Distance Field Computation for Geological Slab Surface Data Sets

Marek Vančo Bernd Hamann Oliver Kreylos
Institute for Data Analysis and Visualization
Department of Computer Science
University of California Davis
Davis, CA, USA

Magali I. Billen Margarete A. Jadamec
Department of Geology
University of California Davis
Davis, CA, USA

Abstract

The three-dimensional shapes of tectonic plates that sink into the Earth's mantle (slabs) are the starting point for a range of geoscience studies, from determining the forces driving the motion of tectonic plates, to potential seismic and tsunami hazards, to the sources of magmas beneath active volcanos. For many of these applications finite element methods are used to model the deformation or fluid flow, and therefore the input model parameters, such as feature geometries, temperature or viscosity, must be defined with respect to a smooth, continuous distance field around the slab.

In this paper we present a framework for processing sparse and noisy seismic data (earthquake locations), defining the shape of the slab and computing a continuous distance function on a mesh with variable node spacing. Due to the inhomogeneous volumetric distribution of earthquakes within the slab and significant inaccuracies in the locations of earthquakes occurring hundreds of kilometers below the Earth's surface, the seismicity data set is extremely noisy and incomplete. Therefore the preprocessing is the major part of the framework consisting of several steps including a point based smoothing procedure, a powerful method to use other observational constraints on slab location (e.g., seismic tomography or geologic history) to extend of the slab shape beyond earthquake data set and continuous resampling using moving least squares method. For the preprocessed point data we introduce approaches for finding the three-dimensional boundary of the slab and a subdivision of the slab into quadric implicit polynomials. The resulting distance field is then compiled from distances to the piecewise continuous approximation of the slab and distances to slab boundary.

1 Introduction

The Earth's surface is divided into tectonic plates whose motion is driven by thermo-chemical convection within the underlying mantle [Tac00]. Over long periods of time (millions of years) the mantle deforms as a fluid, i.e., its motion can be described as flow of a viscous material. Tectonic plates are formed and diverge at mid-ocean spreading centers, slide past one another along transform boundaries or converge at subduction zones. When two tectonic plates converge, one plate sinks asymmetrically beneath the other plate driven by gravitational instability of the cold, dense tectonic plate. The portion of a tectonic plate that has subducted into the mantle is referred to as the *slab*. The thermal signature of tectonic plates and slabs is approximately 100 km thick, and the subducted portions of slabs can extend hundreds to thousands of kilometers laterally and sink to depths of up to 2890 km (core-mantle boundary). The location of the slab in the mantle is discerned from the locations of earthquakes within the slab, which extend to depths of 670 km [IB77], and from seismic tomography, which detects three-dimensional regions of low (warm) and high (cold) seismic velocity [vdHESN91, Rom03] throughout the mantle. The shape of slabs in the mantle combined with mechanical analysis (analytic or numerical modeling) can provide information on the state of stress in the slab (e.g., compressional versus tensional), the history of subduction (e.g., rate and direction), and interaction of the slab with mantle structure (e.g., lay-

ers with different material properties) or with large-scale mantle deformation driven by heating from within [VR84, wZ90, NMHJ96, BGS03, JB10, MGK06]. Therefore, an accurate determination of three-dimensional slab shape is the starting point for several Earth science applications.

The primary source of information about slab location is seismicity. Therefore, the shape of slabs must be reconstructed from inhomogeneously distributed and noisy observations. In publicly available data sets, ten thousand earthquake locations may be known for a slab of size of 2000×700 km^2. However, the error in the location of individual earthquakes may exceed 5–50 km depending on when the earthquake occurred (older events having larger errors) and its depth (deeper events being more difficult to locate) [SPP00]. However, in some regions, earthquake relocation studies have resulted in data sets with location errors of less than one to five kilometers [ZTS*03, RH02]. In addition, even if the locations of earthquakes are known exactly, they occur in a volume of the slab with thickness of 10 to 50 km, rather than along a single surface.

Unlike engineered structures, whose shape is often constructed from standard geometric objects (e.g., planes, cylinders), because slabs deform as stiff viscous sheets, their shape can vary substantially on both short and long length-scales without a simple relationship to standard geometric objects. Previous attempts to reconstruct three-dimensional slab shapes have relied on approximating the entire slab structure using the analytic solution for a thin elastic sheet [Chi91], or smooth splines [GS98b, SA06]. However, both these approaches enforce a long wavelength smooth shape on the slab structure that can miss shorter-length-scale features, or can introduce short wavelength undulations that are unrelated to the actual slab shape. As the slab structure is often used to create input for analytic or numerical models (e.g., thermal and mechanical structure for a finite element model of deformation), errors in the slab structure can lead to errors in the predicted deformation [BGS03, JB10]. In addition, assignment of properties (e.g., temperature or viscosity) to a finite element grid, which being spatially related to the location of the slab surface (e.g., by Euclidean distance), also requires a method to create a smoothly varying field without aliasing on the grid and a method for defining the three-dimensional slab edge.

The main focus of the paper is a case study of complex point-based methods for processing noisy data in the context of generating a continuous distance field function from the data. In Section 2 we describe our preprocessing methods, i.e., point-based smoothing, extending the slab beyond earthquake centers (using sparely sampled contour and/or boundary lines), and resampling the point set based on a subdivision using moving least squares (blended with elliptic weight functions resulting in a smooth continuous discrete approximation of the original data). Finally, the slab boundary is determined. Section 3 describes steps necessary for computation of the distance field. The point set is segmented into quadric patches, which are used as a continuous approximation of the underlying surface for distance computation. For points outside the slab domain, a point-curve distance calculation is performed. Section 4 provides results obtained with our method when applied to a digitized slab beneath southern Alaska. In Section 5 we summarize the main points of the paper and present ideas for potential future research.

2 Data preprocessing

We consider an input point set $\mathcal{X} = \{\mathbf{P}_1, \ldots, \mathbf{P}_n\}$ that consists of three-vectors defined as (*latitude*, *longitude*, *radius*) with the origin in the Earth's center or (*latitude*, *longitude*, *depth*) coordinates with a zero depth on the Earth's surface. As mentioned in the Introduction, the input data reflects the positions of earthquake hypocenters within a slab. Compared to point sets obtained via conventional scanning of an object's surface using range scanners, these data do not lie on a two-manifold surface, they rather represent a volumetric region, as they can appear at an arbitrary place within the slab with thickness of several tens of kilometers. An illustration of the difference between data scanned directly from an object's surface and the slab data is shown in Figure 1. Whereas the noise in the data scanned by a range scanner is determined mainly by the accuracy and the properties of the scanning device, the noise level in the slab data is additionally increased by the diameter of the slab.

