AN IMPROVED ALGORITHM FOR
ALGEBRAIC CURVE AND SURFACE FITTING

Gabriel Taubin
IBM T.J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Abstract
Recent years have seen an increasing interest in algebraic curves and surfaces of high degree as geometric models or shape descriptors for model-based computer vision tasks such as object recognition and position estimation. Although their invariant-theoretic properties make them a natural choice for these tasks, fitting algebraic curves and surfaces to data sets is difficult, and fitting algorithms often suffer from instability, and numerical problems. One source of problems seems to be the performance function being minimized. Since minimizing the sum of the squares of the Euclidean distances from the data points to the curve or surface with respect to the coefficients of the defining polynomials is computationally impractical, because measuring the Euclidean distances requires iterative processes, approximations are used. In the past we have used a simple first order approximation of the Euclidean distance from a point to an implicit curve or surface which yielded good results in the case of unconstrained algebraic curves or surfaces, and reasonable results in the case of bounded algebraic curves and surfaces. However, experiments with the exact Euclidean distance have shown the limitations of this simple approximation. In this paper we introduce a more complex, and better, approximation to the Euclidean distance from a point to an algebraic curve or surface. Evaluating this new approximate distance does not require iterative procedures either, and the fitting algorithm based on it produces results of the same quality as those based on the exact Euclidean distance.

Key words: Algebraic surfaces. Fitting. Surface reconstruction. Distance to algebraic surfaces.

1 Introduction
In the last few years several researchers have started using algebraic curves and surfaces of high degree as geometric models or shape descriptors in different model-based computer vision tasks. Typically, the input for these tasks is either an intensity image or dense range data [19, 22, 27, 28].

One of the fundamental problems in building a recognition and positioning system based on implicit curves and surfaces is how to fit these curves and surfaces to data. This process will be necessary for automatically constructing object models from range or intensity data and for building intermediate representations from observations during recognition. Several methods are available for extracting straight line segments [13], planar patches [14], quadratic arcs [1, 2, 5, 9, 10, 15, 17, 21, 20, 25, 32], and quadric surface patches [3, 4, 7, 14, 16, 18] from 2D edge maps and 3D range images. Recently, methods have also been developed for fitting algebraic curve and surface patches of arbitrary degree [8, 19, 24, 22, 26, 27, 28, 30].

Although the properties of implicit algebraic curves and surfaces make them a natural choice for object recognition and positioning applications, least squares algebraic curve and surface fitting algorithms often suffer from instability problems. One of the sources of problems seems to be the performance function that is being minimized [23]. Since minimizing the sum of the squares of the Euclidean distances from the data points to the implicit curve or surface with respect to the continuous parameters of the implicit functions is computationally impractical, because measuring the Euclidean distances already require iterative processes, approximations are used. In the past we have used a simple approximation of the Euclidean distance from a point to an implicit curve or surface which yielded good results in the case of unconstrained algebraic curves or surfaces of a given degree, and reasonable results in the case of bounded algebraic curves and surfaces. However, experiments with the exact Euclidean distance have shown that the simple approximation has limitations, in particular for this.
family [23]. Taubin [29], defining simple algorithms for rendering planar algebraic curves on raster displays, introduced a more complex, but more accurate, two dimensional higher order approximate, which can still be evaluated directly from the coefficients of the defining polynomials and the coordinates of the point. In this paper we extend the definitions to dimension three, within the context of implicit curve and surface fitting. We show that this new approximate distance produces results of the same quality as those based on the exact Euclidean distance, and much better than those obtained using other available methods.

2 Algebraic curve and surfaces

In this section we review some previous results on fitting algebraic curves and surfaces to measured data points. An implicit surface is the set of zeros of a smooth function \( f : \mathbb{R}^3 \to \mathbb{R} \) of three variables:

\[
Z(f) = \{ (x_1, x_2, x_3) : f(x_1, x_2, x_3) = 0 \}.
\]

