

Geometric Distance Fields of Plane Curves*

Róbert Bán^a and Gábor Valasek^a

Abstract

This paper introduces a geometric generalization of signed distance fields for plane curves. We propose to store simplified geometric proxies to the curve at every sample. These proxies are constructed based on the differential geometric quantities of the represented curve and are used for queries such as closest point and distance calculations. We derive the theoretical approximation order of these constructs and provide empirical comparisons between geometric and algebraic distance fields of higher order. We validate our theoretical results by applying them to font representation and rendering.

Keywords: computer graphics, signed distance fields, plane curves

1 Introduction

Signed distance functions (SDF) are special implicit representations of shapes. They map a real number to every point in space and this scalar encodes two attributes of the query position: (i) its distance to the boundary of the geometry represented by the SDF and (ii) whether the query point is inside, outside, or on the boundary of the geometry. The former is the magnitude of the scalar mapped to the point and the latter is determined by its sign.

The construction and evaluation of the exact SDF of a complex scene is computationally expensive. As such, most applications settle on using discrete samples and various reconstruction filtering techniques to infer an approximate signed distance value for every query point in space. We refer to these as discrete signed distance fields (DSDF) and our present work is a generalization of this approach.

In a recent work [3], we considered the algebraic generalization of a signed distance sample. We proposed the use of degree one Taylor approximations to the signed distance function and showed that this allows considerable reductions in storage. That is, even though the size of a single sample increased, the approximation properties of the field itself have improved enough so that in total less scalars were needed to retain a prescribed accuracy.

*EFOP-3.6.3-VEKOP-16-2017-00001: Talent Management in Autonomous Vehicle Control Technologies – The Project is supported by the Hungarian Government and co-financed by the European Social Fund.

^aEötvös Loránd University, Budapest, Hungary, E-mail: {rob.ban, valasek}@inf.elte.hu, ORCID: 0000-0002-8266-7444, 0000-0002-0007-8647

The generalization of this approach, i.e. increasing the degree of the Taylor approximation is hindered by the coefficient explosion of Taylor polynomials. Since a degree n polynomial in \mathbb{R}^d is represented by $\binom{n+d}{n}$ coefficients, a naive representation of a degree 1 and 2 Taylor polynomial in the plane requires 3 and 6 scalars respectively. Unfortunately, exceeding the per sample storage capabilities of GPU texture formats limits the immediate applicability of texture filtering based approaches, so even degree 2 polynomials need additional techniques to retain their practical value in real-time use cases.

In this paper, we propose an alternative higher order sample construction for planar DSDFs. This technique uses per sample geometric proxies of the boundary curves. These proxies are based on the differential geometric properties of the closest boundary point and they are a generalization of the approach presented in [16].

The intuition comes from recognizing that in the plane, a degree 1 Taylor polynomial is a line in \mathbb{E}^2 that also coincides with the tangent line at the closest boundary point to the sample position.

As such, a second order geometric approximation to the boundary is an osculating circle. Clearly, this does not coincide with a second order algebraic sample, whose zero level set determines a conic section in the plane. Moreover, a circle can be represented by its center and radius, i.e. 3 scalars, whereas a degree two polynomial in two variables is determined by 6 scalar coefficients.

Our main theoretical contribution is that this storage reduction does not cost us approximation power: the signed distance function of the osculating circle is a similarly second order approximation to the signed distance function of the original geometry as a second degree Taylor polynomial. This is proven in Section 6.

More generally, we show that entities possessing an order n geometric contact have equal SDF derivatives up to order n .

We validate our theoretical results by applying this representation to the storage and rendering of vector fonts in Section 9.

2 Prior work

Discrete signed distance field constructions have found uses in many applications, including font rendering [7], collision detection [6], and various other areas [11, 17, 5, 1].

Our focus is in the planar use cases of DSDFs, and more specifically, font and vector art rendering. DSDF based techniques have received much attention in this venue and they have been improved both in terms of performance and quality.

A notable work is that of Loop and Blinn [9], that takes GPU architecture specific considerations into account and contributes in both areas of improvements. They propose a general framework for the rendering of vector art composed of up to cubic Bézier curves. For anti-aliasing, they approximate the signed distance to the boundary using a first order expansion and show how these data can be stored at the primitive vertices such that hardware bilinear interpolation yields correct

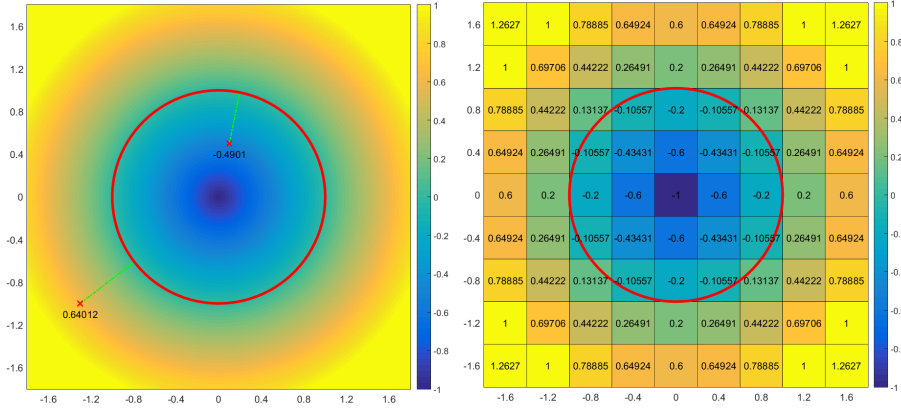


Figure 1: The signed distance function of the unit circle (left) and a signed distance field of the same geometry (right).

approximations.

