Nice and Fast Implicit Surfaces over Noisy Point Clouds

Jan Klein *
University of Paderborn

Gabriel Zachmann†
University of Bonn

Abstract

We propose a new definition of the implicit surface for a noisy point cloud that allows for high-quality reconstruction of the surface in all cases. It is based on proximity graphs that provide a more topology-based measure for proximity of points. The new definition can be evaluated very fast, but, unlike other definitions based on the moving least squares (MLS) approach, it does not suffer from artifacts.

1 Moving Least Squares

An appealing definition of the surface over a noisy point cloud $\mathscr{P} \in \mathbb{R}^3$ is the zero set $S = \{\mathbf{x} | f(\mathbf{x}) = 0\}$ of the implicit function $f(\mathbf{x}) = \mathbf{n}(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{a}(\mathbf{x}))$ [Alexa et al. 2003], where $\mathbf{a}(\mathbf{x})$ is the weighted average of all points $\mathscr{P}$, and $\mathbf{n}(\mathbf{x})$ is determined by moving least squares. Usually, a Gaussian kernel (weighting function) $\theta(d) = e^{-d^2/h^2}, d = \|\mathbf{x} - \mathbf{p}\|$, is used, but other kernels work, too.

There are several variations of this simple definition, but for sake of clarity, we will stay with this basic one. Our new method works with all of its variants.

The problem with all of them is that the influence of cloud points $\mathbf{p} \in \mathscr{P}$ is based solely on the Euclidean distance $\|\mathbf{x} - \mathbf{p}\|$. This produces artifacts in the surface S (see Fig. 1).

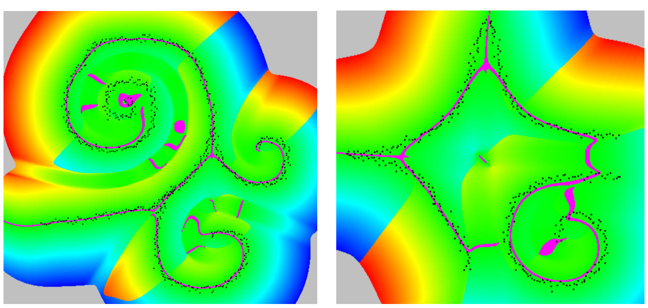

Figure 1: Surface artifacts produced by naïve kernel, shown here for a 2D point cloud, represented by black dots. Colors encode the sign and magnitude of $f(\mathbf{x})$. Magenta denotes $f(\mathbf{x}) \approx 0$, i.e., points that would be considered on the surface.

2 New Surface Definition

We propose to use a different kernel that is based on a much more reasonable proximity measure than the Euclidean distance. It approximates geodesic distances on S by utilizing a geometric proximity graph over $\mathscr{P}$. From the variety of different proximity graphs, we have examined the Delaunay graph DG($\mathscr{P}$), with additional pruning, and a variant of the sphere-of-influence graph SIG($\mathscr{P}$) [Jaromczyk and Toussaint 1992]. In such proximity graphs, nodes $\mathbf{p}$ and $\mathbf{q}$ are connected by an edge if some geometric predicate holds. The length of an edge is the Euclidean distance $\|\mathbf{p} - \mathbf{q}\|$ (or any other metric). The length of the path between two indirectly connected nodes is a function over its edges' lengths (not necessarily just the sum).

*e-mail: janklein@uni-paderborn.de
†e-mail: zach@cs.uni-bonn.de

Based on the proximity graph, we compute a *close-pairs shortest-paths* (CPSP) matrix, which is a subset of the all-pairs shortest-paths matrix. Under reasonable assumptions about the point cloud, this is sufficient and can be computed in time $O(n)$ for a 3D point cloud ($n = |\mathscr{P}|$).

We now define the new kernel as follows. Given some location $\mathbf{x}$, we compute its approximate nearest neighbor $\mathbf{p}^* \in \mathscr{P}$; using a simple k-d tree, this can be done in $O(\log^3 n)$ in 3D.

Starting from $\mathbf{p}^*$, we determine the distance $d(\mathbf{p}^*, \mathbf{p})$ for any $\mathbf{p} \in \mathscr{P}$ as the accumulated length of the shortest path from $\mathbf{p}^*$ to $\mathbf{p}$. This can be retrieved readily from the precomputed CPSP matrix.

Both the DG($\mathscr{P}$) and the SIG($\mathscr{P}$) can be computed efficiently in $O(n)$ time in 3D for uniformly sampled point clouds. Since the DG($\mathscr{P}$) yields a neighbor relation that also includes "long distance" neighborhoods, some shortest paths can "tunnel" through space that should really be a gap in the model. Therefore, we prune edges from DG($\mathscr{P}$) based on the computation of the local density of the point cloud.

In contrast, the SIG($\mathscr{P}$) must be "augmented" by edges, in order to prevent too many, unreasonably unconnected components. Therefore, we increase the influence of the nodes in the graph based on the radius of the k-th nearest neighbor.

3 Results and Conclusion

Our new surface definition over point clouds yields implicit functions, the zero sets of which are much closer to the original surface (see Fig. 2). This is verified by a series of tests for a number of different point clouds.

It is easy to implement and can be easily integrated with existing frameworks based on MLS, such as local polynomial approximations. The only additional data structure is a matrix of size $O(n)$ encoding all close-pairs shortest-paths in a proximity graph and a k-d tree. Both data structures can be constructed efficiently.

The asymptotic complexity of our new geodesic kernel is of the same order as the Euclidean kernel, and our experimental results show that $f(\mathbf{x})$ can be computed as fast as before.

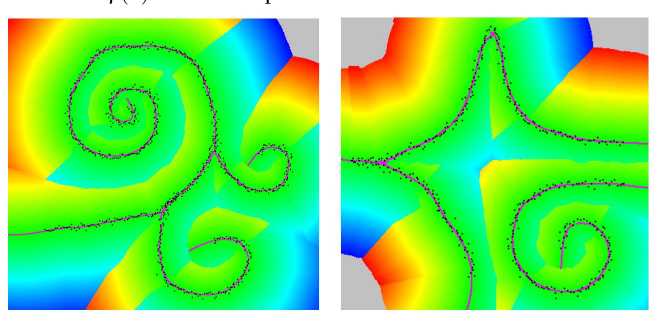

Figure 2: With our new definition, there are far fewer artifacts and there is much less bias (same point clouds as in Fig. 1).

References

ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND SILVA, C. T. 2003. Computing and rendering point set surfaces. *IEEE Trans. on Visualization and Comp. Graphics 9*, 1 (Jan.–Mar.), 3–15.

JAROMCZYK, J. W., AND TOUSSAINT, G. T. 1992. Relative neighborhood graphs and their relatives. In *Proc. of the IEEE*, vol. 80, 1502–1571.