Similarly, an implicit 2D curve is the set \( Z(f) = \{ (x_1, x_2) : f(x_1, x_2) = 0 \} \) of zeros of a smooth function \( f : \mathbb{R}^2 \to \mathbb{R} \) of two variables. The curves or surfaces are algebraic if the functions are polynomials. Although the methods to be discussed will be valid for dimension 2,3, and above, we will concentrate on the case of surfaces, i.e., dimension 3. In what follows, a polynomial of degree \( d \) in three variables \( x_1, x_2, x_3 \) will be denoted as follows

\[
f(x) = \sum_{0 \leq |\alpha| \leq d} F_\alpha x^\alpha,
\]

where \( x = (x_1, x_2, x_3)^T \), \( \alpha = (\alpha_1, \alpha_2, \alpha_3) \) is a multiindex, a term of nonnegative integers, \( |\alpha| = \alpha_1 + \alpha_2 + \alpha_3 \) is the size of the multiindex, \( x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} x_3^{\alpha_3} \) is a monomial, and \( F = \{ F_\alpha : 0 \leq |\alpha| \leq d \} \) is the set of coefficients of \( f \).

3 Fitting

Given a finite set of three dimensional data points \( D = \{ p_1, \ldots, p_q \} \), the problem of fitting an algebraic surface \( Z(f) \) to the data set \( D \) is usually cast as minimizing the mean square distance

\[
\Delta(F) = \frac{1}{q} \sum_{i=1}^q \text{dist}(p_i, Z(f))^2
\]

from the data points to the curve or surface \( Z(f) \), a function of the set of coefficients \( F \) of the polynomial.

Unfortunately, the minimization of (2) is computationally impractical, because there is no closed form expression for the distance from a point to an algebraic surface, and iterative methods are required to compute it.

Thus, we seek approximations to the distance function. Often, the distance from \( p \) to \( Z(f) \) is taken to simply be the result of evaluating the polynomial \( f \) at \( p \), since without noise \( f(p) \) will vanish for all \( p \) on \( Z(f) \)

\[
\Delta_1(F) = \frac{1}{q} \sum_{i=1}^q f(p_i)^2.
\]

While computationally attractive, fitting based on this distance is biased [5].

An alternative to approximately solve the original computational problem, i.e., the minimization of (2), is to replace the real distance from a point to an implicit curve or surface by the first order approximation [26, 27]

\[
\text{dist}(x, Z(f))^2 \approx \frac{f(x)^2}{||\nabla f(x)||^2}.
\]

The mean value of this function on a fixed set of data points

\[
\Delta_1(F) = \frac{1}{q} \sum_{i=1}^q \frac{f(p_i)^2}{||\nabla f(p_i)||^2}.
\]

is a smooth nonlinear function of the parameters, and can be locally minimized using well established nonlinear least squares techniques. However, one is interested in the global minimization of (4), and wants to avoid a global search. A method to choose a good initial estimate was introduced in [26, 27] as well. By replacing the performance function again the difficult multimodal optimization problem turns into a generalized eigenproblem. There exist certain families of implicit curves or surfaces, such as those defining straight lines, circles, planes, spheres and cylinders, which have the value of \( ||\nabla f(x)||^2 \) constant on \( Z(f) \). In those cases we have

\[
\forall i: ||\nabla f(p_i)||^2 \approx \frac{1}{q} \sum_{j=1}^q ||\nabla f(p_j)||^2
\]

\[
\frac{1}{q} \sum_{i=1}^q f(p_i)^2 \approx \frac{1}{q} \sum_{i=1}^q ||\nabla f(p_i)||^2.
\]

In the linear model, the right hand side of the previous expression reduces to the quotient of two quadratic functions of the vector of parameters \( F \).

\[
\frac{1}{q} \sum_{i=1}^q f(p_i)^2 = F M F^T \quad \frac{1}{q} \sum_{i=1}^q ||\nabla f(p_i)||^2 = F N F^T,
\]

659
where the matrices $M$ and $N$, are nonnegative definite, symmetric, and only functions of the data points:

$$
M = \frac{1}{q} \sum_{i=1}^{q} [X(p_i)X(p_i)^T]
$$

$$
N = \frac{1}{q} \sum_{i=1}^{q} [DX(p_i)DX(p_i)^T]
$$

where $DX: \mathbb{R}^n \rightarrow \mathbb{R}^{k \times n}$ is the Jacobian matrix of $X$. In the algebraic case, the entries of $M$ and $N$ are linear combinations of moments of the data points. The new problem, the minimization of (5), reduces to a generalized eigenvalue problem, with the minimizer being the eigenvector corresponding to the minimum eigenvalue of the pencil $F(M - \lambda N) = 0$. Note that the complexity of this generalized eigenvalue fit method is polynomial in the number of variables, while the local minimization of (4) is a function of the number of data points, which is usually much larger than the number of variables. This method directly applies to fitting unconstrained algebraic curves and surfaces, and there is an extension to the case of bounded surfaces as well [30].