Furthermore, the accuracy of the estimation of the earthquake positions is very poor. The measurement error ranges up to several kilometers depending on the elevation of the Earth's surface over the earthquake center. In Figure 2, the underlying slab is visualized using a triangulation of the earthquake centers. The triangulation (which is non-manifold and discontinuous at several places) was generated for visualization purposes only. To measure distances to the slab (based on a distance field around the slab) or extend the slab shape beyond the earthquakes centers (section 2.2), the point set has to be preprocessed. Depending on the noise level in the

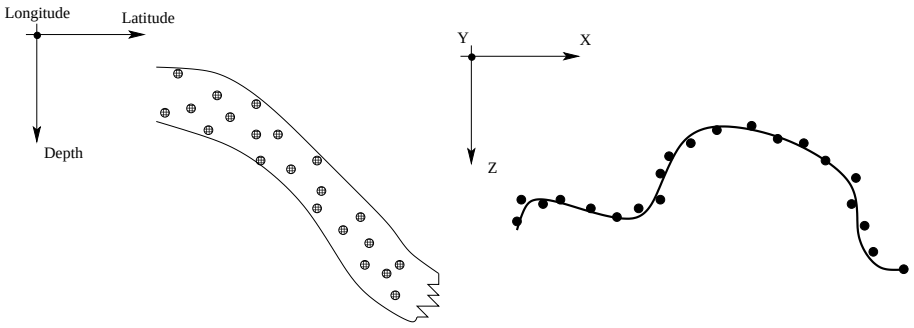

Figure 1: Difference between points approximating a manifold surface using, for example, laser range scanning (*right*) and a point set corresponding to earthquake centers (*left*); viewed from the side.

data, the point set is smoothed (section 2.1) and/or resampled (section 2.3).

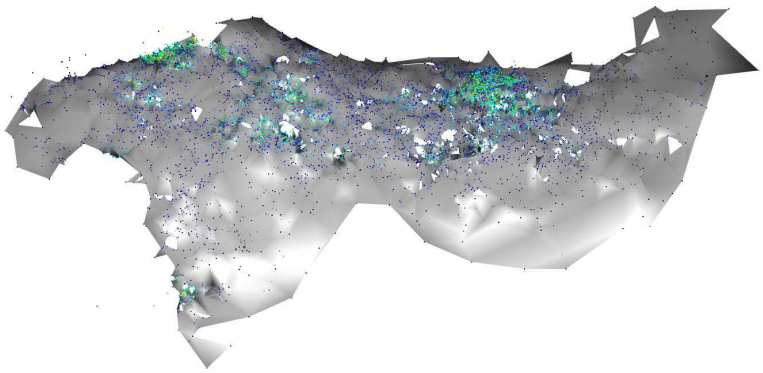

Figure 2: Triangulation of earthquake centers.

2.1 Smoothing

Denoising, smoothing or fairing of noisy data is one of the most important tasks during preprocessing of experimentally obtained data. In recent years, many authors dealt with mesh-based smoothing [BO03, DMSB99, FDCO03, JDD03, KCVS98, LW05, OBB01, SRML07a, SRML07b, Tau95]. All these methods have a common feature: they operate on triangulated meshes. In [VB07] a few of these approaches were modified in order to work directly on neighborhoods, and their smoothing quality and preservation of sharp features has been examined and compared.

For point sets consisting of earthquake centers we considered the following facts:

1. No consistent triangular mesh can be generated from slab data, as the data does not form a two-manifold surface. The result of a triangulation generation is a "triangle soup", as can be seen in Figure 2. Therefore, smoothing has to be performed prior the triangulation step.

2. Slabs do not contain sharp features, so we can use fairing methods suppressing all frequencies and producing smooth surfaces without sharp edges or corners.

3. The noise level in the data is extremely high so that it is often impossible to distinguish between small features and noise. A trade-off between strong smoothing and preservation of important features must be found. Due to the extreme noise level, optimal smoothing parameters cannot be reliably estimated automatically, so the user should alter by visual inspection the parameters in order to achieve the desired smooth shape of a slab.

The most appropriate fairing method for our data sets is a slight modification of the general purpose *moving least squares method* [ABCO*01]. Alexa et al. proposed a two-step approximation. In the first step, a local coordinate system is determined, which is in the second step used for the approximation of the point neighborhood with a high-order polynomial $z = f(x, y)$. Apart from the order of the approximating polynomial, the chosen neighborhood size is the most important parameter affecting the smoothing strength and quality. In order not to approximate the noise and to find the underlying surface the neighborhood size has to be set dependent on the noise level, ranging from 100 (almost "clean" data) to circa 2,000 (extremely noisy, large slab diameter) nearest neighbors.

For large point sets, neighborhoods consisting of ∼1,000 and more points cannot be stored in the memory for every point, and therefore the neighborhoods have to be efficiently computed on-the-fly within the smoothing procedure. This *large* neighborhood is compiled from "direct neighborhoods" (30–50 neighbors per point [VBS99], which we store in main memory) in a similar manner to the finding of geodetic neighbors described in Vanco and Brunnett [VB07]. The large neighborhood is created using a combination of a hash table $\mathcal{H}$ and a queue $\mathcal{Q}$.

Initially the direct neighbors $\mathcal{N}(\mathbf{P}) = \{\mathbf{Q}_1, \ldots, \mathbf{Q}_k\}$ of a point $\mathbf{P}$ are projected into the hash table. The hash function projects the points into the table sorted according to their distance to $\mathbf{P}$ (with the shortest distance at the beginning of the table). The points are successively taken from the hash table, starting at the smallest used table index, and their neighbors are inserted into the hash table (if they are not already included). If a neighbor should be projected outside of the hash table range, it is instead inserted into the queue $\mathcal{Q}$. The process is stopped when the desired number of neighbors k has been found or when all points from the hash table have been processed. In the latter case, if the desired neighbors have not been yet found, the points from the queue $\mathcal{Q}$ are projected into a new hash table using a new hash function until k neighbors are found.

