator, giving us p^{i+1}. An interval test guarantees that we stay on the same component, without skipping a part. If the test fails, we decrease the step size and try again, until the interval test succeeds. If we reach the initial point p^0, the component is fully traced. See Figure 9 for some results of the algorithm.

Finding initial points

A tangent vector to the contour generator at p can be found by computing

$$w(p) = \nabla F(p) \wedge \nabla F_z(p) = \begin{pmatrix} F_y F_{zz} \\ -F_x F_{zz} \\ F_x F_{yz} - F_y F_{xz} \end{pmatrix}.$$

Since the components of the contour generator are bounded, closed curves, there are at least two points on each component where the x-component of $w(p)$ disappears, i.e. where $F_y F_{zz} = 0$.

Let $R, S\colon \mathbb{R}^3 \to \mathbb{R}^3$ be the functions

$$R(p) = \begin{pmatrix} F(p) \\ F_z(p) \\ F_y(p) \end{pmatrix} \text{ and } S(p) = \begin{pmatrix} F(p) \\ F_z(p) \\ F_{zz}(p) \end{pmatrix}$$

The Interval Newton Method can find all roots of R and S. These roots are used to create a list of (regular) initial points.

Tracing step

Let $N(x)$ be the normalized vector field

$$N(x) = \frac{\nabla F(x) \wedge \nabla F_z(x)}{\|\nabla F(x) \wedge \nabla F_z(x)\|}.$$

For x on the contour generator, $N(x)$ is a tangent vector at x. From p^i we first move to $q^0 = p^i + \delta N(p^i)$, where δ is the step size. To move back to the contour generator, we alternately move towards $F = 0$ and $F_z = 0$ by replacing q^i by

$$\begin{cases} q^{i+1} = q^i - \dfrac{F(q^i)\nabla F(q^i)}{\|\nabla F(q^i)\|^2} & \text{towards } F \\ q^{i+1} = q^i - \dfrac{F_z(q^i)\nabla F_z(q^i)}{\|\nabla F_z(q^i)\|^2} & \text{towards } F_z \end{cases}$$

until $\|q^{i+2} - q^i\|$ is sufficiently small. The resulting point is the next point on the contour generator, p^{i+1}. For this new point we perform the interval test (explained below) to determine whether $p^i p^{i+1}$ is a good approximation of the contour generator. If not, we decrease δ (e.g. by setting it to $\delta/2$), and repeat the tracing step from p^i.

Interval test

For a fixed step size, there is always a possibility of accidentally jumping to another component of the contour generator, or of skipping a part (figure 6).

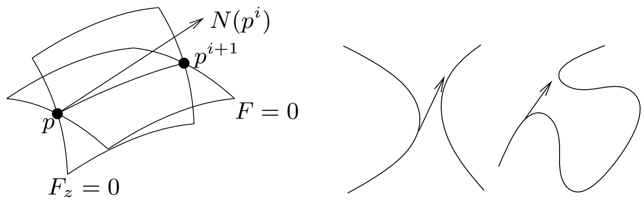

Figure 6: Left: $N(p^i)$ is a tangent to the intersection of $F = 0$ and $F_z = 0$. Right: a fixed step size can miss part of the contour generator.

To assure that $p^i p^{i+1}$ is a good approximation for the contour generator, first we construct a sphere S with centre p^i, that contains p^{i+1}. Then we take the bounding box B of S:

$$B = [p^i_x - \Delta, p^i_x + \Delta] \times [p^i_y - \Delta, p^i_y + \Delta] \times [p^i_z - \Delta, p^i_z + \Delta],$$

where $\Delta = \|p^{i+1} - p^i\|$. Over this box we compute the interval

$$I = \langle N(p^i), N(B) \rangle,$$

where $N(B)$ contains all normalized vectors $\dfrac{\nabla F(s) \wedge \nabla F_z(t)}{\|\nabla F(s) \wedge \nabla F_z(t)\|}$, with $s, t \in B$.

LEMMA 6.1. *If $I > \frac{1}{2}\sqrt{2}$, (i.e. $\forall i \in I\colon i > \frac{1}{2}\sqrt{2}$), then the part of the contour generator within S consists of a single component (fig. 7).*

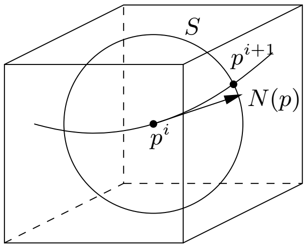

Figure 7: The sphere containing the line segment, and its bounding box.