It is important to note that the approximate distance (3) is also biased in some sense. If one of the data points $p_i$ is close to a critical point of the polynomial $f$, i.e., $\|\nabla f(p_i)\| \approx 0$, but $f(p_i) \neq 0$, the ratio $f(p_i)^2/\|\nabla f(p_i)\|^2$ becomes large. This implies that the minimizer of (4) must be a polynomial with no critical points close to the data set. This is certainly a limitation. Besides, recent experiments with the exact Euclidean distance [23] have shown much better results, at the expense of a very long computation. We will now introduce a better approximation.

4 A better approximation to the Euclidean distance

The next problem that we have to deal with is how to correctly approximate the distance $\text{dist} (p, Z(f))$ from a point $p \in \mathbb{R}^n$ to the set of zeros of a polynomial $f$. In general, the Euclidean distance from $p$ to the set of zeros $Z(f)$ of a function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is defined as the minimum of the distances from $p$ to points in the zero set

$$
\text{dist} (p, Z(f)) = \min \{ \|x - p\| : f(x) = 0 \}. \quad (6)
$$

In this section we show that in the general case we will need to explicitly compute the distance using numerical methods, but in the algebraic case (when $f(x)$ is a polynomial of low degree) different combinations of symbolic and numerical methods can be used to solve the problem. Then, and since even in the algebraic case computing the Euclidean distance is computationally expensive, and sometimes numerically unstable, in following sections we introduce approximations which can be less expensively evaluated.

5 Simple approximate distance

The simple approximate distance of equation (3) is a first order approximation of the Euclidean distance from a point to a zero set of a smooth function. It is a generalization of the update step of the Newton method for root-finding [11]. In this section we review the derivation of this simple distance, which from now on will be denoted $\delta_1(p, f)$, because we will need it to derive the higher order approximate distances.

Let $p \in \mathbb{R}^n$ be a point such that $\|\nabla f(p)\| \neq 0$, and let us expand $f(x)$ in Taylor series up to first order in a neighborhood of $p$

$$
f(x) = f(p) + \nabla f(p)^T(x - p) + O(||x - p||^2). \quad (7)
$$

Now, truncate the series after the linear terms, apply the triangular, and then the Cauchy-Schwartz inequality, to obtain

$$
\|f(x)\| \approx \|f(p) + \nabla f(p)^T(x - p)\| \quad (8)
$$

$$
\geq \|f(p)\| - \|\nabla f(p)^T(x - p)\| \quad (9)
$$

$$
\geq \|f(p)\| - \|\nabla f(p)\||x - p||. \quad (10)
$$

The simple approximate distance from $p$ to $Z(f)$ is defined as the value of $\|x - p\|$ that annihilates the expression on the right side

$$
\delta_1(p, f) = \frac{|f(p)|}{\|\nabla f(p)\|}. \quad (11)
$$

Besides the fact that it can be computed very fast in constant time, requiring only the evaluation of the function and its first order partial derivatives at the point, the fundamental property of the simple approximate distance is that it is a first order approximation to the exact distance in a neighborhood of every regular point (a point $p \in \mathbb{R}^2$ such that $f(p) = 0$ but $\|\nabla f(p)\| > 0$). In our context, this is a very desirable property.

Lemma 1 Let $f: \mathbb{R}^n \rightarrow \mathbb{R}$ be a function with continuous derivatives up to second order, in a neighborhood of a regular point $p_0$ of $Z(f)$. Let $w$ be a unit length normal vector to $Z(f)$ at $p_0$, and let $p_t = p_0 + tw$, for $t \in \mathbb{R}$. Then

$$
\lim_{t \to 0} \frac{\delta_1(p_t, f)}{\delta_1(p_0, f)} = 1, \quad (12)
$$

where $\delta(p, f)$ denotes the Euclidean distance from the point $p$ to the set $Z(f)$. 