The obtained large neighborhood (denoted by $\mathcal{M}^L$) is first approximated by a best-fit plane in order to determine a domain for the approximation of $\mathcal{M}^L$ with a high-order polynomial in a least-squares manner, i.e.,

$$\sum_i (f(x_i, y_i) - z_i)^2 w_i \to \min,$$

where w_i are weights associated with the neighbors $\mathbf{Q}_i$ and $[x_i, y_i, z_i]$ are the Cartesian coordinates of $\mathbf{Q}_i$. The surface degree is an important parameter affecting the smoothing power. The higher the degree is, the better is the neighborhood approximated by the surface. Unfortunately, higher-order polynomials can produce unwanted waves and are more sensitive to noise. Thus, the surface degree should be chosen in conjunction with the neighborhood size, and in accordance with the estimated noise level in the data.

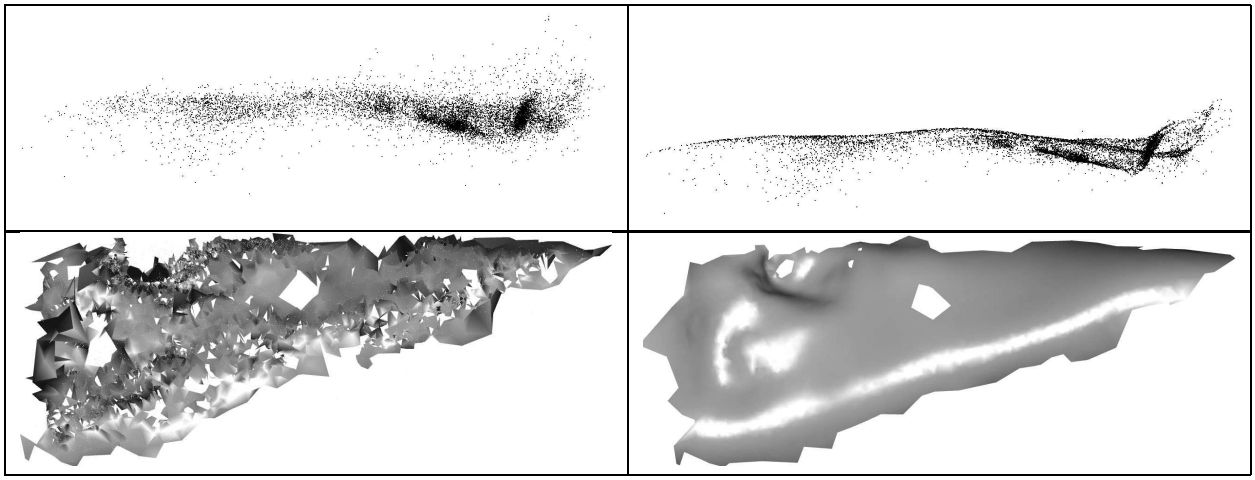

Figure 3: *Top row*: side view of a digitized Tonga slab (north of New Zealand); *bottom row*: top view of a triangulation of the point set. *Left*: original data set; *right*: data set after smoothing.

In Figure 3 the result of the smoothing procedure is demonstrated for an extremely noisy digitized slab

north of New Zealand using a neighborhood containing 1,000 points and a cubic surface, i.e.,

$$f(x,y) = a_0 + a_1 x + a_2 y + \frac{1}{2}\left(a_3 x^2 + 2a_4 xy + a_5 y^2\right) + \frac{1}{6}\left(a_6 x^3 + 3a_7 x^2 y + 3a_8 xy^2 + a_9 y^3\right).$$

The triangulation was used for visualization purposes only.

The smoothing quality is superior compared to the standard smoothing methods, such as Laplacian, bi-Laplacian, Taubin or bilateral smoothing. The only disadvantage is computation time; only 850 points per second were processed. As neighborhood assembling is the most time-consuming part of the smoothing, we project for each neighborhood more than one point onto the polynomial surface. This small optimization has only a marginal influence on the smoothing quality. The larger the point neighborhood is, the more points can be projected onto the surface without changing the smoothing effect. This approach reduces dramatically computation time, see Table 1 [1].

# of projected points	1	2	3	4
Time [sec]	11.69	4.66	3.94	3.45

Table 1: Optimization of the smoothing procedure, demonstrated for digitized Tonga slab containing 9,941 points, using 1,000 nearest neighbors.

2.2 Slab extension

The positions of earthquakes within a slab are determined from observations of arrival time of seismic waves at many seismometers located globally. Outside the seismically active regions the locations of the slab can be inferred through kinematic constraints on where the slab has been entering the mantle or seismic tomography.

The measured earthquake locations provide only a rough approximation of the form of seismically active parts of the slab. Seismically inactive parts of slabs do not contain any earthquakes, and thus they have no data. For the case, where the region of interest of a slab is bigger than the seismically active part, we provide a few simple procedures for extending the digitized slab surface beyond the acquired earthquake data.

As input the extension procedure expects, apart from the earthquakes centers, sparsely sampled contours and/or boundary lines. Boundary lines (denoted by $\mathcal{C}_B$ in the following) define the borders of the region of interest, and contours (denoted by $\mathcal{C}_L$) contain random points belonging to the slab surface. An example of a slab (near Alaska) with several contour lines (middle of the figure) and a boundary line (lower-left corner) is shown in Figure 4, on the left side.

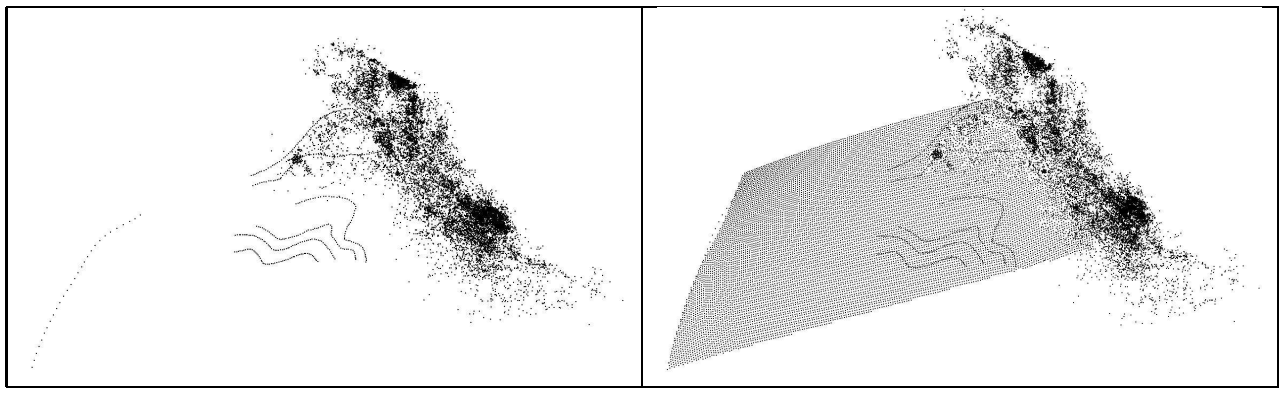

Figure 4: *Left*: slab with contour lines (middle) and a boundary line (left-lower corner), *right*: extended slab.

