Eurographics Symposium on Geometry Processing (2007)
Alexander Belyaev, Michael Garland (Editors)

Ridge Based Curve and Surface Reconstruction

Jochen Süßmuth[1] and Günther Greiner[1]

[1]Computer Graphics Group, University Erlangen-Nuremburg, Germany

Abstract

This paper presents a new method for reconstructing curves and surfaces from unstructured point clouds, allowing for noise in the data as well as inhomogeneous distribution of the point set. It is based on the observation that the curve/surface is located where locally the point cloud has highest density. This idea is pursued by a differential geometric analysis of a smoothed version of the density function. More precisely we detect ridges of this function and have to single out the relevant parts. An efficient implementation of this approach evaluates the differential geometric quantities on a regular grid, performs local analysis and finally recovers the curve/surface by an isoline extraction or a marching cubes algorithm respectively.

Compared to existing surface reconstruction procedures, this approach works well for noisy data and for data with strongly varying sampling rate. Thus it can be applied successfully to reconstruct surface geometry from time-of-flight data, overlapping registered point clouds and point clouds obtained by feature tracking from video streams. Corresponding examples are presented to demonstrate the advantages of our method.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Curve, surface, solid, and object representations

1. Introduction

Surface reconstruction is an important step in generating a three-dimensional virtual representation of real life objects. Typical applications include the digitalization of clay prototypes in car manufacturing, artifacts in archeology and human bodies in movie industry. Modern optical 3D scanners acquire millions of surface points from the object within minutes. These scanners typically produce a set of unordered points which is unsuitable for direct visualization and further processing.

The problem of reconstructing a surface from an unstructured set of points can be formulated as follows: Given a set of sample points $\mathcal{P} = \{\mathbf{p_0}, \ldots, \mathbf{p_{n-1}}\}$ which have been measured from an unknown surface $\mathcal{S} \in \mathbb{R}^3$, possibly with noise; find a surface (usually a polygonal mesh) $\mathcal{M}$ which interpolates resp. approximates the points from $\mathcal{P}$ and is topologically homeomorphic to the scanned object $\mathcal{S}$.

Some methods for shape acquisition, such as shape from motion or time-of-flight (TOF) sensors require algorithms that are robust to noise and missing samples. We propose a new method which can handle both situations, strong noise

Figure 1: *Surface mesh and outliers, reconstructed from a point cloud with 2.008.414 points and 10K random outliers. Data set courtesy of the U.C. Berkeley CAM Group.*

and highly non uniform sampling rates. Our method computes a smooth density function f from the input data points and uses ridge extraction methods to determine a connected surface of maximal local density which will be considered as the surface described by the noisy input samples.

Ridges of two- and three-dimensional data have been successfully used for feature detection and image analysis, face recognition and fingerprint analysis [AL00, LLS98, Mor95, Lin98], segmentation, registration and visualization of medical data [SAN*04, KTW06] and geometric modeling [SF04, OBS04] (and references therein). In this paper we will show that ridges are also a powerful instrument for locating surfaces in point clouds with noise.

This paper is organized as follows: In section 2, we describe related work and in section 3, an overview of our method is presented and the basic steps will be explained. In section 4, we describe our algorithm for reconstructing curves from noisy point clouds in $\mathbb{R}^2$. In section 5, we show how the algorithm can be extended to reconstruct surfaces from three-dimensional data. Results are then presented in section 6 and our method is compared to other algorithms for surface reconstruction from noisy point clouds. Future improvements to our algorithm are finally outlined in section 7 and in an appendix, the mathematics of curvature and ridges is outlined.

2. Related work

The problem of reconstructing a polygonal surface from an unstructured point set that had been produced by highly accurate 3D range sensors has gained a tremendous amount of attention during the past decades. Many excellent algorithms have been proposed for this problem.

Two classes of algorithms have been found to outperform most other methods: *implicit* and *Delaunay-based* methods.

Delaunay based methods

Delaunay based methods build upon a Delaunay tetrahedrization of the initial point cloud. An early Delaunay based approach are the $\alpha-$shapes by Edelsbrunner and Mücke [EM92]. Their algorithm removes all tetrahedra which have an enclosing circumsphere smaller than α. The surface is then obtained from the boundary triangles of the remaining tetrahedra. Another idea is to label each tetrahedron of an initial tetrahedrization either inside or outside. The (guaranteed closed) surface is then defined by all triangle-faces where inside and outside tetraheda meet. This idea first appeared in [Boi84] and was later developed further by the Powercrust [ACK01] and the Tight Cocone algorithms [DG03]. Both algorithms provide theoretical guarantees and have recently been extended to reconstruct noisy data [MAVdF05, DG04].

The advantage of most Delaunay based methods is that they produce watertight triangulations. The drawback is that they seek to interpolate the given point set, which is - especially in the presence of noise - not always desirable.