A particular difficulty with DSDF approaches is that they cannot deal with hard corners as bilinear interpolation tends to smooth them out. More general approaches, such as edge-aware sampling [2] or feature based textures [13] can improve on this, but there are font-specific solutions to this problem as well. For example [10] replaces the Bézier segments by linear and circular arcs that also improves performance. This latter approach is the most similar work to ours but our argument is based on the functional approximation of the SDF and the local differential geometry of the boundary.

Recently, results have been published about a more accurate anti-aliasing of vector based fonts in 3D scenes [4].

3 Theoretical background

Notation We denote the n -dimensional Euclidean space by \mathbb{E}^n and $\|\cdot\|_2$ is the Euclidean norm. The partial derivatives of an $f : \mathbb{E}^2 \rightarrow \mathbb{R}$ function are $\partial_1 f, \partial_2 f$ or f_x, f_y . The scalar product of vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ is written as $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^T \mathbf{b}$.

Definition 1. Let $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$ be a multi-index. Then we define the following operations:

- $|\alpha| = \alpha_1 + \alpha_2 + \dots + \alpha_n$
- $\alpha! = \alpha_1! \cdot \alpha_2! \cdot \dots \cdot \alpha_n!$, where $0! = 1$
- $\mathbf{x}^\alpha = x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot \dots \cdot x_n^{\alpha_n}$ ($\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{E}^n$)
- $\partial^\alpha f = \partial_1^{\alpha_1} \partial_2^{\alpha_2} \dots \partial_n^{\alpha_n} f$ ($f : \mathbb{E}^n \rightarrow \mathbb{R}$)

Definition 2 (Regular curves). An $\mathbf{r} \in \mathbb{R} \rightarrow \mathbb{E}^n$ parametric curve is regular iff $\forall t \in \mathcal{D}_{\mathbf{r}} : \mathbf{r}'(t) \neq \mathbf{0}$.

Definition 3 (Natural parametrization). $\hat{\mathbf{r}} \in \mathbb{R} \rightarrow \mathbb{E}^n$ is the natural or arc-length parametrization of $\mathbf{r} \in \mathbb{R} \rightarrow \mathbb{E}^n$, which means $\forall s \in \mathcal{D}_{\hat{\mathbf{r}}} : \|\hat{\mathbf{r}}'(s)\|_2 = 1$.

Definition 4 (Arc-length function). $s : [a, b] \rightarrow [0, L]$ is called the arc-length function of $\mathbf{r} : [a, b] \rightarrow \mathbb{E}^n$, if $s(t) = \int_a^t \|\mathbf{r}'(x)\|_2 dx$ and $L = \int_a^b \|\mathbf{r}'(x)\|_2 dx$.

Corollary 1 (Natural parametrization). Let $\mathbf{r} : [a, b] \rightarrow \mathbb{E}^n$ be a regular parametric curve, then $\mathbf{r} = \hat{\mathbf{r}} \circ s$ and $\hat{\mathbf{r}} = \mathbf{r} \circ s^{-1}$.

Definition 5 (G^n continuity). Two curves are $G^n, n \geq 1$ at a common point \mathbf{x} iff there exists a regular parametrization with respect to which they are C^n at \mathbf{x} .

Definition 6 (Signed distance function). The signed distance function $f : \mathbb{E}^2 \rightarrow \mathbb{R}$ of an $F \subset \mathbb{E}^2$ two dimensional shape is defined as $f(\mathbf{x}) = \text{sgn}(\mathbf{x}) \cdot d(\mathbf{x}, \partial F)$, where $\partial F := \overline{F} \cap (\mathbb{E}^2 \setminus F)$ is the boundary of F , $d(\mathbf{x}, G) = \inf_{\mathbf{y} \in G} \|\mathbf{x} - \mathbf{y}\|_2$ and $\text{sgn}(\mathbf{x})$ determines if \mathbf{x} is inside or outside of F :

$$\text{sgn}(\mathbf{x}) = \begin{cases} -1 & \text{if } \mathbf{x} \in F, \\ 1 & \text{if } \mathbf{x} \notin F. \end{cases}$$

Definition 7 (Footpoint parameter relation). Let $\mathbf{p} : [a, b] \rightarrow \mathbb{E}^2$ be a regular parametric curve. Then $r \subset \mathbb{E}^2 \times [a, b]$ contains all (\mathbf{x}, t) pairs, where t is a parameter for a closest point on the curve for \mathbf{x} :

$$r = \left\{ (\mathbf{x}, t) \in \mathbb{E}^2 \times [a, b] \mid \|\mathbf{p}(t) - \mathbf{x}\|_2 = \min_{u \in [a, b]} \|\mathbf{p}(u) - \mathbf{x}\|_2 \right\}$$

The above relation is usually almost a function. The points \mathbf{x} that have multiple parameters associated with are the ones where there is no unique closest point, such as the center of a circle. These points lie on the cut locus of the curve.

Definition 8 (Footpoint parameter mapping). A $t : \mathbb{E}^2 \rightarrow [a, b]$ footpoint parameter mapping of a regular parametric curve is any narrowing of the footpoint parameter relation of the curve, so that it is a function on the whole domain: $t \subset r : \forall \mathbf{x} \in \mathbb{E}^2 |\{u \in [a, b] : (\mathbf{x}, u) \in t\}| = 1$.

4 Algebraic SDFs

Algebraic signed distance fields are based on the Taylor approximation theorem [3]. Instead of simple function values, we store higher order Taylor polynomial approximations to the shape in the samples.