The slab extension procedure is sketched in Figure 5. First, a subset $\mathcal{S}_u$ of the slab surface $\mathcal{S}$ is determined so that a high-order polynomial surface in a local coordinate frame approximates all $\mathcal{S}_u$, $\mathcal{C}_L$ and $\mathcal{C}_B$ well, i.e., the average distance of all points stays within a user-specified threshold.

[1] Test system: Dual Core Pentium Core 2 Duo, 2.16 GHz with 2 GB RAM.

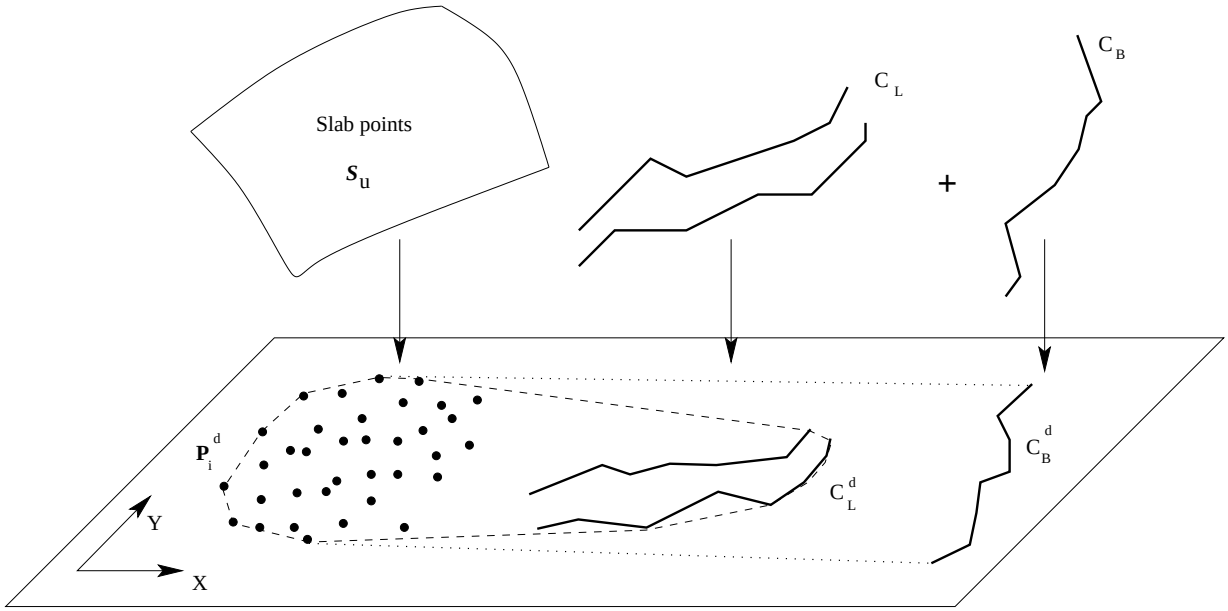

Figure 5: Slab extension.

All points of $\mathcal{S}_u$ and $\mathcal{C}_L$ are projected onto the xy-plane of the local coordinate system. To each projected point, its 2D coordinates with regard to the new local coordinate system of the xy-plane are assigned, denoted by $\mathbf{P}_i^d$. A 2D convex hull of the $\mathbf{P}_i^d$ is computed (dashed lines in Figure 5), and the points of $\mathcal{C}_B$ are projected onto the xy-plane of the local coordinate system; their 2D coordinates $\mathbf{Q}_j^d$ are determined. The convex hull and the 2D polygonal chain $\mathbf{Q}_j^d$ are merged forming a simple polygon $\mathcal{P}$ (dotted lines in Figure 5). For the merging step, a slight modification of the merge step of a divide-and-conquer algorithm for generation of a 2D convex hull can be used. The interior of $\mathcal{P}$ is used as the domain of the high-order polynomial $f(x, y)$. Subsequently, the domain is rasterized according to a user-specified sampling density parameter U_d, and new 3D points $[x_i, y_i, f(x_i, y_i)]$ are inserted into the original data set. In order not to increase the sampling on overlapping parts with the original slab surface, the distance of a point $\mathbf{P}$, which is to be added to its nearest neighbor of $\mathcal{S}$, is computed. If the distance is smaller than U_d, the point $\mathbf{P}$ is not inserted. A result of the extension procedure after one step is shown in Figure 4, *right* side.

2.3 Resampling

In addition to a high noise level in the data provided by global seismic tomography, there is another issue making the processing of this data difficult. For the parts of the slab with no earthquakes, global seismicity provides no data, and on the other side there are regions with extremely high point density. Figure 6 demonstrates strong irregular point density of a slab beneath Alaska.

To be able to create a continuous distance transform for the underlying unknown surface, the boundary of the surface (slab) must be determined. Finding the boundary of an irregularly sampled point set is a difficult and ambiguous task requiring a user intervention. To avoid consideration of many special cases during the implementation of the boundary-finding procedure, we resample the whole point set providing an (almost) regular sampling. The resampling procedure is based on moving least squares (MLS) surfaces [ABCO*01]. In the first step the slab is decomposed into overlapping MLS patches, of which domains are subsequently uniformly sampled and a new point set is created.

Starting with an arbitrary unmarked point $\mathbf{P}$ (with all points being initially unmarked), its large neighborhood $\mathcal{N}_L$ is assembled using small point neighborhoods (section 2.1) and a local coordinate system for $\mathcal{N}_L$ is determined. The points of $\mathcal{N}_L$ are approximated with a high-order polynomial staying within a user-prescribed approximation threshold T_a, i.e., the average distance of the points to the surface must be less than T_a. If the distance criterion is violated, outer points (boundary points) are excluded from $\mathcal{N}_L$ and the approximation is repeated. When an MLS patch has been successfully approximated, the following computations are performed:

Figure 6: Highly irregular point density of the slab beneath Alaska.

