Geometry of the Squared Distance Function to Curves and Surfaces

Helmut Pottmann and Michael Hofer

Institute of Geometry
Vienna University of Technology
Wiedner Hauptstraße 8–10, A-1040 Wien
pottmann@geometrie.tuwien.ac.at
hofer@geometrie.tuwien.ac.at

Summary. We investigate the geometry of that function in the plane or 3-space, which associates to each point the square of the shortest distance to a given curve or surface. Particular emphasis is put on second order Taylor approximants and other local quadratic approximants. Their key role in a variety of geometric optimization algorithms is illustrated at hand of registration in Computer Vision and surface approximation.

Key words: Squared distance function, local quadratic approximants, registration, Computer Vision, geometric optimization, surface approximation, differential geometry

1 Introduction

The distance function of a curve or surface Φ, which assigns to each point $\mathbf{p}$ of the embedding space the shortest distance of $\mathbf{p}$ to Φ frequently appears in problems of geometric computing. Examples include curve and surface approximation in CAD and Geometric Modeling, registration in Computer Vision and positioning problems in Robotics [9].

A variety of contributions deals with the computation of the distance function; in many cases this computation aims towards the singular set of the function, i.e., towards points where the function is not smooth, since those points lie on the medial axis (or skeleton) of the input shape.

Early work on the geometry of the distance function comes from the classical geometric literature of the 19th century. One looks at its graph surface, which consists of developable surfaces of constant slope and applies results of classical differential geometry, line and sphere geometry (for a modern presentation, see e.g. [21]). This approach is also naturally related to the medial axis transform (MAT) which appears as cyclographic preimage in the classical language. More recent work on the MAT comes from H. I. Choi [8], especially the decomposition result for the efficient computation, which also appears in Kimmel et al. [15]. The importance of the skeleton, medial axis and medial axis transform as a shape descriptor has been realized by Blum

[4], and this created a large body of literature that deals with its computation and applications in object recognition, registration and mathematical morphology [14, 24, 25]. Also in a discrete setting (pixel plane, voxel space), distance transforms and the skeleton received a lot of attention (see e.g. [18, 24, 25, 28]).

The distance function is also the (viscosity) solution of the so-called eikonal equation. Its numerical computation is not trivial because it is a hyperbolic equation and an initially smooth front may develop singularities (shocks) as it propagates. Precisely the latter belong to the medial axis and are of particular interest. The computation of viscosity solutions with the level set method of Osher and Sethian [19] proved to be a very powerful approach (see e.g. [23, 26, 27]). Distance functions also play a role in other curve and surface evolutions [23] and in the visualization of large sets of unorganized data sets [29]. The vector distance function of an object, which connects each point in space to its closest point on the object, has been used by Faugeras and Gomes [10] for the evolution of objects of arbitrary dimension and codimension.

The level sets of the distance functions are the *offsets* which are of particular importance in Computer Aided Design (see e.g. [20, 21] and the literature therein) and also appear in mathematical morphology [24]. Results on the distance function and its application to shape interrogation for CAD/CAM may also be found in the very recent book by Patrikalakis and Maekawa [20].

In the present paper, instead of the distance function we rather study the *squared distance function*. Clearly, also the latter is singular in the points of the medial axis and thus the relation to the literature just pointed out is important. However, we are mainly interested in local quadratic approximants at points different from the medial axis. A contribution dealing with local properties of the squared distance function of a manifold has been given by Ambrosio and Mantegazza [1]. Among other interesting results, the authors describe the relation between third order directional derivatives of the squared distance function (taken at points of the manifold) and the curvature invariants of the manifold. Despite the importance of the squared distance function, little effort has so far been made in better understanding, approximating and representing this function for efficient computing.

In the present paper, we will start an investigation of this subject. To get a better understanding of the nature of the problem, we first deal with planar curves. There, we can give a simple kinematic generation of the 3D graph surface of the squared distance function d^2. This is important for visualization of the behaviour of d^2 and it greatly helps us to find simple derivations for *local quadratic approximants* to d^2. The latter are of particular importance for optimization algorithms involving d^2 and therefore they are also studied for surfaces and space curves.

At hand of two examples we outline how to use the results of the present paper in certain types of geometric optimization algorithms which involve

quadratic approximants to the squared distance function of a surface. The two applications we look at are registration in Computer Vision and surface approximation. An important advantage of the new approach to approximation with parametric surfaces, such as B-spline surfaces, is that we avoid the parameterization problem. The registration problem we look at is the optimal matching of a point cloud to a surface. The essential difference in our approach to standard techniques, such as the *iterative closest point (ICP) algorithm* [2] is, that we do not have to compute pairs of corresponding points.

2 Graph Surface of the Squared Distance Function to a Planar Curve

In Euclidean 3-space $\mathbb{R}^3$, we consider a planar C^2 curve $\mathbf{c}(t)$ with parameterization $(c_1(t), c_2(t), 0)$. The tangent and normal line at a curve point $\mathbf{c}(t)$ are denoted by T and N respectively. To each point $(x, y, 0)$ we compute the shortest distance $d(x, y)$ of that point to the curve $\mathbf{c}$. For visualization of the squared distance function d^2 we investigate the graph surface $\Gamma : (x, y, d^2(x, y))$ of the function d^2. By neglecting global effects of the distance function we will at first construct a surface Φ, part of which is the desired graph surface Γ. In a second step we will perform the appropriate trimming which reduces Φ to Γ.

The Frenet frame at a curve point $\mathbf{c}(t)$ consists of the unit tangent vector $\mathbf{e}_1 = \dot{\mathbf{c}}/\|\dot{\mathbf{c}}\|$ and the normal vector $\mathbf{e}_2(t)$. The two vectors form a right-handed Cartesian system in the plane. With $\mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2 = (0, 0, 1)$ this system is extended to a Cartesian system Σ in $\mathbb{R}^3$. Coordinates with respect to Σ are denoted by (x_1, x_2, x_3). The system Σ depends on t and shall have the curve point $\mathbf{c}(t)$ as origin. At least locally, the shortest distance of a point $(0, u, 0)$ on the x_2-axis (curve normal) is its x_2-coordinate u. For each t, locally the graph points $(0, u, u^2)$ form a parabola p in the normal plane of $\mathbf{c}(t)$. This parabola can be considered fixed in Σ. Varying t, the positions of the parabola in the original system generate a surface Φ. A parameterization of Φ is

$$\mathbf{x}(t, u) = \mathbf{c}(t) + u\mathbf{e}_2(t) + u^2\mathbf{e}_3. \tag{1}$$

It is well-known and an immediate consequence of the Frenet equations that the instantaneous motion of the system Σ is a rotation about the axis of the osculating circle of $\mathbf{c}$. These curvature axes form a cylinder surface Λ with the evolute of $\mathbf{c}$ as orthogonal cross section and rulings parallel to $(0, 0, 1)$. Hence, the motion of Σ with respect to the fixed system can be considered as a rolling motion of the normal plane of $\mathbf{c}$ on the evolute cylinder Λ of $\mathbf{c}$. In usual terminology, the surface Φ is a *moulding surface* (cf. Fig. 1). Note that the contour lines of Φ at height u^2 are translates of the offsets of $\mathbf{c}$ at distance u.