Definition 9 (Taylor polynomials). Let $f : \mathbb{R}^n \rightarrow \mathbb{R}, \mathbf{a} \in D_f \subset \mathbb{R}^n, f \in C^k[\mathbf{a}]$. The order k Taylor polynomial of f around \mathbf{a} is

$$T_f^{(k)}(\mathbf{x}) = \sum_{|\alpha| \leq k} \frac{\partial^\alpha f(\mathbf{a})}{\alpha!} (\mathbf{x} - \mathbf{a})^\alpha \quad (\mathbf{x} \in \mathbb{R}^n)$$

Taylor polynomials have a good local approximation property characterized as

Theorem 1 (Taylor Approximation Theorem). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $S \subset \mathbb{R}^n$ open and convex, $f \in C^{k+1}[S]$. If $\mathbf{a}, \mathbf{a} + \mathbf{h} \in S$, then*

$$f(\mathbf{a} + \mathbf{h}) = T_f^{(k)}(\mathbf{a} + \mathbf{h}) + R_{\mathbf{a},k}(\mathbf{h})$$

where the residual $R_{\mathbf{a},k}$ can be expressed using an adequate $c \in (0, 1)$:

$$R_{\mathbf{a},k}(\mathbf{h}) = \sum_{|\boldsymbol{\alpha}|=k+1} \partial^{\boldsymbol{\alpha}} f(\mathbf{a} + c \cdot \mathbf{h}) \frac{\mathbf{h}^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!}$$

or with an integral form:

$$R_{\mathbf{a},k}(\mathbf{h}) = (k+1) \sum_{|\boldsymbol{\alpha}|=k+1} \frac{\mathbf{h}^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \int_0^1 (1-t)^k \partial^{\boldsymbol{\alpha}} f(\mathbf{a} + t\mathbf{h}) dt .$$

In the two dimensional case, the Taylor polynomials are written in the following way. Let $\mathbf{x} = [x, y]^T$, $\mathbf{a} = [a, b]^T \in \mathbb{E}^2$, then

$$T_f^{(k)}(x, y) = \sum_{i=0}^k \sum_{j=0}^i \frac{\partial_1^j \partial_2^{i-j} f(a, b)}{j!(i-j)!} (x-a)^j (y-b)^{i-j}$$

In algebraic signed distance fields, these local approximations are used as samples. In the simplest construction, all of the samples share a common polynomial degree. This degree is referred to as the order of the algebraic distance field, since it is the approximation order of the stored polynomials.

An order 1 algebraic distance field uses the distance function value and the first partial derivatives. In the local polynomial basis, it is written as

$$T_f^{(1)}(\mathbf{x}) = f(\mathbf{a}) + f_x(\mathbf{a})(x-a) + f_y(\mathbf{a})(y-b)$$

Local refers to that the basis functions $x-a$, $y-b$ are relative to the sample position $\mathbf{a} = [a, b]^T$, and the global basis would be independent of it. Similarly, an order 2 algebraic distance field sample is derived from the distance function value, the first and the second order derivatives:

$$T_f^{(2)}(\mathbf{x}) = f(\mathbf{a}) + f_x(\mathbf{a})(x-a) + f_y(\mathbf{a})(y-b) + \frac{1}{2} f_{xx}(\mathbf{a})(x-a)^2 + f_{xy}(\mathbf{a})(x-a)(y-b) + \frac{1}{2} f_{yy}(\mathbf{a})(y-b)^2$$

Note that the classical signed distance field coincides with the order 0 algebraic signed distance field, as the degree 0 polynomial approximation of a function is a constant function value, i.e. $T_f^{(0)}(\mathbf{x}) \equiv f(\mathbf{a})$. This way, the algebraic signed distance fields can be viewed as a generalizations of the classical one.

Also note that an algebraic signed distance field sample is not necessary represented directly by the constant function value and the derivatives; see Section 5.

5 Representation of algebraic samples

The samples in an algebraic signed distance field are polynomials. The order zero case coincides with traditional signed distance fields, which store signed distance values. Polynomials are usually represented with their power basis coefficients, but any basis of representation can be chosen. The two main options are local and global basis.

By local polynomial basis we mean that the basis is relative to the sample position. In a global basis all of the basis functions are given in the same coordinate system, i.e. same axes and origin. For a single sample, there is no real difference between the two options. The practical difference appears when multiple samples interact in some way. This is the case when querying SDFs, which is usually done using hardware accelerated linear blending. We show that the global basis is invariant under affine combinations, which helps leveraging this GPU texture filtering.

To prove this, let $\lambda_i(\mathbf{x})$ be an arbitrary collection of barycentric weight functions, i.e. $\lambda_i : \mathbb{E}^n \rightarrow \mathbb{R}$ such that $\forall \mathbf{x} \in \mathbb{E}^n : \sum_i \lambda_i(\mathbf{x}) = 1$. Let $b_j(\mathbf{x})$ be a polynomial basis and $P_i(\mathbf{x}) = \sum_j a_{ij} b_j(\mathbf{x})$ arbitrary polynomials. Let us now consider the barycentric combination of these polynomials as

$$\begin{aligned} \sum_i \lambda_i(\mathbf{x}) P_i(\mathbf{x}) &= \sum_i \lambda_i(\mathbf{x}) \left(\sum_j a_{ij} b_j(\mathbf{x}) \right) = \\ &= \sum_i \sum_j \lambda_i(\mathbf{x}) a_{ij} b_j(\mathbf{x}) \\ &= \sum_j \underbrace{\left(\sum_i \lambda_i(\mathbf{x}) a_{ij} \right)}_{\hat{a}_j(\mathbf{x})} b_j(\mathbf{x}) \\ &= \sum_j \hat{a}_j(\mathbf{x}) b_j(\mathbf{x}) \end{aligned}$$

This means that if the polynomials are all stored in the same global basis, we can first interpolate the coefficients of the polynomials and then evaluate one polynomial. If the texture stores polynomial coefficients, the GPU bilinear interpolation returns $\hat{a}_j(\mathbf{x})$. This implicitly uses a linear λ_i weighting function, but we can achieve different non-linear λ_i weights by modifying the texture lookup coordinates.