- Project all points of $\mathcal{N}_L$ onto the MLS surface; for each point $\mathbf{P}_i$ store its 2D domain point $\mathbf{P}_i^d$.
- Compute a convex hull $\mathcal{CH}$ of the domain points $\mathbf{P}_i^d$.
- Store the points $\mathbf{P}_{\min}^d$ and $\mathbf{P}_{\max}^d$ of $\mathcal{CH}$ with the minimum and maximum x-coordinate, respectively. Knowing these points speeds up the query whether an arbitrary 2D point is inside or outside the convex hull ($O(\log n)$, where n is the number of points of the convex hull).
- Store the distance $d_{\max}$ from the center of mass of the domain points to the most distant 2D point $\mathbf{P}_m^d$ of $\mathcal{N}_L$.
- Estimate the best enclosing ellipse of the domain points $\mathbf{P}^d$ and determine its two axes $\mathbf{u}^d$ and $\mathbf{v}^d$ and the lengths of the axes. The axes are extended from 2D into 3D space (assigning $z = 0$), and transformed from the local coordinate system of the MLS surface into the global coordinate system $(\mathbf{U}, \mathbf{V})$.
- Create and store an empty quadtree $\mathcal{QT}$. (See Algorithm 1 described below.)

All points of $\mathcal{N}_L$ are marked, and the procedure proceeds with the next unmarked point until all points are marked.

At the end of the first step, a neighborhood graph of the MLS surfaces is created. Each MLS surface stores all its direct neighboring MLS surfaces, i.e., an MLS surface $\mathcal{S}_i$ is adjacent to a surface $\mathcal{S}_j$ if there exists a point $\mathbf{P} \in \mathcal{S}_i$ so that a direct neighbor $\mathbf{Q}$ of $\mathbf{P}$ belongs to the surface $\mathcal{S}_j$, i.e., $\mathbf{Q} \in \mathcal{S}_j$.

Having decomposed the surface into MLS patches, all patches are successively rasterized (sampled) creating one smooth continuous surface. Algorithm 1 consists of the steps performed during the rasterization of the convex hulls of individual MLS surfaces.

The quadtree $\mathcal{QT}$ stores for each MLS surface $\mathcal{S}$ all its 2D points inside the domain of $\mathcal{S}$, which has already been rasterized, either directly by $\mathcal{S}$ or by any of the neighbors of $\mathcal{S}$ on the overlapping area with $\mathcal{S}$. The quadtree test prevents increase of sampling density in areas where two or more MLS surfaces overlap. For instance, let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two overlapping MLS surfaces. During rasterization of $\mathcal{S}_1$ all sampled points on the overlapping area between $\mathcal{S}_1$ and $\mathcal{S}_2$ are transformed to the local coordinate system of $\mathcal{S}_2$ and inserted into its quadtree $\mathcal{QT}(\mathcal{S}_2)$. During rasterization of $\mathcal{S}_2$ no points in the overlapping area, i.e., points with the distance to the nearest neighbor from $\mathcal{QT}$ smaller than the sampling parameter, are processed.

In steps 3 and 4d of Algorithm 1 the weights of the sampled points are computed. In order to obtain a continuous surface, the piecewise-continuous patches have to be blended together using proper blending functions. If we can assume circular patches (i.e., circular domains of the large neighborhoods), an approach based on *Gaussian radial basis functions* of the form

$$w(\mathbf{x}) = e^{\frac{-\|\mathbf{x}-\mathbf{C}\|^2}{2\sigma^2}}$$

provides a smooth and satisfactory blending. Here, $\mathbf{x}$ is the point of interest, $\mathbf{C}$ is the midpoint of the sphere and σ is a weighting radius. The choice of σ is very important; for smooth transitions between neighboring

Algorithm 1 Rasterization of an MLS surface

For each rasterized point $\mathbf{P}_i^d = [x, y]$ in the convex hull $\mathcal{CH}$ of an MLS surface $\mathcal{S}$ do:

1. Search in the quadtree $\mathcal{QT}$ for the nearest neighbor $\mathbf{Q}^d$ to $\mathbf{P}_i^d$. If $\|\mathbf{P}_i^d - \mathbf{Q}^d\| < T_s$, where T_s is a user-defined sampling parameter, skip processing of $\mathbf{P}_i^d$ and select the next rasterized point. Otherwise, insert the point $\mathbf{P}_i^d$ into the quadtree.
2. For $\mathbf{P}_i^d$ compute the corresponding 3D point $\mathbf{P}_i$ by evaluating the MLS surface at $[x, y]$ and transforming the point $[x, y, f(x, y)]$ into world coordinates. Let $\mathbf{X}_0 = \mathbf{P}_i$.
3. Compute the point weight w_0 according to the chosen blending function; explained below.
4. For all neighboring MLS surfaces $\mathcal{S}_j,\ j = 1, \ldots, k$, of $\mathcal{S}$ do:
   (a) Project $\mathbf{X}_0$ onto $\mathcal{S}_j$ yielding a projected 3D point $\mathbf{X}_j$ and its domain point $\mathbf{X}_j^d$.
   (b) If $\mathbf{X}_j^d$ lies outside the convex hull $\mathcal{CH}$ of $\mathcal{S}_j$, goto to the next neighboring MLS surface.
   (c) Search the quadtree $\mathcal{QT}$ of $\mathcal{S}_j$ for the nearest neighbor $\mathbf{Q}^d$ to $\mathbf{X}_j^d$. If $\|\mathbf{X}_j^d - \mathbf{Q}^d\| > T_s$, insert $\mathbf{X}_j^d$ into $\mathcal{QT}$.
   (d) Compute, according to the chosen blending function, the weights of both points $\mathbf{X}_0$ and $\mathbf{X}_j$ in the local coordinate system of $\mathcal{S}_j$ and choose the minimum of both (denoted by w_j).
5. Compute the weighted result point as

$$\mathbf{P}^r = \frac{\sum_{i=0}^{k} w_i \mathbf{X}_j}{\sum_{i=0}^{k} w_i}$$

patches the points on patch boundaries should have small non-zero weights. Considering our extensive empirical tests, we chose $\sigma^2 = \frac{1}{14} d_{\max}^2$.

If the domain of some patches is not circular but elliptical, the radial basis functions can cause unwanted cracks (discontinuities) in the vicinity of the elliptical patches, as shown in Figure 8. In Figure 7, the problem is sketched using a *top* and a *side* view. The domain of the MLS surface $\mathcal{S}_1$ is elliptic (*top* view), but the weighting function provides non-zero weights within a circular area (dotted circle). Point $\mathbf{X}_0$ lies outside the approximation domain of $\mathcal{S}_1$, i.e., the distance of the nearest point from the original point set to $\mathbf{X}_0$ does not have to fulfill the approximation threshold criterion T_a. Thus, the weight contribution for $\mathbf{X}_0$ coming from $\mathcal{S}_1$ should be zero. Unfortunately, $\mathbf{X}_0$ lies inside the influence area of the radial basis function of $\mathcal{S}_1$.