The surface Φ contains a singular curve (*curve of regression*) $\mathbf{l}(t)$, which is contained in the evolute cylinder Λ. This is in accordance to the well-known
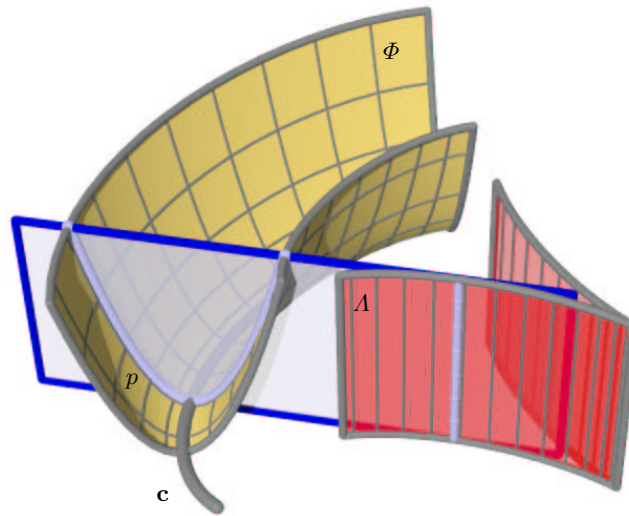
Fig. 1. When the normal plane of the planar curve **c** rolls on the evolute cylinder Λ of **c**, the parabola p generates the moulding surface Φ

result that singularities of the offsets of a curve occur at points of its evolute. The rolling motion also shows that the development of the cylinder Λ maps the curve **l** to a parabola which is congruent to the profile parabola p.
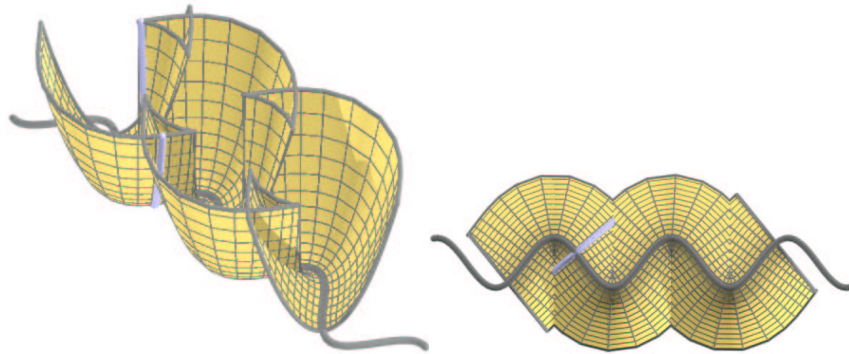

Fig. 2. A moulding surface Φ with self-intersections in an axonometric view (**left**) and viewed from below, where the visible points correspond to the graph Γ of the squared distance function to a sine curve (**right**)

The generated surface Φ is in general not the graph of a function in the horizontal xy-plane. It may happen, that a vertical line through $(x, y, 0)$ intersects Φ in several points. Among those points, the one with the smallest z-coordinate lies on the desired graph surface Γ, whereas the others do not.

Thus Φ can be trimmed so as to form exactly the graph Γ of the squared distance function to **c**. The trimming can be performed with a *visibility test*. We view Φ in direction $(0,0,1)$, i.e., from below. Then, exactly the visible points are those closest to the xy-plane π and therefore the points of Γ (cf. Fig. 2). Trimming has to be performed at self-intersections of Φ. Projecting these curves orthogonally into π we obtain the *cut locus* of **c**. It is formed of those points in π, for which the shortest distance occurs at more than one normal. The limit points of this set are usually added to the cut locus: these points are curvature centers to points of **c** with locally extremal curvature. We should mention that for a closed boundary curve **c** of a planar domain D, the part of the cut locus which lies in D is also called the *medial axis* of D. It is also well-known that the trimming procedure cuts apart the singular curve **l**. Just special points of it, namely those which belong to the end points of branches of the cut locus, remain. We summarize the basically well-known results as follows.

Proposition 1. *The graph surface Γ of the squared distance function to a planar curve **c** is contained in a moulding surface Φ. The surface Φ is generated by a parabola p with parameter 1 whose plane rolls on the evolute cylinder Λ of **c**, such that p's vertex moves along **c** and p's axis remains orthogonal to the plane π of **c**. Exactly those parts of Φ lie on the graph surface Γ which are visible for an orthogonal projection onto π when viewing Φ from below (i.e., from that side of π which does not contain points of Φ).*

Remark 1. In an analogous way we may construct the graph surface of any other function $f(d)$ of the distance d. There, the profile curve has in Σ the parameterization $(0, u, f(u))$. One has to be careful with signs, however. This is easily understood at hand of the simplest example, namely the *distance function d* itself. With a signed distance function whose sign is given by the orientation of the curve normal $\mathbf{e}_2$, the profile is simply the line $(0, u, u)$ and the generated moulding surface is a developable surface of constant slope (see e.g. [21]). For a nonnegative distance, the profile is $(0, u, |u|)$ and the moulding surface is generated from the developable surface described above by reflecting the part below π at π.

In case that f is nonnegative and monotonically increasing, we perform the trimming operation as in the case of $f(d) = d^2$, i.e., with a visibility algorithm viewing the graph orthogonal to π from below. For a monotonically decreasing function, we have to view from above.

An interesting example of this type is the inverse distance function $f(d) = 1/|d|$, which is sometimes built around obstacles in robot motion planning. There, with a nonnegative distance, the profile curve of the moulding surface Φ is formed by two hyperbola segments. An image of the trimmed graph to an ellipse **c** is shown in Fig. 3. Note that for a **c** which bounds a convex domain D, there is no trimming required outside D. Regardless of f, trimming happens at the points of π which lie on the cut locus, and thus global results

on cut locus, medial axis and singularities of offsets can be used to identify regions where no trimming is required (see [21]).
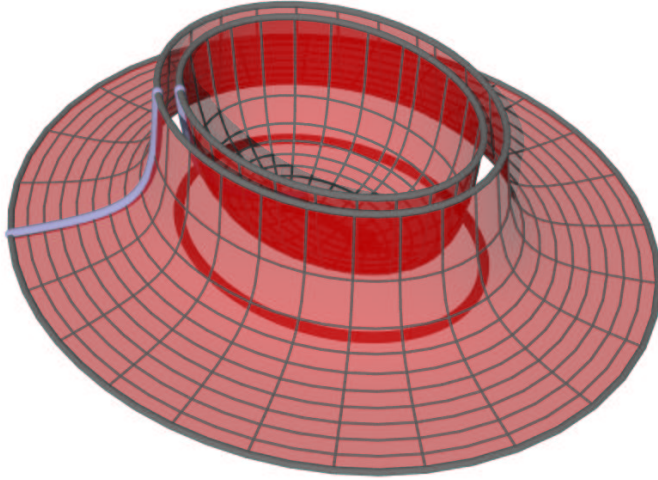

Fig. 3. Graph of the inverse distance function to an ellipse

3 Quadratic Approximations to d^2

The previous section visualized a problem of the squared distance function, namely its non-smoothness at points of the cut locus. When we now study local quadratic (Taylor) approximants, we do not consider the global effects, and give formulae for local approximants which work on the local distance function. This means that in determining d for neighboring points of $\mathbf{p}$ we are only locally varying the footpoint of the normal to the curve $\mathbf{c}$. In other words, at points of the medial axis we work with just one sheet of the surface Φ.