660
As it can be seen already in the one-dimensional case in figure 1, the approximate distance either underestimates or overestimates the exact distance. Underestimation is not a problem for the algorithms described above, but overestimation is a real problem. It is not difficult to find examples where the simple approximate distance is not a problem for the algorithms described in the next section will solve the problem.

6 Higher order approximate distance

In this section we derive a new approximate distance. The simple approximate distance involves the value of the function and the first order partial derivatives at the point. This will be coincide with the new approximate distance for polynomials of degree one. In general, we will define the approximate distance of order \(d\) as a function \(\delta_d(p, f)\) of the coordinates of the point, and all the partial derivatives of the polynomial \(f\) of degree \(d\) at the point \(p\), satisfying the following inequality

\[
0 \leq \delta_d(p, f) \leq \delta(p, f)
\]

when \(f\) is a polynomial of degree \(\leq d\). That is, we will show that \(\delta_d(p, f)\) is a lower bound for the Euclidean distance from the point to the algebraic curve \(Z(f)\). Since we will also show that all the approximate distances are asymptotically equivalent to the Euclidean distance near regular points, we will be able to replace the Euclidean distance with the approximate distance of order \(d\) in our performance function.

Our approach is to construct a univariate polynomial \(F_p(\delta)\) of the same degree \(d\), such that \(F_p(0) = |f(p)|\), and \(|f(x)| \geq F_p(||x - p||)\) for all \(x\). The approximate distance \(\delta_d(p, f)\) will be defined as the smallest nonnegative root of \(F_p(\delta)\). This is exactly what we did to define the first order approximate distance (7). We first rewrite the polynomial (1) in Taylor series around \(p\)

\[
f(x) = f'(x-p) = \sum_{a,d \leq d} F_a(p)(x-p)^a,
\]

where

\[
F_a(p) = \frac{1}{a!} \left. \frac{\partial^{\alpha_1+\cdots+\alpha_n} f}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}} \right|_{x=p},
\]

where \(\alpha! = \alpha_1! \alpha_2! \cdots \alpha_n!\). Horner’s algorithm can be used to compute these coefficients stably and efficiently [6] from the coefficients of the polynomial expanded at the origin \(F = \{F_a : 0 \leq |\alpha| \leq d\}\), and even parallel algorithms exist to do so [12]. We now define the coefficients \(F_0(p), F_1(p), \ldots, F_d(p)\) of the univariate polynomial \(F(\delta)\). We define the first coefficient as \(F_0(p) = |f(p)|\), and for \(h = 1, 2, \ldots, d\) we set

\[
F_h(p) = -\left\{ \sum_{|\alpha| = h} \binom{h}{\alpha}^{-1} F_a(p)^2 \right\}^{1/2}.
\]

We can now state the first fundamental property of the bounding polynomial

\[
F_p(\delta) = \sum_{h=0}^d F_h(p) \delta^h.
\]

Lemma 2 If \(f(x)\) is a polynomial of degree \(d\), and \(F_p(\delta)\) is the univariate polynomial defined above, then

\[
|f(x)| \geq F_p(||x - p||)
\]

for every \(x \in \mathbb{R}^d\).

Proof: In the appendix.

Note that, since it attains the nonnegative value \(|F_0(p)| = |f(p)|\) at \(\delta = 0\), and it is monotonically decreasing for \(\delta > 0\), the polynomial \(F_p(\delta)\) has a unique nonnegative root \(\delta_d(p, f)\). This number, \(\delta_d(p, f)\), is a lower bound for the Euclidean distance from \(p\) to \(Z(f)\), because for \(|x - p| < \delta_d(p, f)\), we have

\[
|f(x)| \geq F_p(||x - p||) > 0.
\]

Also, note that, if \(F_p(\delta) > 0\) then the distance from \(p\) to \(Z(T^df_p)\) is larger than \(\delta\), and in the case of polynomials of degree \(\leq d\), this means that the set \(Z(f)\) does not cut the circle of radius \(\delta\) centered at \(p\).