Implicit methods

Implicit or *zero-set* methods usually reconstruct the surface from a distance function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ which assigns each point in space a signed distance to the surface $\mathcal{S}$ from which the points have been sampled. A polygonal model of the object is then obtained by extracting the zero-set of f using a contouring algorithm. Thus the problem of reconstructing a surface from an unordered point cloud can be reduced to that of finding an appropriate function f which is zero at the sample points and non-zero everywhere else. Many different methods for computing the distance function f have been proposed. Hoppe et. al. [HDD*92] used a discrete function f. In [CBC*01], polyharmonic Radial Basis functions are fitted to the initial point set. More recently, Ohtake et. al. [OBA*03] fit piecewise quadratic functions locally to the data which they then blended globally. Other approaches include the moving least squares [SOS04, FCOS05], basis functions with local support [WSC06, MYC*01], Poisson based reconstruction [KBH06] and the method of Curless and Levoy [CL96] which is designed to construct a surface from several range images.

The drawback of implicit methods is that usually normal vectors have to be assigned to all points. These normal vectors can be computed by fitting a plane to the local neighborhood of a point, however, only if the point cloud is sampled sufficiently uniformly and free of noise.

The work which is most closely related to our approach is the algorithm presented in [HK06]. Hornung et. al. compute an unsigned distance function and extract the surface from the minimum cut of a weighted spatial graph structure.

Point cloud filtering

A popular approach to handle noisy point clouds is to filter the point cloud in a pre-processing step to remove outliers and reduce noise (see [SBS05, OBS05] and references therein). The surface may then be reconstructed from the noise reduced point cloud using one of the standard methods mentioned above.

3. Overview of the ridge based approach

A point $\mathbf{v} \in \mathcal{P}$ that has been sampled from a surface $\mathcal{S}$ by an imperfect device does not represent one distinct point υ on the surface $\mathcal{S}$. It is rather the center of a probability distribution describing the location of the point υ. Summing up all the point distributions gives a probability distribution of the location of the surface $\mathcal{S}$. For example, if the point set $\mathcal{P}$ is a noisy sampling of the circle in $\mathbb{R}^2$, the corresponding probability distribution is a crater like set and the initial

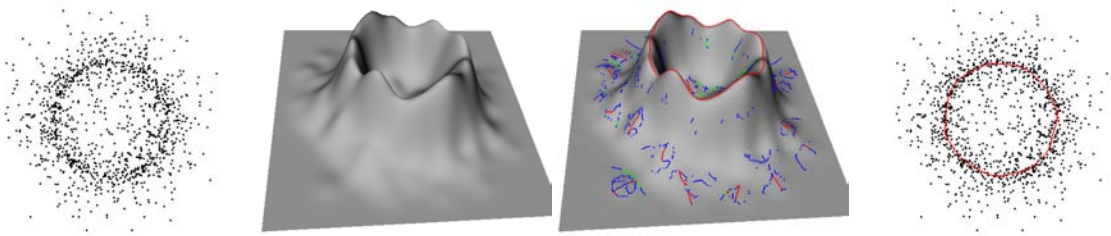


Figure 2: *The algorithm in $\mathbb{R}^2$: (left) noisy point set. (middle left) the density function f generated by placing Gaussian kernels at the sample points. (middle right) the ridges of f; red lines mark ridges, green lines valleys and blue lines spurious ridge lines. (right) the ridges projected back onto the plane of the point set.*

contour can be recovered approximately as the ridge of this crater (see Fig. 2)

Our method can be separated into three distinct steps. In the first step, we consider a smooth version of the density distribution of the point set.

In the second step, we compute the ridges of this density function which can be interpreted as the curve/surface we are looking for. Ridges of a bi-variate function f can be characterized geometrically as follows. Consider the graph of the density function f as a surface in $\mathbb{R}^3$. Then a surface point $(x,y,f(x,y))$ belongs to the ridge, provided that the gradient of f and the direction of largest absolute curvature are perpendicular and the curvature at (x,y) is negative (details and mathematical definitions are given in the appendix).

Due to noise and outliers, the density functions will exhibit local ridges, points that formally fulfill the ridge condition, but do not belong to the principal ridge. The removal of these points requires an additional cleaning step. The cleaning is done by starting at the absolute maximum and tracing the ridge from there. Thus we assume that the surface to be reconstructed is one connected patch.

4. Reconstructing curves in $\mathbb{R}^2$

In this section, we describe how our method can be used to reconstruct curves from two-dimensional scattered points. The method will then be extended to extract surfaces from three-dimensional sample points in the next section.

The first step of our algorithm consists of defining a smooth density function f which results from the convolution of the discrete Dirac distribution of the ($p = 2$-dimensional) point cloud with a Gaussian kernel:

$$f(\mathbf{x}) = \sum_{\mathbf{v}\in\mathcal{P}} (\delta_v * G_\sigma)(\mathbf{x}) = \frac{1}{(2\pi)^{2/p}\sigma^2} \sum_{\mathbf{v}\in\mathcal{P}} e^{\left(-\frac{\|\mathbf{x}-\mathbf{v}\|^2}{2\sigma^2}\right)}, \quad (1)$$

the first and second derivative of f are then given by

$$\nabla f(\mathbf{x}) = \frac{-1}{(2\pi)^{2/p}\sigma^4} \sum_{\mathbf{v}\in\mathcal{P}} (\mathbf{x}-\mathbf{v}) \cdot e^{\left(-\frac{\|\mathbf{x}-\mathbf{v}\|^2}{2\sigma^2}\right)} \quad (2)$$

$$\mathrm{H}(f)(\mathbf{x}) = \frac{1}{(2\pi)^{2/p}\sigma^6} \sum_{\mathbf{v}\in\mathcal{P}} \left((\mathbf{x}-\mathbf{v})\otimes(\mathbf{x}-\mathbf{v}) - \sigma^2 \cdot Id\right) \cdot e^{\left(-\frac{\|\mathbf{x}-\mathbf{v}\|^2}{2\sigma^2}\right)}. \quad (3)$$

