

Title:**Higher Order Algebraic Signed Distance Fields**Authors:

Gábor Valasek, valasek@inf.elte.hu, Eötvös Loránd University, Hungary

Róbert Bán, rob.ban@inf.elte.hu, Eötvös Loránd University, Hungary

Keywords:

Surface and Volume Representation, Implicit Representation, Geometric Modeling, Collision Detection

DOI: 10.14733/cadconfP.2022.287-291Introduction:

Signed distance functions (SDFs) are powerful implicit representations of curves and surfaces. Beyond encoding volume boundaries as their zero level-set, they also convey global geometric information about the scene as they map signed distances to all points in space. In real-time applications, their most common numerical representation is a regular grid of two or three dimensions. In conjunction with an interpolation method to infer a continuous approximation defined on all points of space, these can be efficiently evaluated on the GPU such that even the most demanding applications can utilize them [4, 5].

Several authors proposed to use first order approximations to the SDF at the samples, e.g. gradients [3] or plane equations [1]. In this paper, we present a straightforward generalization of this approach to higher orders and discuss various alternatives to the appropriate filtering of samples such that the inferred SDF reconstructs the given higher order derivatives at the samples. Our focus is on applications in high performance visualizations, as such, we prioritize run-time performance over optimal storage and restrict sampling topologies in our measurements to regular grids. We also discuss the shortcomings of this approach as means to decrease storage for complex shapes.

Higher order distance field samples:

We denote multi-indices by an $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$ tuple. Let $\mathbf{x} \in \mathbb{R}^n$ and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a sufficiently smooth function. Then we define the following operations with multi-indices:

$$|\alpha| = \alpha_1 + \alpha_2 + \dots + \alpha_n, \quad \alpha! = \alpha_1! \cdot \alpha_2! \cdot \dots \cdot \alpha_n!,$$

$$\mathbf{x}^\alpha = x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot \dots \cdot x_n^{\alpha_n}, \quad \partial^\alpha f = \partial_1^{\alpha_1} \partial_2^{\alpha_2} \dots \partial_n^{\alpha_n} f = \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \dots \partial x_n^{\alpha_n}}.$$

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a function that is sampled at an $\mathbf{x}_i \in \mathbb{R}^n$ set of points, i ranging over an appropriate set of multi-indices. Although our discussion holds for arbitrary spacing of samples, the remainder of this paper focuses on regular grids. By an order k sample, we refer to an $f_i(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ n -variate polynomial of total degree k , i.e. $f_i(\mathbf{x}) = \sum_{|\alpha| \leq k} a_{\alpha} \mathbf{x}^\alpha$. We consider three types of higher order samples:

1. Taylor expansions about the sample positions: $f_i(\mathbf{x}) = \sum_{|\alpha| \leq k} \frac{\partial^\alpha f(\mathbf{x}_i)}{\alpha!} (\mathbf{x} - \mathbf{x}_i)^\alpha$
2. Least-squares fit (LSQ) polynomials over regions around sample positions: $f_i(\mathbf{x}) = \sum_{|\alpha| \leq k} a_{i,\alpha} \cdot \mathbf{x}^\alpha$