To compute the value of \(\delta_d(p, f)\), due to the monotonicity of the polynomial \(F_p(\delta)\) and the uniqueness of the positive root, any univariate root finding algorithm will converge very quickly to \(\delta_d(p, f)\). Even Newton’s
algorithm converges in a couple of iterations. But in order to make such an algorithm work, we need a practical method to compute an initial point, or to bracket the root. Since we already have a lower bound \((\delta = 0)\), we only need to show an upper bound for \(\delta_d(p, f)\). For this, note that

\[
F_p(\delta) = \sum_{h=0}^{d} F_h(\delta) \leq F_0(p) + F_1(p) \delta
\]

for every \(\delta \geq 0\), and so \(\delta_d(p, f) \leq \delta_1(p, f)\), the simple approximate distance of section 5 is an upper bound. But, as we have observed above, this upper bound could be infty. Since the degree of \(f\) is \(d\), at least one of the coefficients of degree \(d\) must be nonzero, and so, \(F_d(p) \neq 0\). More generally, if \(F_h(p) \neq 0\) the following inequality

\[
F_p(\delta) = \sum_{h=0}^{d} F_h(\delta) \leq F_0(p) + F_h(p) \delta^h
\]

can be used to obtain the following bound

\[
0 \leq \delta_d(p, f) \leq \left| \frac{F_0(p)}{F_h(p)} \right|^{1/h}.
\]

Finally, the asymptotic behavior of the new approximate distance near regular points is determined by the behavior of the simple approximate distance, i.e., the first order approximate distance.

**Lemma 3** Let \(f : \mathbb{R}^2 \to \mathbb{R}\) be a function with continuous partial derivatives up to order \(d+1\), in a neighborhood of a regular point \(p_0\) of \(Z(f)\). Let \(w\) be a unit length normal vector to \(Z(f)\) at \(p_0\), and let \(p_t = p_0 + tw\), for \(t \in \mathbb{R}\). Then

\[
\lim_{t \to 0} \frac{\delta_d(p_t, f)}{\delta(p_t, f)} = 1,
\]

**Proof:** In the appendix. \(\Box\)

### 7 Implementation details

For the minimization of the approximate mean square distance of order \(d\)

\[
\Delta_d(F) = \frac{1}{q} \sum_{i=1}^{q} \delta_d(p_i, f)^2
\]

using the Levenberg-Marquardt algorithm, we need to provide a routine for the evaluation of \(\delta_d(p, f)\) and all its partial derivatives with respect to the parameters \(F = \{ F_0, \cdots, F_d \} \). Now we show how to compute these partial derivatives. By differentiating the equation

\[
F_p(\delta_d(p, f)) = 0
\]

we obtain

\[
0 = \frac{\partial}{\partial F_p} \left\{ \sum_{h=0}^{d} F_h(p) \delta_d(p, f)^h \right\} = \left\{ \sum_{h=0}^{d} \frac{\partial \{ F_h(p) \}}{\partial F_p} \delta_d(p, f)^h \right\} + \left\{ \sum_{h=1}^{d} h F_h(p) \delta_d(p, f)^{h-1} \right\} \frac{\partial \{ \delta_d(p, f) \}}{\partial F_p},
\]

where \(\nu = (\nu_1, \nu_2, \nu_3)\) is another multiindex, from where we get

\[
\frac{\partial \{ F_h(p) \}}{\partial F_p} = \left\{ \sum_{h=1}^{d} \frac{\partial \{ F_h(p) \}}{\partial F_p} \delta_d(p, f)^{h-1} \right\}.
\]

Note that the denominator of the expression on the right side is the derivative \(F_p(\delta)\) of the univariate polynomial \(F_p(\delta)\) with respect to \(\delta\), evaluated at \(\delta = \delta_d(p, f)\). To compute these derivatives we also need to compute the partial derivatives of \(F_0(p), \ldots, F_d(p)\) with respect to the coefficients of \(f\). For \(h = 0, 1, \ldots, d\), we have

\[
\frac{\partial \{ F_h(p) \}}{\partial F_p} = \frac{1}{F_h(p)} \sum_{l=1}^{d} \binom{h}{l}^{-1} F_h(p) \frac{\partial \{ F_l(p) \}}{\partial F_p},
\]

where

\[
\frac{\partial \{ F_l(p) \}}{\partial F_p} = \begin{cases} \left( \nu \right)^\beta & \text{if } \exists \beta : \nu = \alpha + \beta \\ 0 & \text{otherwise} \end{cases}.
\]

In the last equation addition of multiindices is defined coordinatewise \(\nu_i = \alpha_i + \beta_i\) for \(i = 1, 2, 3\), and \(\beta\) is also a vector of nonnegative integers.