In a second step, we compute the ridges of the density function f. A point $\mathbf{p}$ on f lies on a ridge if f has a local maximum in the direction of maximum curvature at $\mathbf{p}$. (i.e. $\langle \nabla f, \mathbf{e_1} \rangle = 0$ and $\kappa_1 < 0$, cf. Appendix). To compute the zero set of $\langle \nabla f, \mathbf{e_1} \rangle$, we sample ∇f and $\mathrm{H}(f)$ on a regular grid $\mathcal{G}$ with grid size δ. At each vertex we perform an eigenvector/-value analysis to determine κ_1 and $\mathbf{e_1}$. Since the eigenvectors $\mathbf{e_1}$ are only determined up to a scaling factor (if $\mathbf{e_1}$ is an eigenvector of M, so is $-\mathbf{e_1}$ also an eigenvector of M), we need to establish a consistent orientation of the $\mathbf{e_1}$'s in each cell. This can be obtained by retaining one of the cells $\mathbf{e_1}$ and inverting the remaining $\mathbf{e_{1,i}}$ if they point in the opposite direction, i.e. if $\langle \mathbf{e_1}, \mathbf{e_{1,i}} \rangle < 0$ (for more sophisticated methods to solve the orientation problem see [Mor95, CFPR06]). A simple contouring algorithm is then applied to compute the zero crossings of $\langle \nabla f, \mathbf{e_1} \rangle$ in the current cell. As you can see from figure 2(c), this approach does not only detect ridge and valley lines, but also some so called *spurious ridges* (blue lines) in regions where the curvature direction varies.

Finally, the curve described by the noisy point set is extracted by tracing the ridge line starting at the line segment where the density f is maximal, following the line segments where κ_1 is negative; valleys and spurious ridges are thereby removed.

The steps of the algorithms for reconstructing curves from two-dimensional point clouds are depicted in figure 2. Figure 2(a) shows a point set sampled from a circle with noise. In figure 2(b), the density function f, sampled on a regular grid, is shown. Figure 2(c) shows the ridges, valleys and spurious ridges of f as computed and classified in the second step of our algorithm. The reconstructed curve together with the input data is shown in figure 2(d).

5. Reconstructing surfaces in $\mathbb{R}^3$

The extension of our algorithm to reconstruct surfaces from point clouds in $\mathbb{R}^3$ is straightforward. Hence, we will focus

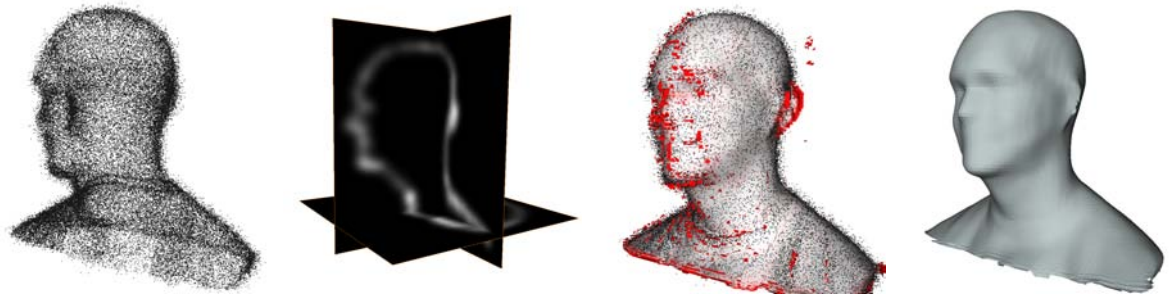


Figure 3: *The algorithm in $\mathbb{R}^3$: (left) a noisy point set. (middle left) the 3D density function f. (middle right) all ridges of f, including valleys and spurious ridges. (right) surface reconstructed by tracing the maximum ridge.*

on some methods to improve the basic algorithm presented in the previous section. Figure 3 shows the steps analogously to the method in two dimensions.

The exact computation of the gradient ∇f and the Hessian matrix $\mathrm{H}(f)$ requires us to sum up over all kernel functions, one for each point in $\mathcal{P}$. Computing these sums for each cell of a reasonable grid with dimension 500^3 leads to an enormous computational effort. To reduce computation time we propose to only sum up over the kernel functions which have a reasonable impact on the computation of f, ∇f and $\mathrm{H}(f)$. Therefore we omit all kernel functions $f_{\mathbf{v}}$ that would add values less than a threshold ε to the density function, its gradient and its Hessian. (The ε used for our implementation ranges from 0.0005 for the *angel* data set to 0.1 for the *TOF* data set). Given ε and the variance σ of the Gaussian kernels, a range of interest d can be computed. In order to evaluate the function and its derivatives at a particular grid point $\mathbf{x}$, we estimate all points $\mathbf{p_j} \in P$ that lie within distance d from $\mathbf{x}$. This search can be efficiently implemented by using a *kD*-tree in which the point cloud P is stored. The density f is then computed from

$$f(\mathbf{x}) = \sum_{\{\mathbf{v}\in\mathcal{P}\ ,\ \|\mathbf{x}-\mathbf{v}\|^2\leq d^2\}} f_{\mathbf{v}}(\mathbf{x}).$$

The formulas (2) and (3) for computing ∇f and $\mathrm{H}(f)$ are modified accordingly.