Consider a point $\mathbf{p}$ in π whose coordinates in the Frenet frame at the normal footpoint $\mathbf{c}(t_0)$ are $(0, d)$. The curvature center $\mathbf{k}(t_0)$ at $\mathbf{c}(t_0)$ has coordinates $(0, \varrho)$. Here, ϱ is the inverse curvature $1/\kappa$ and thus has the same sign as the curvature, which depends on the orientation of the curve. Since all level sets of d^2 (offsets) to points on the curve normal share $\mathbf{k}$ as curvature center, we see that *the squared distance function to the given curve and to its osculating circle at the normal footpoint agree up to second order*.

A visualization is as follows (see Fig. 4): rotating the profile parabola $p(t_0)$ around the curvature axis (vertical line through $\mathbf{k}$) results in a surface of revolution Ψ which has second order contact with Φ at all points of $p(t_0)$. This is a well-known curvature property of moulding surfaces. In the Frenet

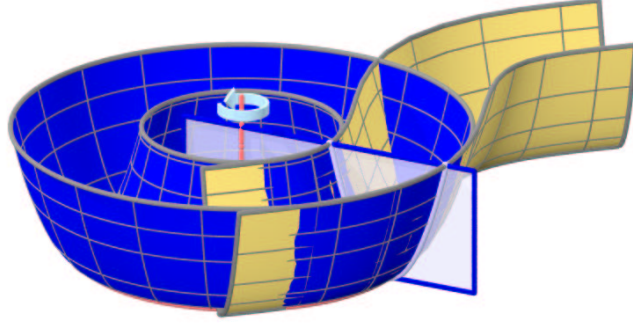**Fig. 4.** Surface of revolution Ψ which has second order contact with the moulding surface Φ at all points of $p(t_0)$

frame, the function d^2 to the osculating circle is

$$f(x_1, x_2) = \left(\sqrt{x_1^2 + (x_2 - \varrho)^2} - |\varrho| \right)^2. \tag{2}$$

The graph of this function is the surface of revolution Ψ. The second order Taylor approximant F_d of f at $(0, d)$ is found to be

$$F_d(x_1, x_2) = \frac{d}{d - \varrho} x_1^2 + x_2^2. \tag{3}$$

Let us discuss the various cases.

- For $d = 0$ we get the Taylor approximant $F_0 = x_2^2$ at the normal footpoint. This shows the following interesting result: *At a point* $\mathbf{p}$ *of a curve* $\mathbf{c}$ *the second order approximant of the squared distance function to* $\mathbf{c}$ *and to the curve tangent* T *at* $\mathbf{p}$ *are identical.* Visually, this is not unexpected since curvature depends on the scale. Zooming closer to the curve it appears less and less curved. The graph surface Γ_0 of F_0 is a parabolic cylinder with rulings parallel to the curve tangent (see Fig. 5).
- For $d \to \infty$, the Taylor approximant tends to $F_\infty = x_1^2 + x_2^2$. This is the squared distance function to the footpoint $\mathbf{c}(t_0)$. The graph Γ_∞ of F_∞ is a paraboloid of revolution.
- For general d, it may be advantageous to view F as combination of F_0 and F_∞,

$$F_d(x_1, x_2) = \frac{d}{d - \varrho} \left(x_1^2 + x_2^2 \right) - \frac{\varrho}{d - \varrho} x_2^2 = \frac{d}{d - \varrho} F_\infty - \frac{\varrho}{d - \varrho} F_0. \tag{4}$$

This form is particularly useful for computing the second order approximant in the original x, y-system. Clearly, F_d is not defined at the curvature
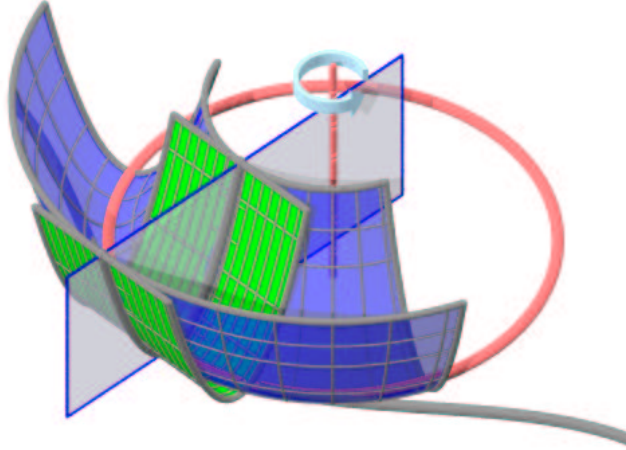
Fig. 5. The graph surface Γ_0 of F_0 is a parabolic cylinder with rulings parallel to the curve tangent

center $d = \varrho$, where we have a singularity. Otherwise, we see that the type of the graph surface Γ_d depends on $s = sign[d/(d - \varrho)]$. A value $s > 0$ yields an elliptic paraboloid (see Fig. 6), $s = 0$ the parabolic cylinder Γ_0 (see Fig. 5), and $s < 0$ a hyperbolic paraboloid (see Fig. 7). The latter case belongs exactly to points between the curve point and the curvature center, i.e., $0 < d < \varrho$ or $0 > d > \varrho$. Here, the quadratic approximant also assumes negative values. In all other cases this does not happen.

Note also that all quadratic approximants F_d agree along the curve normal and are symmetric with respect to it. The graph paraboloids Γ_d touch the cylinder Γ_0 along the profile parabola $p(t_0)$.

 – For an *inflection point* we have $\kappa = 0$ and thus $F_d = x_2^2$. This reflects the trivial fact that the squared distance function to the tangent is a second order approximant along the whole curve normal.

For the applications we have in mind, it can be important to employ *nonnegative* quadratic approximants to d^2. Thus, we briefly address a convenient way to deal with those approximants.

Negative function values of the quadratic approximant F_d arise for a distance d with $0 < d < \varrho$ or $0 > d > \varrho$. Without loss of generality, we may assume an appropriate local orientation of the curve **c** such that $\varrho > 0$. Thus, only the case $0 < d < \varrho$ needs to be discussed. We fix such a distance and call it D. Our goal is to replace the local quadratic approximant F_D by a nonnegative quadratic approximant F_D^+ with the following property: For all points **x** whose distance d to **c** is less than the given value D, the local quadratic approximant F_D^+ also returns a value $< D^2$.
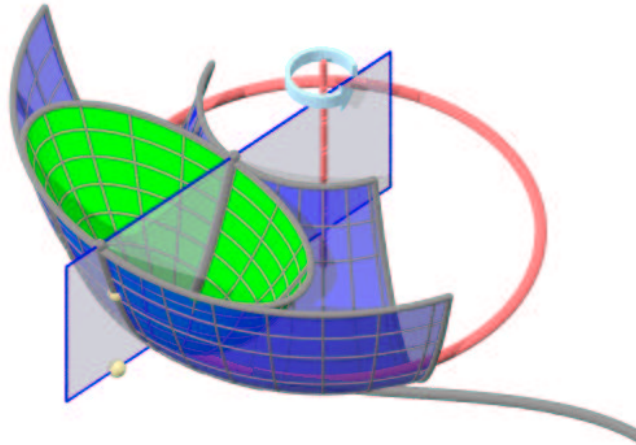
Fig. 6. The graph surface Γ_d of F_d is an elliptic paraboloid for $s > 0$
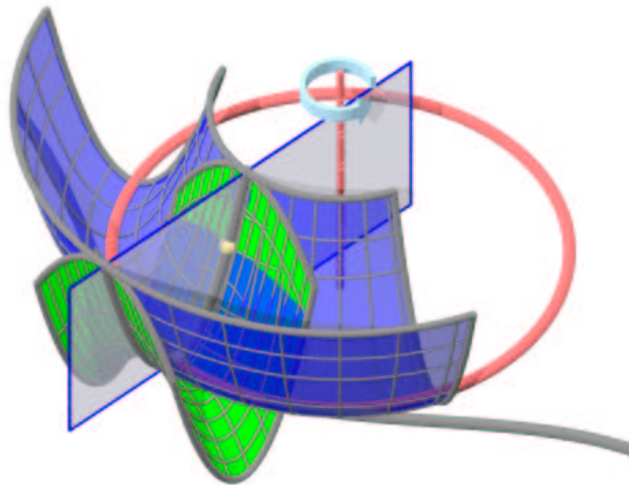