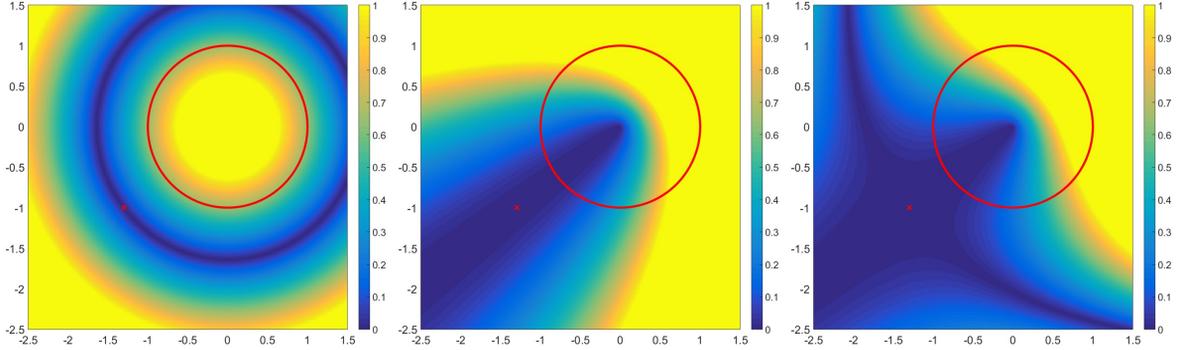


Fig. 1: Error heatmaps of planar signed distance fields extrapolated from a single order zero (left), order one (middle), and order two (right) Taylor sample. Larger numbers (yellow) mean higher error.

3. SDF partial derivatives in conjunction with Hermite interpolation: $\{\partial^\alpha f(\mathbf{x}_i)\}_{|\alpha| \leq k}$

Taylor samples can be represented by their coefficients in an appropriate basis. By choosing one where evaluation is invariant under barycentric combination, we can leverage GPU hardware filtering to optimize queries. Depending on the particular GPU, this may be an essential performance optimization.

In the 1D case, the above invariance means that $(1 - \lambda(x))f_i(x) + \lambda(x)f_{i+1}(x)$ can be reduced to a single evaluation after interpolating the coefficients of $f_i(x)$ and $f_{i+1}(x)$. It follows immediately that, for example, the global monomial basis satisfies this property, that is,

$$(1 - \lambda(x)) \sum_j a_{i,j} x^j + \lambda(x) \sum_j b_{i,j} x^j = \sum_j \left((1 - \lambda(x))a_{i,j} + \lambda(x)b_{i,j} \right) x^j \quad (2.1)$$

holds. As long as the data was normalized, this basis was sufficient using IEEE binary32 numbers and for the sake of convenience, we will use them throughout the paper. Regardless of basis, higher order Taylor expansions provide better accuracy over an increasing area, as long as the SDF is sufficiently continuous. Fig. 1. illustrates this for an origin centered circle. There, samples are represented by the $a_{ij} \in \mathbb{R}$ coefficients of the polynomials a_{00} , $a_{10}x + a_{01}y + a_{00}$, and $a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{10}x + a_{01}y + a_{00}$ respectively. The sample position is denoted by the red X. The geometry is the origin centered, radius 1 red circle. Blue corresponds to low error, yellow is high error.

LSQ samples are also stored by their coefficients, however, we compute them as the solution to an optimization problem. We detail a general framework for such constructs in the next point.

For Hermite reconstruction, partial derivatives may be stored by their coordinates. Note that we only use derivatives up to the total order of k . That is, for example in the case of first order samples in the plane, we store $f(\mathbf{x}_i)$, $\partial_x f(\mathbf{x}_i)$, $\partial_y f(\mathbf{x}_i)$ but we do not encode $\partial_{xy}^2 f(\mathbf{x}_i)$.

All of the above representations may be made more efficient by noting that an SDF gradient is of unit length everywhere where the SDF itself is differentiable.

Least squares SDF construction framework:

For every \mathbf{x}_i sample position of the grid, let us construct a small *fine grid*, that collects a number of distance samples in the neighborhood of \mathbf{x}_i . Let this fine grid be $\mathbf{s}_{ij} = \mathbf{x}_i + [i \cdot \Delta a, j \cdot \Delta b, k \cdot \Delta c]^T$, $\mathbf{j} = (i, j, k) \in \{-H, \dots, H\}^3$, i.e. symmetric about \mathbf{x}_i and let us denote the collection of fine grid distance samples by f_{ij} . Let N denote the total number of samples, $N = (2H + 1)^3$. Then the best fit