During the rasterization of $\mathcal{S}_2$ (Figure 7, *side* view) the point $\mathbf{X}_0$ is blended with its weighted projections on the neighboring MLS surfaces, which overlap with $\mathcal{S}_2$. Note that the decision whether two MLS surfaces overlap is done based on the weights of projected points: if the weight of a point projected onto a neighboring MLS surface is non-zero, we assume overlapping of both surfaces. In our example, the point $\mathbf{X}_0$ is projected onto $\mathcal{S}_1$, yielding point $\mathbf{X}_1$, and the weight $w(\mathbf{X}_1)$ in the local coordinate system of $\mathcal{S}_1$ is computed. $\mathbf{X}_1$ lies outside the approximation domain of $\mathcal{S}_1$ (dotted part of $\mathcal{S}_1$) but inside the perimeter of its radial basis function. Therefore, $\mathbf{X}_1$ is assigned a non-zero weight contribution from $\mathcal{S}_1$. Thus, a part of the MLS surface $\mathcal{S}_2$ is pushed in an undesired direction creating a crack, see Figure 8.

To resolve this problem, we decided to use weight functions that provide non-zero values only in the approximation domain of the MLS surface, i.e., in the area, where the points of the original point set fulfill the approximation threshold T_a. *Elliptic weight functions* offer the desired flexibility, but their evaluation is computationally more expensive and they make necessary calculation and storage of additional structures.

During the first step (creation of MLS patches) the 2D domain of each MLS surface has to be enclosed with an ellipse. For efficiency we estimate the axes of the ellipse using principal component analysis (PCA). PCA is a good trade-off between accuracy and computation speed. The result of PCA is not the best smallest enclosing ellipse, but it is very close to the best one. Further, PCA is much faster compared to methods for computing a smallest enclosing ellipse [GS98a, Wel91]. Both 2D axes $[\mathbf{u}^d, \mathbf{v}^d]$ are extended to 3D space (assigning $z = 0$) and transformed from the local coordinate system of the MLS surface to global coordinates.

The weight of an arbitrary 3D point $\mathbf{P}$ (in the global coordinate system) with regard to an MLS surface

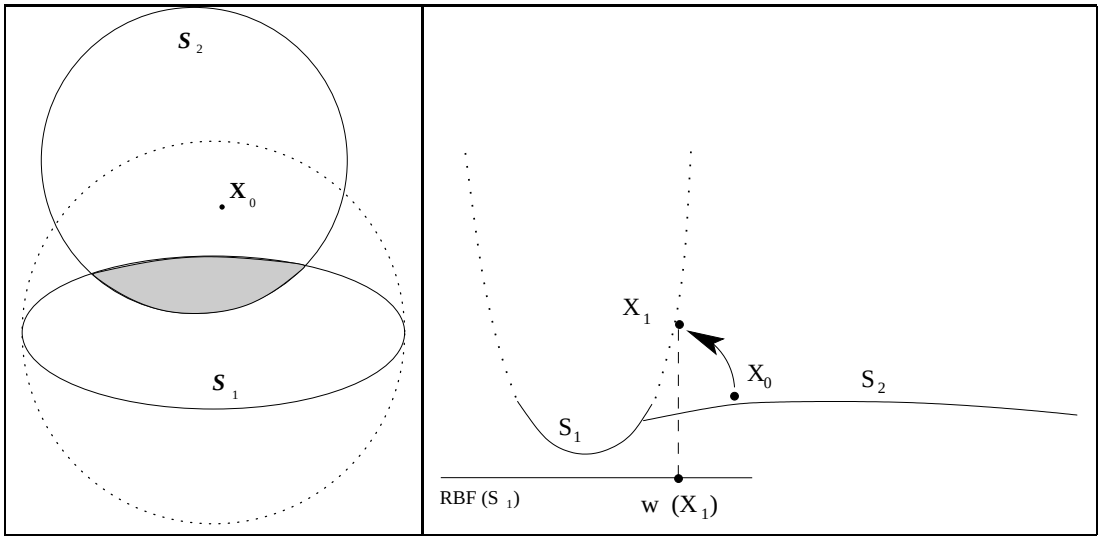

Figure 7: Problems with elliptic-shaped MLS patches using radial basis functions.

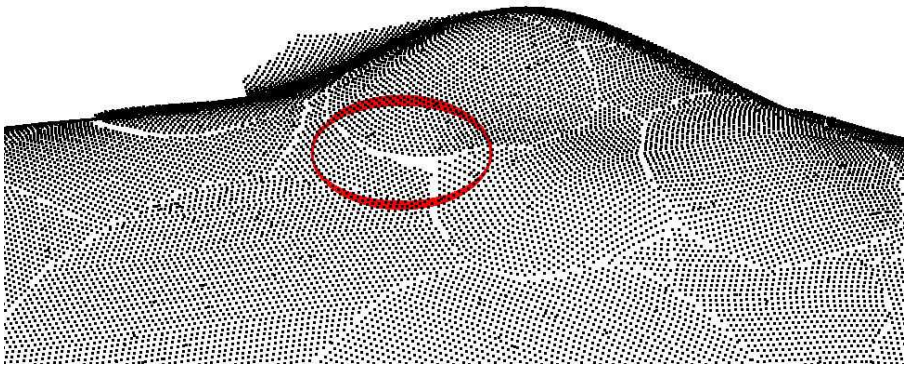

Figure 8: Slab data with a discontinuity in vicinity of an elliptic patch.