6 Geometric interpretation

The main issue with the algebraic approach is that the number of coefficients grow quadratically with the order. For order n , the number of coefficient is $\binom{n+2}{n} = \frac{(n+2)(n+1)}{2}$. Thus a second order algebraic sample consists of six coefficients which already do not fit into a single texture sample. This motivated us to investigate

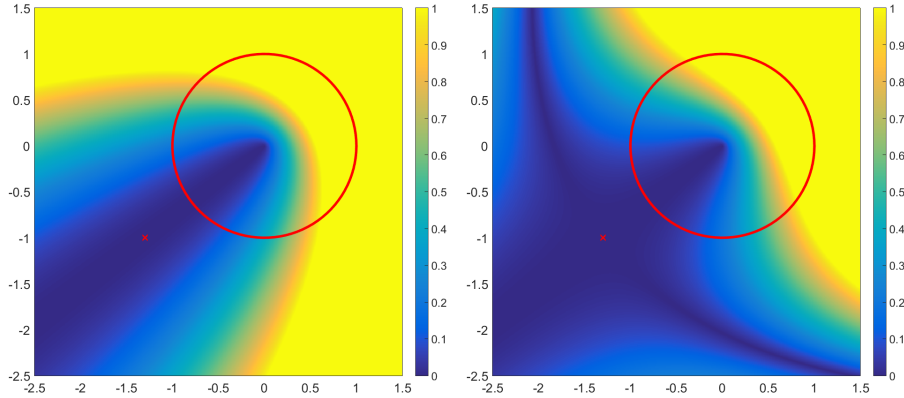


Figure 2: Error heat map of the signed distance function inferred from a first and a second order algebraic sample – Taylor polynomial – approximating the SDF of the unit circle. The sample position is the red x and the geometry is the red circle.

alternative approximations with a lower scalar footprint that still retain a proven approximation order.

The geometric interpretation of the order 0 algebraic sample is already established in the literature: a value $d = f(\mathbf{x})$ sampled from the distance function is the signed distance from the closest point to the curve (or surface in 3D) and defines an unbounding circle (sphere) around the sample position with a radius of $|d|$. This unbounding volume contains no surface point.

The first order algebraic sample is a linear polynomial with three coefficients, which naturally describes a line with its zero level set. However, in our case, this line has a geometric interpretation: this line is the tangent line of the curve at the closest point to the sample point. Moreover, the polynomial is the exact signed distance function of the represented line, since the gradient of the function is unit length and constant for the entire domain.

A second order algebraic sample is a second order polynomial with six coefficients, which means that the level sets are quadrics or, in special cases, degenerate quadrics. However the polynomial here – unlike in the order one case – is not the signed distance function of the zero level set, but only a local approximation to it. The nearest point on the curve is usually part of the zero level set of the second degree polynomial and the close neighbourhood matches the approximated curve.

These geometric interpretations open a way to using geometric objects as approximations. To see what kind of geometries could replace the algebraic samples, we show that two curves connecting with G^n continuity have a C^n continuous signed distance function around the connection. For this let us first prove

Theorem 2. *The signed distance function of any regular parametric curve is independent of the parametrization of the curve.*

Proof. Let us start with an arbitrary parametrization, and show that the signed distance function is the same as for the natural parametrization. Let $\mathbf{p} : [a, b] \rightarrow \mathbb{E}^2$ be a regular parametric curve, and consider one of its footpoint parameter mappings as defined in Definition 8. A closest point function $\mathbf{p}^* : \mathbb{E}^2 \rightarrow \mathbb{E}^2$ is then $\mathbf{p}^* = \mathbf{p} \circ t$. This maps all points in the plane to the closest point – or one of the closest points – on the curve. With the help of the closest point function, we can define the (unsigned) distance function $\tilde{f} : \mathbb{E}^2 \rightarrow \mathbb{R}_0^+$ of the curve as $\tilde{f}(\mathbf{x}) = \|\mathbf{p}^*(\mathbf{x}) - \mathbf{x}\|_2$. For the signed distance function, we also need to partition the plane into inside and outside sets. Let $F \subset \mathbb{E}^2$ be the inside of our shape (bounded by \mathbf{p}), then the signed distance function $f : \mathbb{E}^2 \rightarrow \mathbb{R}$ is

$$f(\mathbf{x}) = \text{sgn}(\mathbf{x}) \cdot \tilde{f}(\mathbf{x}) \quad (\mathbf{x} \in \mathbb{E}^2).$$