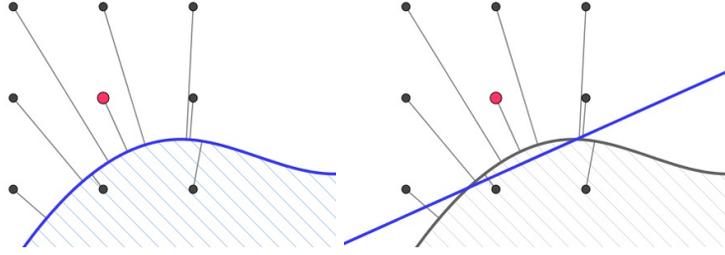


Fig. 2: The first order grid sample \mathbf{x}_i (in red) uses distance samples on the fine grid (in black) to fit a plane (blue line).

polynomial satisfies the following

$$\sum_{|\alpha| \leq k} \mathbf{a}_{i\alpha} \mathbf{s}_{ij}^\alpha \approx f_{ij} \quad , j \in \{-H, \dots, H\}^3 \quad (2.2)$$

where $(a_{i1}, a_{i2}, \dots, a_{iK})$, $K = \binom{k+3}{3}$ are the sample coefficients and f_{i1}, f_{i2}, \dots fine grid distance samples.

The best fit polynomial can be sought in various norms. For the ease of GPU implementations, we propose the use of the least-squares (LSQ) optimal fit, as optimization simplifies to a matrix-vector multiplication in this case. Moreover, the matrix in question only needs to be computed once. This follows from two observations. First, the Moore-Penrose pseudoinverse is the least-squares solution to Eqn. (2.2). If $\mathbf{X}_i^T \cdot \mathbf{X}_i$ is invertible, it can be written as $\mathbf{X}_i^+ = (\mathbf{X}_i^T \cdot \mathbf{X}_i)^{-1} \cdot \mathbf{X}_i^T$ and the optimal coefficients are $\mathbf{a}_i = \mathbf{X}_i^+ \cdot \mathbf{f}_i$. Second, we only need to store the \mathbf{X}_O^+ origin centered version of the above matrix. A fit using \mathbf{X}_O^+ merely translates the origin to the sample position without affecting the fit and a straightforward computation can transform those coefficients to the world origin. As we are only focusing on real-time GPU-viable degrees, i.e. up to 3, the condition number of the normal matrices is manageable with IEEE binary32 formats.

Fig. 2. illustrates the fitting process. This construction may be carried out on any representation where point-boundary distance queries can be evaluated.

Regardless of the norm chosen, the best fit polynomial converges to the Taylor polynomial as the fine grid extent tends to zero. Intuitively speaking, the size of the fine grid acts as a low pass filter on the geometry. In this sense, the fine grid is a level-of-detail control.

We found that the estimated gradients for first order samples are almost always of unit-length. They only deviate around the cut locus, where the samples collapse to piece-wise constant approximations consisting of the distance to the boundary. However, this deviation may be particularly large and reduce the overall precision of the field, especially when filtering is taken into account.

Filtering higher order samples:

Sample field queries usually aggregate several samples to construct an approximation to the sampled function. More generally, filtering can be formulated as a $\sum_j \lambda_j(\mathbf{x}) f_j(\mathbf{x})$ barycentric combination of $f_j(\mathbf{x})$ samples with some, not necessarily linear, barycentric weighting functions $\lambda_j(\mathbf{x})$, $\sum_j \lambda_j(\mathbf{x}) \equiv 1$.

For order k samples, the $\lambda_i(\mathbf{x})$ barycentric filtering functions have to be such that they preserve the higher order derivatives specified by the samples at the sample positions, that is,

$$\partial^\alpha \sum_j \lambda_j(\mathbf{x}_i) f_j(\mathbf{x}_i) = \partial^\alpha f_i(\mathbf{x}_i) \quad (\forall 0 \leq |\alpha| \leq k) \quad (2.3)$$

should hold under the \mathcal{J} filtering footprint, $i \in \mathcal{J}$. This can be done by combining univariate Hermite solutions to the same problem, effectively doing G^k blends of the implicit surfaces in the samples.