The approach presented in the previous section evaluates the function uniformly on a regular grid, even in regions that are of little interest because they are far away from being traversed by a ridge surface. The subsequent ridge extraction step however, only considers the cells which are traversed by the maximum ridge. We propose to start the iso-surface extraction at a *significant* local maximum of f. To find such a local maximum we sample f at a low resolution and determine the cell containing the largest value of f.

We then use a modified marching cubes [LC87] algorithm which takes care of the orientation of the curvature vectors $\mathbf{e_1}$ to extract the zero set of $\langle \nabla f, \mathbf{e_1} \rangle$. The algorithm starts at the previously detected maximum. Only cells containing the zero set are visited by pushing solely the appropriate neighboring cells onto a stack. To skip valleys and spurious ridges (figure 3(c)), we only add those triangles to the reconstruction that are connected to the extracted ridge so far.

The variable σ should be chosen such that the sum of two neighboring Gaussian kernels forms one common maximum. Let the distance between two points be d. The radial kernel functions centered at the points do then form one common maximum, if $\sigma > (d/\sqrt{2})$. From this observation we can derive a rough rule for choosing $\sigma > (\rho/\sqrt{2})$ if the spacing ρ between two neighboring points in $\mathcal{P}$ is known.

5.1. Anisotropic kernel functions

Using radial kernel function works well for point sets that are disturbed by a constant level of noise. Though it has problems at sharp features and areas where the object is thin. Figure 4 shows an example where the reconstruction using spherical kernel functions fails. The ears of the bunny model are quite thin. If σ is too large, the density function f will be maximal in between the actual surface sheets. This results in one ridge for the ear instead of one ridge for either side. To overcome this problem, we recommend using kernel functions that describe the probability distribution of the points better. Let the neighborhood $N_{\mathbf{v}}$ be the $k-$nearest neighbors of $\mathbf{v}$ (k typically lies between 20 and 100). If the neighborhood $N_{\mathbf{v}}$ is completely flat then the probability that $\mathbf{v}$ was actually sampled from that plane should be high. To get such an anisotropic probability distribution, we compute the covariance matrices

$$\mathbf{C_v} = \sum_{\mathbf{p_j}\in N_{\mathbf{v}}} (\mathbf{p_j}-\mathbf{v}) \otimes (\mathbf{p_j}-\mathbf{v})$$

of the neighborhoods $N_{\mathbf{v}}$ for each point $\mathbf{v} \in \mathcal{P}$. The anisotropic probability distribution for the point $\mathbf{v}$ is then given as:

$$\tilde{f}_{\mathbf{v}}(\mathbf{x}) = \frac{\det(\mathbf{C_v^{-1}})}{(2\pi)^{2/3}\sigma^2} \cdot e^{\left(-\frac{(\mathbf{x}-\mathbf{v})^T \mathbf{C_v^{-1}} (\mathbf{x}-\mathbf{v})}{2\sigma^2}\right)}$$

$$\nabla \tilde{f}_{\mathbf{v}}(\mathbf{x}) = -\frac{\mathbf{C}_{\mathbf{v}}^{-1}(\mathbf{x}-\mathbf{v})}{\sigma^2} \cdot \tilde{f}_{\mathbf{v}}(\mathbf{x})$$

$$\tilde{\mathrm{H}}(f)_{\mathbf{v}}(\mathbf{x}) = \left(\frac{\left(\mathbf{C}_{\mathbf{v}}^{-1}(\mathbf{x}-\mathbf{v})\right) \otimes \left(\mathbf{C}_{\mathbf{v}}^{-1}(\mathbf{x}-\mathbf{v})\right)}{\sigma^4} - \frac{\mathbf{C}_{\mathbf{v}}^{-1}}{\sigma^2} \right) \cdot \tilde{f}_{\mathbf{v}}(\mathbf{x}).$$

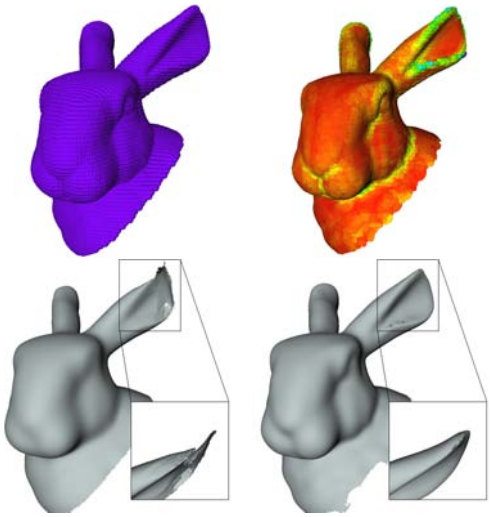


Figure 4: *Head of the stanford bunny; (1) radial kernel functions $f_{\mathbf{v}}$ (2) anisotropic kernel functions $\tilde{f}_{\mathbf{v}}$ colored according to their anisotropy; (3) reconstruction using radial kernels; (4) reconstruction using anisotropic kernels.*

The first row of figure 4 visualizes the kernel functions that were used to reconstruct the bunny model in the lower row. The left image shows isotropic kernel function while the right image shows kernel functions that have been scaled and rotated according to the neighborhoods of the points.