$\mathcal{S}\,[\mathbf{C},\mathbf{N},\mathbf{U},\mathbf{V}]$ - with $\mathbf{C}$ being the origin of the local coordinate system of $\mathcal{S}$, $\mathbf{N}$ being the normalized z-axis of the local coordinate system, and $\mathbf{U},\mathbf{V}$ being the non-normalized x, y axes of the local coordinate system - is computed as follows:

$$\mathbf{P}' = \mathbf{P} + \langle \mathbf{C}-\mathbf{P},\mathbf{N}\rangle\,\mathbf{N} \tag{1}$$

$$d^2 = \left(\frac{\langle \mathbf{P}'-\mathbf{C},\mathbf{U}\rangle}{\langle \mathbf{U},\mathbf{U}\rangle}\right)^2 + \left(\frac{\langle \mathbf{P}'-\mathbf{C},\mathbf{V}\rangle}{\langle \mathbf{V},\mathbf{V}\rangle}\right)^2 \tag{2}$$

$$w(\mathbf{P}) = e^{-\frac{d^2}{\sigma_e^2}} \tag{3}$$

Here, $\langle \mathbf{A},\mathbf{B}\rangle$ denotes the dot product of two vectors $\mathbf{A}$ and $\mathbf{B}$. In equation (1), the point $\mathbf{P}$ is orthogonally projected onto the xy-plane of the local coordinate system of $\mathcal{S}$. Equation (2) calculates the squared elliptic algebraic distance of $\mathbf{P}'$ to the ellipse center.

When defining the parameter σ_e of equation (3), one must consider the fact that one is not dealing with Euclidean distances (radial basis functions) but with elliptic algebraic distances, i.e., the distance of boundary points of $\mathcal{S}$ to the coordinate system origin is always one. The calculation of weights of projections of $\mathbf{P}$ onto the neighboring MLS surfaces of $\mathcal{S}$ is described in step 4d of Algorithm 1. The minimum of two weights -

corresponding to the point $\mathbf{P}$ and to the point $\mathbf{X}_j$, which is a projection of $\mathbf{P}$ to $\mathcal{S}_j$ - is chosen.

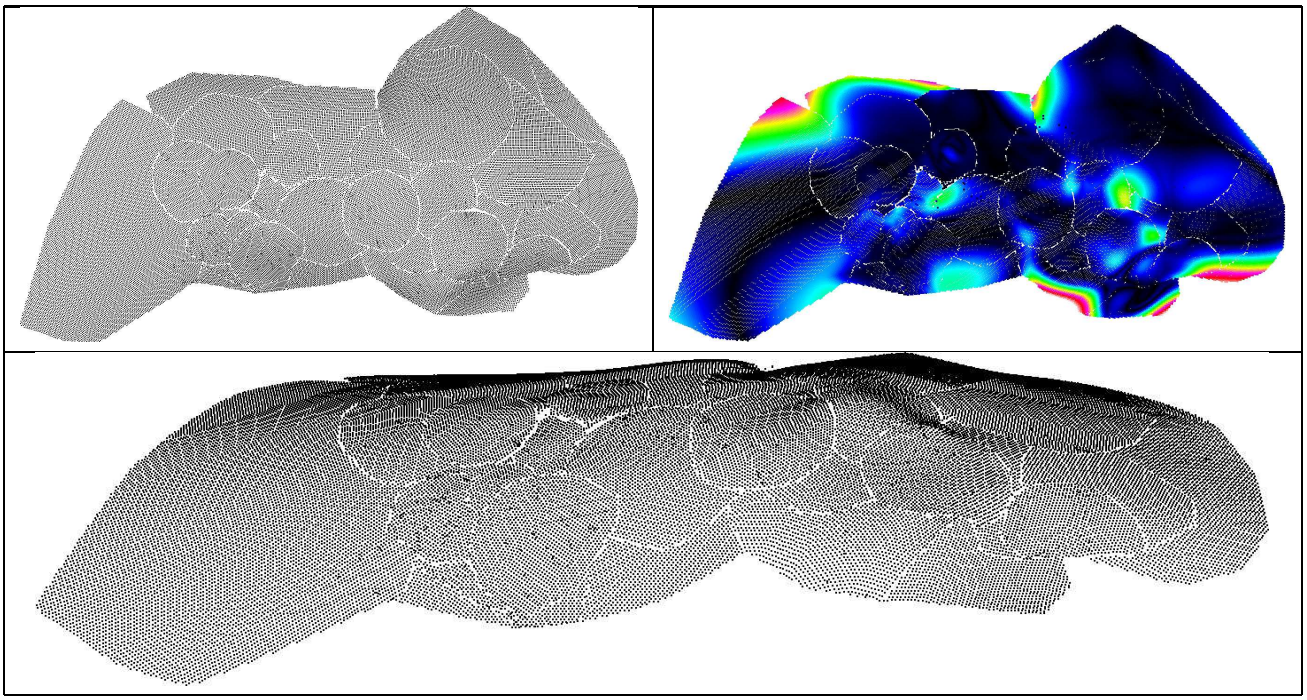

Figure 9: Resampling of slab close to Alaska. *Upper-left corner*: top view; *bottom row*: side view; *upper-right corner*: curvature plot of resampled point set.

The elliptic weight functions provide smooth, curvature-continuous transitions between adjacent MLS surfaces, as demonstrated on the slab surface close to Alaska, shown in Figure 9. The original point set contained 13,918 points. The resampling procedure produced 33,501 points in 5.01 seconds[2] using an approximation threshold $T_a = 4.057$ km. The *upper-right* part of the figure shows the curvature plot of the resampled point set. Note that the principal curvatures were not estimated from the point set. They were computed exactly from the MLS surfaces. Curvature discontinuities can be observed for a few transitions where the shape of the approximation domain is different from an ellipse. The visible boundaries between MLS surfaces are due to the fact that the sampling (rasterization) is performed in the local coordinate system of the elliptic domains. For our purposes these boundaries have no influence on further data processing.

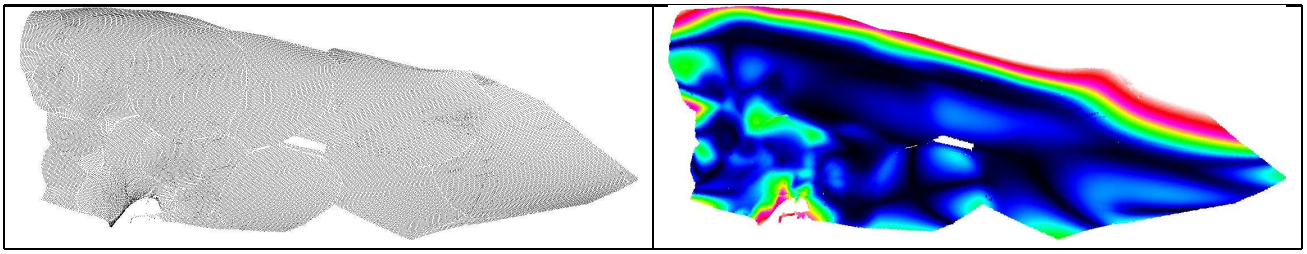

Figure 10: Resampling of Tonga slab north of New Zealand. *Left*: top view; *right*: top view of the maximum curvature plot.