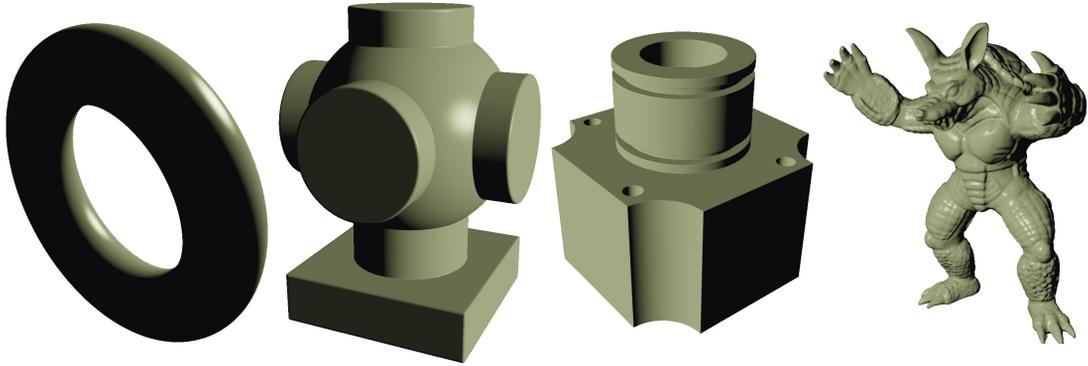


Fig. 3: The test models used in our accuracy and performance tests.

Another important aspect of high performance queries is whether the GPU accelerated bi- and trilinear interpolation units can be used for lookup. This is not the case for Hermite interpolation, however, higher order tensor-product blends of Taylor and LSQ polynomials may be carried out by perturbing the fractional part of the query texture coordinates, which was used in our performance tests.

Test results:

Accuracy tests resampled lower resolution (8^3 to 128^3) higher order discrete SDFs (DSDF) to 257^3 zero order ground-truth DSDFs. The vertices were normalized to be zero centered and within $[-1, 1]^3$. We computed the mean, median, and maximum absolute error metrics between the ground truth and the upsampled distance fields. A torus, two procedural geometries (SDF1 and SDF3), and an Armadillo mesh model were used in the accuracy tests, see Fig. 3.

Tab. 1. shows our total storage percentages such that the higher order field with the given filtering produces at most the same error as the trilinearly filtered zero order DSDF. Percentages are relative to trilinearly filtered zero order fields. O1c, O2q, O3s refers to DSDFs of order 1, 2, and 3 with cubic, quintinc, and septic Hermite blending interpolation. O1F and O2F are tensor tricubic and triquintic Ferguson-like solutions to first and second order Hermite interpolation problems.

		O1c	O1F	mean O2q	O2F	O3s	O1c	O1F	median O2q	O2F	O3s	O1c	O1F	max O2q	O2F	O3s
Torus	64	105%	7%	19%	8%	13%	98%	3%	8%	2%	4%	21%	14%	31%	19%	31%
	128	105%	6%	10%	4%	8%	98%	2%	4%	1%	1%	21%	13%	26%	20%	26%
	192	—	4%	7%	3%	5%	108%	2%	2%	0%	1%	21%	14%	25%	17%	21%
Arma.	64	139%	65%	114%	67%	105%	139%	41%	125%	53%	119%	98%	50%	53%	41%	61%
	128	139%	60%	103%	60%	105%	121%	32%	88%	38%	99%	98%	55%	46%	50%	57%
	192	—	50%	96%	53%	97%	119%	25%	78%	31%	89%	79%	39%	37%	34%	42%
SDF1	64	149%	55%	103%	60%	105%	130%	14%	75%	26%	81%	27%	24%	41%	19%	44%
	128	149%	45%	88%	50%	93%	98%	6%	28%	7%	31%	41%	20%	41%	14%	26%
	192	—	47%	96%	48%	97%	88%	4%	14%	4%	16%	31%	12%	18%	15%	28%
SDF3	64	158%	55%	125%	67%	119%	202%	21%	125%	46%	150%	71%	55%	53%	46%	52%
	128	169%	60%	119%	63%	127%	139%	8%	43%	13%	52%	50%	35%	31%	28%	26%
	192	—	57%	125%	62%	124%	—	6%	24%	6%	26%	47%	57%	28%	29%	24%