Figure 5: *Results for point sets acquired from different devices: (1) Gall bladder; (2) TOF Scene; (3) Hand; (4) Disposable tip.*

name	# of pts	σ	grid	time (sec)
gall bladder	1.904	0.04	100^3	6.1
TOF scene	230.400	0.03	100^3	653.1
hand ridges	38.219	0.02	200^3	172.9
hand valleys	38.219	0.03	50^3	22.1
disposable tip	70.052	0.01	100^3	11.2
angel	2.018.414	0.002	500^3	1089.4

Table 1: *Computation time and parameters for the data sets shown in figure 5 and figure 1*

6. Results

We have employed our algorithm to reconstruct the surfaces from point clouds which were generated using different methods for shape acquisition. The reconstructed surfaces and the input point clouds are shown together in figure 5. The parameters used for reconstruction and the time of computation required to reconstruct the models on a Pentium IV 3.4 GHz are listed in table 1.

In figure 5(1), the model of a gall bladder is shown. The point cloud has been generated from a video sequence captured by an endoscope using shape from motion technique. The point cloud suffers from both, strong noise and varying sample density. Figure 5(2) shows a cooling fan in a room corner. The point cloud has been generated by registering 10 overlapping depth maps acquired by a time-of-flight camera. Due to the distortion introduced by the time-of-flight camera, the single depth maps are not congruent and the point cloud therefore contains multiple tiers. The hand in figure 5(3) was scanned by Polhemus *FastSCAN*. The polygonal surface model was reconstructed using relative small grid spacing and anisotropic kernel functions to recover small features and thin areas. Also depicted are the valleys of the density function (red). Valleys occur where the density of a point cloud is minimal; hence they can be understood as a medial structure of the object. Figure 5(4) shows a fraction of a disposable tip that has been scanned using white light interferometry. The dataset is composed of 7 individual range scans and contains natural noise and non-uniform sample density.

Figure 1 shows a mesh reconstructed from a point cloud with 2 million points and 10.000 randomly added outliers. The point samples are very non uniformly distributed and the point cloud contains natural noise. Our reconstruction captures even fine detail, yet it leaves holes in areas where no samples were available.

To demonstrate the noise robustness of our method we

Figure 6: *Comparison to other reconstruction algorithms. From left to right: Initial point cloud with 1 percent gaussian noise; reconstruction using* Powercrust *[ACK01],* Robust Cocone *[DG04],* QPoly *[OBS05] and our method.*

have tested it with several synthetic data sets (figure 7). Gaussian noise with a variance of x percent of the point cloud's diagonal length was added to the male data set. The noisy point clouds were then reconstructed using different smoothing parameters σ. As you can see from the bottom row in figure 7, we can still produce reasonable results even if there is an enormous amount of noise. You can also see that the variable σ used for smoothing the point distribution does not only depend on the noise level but also on the sample density of the point cloud. The denser the point cloud is sampled the smaller the sigma may be chosen.

Comparison with other algorithms for noisy point clouds

We have further tested our method against some popular surface reconstruction algorithms for point clouds without normals. The results are shown in figure 6.
The input point cloud does clearly not fulfill the sample requirements of the Powercrust algorithm [ACK01], the surface could therefore not be reconstructed appropriately. Some of the problems associated with noise have been resolved in [MAVdF05] and [KSO04]. But because of the interpolation property of these methods the resulting surfaces still look very crinkled.
The Robust Cocone algorithm [DG04] is able to restore a rough approximation of the initial shape. However the method tends to increase the volume of the reconstructed surface.
The results produced by the QPoly algorithm [OBS05] are better than the previous ones, yet the surface is very coarse and not two-manifold.
Our algorithm yields the best results. The reconstructed surface is two-manifold with boundaries and we could recover more detail than the methods discussed above.

The method of Hornung et. al. for reconstructing surfaces from point clouds without normals in [HK06] looks quite promising in terms of results and computation time. Yet it is unclear how robust their algorithm is to large magnitude noise and outliers. Another limitation of their approach is the fact that they can only reconstruct closed surfaces. Surface sheets like the two shown in the top row of figure 5 are not appropriate for being reconstructed as closed surfaces.

Conclusion and future work

We have presented a new method for reconstructing curves and surfaces from unorganized point clouds with noise and outliers. We showed that extracting the ridges of a density function defined by the point set is a simple and efficient way to reconstruct an approximating curve/surface from scattered data. Our algorithm does not require any point normal information and can successfully handle noise which is more than one order stronger then Delaunay based methods for noisy point clouds. We think that our algorithm is therefore an interesting option for reconstructing point clouds with a large amount of noise that have been generated by inaccurate techniques for shape acquisition.