Fig. 7. The graph surface Γ_d of F_d is a hyperbolic paraboloid for $s < 0$

Let us look at the level sets of F_D, whose graph is a hyperbolic parabloid. Those points $\mathbf{x}$ in the plane whose squared distance to $\mathbf{c}$ is less than D^2 lie between the two offset curves $\mathbf{c}_D$ and $\mathbf{c}_{-D}$ at oriented distances D and $-D$. The level set of F_D is a hyperbola h_D. Its axes agree with those of the Frenet frame at $\mathbf{c}(t_0)$ and it has second order contact with the offset $\mathbf{c}_D$ at the point $(0, D)$, where we are looking for the local quadratic approximant (see Fig. 8).

The region of points $\mathbf{x}$ with $F_D(\mathbf{x}) < D^2$ is bounded by this hyperbola. Of course, there are points with $F_D(\mathbf{x}) < D^2$ and $d^2(\mathbf{x}) > D^2$.
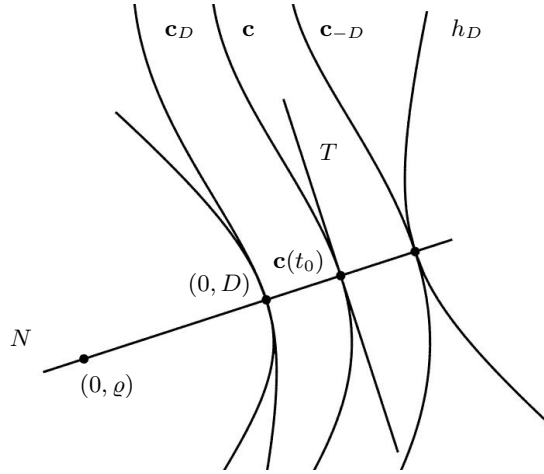
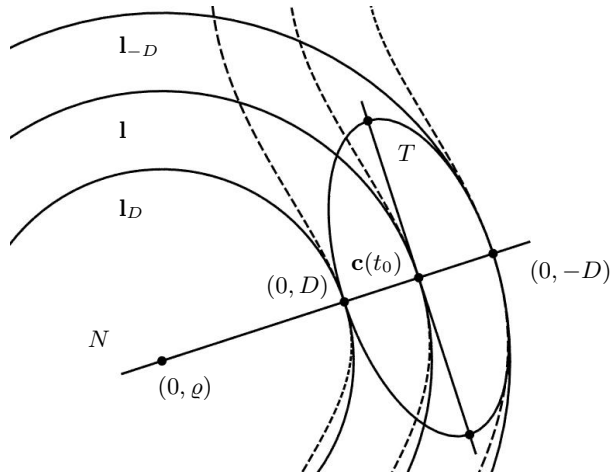**Fig. 8.** Level set of F_D at height D^2 is the hyperbola h_D

Fig. 9. Level set of F_{-D} at height D^2 is an ellipse

To arrive at a practically useful result, we make a further simplification. We replace the curve by its osculating circle $\mathbf{l}$ at $\mathbf{c}(t_0)$. Points whose distance to $\mathbf{l}$ is smaller than D lie in an annulus A bounded by two concentric circles $\mathbf{l}_D$, $\mathbf{l}_{-D}$ with midpoint $(0, \varrho)$ and radii $\varrho - D$ and $\varrho + D$ (see Fig. 9). To

warrant the symmetry and the precision of the approximant along the curve normal, we have to take as F_D^+ an F_d to a point $(0, d)$ with $d < 0$. The region with $F_d(\mathbf{x}) < D^2$ is then an ellipse coaxial with the Frenet frame axes $\mathbf{e}_1, \mathbf{e}_2$, and with points $(0, D)$ and $(0, -D)$ as two vertices. The solution F_D^+ of our problem must give rise to the *largest* ellipse which still lies inside the annulus A. It is easy to prove that it has to have second order contact (in fact, fourth order contact) with the larger bounding circle of A at point $(0, -D)$. Therefore, it is the level set of the quadratic approximant F_{-D}. Thus, our solution is

$$F_D^+ = F_{-D}. \tag{5}$$

Hence, at points with an indefinite quadratic approximant we just use the nonnegative quadratic approximant at the point which is obtained by reflection at $\mathbf{c}(t_0)$. In other words, we may use the formula

$$F_d^+(x_1, x_2) = \frac{d}{d + \varrho} x_1^2 + x_2^2, \tag{6}$$

where we do not have to care about signs and always take $d > 0$ and $\varrho > 0$. This would yield 'wrong' results for distances $d > \varrho > 0$. However, a point $\mathbf{x}$ beyond the curvature center $\mathbf{k}(t)$ at $\mathbf{c}(t)$ always has another normal footpoint which is closer to it than $\mathbf{c}(t)$ and thus this case does not occur when we are looking at globally shortest distances.

4 Squared Distance Function to a Surface and its Second Order Approximants

What we said about d^2 to a planar curve can in principle be extended to the squared distance function to a surface in $\mathbb{R}^3$. However, the visualization with help of a graph in $\mathbb{R}^4$ becomes harder. Again we have a local behaviour of the distance function d^2, which we will investigate up to second order, and a global behaviour. The global effects cause non-smoothness at points of the cut locus, which in general consists of surfaces.

Consider an oriented surface $\mathbf{s}(u, v)$ with a unit normal vector field $\mathbf{n}(u, v) = \mathbf{e}_3(u, v)$. At each surface point $\mathbf{s}(u, v)$, we have a local right-handed Cartesian system whose first two vectors $\mathbf{e}_1, \mathbf{e}_2$ determine the principal curvature directions. The surface normal and the principal tangents are denoted by N, T_1 and T_2, respectively. The latter are not uniquely determined at an umbilical point. There, we can take any two orthogonal tangent vectors $\mathbf{e}_1, \mathbf{e}_2$. We will refer to the thereby defined frame as *principal frame* $\Sigma(u, v)$. Let κ_i be the (signed) principal curvature to the principal curvature direction $\mathbf{e}_i$, $i = 1, 2$, and let $\varrho_i = 1/\kappa_i$. Then, the two principal curvature centers at the considered surface point $\mathbf{s}(u, v)$ are expressed in Σ as $\mathbf{k}_i = (0, 0, \varrho_i)$. The quadratic approximant F_d to d^2 at $(0, 0, d)$ is the following.