### 8 Experimental results

We have implemented the methods described above for fitting unconstrained polynomials of an arbitrary degree to range data. The Levenberg-Marquardt algorithm requires an initial estimate, and we have found the results quite dependent on this initial choice. In the past we have use a method based on generalized eigendecomposition [26, 27] to get initial estimates with good results. In this case we have observed better results starting the algorithm from the coefficients of a sphere, a surface defined by a polynomial

\[
(x_1^2 + x_2^2 + x_3^2) + F_{000} x_1 + F_{010} x_2 + F_{001} x_3 + F_{000},
\]

with the coefficients \(F_{000}, F_{010}, F_{001}, F_{000}\) computed using the generalized eigenvalue fit method. For
higher degree surfaces, we also tried surfaces defined by polynomials

$$(x_1^2 + x_2^2 + x_3^2)^{d/2} + \sum_{0 \leq |\alpha| \leq d-1} F_\alpha x^\alpha,$$

with the coefficients $F_\alpha$ initialized with the generalized eigenvalue fit method as well. We call these surfaces hyperspheres. In both cases the initial surface is bounded, and although the results are not constrained to be bounded surfaces, they turned out to be bounded in all the examples that we have tried. Starting from the results of the unconstrained generalized eigenvalue fit method usually produced unbounded surfaces, and of much worse quality than those initialized from the bounded surfaces defined above.

The main problem is whether the data set can be well approximated with an algebraic surface of the given degree or not. This is a problem that cannot be solved beforehand. In those cases with a positive answer, the fitted surfaces turn out to be good approximations of the original surfaces. In the cases with a negative answer, the quality of fitted surface varies, some results are good and some are bad.

Figures 2, 3, and 4 show some examples with synthetic data, while figure 5 shows a couple of examples with range and CT data.

![Figure 2: Bean. (a): Large data set. (b): Hypersphere fit to (a). (c) General fit to (a). (d): Small data set. (e): Hypersphere fit to (d). (f) General fit to (d).](image2)

In the synthetic examples the data was generated from algebraic surfaces, but the data points are not on the surfaces, but close by. In fact, the data points are some vertices of a regular rectangular mesh constructed by recursive subdivision of a cube containing the original surface. The nonlinear minimization process was initialized with the hypersurface fit described above. The result of this initial fitting step is shown in the second column of the pictures, along with the data points. The third column shows the results of the nonlinear minimization, the fit based on the approximate distance introduced in this paper, along with the data set.

![Figure 3: Cup. (a): Large data set. (b): Hypersphere fit to (a). (c) General fit to (a). (d): Small data set. (e): Hypersphere fit to (d). (f) General fit to (d).](image3)

In all the cases the resulting surfaces are not constrained to be bounded, but in general they turned out to be bounded. An exception can be observed in figure 4, where the fit to the low resolution data set is unbounded. It is important to note that this new approximate distance can be used to fit bounded surfaces as well, using the parameterized families introduced in [30].

Finally, figure 5 show some examples of fitting sur-
faces to real data. The first example is range data from a pepper. The second example is CT data. It is important to note that in the case of CT data, each result is so badly conditioned (very close to an ellipsoid) for a fourth degree surface fit, that almost all methods either fail to converge, or produce bad fits.

9 Conclusions

We have described a new method to improve the algebraic surface fitting process by better approximating the Euclidean distance from a point to the surface. This method is more stable than previous algorithms, and produces much better results.

A Proofs

Proof [Lemma 1]: In the first place, under the current hypotheses, there exists a neighborhood of \( p_0 \) within which the Euclidean distance from the point \( p_t \) to the curve \( Z(f) \) is exactly equal to \(| \nu |\)

\[
\delta(p_t, f) = |p_t - p_0| = |\nu |
\]