Our method is currently not able to reconstruct sharp crest lines from sparsely sampled point sets. We think that an a priory detection of crest lines [DVVR07] and a special treatment for crest points (i.e. using special kernel functions, e.g. superquadrics, for these points) would improve the ability to reconstruct sharp features.

Using an adaptive octree for the computation of the density function and the subsequent ridge extraction could further help to close holes due to missing samples in the point cloud (cf. figure 1). As a positive side effect it would also improve the speed and memory requirements of our algorithm.

Acknowledgments

We would like to thank Nina Amenta, Tamal Dey and Yutaka Ohtake for making their algorithms publicly available. We would also like to thank SFB 603/TP B6 for the TOF and the gallbladder data set, Cyberware for the male data set, Stanford 3D Scanning Repository for the bunny data set, James O'Brien for the angel data set and Polhemus *FastSCAN* for the hand data set.

This work was supported by the German Research Foundation (DFG) under grant SFB 603/TP A2. Only the authors are responsible for the content.

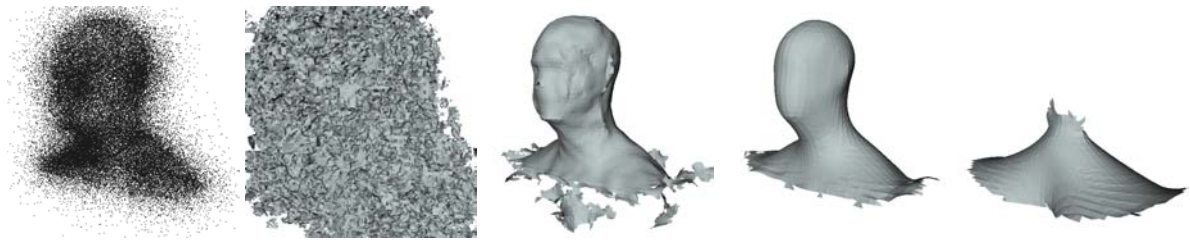
References

[ACK01] AMENTA N., CHOI S., KOLLURI R. K.: The power crust. In *SMA '01: Proceedings of the sixth ACM*

(a) 0.5 percent Gaussian distributed noise.

(b) 2 percent Gaussian distributed noise.

(c) 4 percent Gaussian distributed noise.



(d) 8 percent Gaussian distributed noise.

Figure 7: *Choice of the variable σ; in each row from left to right: noisy point cloud with 63393 points, reconstructed surfaces with $\sigma = 1\%$, $\sigma = 2\%$, $\sigma = 4\%$ and $\sigma = 8\%$ of the point clouds bounding box diagonal.*

symposium on Solid modeling and applications (2001), ACM Press, pp. 249–266.

[AL00] ALMANSA A., LINDEBERG T.: Fingerprint enhancement by shape adaptation of scale-space operators with automatic scale-selection. *IEEE Transactions on Image Processing 9*, 12 (2000), 2027–2042.

[Boi84] BOISSONNAT J.-D.: Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics 3*, 4 (1984), 266–286.

[CBC*01] CARR J. C., BEATSON R. K., CHERRIE J. B., MITCHELL T. J., FRIGHT W. R., MCCALLUM B. C., EVANS T. R.: Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH '01: ACM SIGGRAPH 2001 Papers* (New York, NY, USA, 2001), ACM Press, pp. 67–76.

[CFPR06] CAZALS F., FAUGÈRE J.-C., POUGET M., ROUILLIER F.: The implicit structure of ridges of a smooth parametric surface. *Comput. Aided Geom. Des. 23*, 7 (2006), 582–598.

[CL96] CURLESS B., LEVOY M.: A volumetric method

for building complex models from range images. In *SIGGRAPH '96: ACM SIGGRAPH 1996 Papers* (1996), pp. 303–312.

[DG03] DEY T. K., GOSWAMI S.: Tight cocone: a watertight surface reconstructor. In *SM '03: Proceedings of the 8th ACM symposium on Solid modeling and applications* (New York, NY, USA, 2003), ACM Press, pp. 127–134.

[DG04] DEY T. K., GOSWAMI S.: Provable surface reconstruction from noisy samples. In *SCG '04: Proceedings of the 20th annual symposium on Computational geometry* (New York, NY, USA, 2004), ACM Press, pp. 330–339.

[DVVR07] DEMARSIN K., VANDERSTRAETEN D., VOLODINE T., ROOSE D.: Detection of closed sharp edges in point clouds using normal estimation and graph theory. *Computer-Aided Design 39*, 4 (2007), 276–283.

[Ebe96] EBERLY D. H.: *Ridges in Image and Data Analysis*. Kluwer Academic, Dordrecht, Nederlands, 1996.

[EM92] EDELSBRUNNER H., MÜCKE E. P.: Three-dimensional alpha shapes. In *VVS '92: Proceedings of the 1992 workshop on Volume visualization* (New York, NY, USA, 1992), ACM Press, pp. 75–82.

[FCOS05] FLEISHMAN S., COHEN-OR D., SILVA C. T.: Robust moving least-squares fitting with sharp features. *ACM Trans. Graph. 24*, 3 (2005), 544–552.