Proposition 2. *The second order Taylor approximant of the squared distance function to a surface at a point* **p** *is expressed in the principal frame at the normal footpoint via*

$$F_d(x_1, x_2, x_3) = \frac{d}{d - \varrho_1} x_1^2 + \frac{d}{d - \varrho_2} x_2^2 + x_3^2. \tag{7}$$

Proof. We just give a sketch of the proof. We first have to show that it is sufficient to approximate the surface at the footpoint $\mathbf{s}(u, v)$ up to second order. This can be done with well-known results on the curvature behaviour of offset surfaces. Hence, we may replace the surface locally by an osculating torus T, which is obtained by rotating the first principal curvature circle c_1 (center $\mathbf{k}_1$, radius $|\varrho_1|$, in the plane $x_2 = 0$) around the axis of the other principal curvature circle c_2, which is defined analogous to c_1 (see Fig. 10). Clearly, by exchanging the roles of c_1 and c_2, we obtain two such tori. Any one is fine for our purposes. The level sets of the squared distance function to T are coaxial and concentric tori. Particularly, in the symmetry planes $x_1 = 0$ and $x_2 = 0$, we have the function d^2 to the principal curvature circles c_2 and c_1, respectively. Hence, almost all first and second order partial derivatives of the squared distance function with respect to x_i can be taken from the planar case. The only one, for which this is not true, is $\partial^2 f / \partial x_1 \partial x_2$. However, the second order quadratic approximant must be symmetric with respect to the planes $x_1 = 0$ and $x_2 = 0$. Hence, this mixed partial derivative vanishes. Together with (3) this proves (7). □
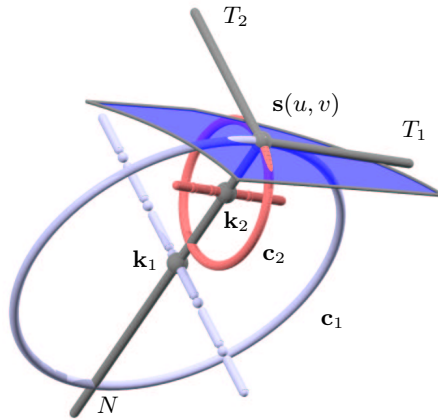

Fig. 10. Principal frame and principle circles at an elliptic surface point $\mathbf{s}(u, v)$

The discussion of the arising cases is as in Sect. 2. At first, let us point to the case $d = 0$.

Proposition 3. *At a point* $\mathbf{p}$ *of a surface* $\mathbf{s}$ *the second order approximant of the squared distance functions to* $\mathbf{s}$ *and to the surface's tangent plane at* $\mathbf{p}$ *are identical.*

More generally, it is easy to show that if two curves or surfaces possess contact of order k at some point, their respective squared distance functions agree of order $k + 1$ there. One example of this is the fact that the second order Taylor approximation of the squared distance function at a point of a surface $\mathbf{s}$ agrees with the squared distance function of the tangent plane. A second example is the result of Ambrosio and Mantegazza [1], which says that third order derivatives of the squared distance function at points of $\mathbf{s}$ are related to the curvature invariants of $\mathbf{s}$.

For a further discussion, we distinguish between the types of surface points.

- In the case of an *elliptic point*, $\varrho_1 \varrho_2 > 0$, we assume a surface normal orientation and an indexing of the principal curvature directions that yield $\varrho_1 \geq \varrho_2 > 0$. Points of the normal, which lie on the other side of the tangent plane than the principal curvature centers, i.e., $d < 0$, cause positive factors $d/(d-\varrho_i)$, and hence a positive definite quadratic form F_d. This case always arises at points outside a closed convex surface $\mathbf{s}$. For points between the principal curvature center $\mathbf{k}_1$ and the surface point, i.e., $0 < d < \varrho_1$, we get an indefinite F_d, and for $d > \varrho_1$, we again have a positive definite second order approximant. Clearly, we exclude evaluation at the principal curvature centers, where we have a singularity.
- For a *hyperbolic point* $\mathbf{s}$, we assume w.l.o.g. $\varrho_1 < 0$ and $\varrho_2 > 0$. Here, points between the principal curvature centers belong to $\varrho_1 < d < \varrho_2$ (excluding $d = 0$) and result in an indefinite F_d, whereas for points outside the principal curvature segment $\mathbf{k}_1\mathbf{k}_2$ the approximants F_d are positive definite. There, the level sets $F_d = c$ to any constant $c > 0$ are homothetic ellipsoids, centered at the surface point $\mathbf{s}$, with the axes of the principal frame as axes. In the indefinite cases, the level sets are hyperboloids (of one or two sheets).
- At a *parabolic point* we may assume $\kappa_1 = 0$ and $\kappa_2 > 0$. Now, the second order approximant to d^2 reads

$$F_d(x_1, x_2, x_3) = \frac{d}{d - \varrho_2} x_2^2 + x_3^2. \tag{8}$$

This shows that we never get a positive definite F_d. The level sets of F_d are, in general, cylinder surfaces with rulings parallel to the x_1-axis (principal curvature direction with vanishing curvature).
- A *flat point* is characterized by $\kappa_1 = \kappa_2 = 0$ and thus yields the obvious result $F_d = x_3^2$. Hence, the squared distance function to the tangent plane agrees along the whole surface normal up to second order with the squared distance function to the surface $\mathbf{s}$.

Analogously to the case of planar curves we derive nonnegative quadratic approximants with

$$F_d^+(x_1, x_2, x_3) = \frac{d}{d + \varrho_1} x_1^2 + \frac{d}{d + \varrho_2} x_2^2 + x_3^2, \tag{9}$$

where d, ϱ_1, ϱ_2 are taken as positive values. Again, points beyond the principal curvature centers are ruled out, but they do not arise anyway when considering global distances.

5 Squared Distance Function to a Space Curve

The study of the squared distance function to a C^2 space curve $\mathbf{c}(t)$ is also interesting in connection with surfaces, namely in regions where the closest points on the surface are on a boundary curve or at a curved edge, i.e., a surface curve with tangent plane discontinuities. Again, we are focussing on the second order approximants, and this we do from a local point of view.

Given a point $\mathbf{p}$ in $\mathbb{R}^3$, the shortest distance to the curve $\mathbf{c}$ occurs along a normal of the curve or at a boundary point of it. The latter case is trivial and thus we exclude it. At the normal footpoint $\mathbf{c}(t_0)$, with tangent line T, we form a Cartesian system with $\mathbf{e}_1$ as tangent vector and $\mathbf{e}_3$ in direction of the vector $\mathbf{p} - \mathbf{c}(t_0)$. This *canonical frame* can be viewed as limit case of the principal frame for surfaces, when interpreting the curve as pipe surface with vanishing radius. By this limit process, we can also show the following result.

Proposition 4. *The second order Taylor approximant of the squared distance function to a space curve* $\mathbf{c}$ *at a point* $\mathbf{p}$ *is expressed in the canonical frame* Σ *at the normal footpoint via*

$$F_d(x_1, x_2, x_3) = \frac{d}{d - \varrho_1} x_1^2 + x_2^2 + x_3^2. \tag{10}$$

Here, $(0, 0, \varrho_1)$ *are the coordinates (in* Σ*) of the intersection point of the curvature axis of* $\mathbf{c}$ *at the footpoint* $\mathbf{c}(t_0)$ *with the perpendicular line* N *from* $\mathbf{p}$ *to* $\mathbf{c}(t_0)$.