(see for example [31, Chapter 16]). Also, since \( p_0 \) is a regular point of \( Z(f) \), we have \( |\nabla f(p_0)| > 0 \), and by continuity of the partial derivatives, \( 1/|\nabla f(p_t)| \) is bounded for small \(| \nu |\). Thus, we can divide by \( |\nabla f(p_t)| \) without remorse. And since \( w \) is a unit length normal vector, \( \nabla f(p_0) = \pm |\nabla f(p_0)| w \). Finally, by continuity of the second order partial derivatives, and by the previous facts, we obtain

\[
\nabla f(p_t) = \nabla f(p_0) + O(|p_t - p_0|)
\]

and

\[
0 = f(p_0)
\]

\[
= f(p_t) + \nabla f(p_t)^T (p_0 - p_t) + O(|p_0 - p_t|^2)
\]

\[
= f(p_t) + |\nabla f(p_t)| \nu + O(|\nu|^2).
\]

Moving \( f(p_t) \) to the other member, dividing by \( |\nabla f(p_t)||\nu| \), and taking absolute value, we obtain

\[
\frac{\delta(p_t, f)}{\delta(p_t, f)} = \frac{|f(p_t)|}{|\nabla f(p_t)||\nu|} = 1 + O(|\nu|),
\]

which finishes the proof. □

Proof [Lemma 2]: We first rewrite the polynomial (8) as a sum of forms

\[
f(x) = \sum_{h=0}^{d} \left\{ \sum_{|\alpha|=h} F_\alpha(p) (x - p)^\alpha \right\},
\]

apply the triangular inequality

\[
|f(x)| \geq |f(p)| - \sum_{h=1}^{d} \left| \sum_{|\alpha|=h} F_\alpha(p) (x - p)^\alpha \right|
\]

and the Cauchy-Schwartz inequality to the terms in the sum

\[
\left| \sum_{|\alpha|=h} F_\alpha(p) (x - p)^\alpha \right|
\]

\[
\leq \left( \sum_{|\alpha|=h} \left| \frac{h!}{\alpha_1!\alpha_2!\alpha_3!} \right| \right)^{1/2} \left( \sum_{|\alpha|=h} \left| F_\alpha(p)^2 \right| \right)^{1/2}
\]

where, for \(|\alpha| = h\) we denote

\[
\left( \frac{h}{\alpha_1!\alpha_2!\alpha_3!} \right)^{1/2} = \frac{h!}{\alpha_1!\alpha_2!\alpha_3!}.
\]

Now, on the right hand side of the previous equation we identify the definition of \( -F_h(p) \), followed by a function of \( x - p \), but from the multinomial formula we get

\[
\left( \sum_{|\alpha|=h} \left( \frac{h}{\alpha_1!\alpha_2!\alpha_3!} \right)^{1/2} \right)^{1/2} = |x - p|^h.
\]

It follows that

\[
|f(x)| \geq F_0(p) + \sum_{h=1}^{d} F_h(p) |x - p|^h = F_p(||x - p||).
\]

664
Proof [Lemma 3]: Since the case $d = 1$ was the subject of Lemma 1, let us assume that $d > 1$. Since for small $t$ $F_0(p_t) = |f(p_t)| \neq 0$ ($t \neq 0$) and $F_1(p_t) = -\|\nabla f(p_t)\| \neq 0$, we can divide by $F_0(p_t)$ and by $F_1(p_t)$. Remembering that the simple approximate distance is $\delta_1(p_t, f) = -F_0(p_t)/F_1(p_t)$, we can rewrite the previous equation as follows:

$$1 - \frac{\delta_d(p_t, f)}{\delta_1(p_t, f)} = \frac{\delta_d(p_t, f)}{\delta_1(p_t, f)} \left\{ \sum_{k=2}^d \frac{F_k(p_t)}{F_1(p_t)} \frac{\delta_d(p_t, f)^{k-2}}{\delta_1(p_t, f)} \right\} \delta_1(p_t, f).$$

Now we observe that the three factors on the right side are nonnegative, the first factor is bounded by 1 because $0 < \delta_d(p_t, f) \leq \delta_1(p_t, f)$, the second factor is bounded by

$$\sum_{k=2}^d \frac{F_k(p_t)}{F_1(p_t)} \frac{\delta_d(p_t, f)^{k-2}}{\delta_1(p_t, f)}$$

for $\delta_1(p_t, f) < 1$, which is continuous at $t = 0$, and so bounded for $t \to 0$, and the last factor is bounded by $\delta_1(p_t, f)$. That is, we have shown that

$$\frac{\delta_d(p_t, f)}{\delta_1(p_t, f)} = 1 + O(\delta_1(p_t, f))$$

which based on Lemma 1 finishes the proof. □

References