[FP98] FURST J., PIZER S. M.: Marching optimal-parameter ridges: An algorithm to extract shape loci in 3d images. In *Proceedings of MICCAI'98* (London, UK, 1998), Springer-Verlag, pp. 780–787.

[Har83] HARALICK R.: Ridges and valleys on digital images. *CVGIP 22*, 1 (1983), 28–38.

[HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. *Computer Graphics 26*, 2 (1992), 71–78.

[HK06] HORNUNG A., KOBBELT L.: Robust reconstruction of watertight 3d models from non-uniformly sampled point clouds without normal information. In *Proc. Symposium on Geometry Processing* (2006), pp. 41–50.

[KBH06] KAZHDAN M., BOLITHO M., HOPPE H.: Poisson surface reconstruction. In *Proc. Symposium on Geometry Processing* (2006), pp. 61–70.

[KSO04] KOLLURI R., SHEWCHUK J., O'BRIEN J.: Spectral surface reconstruction from noisy point clouds. In *Proc. Symposium on Geometry Processing* (2004), pp. 11–21.

[KTW06] KINDLMANN G., TRICOCHE X., WESTIN C.-F.: Anisotropy creases delineate white matter structure in diffusion tensor MRI. In *Proceedings of MICCAI'06* (Copenhagen, Denmark, 2006), LNCS 3749.

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: ACM SIGGRAPH 1987 Papers* (New York, NY, USA, 1987), ACM Press, pp. 163–169.

[Lin98] LINDEBERG T.: Edge detection and ridge detection with automatic scale selection. *Int. J. Comput. Vision 30*, 2 (1998), 117–156.

[LLS98] LÓPEZ A. M., LUMBRERAS F., SERRAT J.: Creaseness from level set extrinsic curvature. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume II* (London, UK, 1998), Springer-Verlag, pp. 156–169.

[MAVdF05] MEDEROS B., AMENTA N., VELHO L., DE FIGUEIREDO L. H.: Surface reconstruction from noisy point clouds. In *Proc. Symposium on Geometry Processing* (2005), pp. 53–62.

[Mor95] MORSE B. S.: *Computation of object cores from grey-level images*. PhD thesis, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

[MYC*01] MORSE B. S., YOO T. S., CHEN D. T., RHEINGANS P., SUBRAMANIAN K. R.: Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *Proceedings of SMI '01* (Washington, DC, USA, 2001), IEEE Computer Society, p. 89.

[OBA*03] OHTAKE Y., BELYAEV A., ALEXA M., TURK G., SEIDEL H.-P.: Multi-level partition of unity implicits. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), ACM Press, pp. 463–470.

[OBS04] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: Ridge-valley lines on meshes via implicit surface fitting. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), ACM Press, pp. 609–612.

[OBS05] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: An integrating approach to meshing scattered point data. In *SPM '05: Proceedings of the 2005 ACM symposium on Solid and physical modeling* (New York, NY, USA, 2005), ACM Press, pp. 61–69.

[SAN*04] STAAL J., ABRÀMOFF M. D., NIEMEIJER M., VIERGEVER M. A., VAN GINNEKEN B.: Ridge-based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging 23*, 4 (2004), 501–509.

[SBS05] SCHALL O., BELYAEV A., SEIDEL H.-P.: Robust filtering of noisy scattered point data. In *IEEE/Eurographics Symposium on Point-Based Graphics* (Stony Brook, New York, USA, 2005), Pauly M., Zwicker M., (Eds.), Eurographics Association, pp. 71–77.

[SF04] STYLIANOU G., FARIN G.: Crest lines for surface segmentation and flattening. *IEEE Transactions on Visualization and Computer Graphics 10*, 5 (2004), 536–544.

[SOS04] SHEN C., O'BRIEN J. F., SHEWCHUK J. R.: Interpolating and approximating implicit surfaces from

polygon soup. In *Proceedings of ACM SIGGRAPH 2004* (New York, NY, USA, 2004), ACM Press, pp. 896–904.

[WSC06] WALDER C., SCHÖLKOPF B., CHAPELLE O.: Implicit surface modelling with a globally regularised basis of compact support. In *Proceedings of Eurographics 2006* (Oxford, 2006), Blackwell, pp. 635–644.

Appendix: Mathematics of ridges

In topography ridges are defined as watersheds that separate between regions where water flows in different directions. Although heavily studied for more than 150 years, researcher still do not agree on a precise mathematical definition of ridges. Intuitively, ridges of a two-dimensional function $f(x,y) = z$ can be understood as the connected locus of maximum value.

In order to give a rigorous mathematical definition for ridges we have to fix some notation. The gradient and Hessian of a smooth scalar function $f(x_1,...,x_d)$ are given by

$$\nabla f = \begin{pmatrix} \partial_1 f \\ \vdots \\ \partial_d f \end{pmatrix}, \; \mathrm{H}(f) = \begin{pmatrix} \partial_{11} f & \cdots & \partial_{1d} f \\ \vdots & \ddots & \vdots \\ \partial_{d1} f & \cdots & \partial_{dd} f \end{pmatrix}$$

Here $\partial_i f$ and $\partial_{ij} f$ denote the partial derivatives of first and second order respectively. To determine curvature and curvature direction we need the first and second fundamental form of the d-dimensional hyper surface $(x_1,\ldots,x_d, f(x_1,\ldots x_d))$ embedded in $\mathbb{R}^{d+1}$. These are given by $d \times d$ matrices.