Proof. It is sufficient to consider the squared distance function to the osculating circle c of the curve at $\mathbf{c}(t_0)$. For any torus with spine circle c, the principal curvature lines are the family of parallel circles and the family of meridian circles. By Meusnier's theorem, the principal curvature centers to the parallel circles lie on the rotational axis of the torus. This is the axis of the spine circle c. If we now shrink the radius of the meridian circles to 0, we get in the limit principal curvature centers $\mathbf{k}_1 = (0, 0, \varrho_1)$ on the axis. Here,
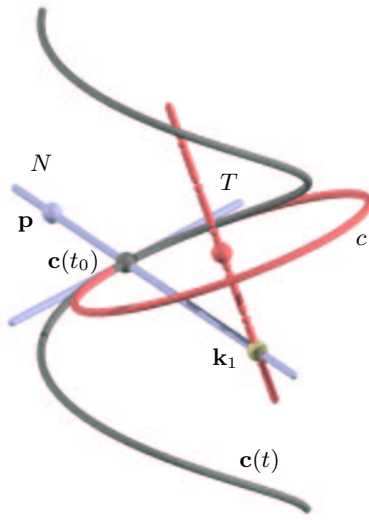
$$\varrho_1 = \varrho / \cos \alpha, \tag{11}$$

PSfrag replacements

Fig. 11. A space curve $\mathbf{c}(t)$ with the osculating circle c at $\mathbf{c}(t_0)$

with ϱ as curvature radius of $\mathbf{c}$ at $\mathbf{c}(t_0)$ and α as angle between the normal (x_3-axis) and the osculating plane at $\mathbf{c}(t_0)$ (see Fig. 11). With $\varrho_2 = 0$, the result follows from (7). □

The discussion of the different cases of F_d is a limit case of the situation for surfaces and thus we omit it here.

6 Application to Geometric Optimization Problems

Optimization problems in geometric computing are frequently nonlinear and involve the squared distance function to a curve or surface. It is therefore natural to apply the results discussed above to the development of geometric optimization algorithms which are based on local quadratic approximants of the function to be minimized, such as quasi-Newton and SQP-type algorithms [5]. We will now outline this idea at hand of two important examples: registration and surface approximation.

6.1 Registration of a Point Cloud to a CAD Model

Suppose that we are given a large number of 3D data points that have been obtained by some 3D measurement device (laser scan, light sectioning, ...) from the surface of a technical object. Furthermore, let us assume that we also have got the CAD model of this workpiece. This CAD model shall describe

the 'ideal' shape of the object and will be available in a coordinate system that is different from that of the 3D data point set. For the goal of shape inspection it is of interest to find the optimal Euclidean motion (translation and rotation) that aligns, or registers, the point cloud to the CAD model. This makes it possible to check the given workpiece for manufacturing errors and to classify the deviations.

A well-known standard algorithm to solve such a registration problem is the *iterative closest point (ICP) algorithm* which has been introduced by Chen and Medioni [7] and Besl and McKay [2].

In the first step of each iteration, for each point of the data point cloud the closest point in the model shape is computed. As result of this first step one obtains a point sequence $Y = (\mathbf{y}_1, \mathbf{y}_2, \ldots)$ of closest model shape points to the data point sequence $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots)$. Each point $\mathbf{x}_i$ corresponds to the point $\mathbf{y}_i$ with the same index.

In the second step of each iteration the rigid motion M is computed such that the moved data points $M(\mathbf{x}_i)$ are closest to their corresponding points $\mathbf{y}_i$, where the objective function to be minimized is

$$\sum_i \|\mathbf{y}_i - M(\mathbf{x}_i)\|^2. \tag{12}$$

This least squares problem can be solved explicitly, cf. [2]. The translational part of M brings the center of mass of X to the center of mass of Y. The rotational part of M can be obtained as the unit eigenvector that corresponds to the maximum eigenvalue of a symmetric 4×4 matrix. The solution eigenvector is nothing but the unit quaternion description of the rotational part of M.

After this second step the positions of the data points are updated via $X_{\text{new}} = M(X_{\text{old}})$. Now step 1 and step 2 are repeated, always using the updated data points, until the change in the mean-square error falls below a preset threshold. The ICP algorithm always converges monotonically to a local minimum, since the value of the objective function is decreasing both in steps 1 and 2. A summary with new results on the acceleration of the ICP algorithm has been given by Rusinkiewicz and Levoy [22], who also suggest that *iterative corresponding point* is a better expansion for the abbreviation ICP than the original *iterative closest point*.

Actually, we want to apply that motion to the data point cloud which brings it into a position where the sum f of squared distances to the CAD-model is minimal. Minimizing f under the constraint that the applied transformation is a rigid body motion can be done with a constrained optimization algorithm. We do not directly apply a standard implementation, but find a geometrically motivated algorithm. It works with local quadratic approximants of f, and a linearization of the motion.

ICP is also based on local quadratic approximants of f, but a major disadvantage of ICP is the following: data points $\mathbf{x}_i$ are moved towards the

normal footpoints $\mathbf{y}_i$. In other words, the squared distance to $\mathbf{y}_i$ is used as a quadratic approximant of the function d^2 at $\mathbf{x}_i$. As we know from the study of d^2, this approximation is only good in the 'far field', i.e., for large distances d. However, in the practical application one has to start with an initial position of the point cloud which is sufficiently close to the CAD model in order to run into the right local minimum. Thus, typically the involved distances are small. As a simple solution, we therefore may use the squared distance function to the tangent plane at the normal footpoint $\mathbf{y}_i$.

Our proposed algorithm works as follows: the first step is similar to that of the ICP algorithm. For each data point $\mathbf{x}_i \in X$ determine the nearest point $\mathbf{y}_i$ of the model surface and determine the tangent plane there. Let $\mathbf{n}_i$ denote a unit normal vector of this tangent plane in $\mathbf{y}_i$. Because $\mathbf{y}_i$ is the nearest point to $\mathbf{x}_i$ on the surface, $\mathbf{x}_i$ lies on the surface normal in $\mathbf{y}_i$, i.e., $\mathbf{x}_i = \mathbf{y}_i + d_i \mathbf{n}_i$ with d_i denoting the oriented distance of $\mathbf{x}_i$ to $\mathbf{y}_i$. Now we locally replace the function d^2 of the CAD model at $\mathbf{x}_i$ by the squared distance to the tangent plane at $\mathbf{y}_i$. Summing up these quadratic approximants results in a quadratic approximant of the function F.

The linearization of the motion is equivalent to using instantaneous kinematics, an idea that already appeared in a similar form in [6]. We estimate the displacements of points via their velocity vectors. The velocity vector field of a rigid body motion is known to have the form

$$\mathbf{v}(\mathbf{x}) = \bar{\mathbf{c}} + \mathbf{c} \times \mathbf{x}. \tag{13}$$

The distance of the point $\mathbf{x}_i + \mathbf{v}(\mathbf{x}_i)$ to the tangent plane at $\mathbf{y}_i$ with unit normal vector $\mathbf{n}_i$ is given by $d_i + \mathbf{n}_i \cdot \mathbf{v}(\mathbf{x}_i)$, where d_i again denotes the oriented distance of $\mathbf{x}_i$ to $\mathbf{y}_i$. The objective function to be minimized is

$$\sum_i \left(d_i + \mathbf{n}_i \cdot (\bar{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i) \right)^2, \tag{14}$$

which is quadratic in the unknowns $(\mathbf{c}, \bar{\mathbf{c}})$. The unique solution can be given explicitly by solving a system of linear equations.