PROOF. We define $G(x) = \langle x - p^i, N(p^i)\rangle$. The level sets of G are planes perpendicular to $N(p^i)$. Let $f\colon \mathbb{R}^3 \to \mathbb{R}^3$ be the function

$$f(x) = \begin{pmatrix} F(x) \\ F_z(x) \\ G(x) \end{pmatrix}.$$

Suppose there are two points x and y of the contour generator in B, lying in a plane perpendicular to $N(p^i)$, i.e. $f(x) = f(y) = (0, 0, \theta)$ for some θ. Then there are points s and t on xy where

$$\begin{pmatrix} \nabla F(s) \\ \nabla F_z(t) \\ N(p^i) \end{pmatrix} (y - x) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

For $x \neq y$, we find that $\langle \nabla F(s) \wedge \nabla F_z(t), N(p^i)\rangle = 0$. Since $s, t \in B$ this contradicts the interval test. Therefore, within box B each plane perpendicular to $N(p^i)$ contains at most one point of the contour generator.

The interval condition $I > \frac{1}{2}\sqrt{2}$ implies that the angle between $N(p^i)$ and $N(x)$ is at most $\frac{\pi}{4}$. The contour generator lies in a cone C around $N(x)$ (fig. 8) with top angle $\frac{\pi}{2}$, for if it leaves the cone at point a, then $\langle N(p^i), N(a)\rangle$ would be smaller than $\frac{1}{2}\sqrt{2}$. It can only leave the sphere in $S \cap C$. In this part of the sphere the contour generator can't reenter S, because that would require an entry point b where $\langle N(p^i), N(b)\rangle < \frac{1}{2}\sqrt{2}$. Therefore, there is only a single connected component of the contour generator within sphere S. $\square$

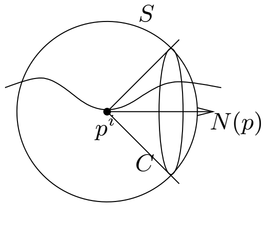

Figure 8: The contour generator lies within a cone.

If the interval test succeeds, we can use the same sphere S to remove redundant points from the initial point list. If there are point in the list that lie within S, they must be part of the component we are tracing, so they can be discarded.

To test whether the component is fully traced, we test if the initial point p^0 is contained in S. If it is, testing

$$\langle p^{i+1} - p^i, p^0 - p^i\rangle > 0$$

tells us whether we're done with this component (otherwise we just started the trace and are still moving away from p^0).

The interval bound is valid if the centre of the cube is on the contour generator. Since p^i is in general close to, but not on the contour generator, in practice we take a slightly larger bound. Also, Δ should be slightly larger than $\|p^{i+1} - p^i\|$, to prevent p^{i+1} from being too close to other parts of the contour generator outside S.

7. CONCLUSION AND FUTURE RESEARCH

We presented a framework for the analysis of an implicit surface near regular and singular points of its contour generator and apparent contour, and also derived conditions for detecting changes of topology of these visibility features in generic one-parameter families of implicit surfaces.

We developed an algorithm to compute a topologically correct approximation of the initial contour generator. A dynamic step size, combined with an interval test, guarantees that no part of the contour generator is skipped.

We plan to extend this work by implementing a robust algorithm for maintaining the contour generator under time-dependent directions of projection or surfaces, in such a way that its topology is guaranteed.

8. REFERENCES

[1] P. Alliez, N. Laurent, H. Sanson, and F. Schmitt. Efficient view-dependent refinement of 3d meshes using sqrt(3)-subdivision, 2001. To appear in the Visual Computer.

[2] V.I. Arnol'd. Wave front evolution and equivariant Morse lemma. *Comm. Pure Appl. Math.*, 29:557–582, 1976.

[3] V.I. Arnol'd. *Catastrophe Theory.* Springer-Verlag, Berlin, 1986.

[4] G. Barequet, C.A. Duncan, M.T. Goodrich, S. Kumar, and M. Pop. Efficient perspective-accurate silhouette computation. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 417–418, 1999.