Now let $\hat{\mathbf{p}} : [0, L] \rightarrow \mathbb{E}^2$ be the natural parametrization of \mathbf{p} . As we have noted in Corollary 1, the arc-length function $s : [a, b] \rightarrow [0, L]$ links \mathbf{p} to the natural parametrization: $\mathbf{p} = \hat{\mathbf{p}} \circ s$. Similarly, the footpoint parameter mapping $\hat{t} : \mathbb{E}^2 \rightarrow [0, L]$ of $\hat{\mathbf{p}}$ is related to t through $s : t = s^{-1} \circ \hat{t}$. These show that \mathbf{p}^* is independent of parametrization and thus the signed distance function as well since

$$\begin{aligned} \mathbf{p}^* &= \mathbf{p} \circ t \\ &= \hat{\mathbf{p}} \circ s \circ s^{-1} \circ \hat{t} \\ &= \hat{\mathbf{p}} \circ \hat{t} \end{aligned}$$

□

Another interpretation of Theorem 2 is that the signed distance function is invariant under regular reparametrizations of the represented curve. For example, the first partial derivatives with respect to the X and Y coordinate axes are

$$\begin{aligned} \partial_k f(\mathbf{x}) &= \partial_k (\text{sgn}(\mathbf{x}) \cdot \|\mathbf{p}^*(\mathbf{x}) - \mathbf{x}\|_2) \\ &= \text{sgn}(\mathbf{x}) \cdot \frac{(\mathbf{p}^*(\mathbf{x}) - \mathbf{x})^T}{\|\mathbf{p}^*(\mathbf{x}) - \mathbf{x}\|_2} \cdot (\partial_k \mathbf{p}^*(\mathbf{x}) - \partial_k \mathbf{x}) \quad (k \in \{1, 2\}) \end{aligned}$$

Corollary 2. *If two curve segments connect G^n continuously, then the distance fields of the curves are C^n continuous at the connection point.*

This is true because G^n continuity is equivalent to the two connecting curves having the same natural parametrization [15].

Remark 1. This continuity extends to the line segment starting from the foot point through the sample, until the normal line intersects the cut locus. See Figure 3 for an example.

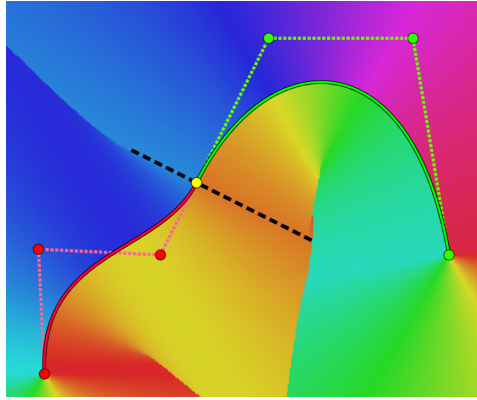


Figure 3: The red and green cubic Bezier-segments connect G^1 continuously at the shared yellow point. The background color encodes the gradient vector of the unsigned distance function of the curves. The black dashed line partitions the plane by which curve segment provides the foot point. As we note in Remark 1, the gradient is continuous on this line segment.

7 Geometric proxies

We have seen in Corollary 2 that G^n continuous curves have C^n continuous signed distance fields outside the cut locus. As a consequence, curves with proper connectivity properties can be used to approximate the signed distance field of a shape instead of algebraic polynomials. We have a few expectations towards the geometric proxies. (i) The proxy has to be a theoretically proven approximation to the original SDF. This is fulfilled by any G^n continuously connected curve at the footpoint of the desired order. (ii) It has to encode the inside/outside partitioning. This will give us the sign of the distance field, which is a binary information but it can be often encoded naturally into the representation of the proxy. (iii) The SDF of the proxy should be easy to evaluate. It is important since it will be used several times when the distance field is sampled. (iv) It should have a low memory cost, as we want a representation that is at least as memory efficient as the algebraic distance fields are.

The order zero geometric proxy is the footpoint itself. In practice, this construction is suboptimal, as the classical signed distance field uses a single scalar value as a sample, but a 2D point is represented by two scalars and the sign needs additional handling.

In first order, the simplest curve is the line which is in our case the tangent line of the approximated curve at the footpoint. Note that this is equivalent to the algebraic first order sample, however in the next chapter we will show a more efficient encoding. As seen with the polynomial encoding, the sign is naturally present in the linear function and the inside/outside partitioning is trivially inferred

from a sample. For the tangent line to exist, the curve must be G^1 continuous at the footpoint. This means that the direction of the gradient of the curve changes continuously. The length of the gradient, however, is allowed to have discontinuities.

The second order geometric proxy is the circle. The approximating circle is again touching the curve at the footpoint and in this case it will be the osculating circle. This differs from the algebraic order two representation, so a comparison is necessary. The inside of the circle represents our 2D shape if the shape is locally convex and the outside if the shape is locally concave. The criterion for the existence of the osculating circle is that the curve is G^2 continuous. This means, that it is G^1 continuous and the osculating circles change continuously. It is important to note that the representation has to fall back to the first order case if the curvature is zero, since that would mean an infinitely large circle, that is a straight line, that are very common in modelling and especially vector art.

Geometric distance fields are sampled similarly to algebraic DSDFs, but this time we cannot rely on automatic GPU-interpolation, because the stored geometric data cannot be interpolated trivially into a new correct proxy geometry. First we take the proxies at the corners of the sampled cell, then calculate the distance from all four proxies and finally, bilinearly interpolate the received values.

8 Encoding of geometric samples

8.1 First order geometric sample

The first order geometric sample is a half plane – or an oriented line. We have already seen that the first order algebraic sample describes the same line with a first order polynomial.

With a geometric approach, we could store the footpoint and the direction vector or normal vector of the line. If we save these, the footpoint would consume two scalars and the line direction/normal one additional scalar. However, note that the tangent line is always perpendicular to the footpoint vector, i.e the vector starting at the sample point and pointing to the footpoint. Using this fact, we can discard the direction component and only store the footpoint.