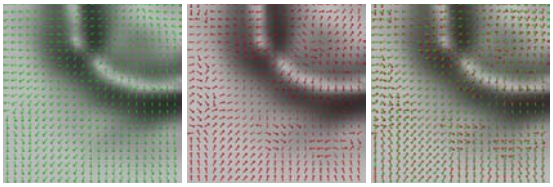


Figure 8: *Gradient ∇f (left) and direction of maximum curvature $\mathbf{e_1}$ (middle) of a bivariate function f. The rightmost image shows both vector fields.*

$$\mathbf{I}_f = \left(g_{ij}\right) \quad \text{with} \quad g_{ij} = \begin{cases} 1+\partial_i f^2 & \text{if} \quad i = j \\ \partial_i f \partial_j f & \text{if} \quad i \neq j \end{cases} \tag{4}$$

$$\mathbf{II}_f = \left(h_{ij}\right) = \frac{1}{\sqrt{(\det(\mathbf{I}_f))}} \left(\partial_{ij} f\right) \tag{5}$$

A principal curvature κ_i and the corresponding principal curvature direction $\mathbf{e_i}$ satisfy the condition

$$\mathbf{II}_f \mathbf{e_i} = \kappa_i \mathbf{I}_f \mathbf{e_i} \tag{6}$$

Thus one has to perform an eigenvalue/-vector analysis of the *Weingarten map* ${\mathbf{I}_f}^{-1}\mathbf{II}_f$. There are precisely d pairwise orthonormal principal curvature directions $\mathbf{e_i}$ $(i = 1\ldots d)$ with corresponding principal curvatures κ_i, $(i = 1\ldots d)$. We assume that $|\kappa_1| \geq |\kappa_2| \geq \cdots \geq |\kappa_d|$.

Definition: A point $\mathbf{x}$ is a *ridge point* provided that at $\mathbf{x}$ we have

$$\kappa_1 < 0 \quad \text{and} \quad \mathbf{e_1} \perp \nabla f \tag{7}$$

The direction of largest negative curvature is also called the *direction of maximal concavity* (see [Har83]).

Equation 7 states that the function $f : \mathbb{R}^d \to \mathbb{R}$ has a ridge at $\mathbf{x}$ iff f has a local maximum in the direction of maximum curvature $\mathbf{e_1}$ at $\mathbf{x}$. Analogously, all points at which f takes on a local minimum in direction $\mathbf{e_1}$ are called *valley points*.

The points which satisfy equation 7 form a hypersurface of $\mathbb{R}^n$, i.e. the ridges of two-dimensional functions are lines and the ridges of three-dimensional functions are surfaces. Higher dimensional ridges (d-k dimensional hypersurfaces) can be defined by extending equation 7 such that the gradient ∇f of a function f is required to be orthogonal to the first k curvature directions [FP98, Ebe96]:

$$\kappa_i < 0 \quad \text{and} \quad \mathbf{e_i} \perp \nabla f, \quad i = 1\ldots k.$$

In the two-dimensional case, the determination of ridges can be simplified. At a ridge point we have $\mathbf{e_1} \perp \nabla f$ hence $\mathbf{e_2} \parallel \nabla f$. Thus $\det(\mathbf{II}_f \nabla f, \mathbf{I}_f \nabla f) = 0$ which simplifies to

$$\sqrt{\det(\mathbf{I}_f)}\det(\sqrt{\det(\mathbf{I}_f)}\mathbf{II}_f \nabla f, \nabla f) = 0. \tag{8}$$

Thus we obtain

$$(\partial_{11} f \partial_1 f + \partial_{12} f \partial_2 f)\partial_2 f - (\partial_{21} f \partial_1 f + \partial_{22} f \partial_2 f)\partial_1 f = 0.$$

Thus ridges satisfy the relation

$$(\partial_{11} f - \partial_{22} f)\partial_1 f \partial_2 f + \partial_{12} f(\partial_2 f 2 - \partial_1 f 2) = 0. \tag{9}$$

The problem with this simple characterization is that not only ridge points but many others also satisfy this relation. All points where the gradient is perpendicular to one principal curvature direction, regardless of whether it is minimal or maximal, positive or negative. An extraction based on this formula requires a more detailed cleaning process to separate spurious crest lines from ridges.

For d-dimensional functions a similar characterization is possible. More precisely, at a ridge point the determinant

$$\det(\mathbf{II}_f^{d-1} \nabla f, \ldots \mathbf{II}_f \nabla f, \nabla f) \tag{10}$$

is zero. Again this condition is satisfied not only for ridge points, but also if ∇f is perpendicular to an arbitrary principal curvature direction

Note that the ridge formulation used in this paper is different to another definition of ridges frequently used for detecting creases and sharp features on two-dimensional manifolds embedded in $\mathbb{R}^3$, which classifies surface points as ridge points iff the curvature κ_i along one of the points principal curvature directions $\mathbf{e_j}$ is maximal [OBS04].