[5] J. Bloomenthal. *Introduction to Implicit Surfaces.* Morgan-Kaufmann, 1997.

[6] D.J. Bremer and J.F. Hughes. Rapid approximate silhouette rendering of implicit surfaces. In *Proceedings of Implicit Surfaces '98*, pages 155–164, 1998.

[7] J.W. Bruce. Seeing - the mathematical viewpoint. *The Mathematical Intelligencer*, 6:18–25, 1984.

[8] J.W. Bruce and P.J. Giblin. Outlines and their duals. *Proc. London Math. Soc. (3)*, 50:552–570, 1985.

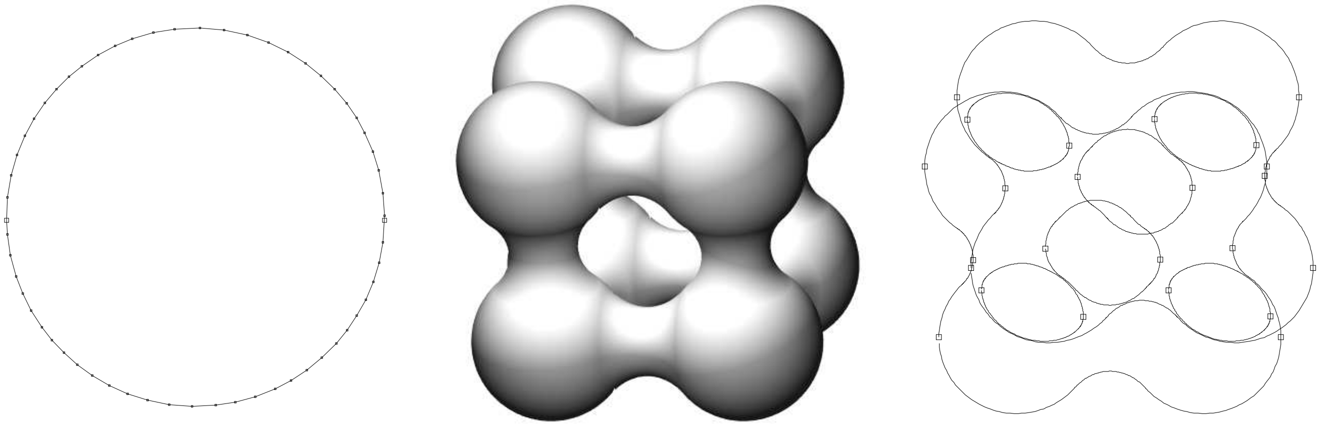

Figure 9: Contour generator of a sphere and of 8 metaballs (see appendix B) near the vertices of a cube. The small squares indicate the initial points. The dots on the sphere show the dynamic step size. Note that the metaballs are close to a singularity under motion of the viewpoint. The sizes of the eight components range from 439 to 879 segments.

[9] R. Cipolla and P.J. Giblin. *Visual Motion of Curves and Surfaces*. Cambridge University Press, 2000.

[10] A.T. Fomenko and T.L. Kunii. *Topological Modeling for Visualization.* Springer-Verlag, Tokyo, 1997.

[11] X. Gu, S.J. Gortler, H. Hoppe, L. McMillan, B.J. Brown, and A.D. Stone. Silhouette mapping. Computer Science Technical Report TR-1-99, Harvard University, March 1999.

[12] E.R. Hansen and R.I. Greenberg. An Interval Newton Method. *Applied Mathematics and Computation*, 12:89–98, 1983.

[13] D. Hearn, M.P. Baker. *Computer Graphics, 2nd edition* Prentice-Hall, Englewood Cliffs, NJ, 1994.

[14] C.M. Hoffman. *Geometric and Solid Modeling.* Morgan-Kaufman, 1989.

[15] J.J. Koenderink. *Solid Shape.* Artificial Intelligence. MIT-Press, Cambridge, Massachusetts, 1990.

[16] L. Markosian, M.A. Kowalski, S.J. Trychin, L.D. Bourdev, D. Goldstein, and J.F. Hughes. Real-time nonphotorealistic rendering. In *SIGGRAPH'97 Proceedings*, pages 415–420, 1997.