To be able to differentiate between the inside and outside, we propose to use a modified polar coordinate system. The change is that the radial coordinate is allowed to have a sign, encoding the insideness of the sample point. For consistency, the angular coordinate has to be rotated 180 degrees if the radial coordinate is negative.

A first order geometric proxy is then encoded by a pair of signed distance and modified polar angle values $(d, \theta) \in \mathbb{R} \times [0, 2\pi)$, and the curve normal $\mathbf{n} \in \mathbb{R}^2$ at the footpoint and the footpoint $\mathbf{t} \in \mathbb{E}^3$ are computed upon query as

$$\mathbf{n} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} \quad (1)$$

$$\mathbf{t} = \mathbf{x} - d \cdot \mathbf{n} \quad (2)$$

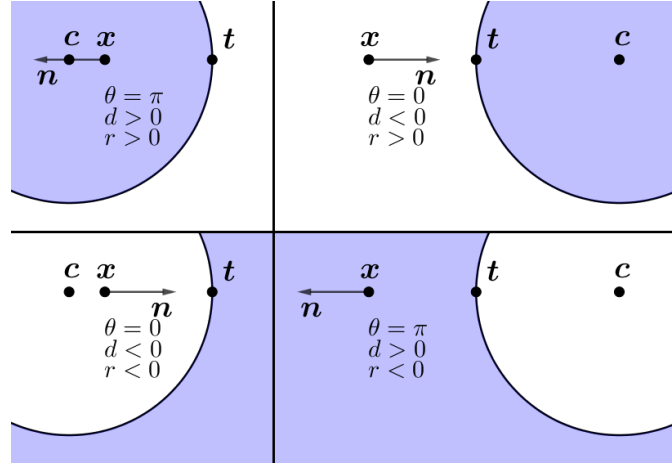


Figure 4: An order 2 geometric sample (d, θ, r) can represent four circles, depending on the signs of d and r . The figure illustrates these four corresponding to the same x sample position and t footprint. The inside of the shape is in blue, the outside is in white.

Note that this representation is an orthogonal extension of the ordinary signed distance field, as the first coordinate is simply the signed distance, and the second, angular coordinate is independent of it.

8.2 Second order geometric sample

The second order geometric proxy is the osculating circle and there are various options for the representation of a circle.

The three parameters representing the circle proxies are $(d, \theta, r) \in \mathbb{R} \times [0, 2\pi) \times \mathbb{R}$, which is an orthogonal extension of d and θ , with the new signed radius parameter, r . The footprint $t \in \mathbb{E}^2$ can be decoded the same way as in Equation (2). And the center $c \in \mathbb{E}^2$ is calculated as

$$c = t + r \cdot n \quad (3)$$

Here, we took advantage of the special geometric setting and noted that the osculating circle is tangent to the tangent line at the footprint. By displacing the footprint along the normal line by the signed radius, we can reconstruct the correct osculating circle. Figure 4 shows the four cases that can correspond to an order 2 geometric sample.

The last problem to solve is when the osculating circle degenerates into the tangent line, i.e. the case of zero curvature. We might encode this as either positive or negative infinity in r if the number format allows it. For example most of the floating point formats have infinities. In a uniformly quantized representation the minimum and maximum values could be treated as such. Another way could be

to use signed curvature $\kappa \in \mathbb{R}$ instead of the radius. The relation between the two quantities is $\kappa = \frac{1}{r}$. This way, the zero curvature sample has zero as a third parameter instead of infinity. In this case, the decoder ignores the curvature and it falls back to the first order mode and the line.

8.3 Evaluation of a sample

To query a value from a sample of the geometric distance field at a query point $\mathbf{p} \in \mathbb{E}^2$, first, the properties of the geometric proxy has to be calculated. Then the signed distance is calculated at the query point to the proxy geometry. In first order, calculate \mathbf{n} and \mathbf{t} using Equations (1) and (2), and the distance as

$$f(\mathbf{p}) = \langle \mathbf{p}, \mathbf{n} \rangle - \langle \mathbf{t}, \mathbf{n} \rangle. \quad (4)$$

For the second order, the first step is to decode the parameters of the osculating circle using Equations (1), (2) and (3). If the sample represents a circle – and not a line as a fallback, i.e. $\kappa \neq 0$ – then the final distance is

$$f(\mathbf{p}) = -\text{sgn}(r) \cdot (\|\mathbf{p} - \mathbf{c}\|_2 - |r|). \quad (5)$$

Blending can be used to combine multiple samples but unlike for algebraic samples, we cannot use hardware accelerated texture filters, as the samples first have to be decoded from their encoded form.

9 Results

9.1 Font representations

We tested our signed distance field constructions by font rendering. We used various TrueType fonts [8] in our tests. Distance fields were generated for each glyph, which were combined into a single 2D texture.

A TrueType glyph consists of one or multiple outlines. The winding direction of the outline defines if the outline is an outside border or a border of an inside hole. Outlines consist of a closed loop of line segments and quadratic Bezier segments, given with their control points. We used the FreeType library [14] to load the font data.

These outlines are the input for the DSDF generation, given as a series of segments. For simplicity, we used a brute force method for finding the closest outline point for the samples. An analytic nearest point solution is calculated for every relevant segment, and the closest one is selected. Then the insideness is decided by the direction of the nearest segment, any other features needed for the different DSDF constructions (derivatives, curvature) are extracted analytically. We implemented the DSDF generation for the GPU, and even though it is a brute force algorithm, the run time for the generation is negligible. Also note, that the generation step is usually only needed to be done once.