Note that the transformation which maps $\mathbf{x}_i$ to $\mathbf{x}_i + \mathbf{v}(\mathbf{x}_i)$ is an affine map and not a rigid Euclidean motion. Nevertheless, the vector field determined by $(\mathbf{c}, \bar{\mathbf{c}})$ uniquely determines a uniform helical motion M. Axis G and pitch p of M are easily computed (see e.g. [21]). The motion we apply to $\mathbf{x}_i$ is the superposition of a rotation about this axis G through an angle of $\alpha = \arctan(\|\mathbf{c}\|)$ and a translation parallel to G by the distance of $p \cdot \alpha$.

Similar to the ICP algorithm we update the data points via $X_{\mathrm{new}} = M(X_{\mathrm{old}})$ and repeat the procedure until the change in mean-square error $\sum d_i^2$ falls below a preset threshold.

Remark 2. We iteratively minimize a quadratic approximant of f under a linearized motion. The resulting transformation would not satisfy the rigidity constraint on the moved point cloud. Thus, we project back to the constraint

manifold by applying a helical motion indicated by the velocity field. Therefore, our method is a feasible point method. An SQP-algorithm [5] would work with quadratic approximants on the Lagrangian function, a linearization of the motion, but it would then directly apply the linearized transformation; it is not a feasible point method. We expect that SQP-algorithms will help in the solution of other geometric optimization algorithms where the projection onto the constraint manifold is not so simple as for the registration problem.
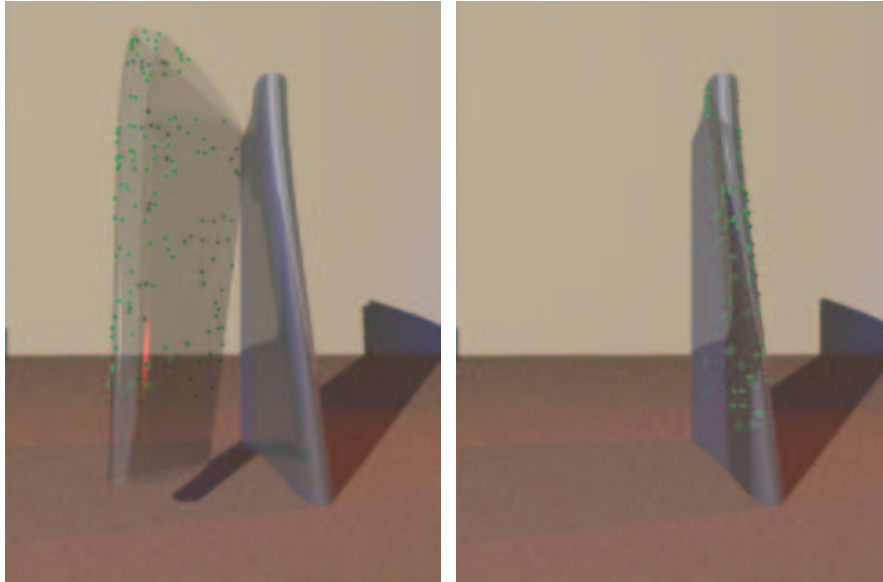
Fig. 12. Matching of a point cloud to the corresponding CAD model: (**left**) initial position of data points and CAD model, (**right**) final position after seven iterations

An example for the registration of a point cloud to a surface is shown in Fig. 12. In order to better visualize the spatial position of the point data, a transparent surface is associated with the data point set. This transparent surface does not enter the computation in any way, it is only displayed for reasons of visualization. The pictures show the data point set in its initial position, after the first iteration step, and in its final position after 7 iterations. In this example the error tolerance reached in the final position will be obtained with the standard implementation of the ICP algorithm only after 45 iterations. This is because in the ICP algorithm the data points move towards their nearest position on the surface in each iteration step. A displacement of the data point set in tangential direction to the model surface therefore needs many iterations.

It is straightforward to extend the objective function (14) to a weighted scheme. There are 3D measurement devices that supply for each data point

a tolerance for the occurring measurement errors. These can be included in the objective function to downweight outliers.

In the description above, we have approximated the squared distance function d^2 to the given model surface by the squared distance function to the tangent plane at the footpoint $\mathbf{y}_i$ of $\mathbf{x}_i$. The latter function is exactly the second order Taylor approximant of d^2 at $\mathbf{y}_i$. In a further improvement we may directly use the second order Taylor approximant of d^2 at $\mathbf{x}_i$ (if necessary, the discussed nonnegative approximant), or another quadratic approximant which is sufficiently accurate at the given position $\mathbf{x}_i$ at an appropriate level of detail. Thus, the results of the investigation on the geometry of the squared distance function can be directly applied to this registration problem and also to other types of registration and positioning problems.

6.2 Application to Surface Approximation

Approximating a given surface (in any representation) or an unstructured cloud of points by a NURBS surface is a widely investigated problem. The main approach uses a least square formulation with a regularization term that expresses the fairness of the final result (see e.g. [11]). Here, the parameterization problem is a fundamental one which largely effects the result (see e.g. [16] and the references therein). Therefore, parameter correction procedures have been suggested [12].

A different approach to the approximation of curves and surfaces are *active contour models*, which are mainly used in Computer Vision and Image Processing. The origin of this technique is the seminal paper by Kass et al. [13], where a variational formulation of parametric curves, coined *snakes*, is presented for detecting contours in images. There are various other applications and a variety of extensions of the snake model (see e.g. [3]).

Instead of a parametric representation of a curve, one may use an implicit form as zero set (level set) of a bivariate function. The formulation of active contour models via level sets goes back to Osher and Sethian [19]. The *level set method* [26] has been successfully applied to the solution of a variety of problems, e.g. for segmentation and analysis of medical images [17]. There are also extensions to surfaces. An application to the surface fitting problem has been given by Zhao et al. [30, 31].

In the registration problems outlined above, the moving object (point cloud) undergoes a rigid body motion. The motion is linearized in each iteration step and guided by the 'flow' imposed by the squared distance function. Basically the same idea applies for surface approximation. Given a very rough initial approximant, the surface is 'deformed' by moving its control points. Since the major parametric surfaces used in CAD systems are B-spline surfaces, which depend linearly on the control points, we may use the same ideas as above.