Tab. 1: Storage usage compared to trilinearly filtered order zero fields with matching accuracy.

The best storage is attained by using Ferguson-Hermite fields, however, the DSDF order appropriate blending proved to be close to them in maximum error metrics. Note that as the model complexity increases, the storage efficiency decreases. For example, while higher order fields improve storage on the torus by 3 – 25 \times , they are less efficient for the Armadillo and the procedural models. Our hypothesis is

that this is mainly due to fitting and combining samples from different sides of the cut locus.

Taylor based samples proved to be too local for blending-type interpolation and all the above figures used LSQ samples, as they provided better accuracy.

In terms of performance, we considered incorrect filtering of higher fields as well, e.g. trilinear filtering of O2, due to their prominence in real-time computer graphics. Tab. 2. contains average FPS figures over 100 frames, reported in milliseconds. Visually, the second order Ferguson Hermite solution was less pleasing due to the flatness introduced by the zeroed out partial derivatives and its evaluation cost was also too high to use beyond order 2.

	AMD RX 5700 FP32				NVIDIA 2080 FP32			
	O0	O1	O2	O3	O0	O1	O2	O3
linear	0.3	0.52	0.92	1.61	0.24	0.74	1.82	3.64
cubic		0.56	0.94	1.64		0.73	1.79	3.7
quintic			0.98	1.68			1.86	3.53
septic				1.69				3.52
FH		1.02	1.94			1.08	2.90	

Tab. 2: Average render times (ms) of the Armadillo model at full HD resolution.

Conclusions:

We presented a general framework to study higher order algebraic signed distance fields and an algorithm to compute LSQ samples from arbitrary representations that can resolve point-boundary distance queries. With specialized GPU implementations, these proved to be viable solutions for real-time graphics, however, their storage gains could be only realized with expensive filtering. Using too local samples, such as Taylor expansions or too narrow fine grids, is not advised with blending. Our experiments suggest that adapting storage to the cut locus of the model could provide further improvements.

Acknowledgement:

We would like to thank Visual Concepts for providing the AMD GPU used in the tests.

Supported by the ÚNKP-21-3 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund.

References:

- [1] Bán, R.; Valasek, G.: First Order Signed Distance Fields. In Eurographics 2020 - Short Papers; Wilkie, A., Banterle, F., Eds.; <https://doi.org/10.2312/egs.20201011>
- [2] Brunton, A.; Rmaileh, L. A.: Displaced Signed Distance Fields for Additive Manufacturing, ACM Trans. Graph. 2021, 40 (4). <https://doi.org/10.1145/3450626.3459827>
- [3] Cohen-or, D.; Levin, D.; Solomovici, A.: Three-Dimensional Distance Field Metamorphosis, ACM Transactions on Graphics 1998, 17, 116–141. <https://doi.org/10.1145/274363.274366>
- [4] Green, C.: Improved Alpha-Tested Magnification for Vector Textures and Special Effects, ACM SIGGRAPH 2007 Courses; SIGGRAPH '07; ACM: San Diego, California, 2007; pp 9–18. <https://doi.org/10.1145/1281500.1281665>
- [5] Wright, D.: Dynamic Occlusion with Signed Distance Fields, Advances in Real-Time Rendering in Games; SIGGRAPH, 2015.