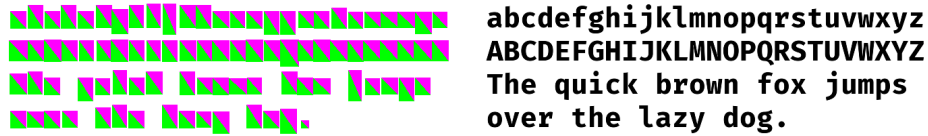


Figure 5: Individual rectangles are rendered for every character. The fragment shader samples the DSDF and determines the alpha level of the pixel. The used font is FiraCode Bold [12].

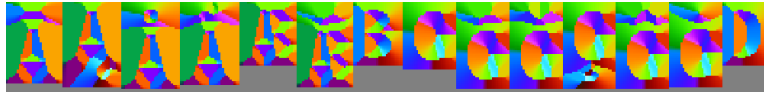


Figure 6: Visualization of a part of an algebraic first order texture atlas. The three color components encode the gradient vector of the distance field.

The generated DSDF is saved in a 2D texture and used as an atlas for rendering. The font characters are stored in rectangular regions in the atlas, see for example Figure 6. At render time each character to be rendered is covered with a rectangle (consisting of two triangles – see Figure 5), and a custom fragment shader is used to sample and calculate the signed distance from the outline of the character. The distance is then mapped to the alpha value of the fragment. The mapping can be simply 1 and 0 for negative and positive values respectively, but a better anti-aliased result is achieved by setting a 1-2 pixel wide band with a gradient between the two values. The gradient is tuned to represent the coverage of a hard edge passing through the pixel in the given distance.

Font rendering is a challenging test case for DSDFs, since they often operate with corners which do not satisfy our assumed higher order continuity properties. Nonetheless, higher order DSDFs provide a better use of memory, as it is shown in Section 9.2.

9.2 Test results

We tested the signed distance field constructions on different fonts and vector arts. The tests rasterized the DSDFs as high resolution classical distance fields (containing only signed distance values) and compared them to the signed distance function values of the original vector image. On the tables and figures of this section we refer to the zero, first and second order algebraic distance fields as A0, A1 and A2. Similarly G1 and G2 are the first and second order geometric distance fields. A0 distance fields are the traditional distance fields.

A1 and G1 theoretically encode the same information and therefore reconstruct the same distance values for any query position apart from a small numerical error.

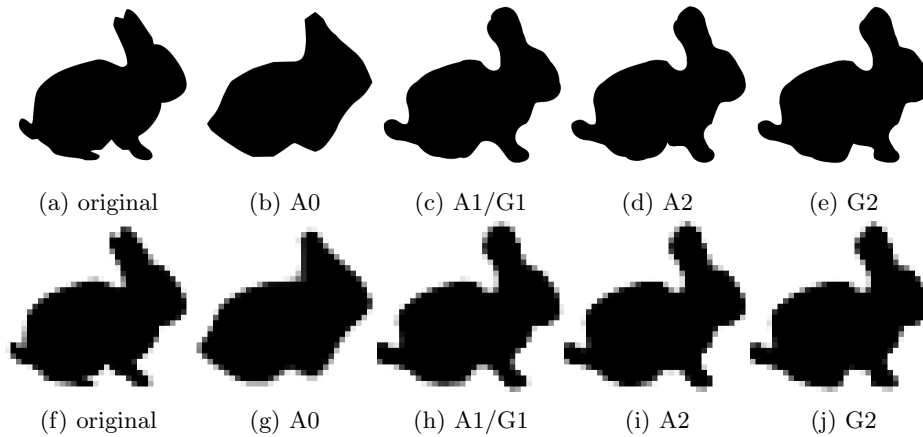


Figure 7: Test case for DSDF representations on a bunny silhouette. (a) shows the original vector art, (b)-(e) are high resolution rasterizations of the tested DSDF representations. (f)-(j) show the corresponding rasterizations at the intended display size (34x30px). The DSDFs have a four times larger sample spacing compared to the display resolution, therefore having 16 times less samples/pixels. (c) and (h) contain both A1 and G1 as their reconstruction always matches exactly.

The rasterized images of A1 and G1 are thus indistinguishable in practice.

The DSDFs queried at their sample positions are exact. This means that if the pixels are aligned with the field samples, the rasterized image is the same for all algebraic and geometric fields. If the pixels and field samples are offset or the pixels are sparser than the samples, the queries of the DSDFs are so close to the true distance function value that the resulting image is stable, giving a robust rendering method.

Our new signed distance field constructions proved to be a useful tool for font and vector art rendering. Example renders of a bunny silhouette can be seen on Figure 7. The shown distance fields have four times lower resolution than the intended display size, meaning that they contain 16 times less samples. This extreme setting is presented here to show that even these sparse fields – with much lower resolution – reconstruct the original vector art closely apart from really fine details. Rasterized at the intended display resolution, these lower resolution higher order distance fields (excluding A0) only differ in a few pixels near the most curved parts.

Table 1 shows the results of two accuracy tests. Test case #1 was some text similar to Figure 5 and test case #2 was calculated on renderings of the bunny seen on Figure 7 but with twice the resolution. The first two columns show the median and mean absolute error of the reconstructed signed distance field values. The third column is the error ratio of the reconstructed sign, i.e. when the inferred inside/outside partitioning is incorrect. Note that A1 and G1 are equal as stated before. G2 usually performs at the same level or better as A2. The first order DSDFs do not seem to lower the absolute distance error compared to the classical

Table 1: Error metrics of the different DSDF representations.

	test #1			test #2		
	median	mean	sign	median	mean	sign
A0	0.0399	0.1053	2.14%	0.0399	0.0871	1.23%
A1	0.0411	0.1119	1.84%	0.0413	0.0861	0.98%
G1	0.0411	0.1119	1.84%	0.0413	0.0861	0.98%
A2	0.0029	0.1002	1.52%	0.0121	0.0785	0.83%
G2	0.0015	0.0869	1.39%	0.0098	0.0743	0.87%

Table 2: Average render times of full screen test texts.