[17] J. Milnor. *Morse Theory,* Vol.51 of Annals of Mathematics Studies, Princeton University Press, Princeton, NJ, 1963.

[18] A. Opalach and S.C. Maddock. An Overview of Implicit Surfaces. *Introduction to Modelling and Animation Using Implicit Surfaces*, pages 1.1–1.13, 1995.

[19] J.M. Snyder. Interval Analysis for Computer Graphics. *SIGGRAPH'92 Proceedings*, pages 121–130, 1992.

[20] B.T. Stander and J.C. Hart. Guaranteeing the topology of an implicit surface polygonization for interactive modeling. *SIGGRAPH'97 Proceedings*, pages 279–286, 1997.

APPENDIX

A. SINGULARITIES OF FUNCTIONS ON SURFACES

Non-degenerate singular points

Consider an implicit surface $S = F^{-1}(0)$, where $F\colon \mathbb{R}^3 \to \mathbb{R}$ is a C^2 function. We assume that 0 is a regular value of F, so according to the Implicit Function Theorem S is a regular C^2-surface.

Our goal is to determine conditions which guarantee that the restriction of a C^2 function $G\colon \mathbb{R}^3 \to \mathbb{R}$ to the surface S has a non-degenerate singular point.

As for notation, the gradient of a function $F\colon \mathbb{R}^3 \to \mathbb{R}$ at $p \in \mathbb{R}$ will be denoted by $\nabla F(p)$. Furthermore, the *Hessian quadratic form* of a function $F\colon \mathbb{R}^3 \to \mathbb{R}$ at $p \in \mathbb{R}$ will be denoted by $H_F(p)$. Usually, we suppress the dependence on p from our notation, and denote this quadratic form by H_F. With respect to the standard euclidean inner product its matrix is the usual symmetric matrix whose entries are the second order partial derivatives of F. We denote partial derivatives using subscripts, e.g., F_x denotes $\frac{\partial F}{\partial x}$, F_{xy} denotes $\frac{\partial^2 F}{\partial x \partial y}$, etc.

THEOREM A.1. *Let $F, G\colon \mathbb{R}^3 \to \mathbb{R}$ be C^2 functions, and let 0 be a regular value of F. Let p be a point on the surface $S = F^{-1}(0)$.*

1. *p is a singular point of $G|S$ iff there is a real number λ such that*

$$\nabla G(p) = \lambda \nabla F(p). \tag{22}$$

2. *Furthermore, the singular point p is non-degenerate iff*

$$(H_G - \lambda H_F) \mid T_p S \tag{23}$$

is a non-degenerate quadratic form, where λ is as in (22).

Remark The scalar λ in (22) is traditionally called a *Lagrange multiplier.*

COROLLARY A.2. *The singularity p of $G|S$ is non-degenerate iff the 2×2-matrix Δ, defined by (24), is non-singular:*

$$\Delta = V^T \cdot \left(\begin{pmatrix} G_{xx} & G_{xy} & G_{xz} \\ G_{xy} & G_{yy} & G_{yz} \\ G_{xz} & G_{yz} & G_{zz} \end{pmatrix} - \lambda \begin{pmatrix} F_{xx} & F_{xy} & F_{xz} \\ F_{xy} & F_{yy} & F_{yz} \\ F_{xz} & F_{yz} & F_{zz} \end{pmatrix} \right) \cdot V, \quad (24)$$

where λ is the Lagrange multiplier defined by (22), and V is a 3×2-matrix whose columns span the tangent space T_pS. Here all first and second order derivatives are evaluated at p. Furthermore, $G|S$, the singular point p is a maximum or minimum if $\det(\Delta) > 0$*, and a saddle point if* $\det(\Delta) < 0$.

In particular, we may take $V = X$, $V = Y$, or $V = Z$ if $F_x(p) \neq 0$, $F_y(p) \neq 0$, or, $F_z(p) \neq 0$, respectively, where

$$X = \begin{pmatrix} F_y & F_z \\ -F_x & 0 \\ 0 & -F_x \end{pmatrix}, Y = \begin{pmatrix} -F_y & 0 \\ F_x & F_z \\ 0 & -F_y \end{pmatrix}, Z = \begin{pmatrix} -F_z & 0 \\ 0 & -F_z \\ F_x & F_y \end{pmatrix}. \quad (25)$$

Proof of Theorem A.1

1. Saying that p is a singular point of $G|S$ is equivalent to $dG_p(v) = 0$, for all $v \in T_pS$. Since $T_pS = \ker dF_p$, we see that this is equivalent to the existence of a scalar λ such that $dG_p = \lambda\, dF_p$.