To explain the principle, we choose an example: The surface Φ in Fig. 13 shall be approximated by a B-spline surface patch. The initial position of
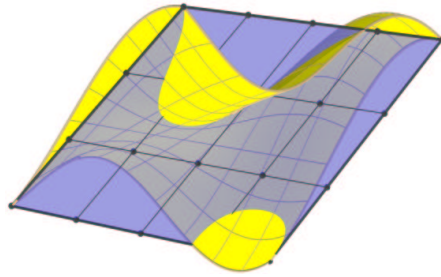
Fig. 13. Approximation of a surface patch (light colored) by a B-spline surface (dark colored): Initial position of B-spline surface
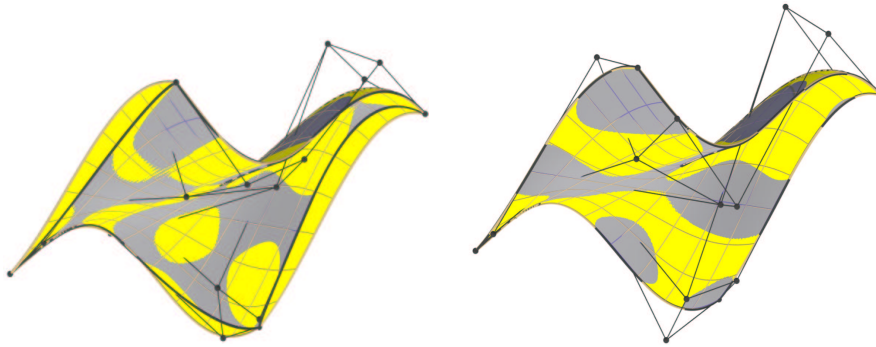

Fig. 14. Approximation of a surface patch by a B-spline surface; final position after five iterations (**left**) without boundary approximation (**right**) with boundary approximation

the B-spline control points was chosen by linear interpolation of Φ's vertex points (Fig. 13). In each iteration step the contol points are recomputed, such that a sufficiently dense sample of points on the B-spline surface is moved towards the target surface Φ. The 'flow' of the sampled points is again guided by the squared distance function. In order to avoid clustering of the control points and self-intersections of the surface (folding), smoothing terms must be incorporated into the function to be minimized. In our example the resulting B-spline patch after the fifth iteration step is given in Fig. 14, *without* boundary approximation (left), and *with* boundary approximation (right), respectively.

An important advantage of this active contour approach to surface approximation is the avoidance of the parameterization problem. Another advantage is the *applicability to subdivision surface fitting*: points at refined levels depend linearly on points of coarse levels in the subdivision procedure. This will be explored in future research.

7 Future Research

There is a large amount of work left for future research. We have to study the squared distance function d^2 to surfaces, which are just given by a dense sample of points. Moreover, computationally efficient ways of working with a piecewise quadratic approximation of d^2 need to be addressed. Furthermore, we will investigate algorithms for the solution of other geometric optimization problems. There, we believe it is important not to use an optimization algorithm as a black box, but adapt an optimization concept (Newton, quasi-Newton, SQP,...) in a geometric way to the special problem.

Acknowledgements

References

1. Ambrosio, L., Mantegazza, C. (1998): Curvature and distance function from a manifold. Journal of Geometric Analysis, **8**, 723–748
2. Besl, P. J., McKay, N. D. (1992): A method for registration of 3D shapes. IEEE Trans. Pattern Anal. and Machine Intell., **14**, 239–256
3. Blake, A., Isard, M. (1998): Active Contours. Springer New York
4. Blum, H. (1973): Biological shape and visual science, Journal of Theoretical Biology, **38**, 205–287
5. Boggs, P. T., Tolle, J. W. (1995): Sequential quadratic programming. Acta Numerica, **4**, 1–52
6. Bourdet, P., Clement, A. (1988): A study of optimal-criteria identification based on the small-displacement screw model. Annals of the CIRP, **37**, 503–506
7. Chen, Y., Medioni, G. (1992): Object modelling by registration of multiple range images. Image and Vision Computing, **10**, 145–155
8. Choi, H. I., Choi, S. W., Moon, H. P. (1997): New algorithm for medial axis transform of plane domain. Graphical Models and Image Processing, **59**, 463–483
9. Faugeras, O. (1993): Three-Dimensional Computer Vision: a Geometric Viewpoint. The MIT Press
10. Faugeras, O., Gomes, J. (2000): Dynamic shapes of arbitrary dimension: the vector distance functions. In: Martin, R. (ed.) The Mathematics of Surfaces IX. Springer, 227–262
11. Hoschek, J., Jüttler, B. (1999): Techniques for fair and shape-preserving surface fitting with tensor-product B-splines. In: Peña, J. M. (ed.) Shape Preserving Representations in Computer-Aided Geometric Design. Nova Science Publ., Commak, New York

12. Hoschek, J., Lasser, D. (1993): Fundamentals of Computer Aided Geometric Design. A. K. Peters, Wellesley, Massachusetts
13. Kass, M., Witkin, A., Terzopoulos, D. (1988): Snakes: Active contour models. International Journal of Computer Vision, **1**, 321–332
14. Kimia, B., Tannenbaum, A, Zucker, S.W. (1995): Shapes, shocks and deformations I: The components of two-dimensional shape and the reaction-diffusion space. International Journal of Computer Vision, **15**, 189–224
15. Kimmel, R., Shaked, D., Kiryati, N., Bruckstein, A. (1995): Skeletonization via distance maps and level sets. Computer Vision and Image Understanding, **62**, 382–391
16. Ma, W., Kruth, J. P. (1995): Parametrization of randomly measured points for the least squares fitting of B-spline curves and surfaces. Computer Aided Design, **27**, 663–675
17. Malladi, R., Sethian, J. A., Vemuri, B. C. (1995): Shape modeling with front propagation: A level set approach. IEEE Transactions Pattern Analysis and Machine Intelligence, **17**, 158–175
18. Nikolaidis, N., Pitas, I. (2001): 3-D Image Processing Algorithms, Wiley
19. Osher, S. J., Sethian, J. A. (1988): Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation. Journal of Computational Physics, **79**, 12–49
20. Patrikalakis, N. M., Maekawa, T. (2002): Shape Interrogation for Computer Aided Design and Manufacturing. Springer
21. Pottmann, H., Wallner, J. (2001): Computational Line Geometry. Springer Berlin Heidelberg New York
22. Rusinkiewicz, S., Levoy, M. (2001): Efficient variants of the ICP algorithm, in Proc. 3rd Int. Conf. on 3D Digital Imaging and Modeling, Quebec, Springer
23. Sapiro, G. (2001): Geometric Partial Differential Equations and Image Analysis. Cambridge University Press
24. Serra, J. (1982): Image Analysis and Mathematical Morphology. Academic Press, London
25. Serra, J., Soille, J., eds. (1994): Mathematical Morphology and Its Applications to Image Processing. Kluwer Academic Publishers, Dordrecht
26. Sethian, J. A. (1999): Level Set Methods and Fast Marching Methods. Cambridge University Press
27. Siddiqi, K., Tannenbaum, A., Zucker, S.W. (1999): A Hamiltonian approach to the eikonal equation. Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, 1–13
28. Toriwaki, J., Yokoi, S. (1981): Distance transformations and skeletons of digitized pictures with applications. In: Kanal, L. N. and Rosenfeld, A. (eds.) Progress in Pattern Recognition. North Holland, 187–264
29. Zhao, H. K. (2002): Analysis and visualization of large sets of unorganized points using the distance function. Preprint
30. Zhao, H. K., Osher, S., Fedkiw, R. (2001): Fast surface reconstruction and deformation using the level set method. Proceedings IEEE Workshop on Variational and Level Set Methods in Computer Vision, Vancouver
31. Zhao, H. K., Osher, S., Merriman, B., Kang, M. (2000): Implicit and non-parametric shape reconstruction from unorganized data using variational level set method. Computer Vision and Image Understanding, **80**, 295–314

Index