A0	A1	G1	A2	G2
0.306 ms	0.325 ms	0.319 ms	0.370 ms	0.349 ms

distance field (A0), but they always improve the sign correctness metric. This can be seen on Figure 7 as well: the average error values might be close, but visually the first order fields perform better. Similarly, the second order fields always outperform the first order ones. Other test cases have shown similar relative¹ numbers for all constructions.

Table 2 shows averaged render times for the different DSDF constructions from FullHD full screen tests. The sampling of higher order distance fields always costs more than the traditional fields, but they provide the possibility to use lower resolution fields or have better precision at the same resolution. G2 rendering is faster than A2 despite the extra calculations needed. This can be explained by the fact that A2 needs two textures for its 6 coefficients.

10 Conclusion

We proposed a geometric generalization of higher order signed distance fields. We have proven that these constructs have the same approximation order as their Taylor-based algebraic counterparts. These theoretical results were also validated by empirical measurements.

The geometric distance field representations proved to be valid and efficient tools for font and vector art representation and rendering. Our empirical tests have shown that the geometric signed distance fields are as good as the algebraic ones, in fact, in most of the cases the second order geometric construction is more precise than the second order algebraic. This comes with a performance cost of 10 – 20%

¹Because the error in value and sign both depend on scale

in render times compared to using traditional signed distance fields, assuming the same field resolution.

In the future, we plan to extend our geometric construction to three dimensional signed distance fields. Three dimensional first order algebraic distance fields already proved their applicability [3].

References

- [1] Aaltonen, Sebastian. GPU-based clay simulation and ray-tracing tech in Claybook, Game Developers Conference 2018. In *Game Developers Conference*, San Francisco, CA, March 2018.
- [2] Bala, Kavita, Walter, Bruce, and Greenberg, Donald P. Combining edges and points for interactive high-quality rendering. *ACM Trans. Graph.*, 22(3):631–640, July 2003. DOI: 10.1145/882262.882318.
- [3] Bán, Róbert and Valasek, Gábor. First order signed distance fields. In Wilkie, Alexander and Banterle, Francesco, editors, *Eurographics 2020 - Short Papers*. The Eurographics Association, 2020. DOI: 10.2312/egs.20201011.
- [4] Ellis, A., Hunt, W., and Hart, J. Real-time analytic antialiased text for 3-D environments. *Computer Graphics Forum*, 38(8):23–32, 2019. DOI: 10.1111/cgf.13757.
- [5] Evans, Alex. Learning from failure: a survey of promising, unconventional and mostly abandoned renderers for ‘Dreams PS4’, a geometrically dense, painterly UGC game. In *Advances in Real-Time Rendering in Games*. MediaMolecule, SIGGRAPH, 2015.
- [6] Fuhrmann, Arnulph, Sobottka, Gerrit, and Groß, Clemens. Distance fields for rapid collision detection in physically based modeling. In *Proceedings of the International Conference Graphicon*, 2003.
- [7] Green, Chris. Improved alpha-tested magnification for vector textures and special effects. In *ACM SIGGRAPH 2007 Courses*, SIGGRAPH ’07, pages 9–18, New York, NY, USA, 2007. ACM. DOI: 10.1145/1281500.1281665.
- [8] Kaasila, Sampo. Method and apparatus for moving control points in displaying digital typeface on raster output devices, 1994. United States Patent No. US5155805A.
- [9] Loop, Charles and Blinn, Jim. *Rendering Vector Art on the GPU*. In *GPU Gems 3*, chapter 25. Addison-Wesley Professional, 2007.
- [10] Michail, Ashraf A., Teitlebaum, David B., and Furtwangler, Brandon C. Arc spline GPU rasterization for cubic Bezier drawing, 2010. United States Patent No. US8624899B2.

- [11] Osher, S. and Fedkiw, R. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, 2003. DOI: 10.1007/b98879.
- [12] Prokopov, Nikita. Fira code: free monospaced font with programming ligatures. GitHub URL: <https://github.com/tonsky/FiraCode>.
- [13] Ramanarayanan, G., Bala, K., and Walter, B. Feature-Based Textures. In Keller, Alexander and Jensen, Henrik Wann, editors, *Eurographics Workshop on Rendering*. The Eurographics Association, 2004. DOI: 10.2312/EGWR/EGSR04/265-274.
- [14] Turner, David, Wilhelm, Robert, and Lemberg, Werner. The FreeType Project. URL: <https://www.freetype.org/index.html>.
- [15] Valasek, Gábor. *Nonlinear Geometric Models*. PhD thesis, Eötvös Loránd University, 2016. DOI: 10.15476/ELTE.2015.123.
- [16] Valasek, Gábor. Generating distance fields from parametric plane curves. In *Annales Mathematicae et Informaticae* 48, pages 83–91, 03 2018. <http://publikacio.uni-eszterhazy.hu/id/eprint/3316>.
- [17] Wright, Daniel. Dynamic occlusion with signed distance fields. In *Advances in Real-Time Rendering in Games*. Epic Games (Unreal Engine), SIGGRAPH, 2015.