2. Since 0 is a regular value of F, we have $\nabla F(p) \neq 0$. We assume that $F_x(p) \neq 0$, and argue similarly in case $F_y(p) \neq 0$ or $F_z(p) \neq 0$. Furthermore, assume that $p = (0, 0, 0)$. According to the Implicit Function Theorem, there is a unique local solution $x = f(y, z)$, with $f(0, 0) = 0$, of the equation $F(x, y, z) = 0$. Implicit differentiation yields

$$F_x\, f_y + F_y = 0, \quad (26)$$

$$F_x\, f_z + F_z = 0, \quad (27)$$

where f_y and f_z are evaluated at (y, z), and F_x, F_y and F_z are evaluated at $(f(y, z), y, z)$. Similarly,

$$F_{xx}\, f_y^2 + 2F_{xy} f_y + F_{yy} + F_x f_{yy} = 0. \quad (28)$$

Similar identities are obtained by differentiating (26) with respect to y, and (27) with respect to z.

Using y and z as local coordinates on S, we obtain the following expression of $G|S$ with respect to these local coordinates:

$$g(y, z) = G(f(y, z), y, z).$$

Differentiating this identity twice with respect to y we obtain

$$g_{yy} = G_{xx}\, f_y^2 + 2G_{xy} f_y + G_{yy} + G_x\, f_{yy}. \quad (29)$$

Since $F_x(p) \neq 0$, we solve f_{yy} from (28), and plug the resulting expression into (29), to get

$$g_{yy} = (G_{xx} - \lambda F_{xx})\, f_y^2 + 2(G_{xy} - \lambda F_{xy})\, f_y + (G_{yy} - \lambda F_{yy}), \quad (30)$$

where

$$\lambda = \frac{G_x}{F_x}$$

is the Lagrange multiplier, cf. (22). We rewrite (30) as

$$g_{yy} = \begin{pmatrix} f_y & 1 & 0 \end{pmatrix} (H_G - \lambda\, H_F) \begin{pmatrix} f_y \\ 1 \\ 0 \end{pmatrix}$$

$$= \frac{1}{F_x^2} \begin{pmatrix} F_y & -F_x & 0 \end{pmatrix} (H_G - \lambda\, H_F) \begin{pmatrix} F_y \\ -F_x \\ 0 \end{pmatrix}$$

We similarly derive

$$g_{yz} = \frac{1}{F_x^2} \begin{pmatrix} F_y & -F_x & 0 \end{pmatrix} (H_G - \lambda\, H_F) \begin{pmatrix} F_z \\ 0 \\ -F_x \end{pmatrix},$$

and

$$g_{zz} = \frac{1}{F_x^2} \begin{pmatrix} F_z & 0 & -F_x \end{pmatrix} (H_G - \lambda\, H_F) \begin{pmatrix} F_z \\ 0 \\ -F_x \end{pmatrix}.$$

Since the vectors $(F_y, -F_x, 0)$ and $(F_z, 0, -F_x)$ span the tangent space T_pS, we see that

$$\begin{pmatrix} g_{yy} & g_{yz} \\ g_{yz} & g_{zz} \end{pmatrix} = \frac{1}{F_x^2} \Delta.$$

B. METABALLS

One approach to modeling using implicit surfaces is to use metaballs. A single metaball consists of a density function around a single point. The function value at the point equals a given weight, and drops to zero at a given distance (the radius) from the centre. By adding individual metaballs and subtracting a threshold value, blobby objects can be joined smoothly.

For a single metaball with weight w and radius R we use the density function

$$D(P) = \begin{cases} w\left(1 - \left(\frac{r}{R}\right)^2\right)^2 & r < R \\ 0 & r \geq R \end{cases}$$

where r is the distance from P to the centre of the metaball.

Using threshold T, the implicit function for the metaball object is given by

$$F(P) = \sum_{i=1}^{n} D_i(P) - T.$$