In Figure 10 the resampling effect is demonstrated using a smoothed point set of the Tonga slab north of New Zealand. For resampling we used the approximation threshold $T_a = 3.457$ km and a cubic surface.

[2]Computer system: Intel Pentium Core 2 Duo, 2.16 GHz, with 2 GB SO-DIMM DDR RAM.

2.4 Boundary of a point set

The computation of a distance field to an open point set requires the knowledge of the boundary of the open point set in order to compute the distance from an arbitrary point $\mathbf{P}$ to the boundary rather than to the underlying surface when $\mathbf{P}$ lies outside the point set domain (Figure 14).

After the resampling step, the boundary of the new resampled point set can be found by using a simple and robust procedure. Algorithm 2 describes the necessary steps.

Algorithm 2 Finding a boundary of an open point set

Initially, all points are unmarked.

1. Pick an unmarked point $\mathbf{P}$. Find large neighborhood $\mathcal{N}_L$ to $\mathbf{P}$.
2. Approximate $\mathcal{N}_L$ with a high-order polynomial $f(x, y)$ in the local coordinate system of $\mathcal{N}_L$.
3. For each point $\mathbf{P}_i$ of $\mathcal{N}_{\text{sub}} \subset \mathcal{N}_L$, $\mathcal{N}_{\text{sub}}$ being a fixed-percentage subset of $\mathcal{N}_L$, do:
   - Project $\mathbf{P}_i$ and its k-nearest Euclidean neighbors $\{\mathbf{Q}_1, \ldots, \mathbf{Q}_k\}$ onto $f(x, y)$ and determine their domain coordinates $\mathbf{P}_i^d$ and $\mathbf{Q}_j^d$, respectively.
   - Sort $\mathbf{Q}_j^d$ with regard to the angle to line $\mathbf{Q}_1^d - \mathbf{P}_i^d$, i.e., $\angle(\mathbf{Q}_1^d, \mathbf{P}_i^d, \mathbf{Q}_j^d),\ j = 1, \ldots, k$.
   - If the angle between two consecutive points $\left(\mathbf{Q}_j, \mathbf{Q}_{(j+1) \mod k}\right),\ j = 1, \ldots, k$, is larger than a user-specified threshold (usually 90° - 120°), mark the point $\mathbf{P}_i$ as a *boundary point*.

The principle of Algorithm 2 is based on the observation that a boundary point $\mathbf{P}_e$ is incident with two boundary edges $\mathbf{e}_1$ and $\mathbf{e}_2$ so that a local neighborhood of $\mathbf{P}_e$, projected onto an appropriate 2D domain (e.g., a plane or a higher-order polynomial), lies on one side of $\mathbf{e}_1$ and $\mathbf{e}_2$. This observation can be transformed to finding the largest angle between two consecutive projected points sorted according to their polar coordinates (angle, radius), with origin $\mathbf{P}_i^d$ and an axis $\mathbf{Q}_1^d - \mathbf{P}_i^d$. For the projection domain we use a high-order polynomial surface in a local coordinate system of large neighborhoods $\mathcal{N}_L$. It turned out to be very robust compared to a plane, especially for points close to edges or highly curved areas.

Let σ be the distance from a point $\mathbf{P}$ to its k_{th} nearest neighbor (we usually choose $k = 50$). The point $\mathbf{P}$ can be incorrectly marked as a boundary point, if the *local feature size* [AB98, ACK01] in the vicinity of $\mathbf{P}$ is $LFS(\mathbf{x}) \geq \frac{\sigma}{2}$, for an arbitrary sample $\mathbf{x} \in \mathcal{F}$, where $\mathcal{F}$ is the smooth underlying surface.

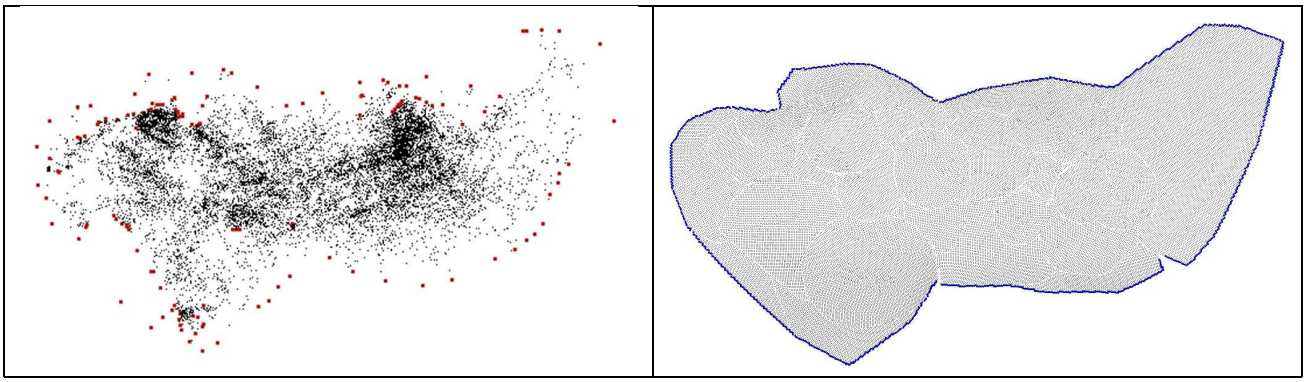

Figure 11: Comparison of boundary detection of original slab (*left*) and resampled point set (*right*).

For regularly sampled point sets the k-neighborhood projected onto the high-order polynomial surface can be small (10–20 neighbors) resulting in short processing times. With growing irregularity the neighborhood size must be increased to obtain stable results. Unfortunately, with growing neighborhood boundary points in highly-curved concave areas are usually not detected correctly. For irregular point sets without any assumptions about point sampling regularity, the algorithm can fail, as illustrated in Figure 11, *left*, using the original earthquake

centers. However, the procedure is successful for the point set after resampling (*right*).

3 Distance field of a slab

The primary need of geophysical simulations of the slab deformation is knowledge of the distance of an arbitrary point in 3D to the slab. For instance, using distances the viscosity and other important slab properties at the point can be estimated.

In the previous sections, we described necessary steps for preparing raw data acquired from global seismicity in order to compute a distance field around the slab. Computation of the distance from an arbitrary point to a discrete approximation of the slab (i.e., to the original point set) can suffer from aliasing effects, which grow with decreasing distance to the slab, see Figure 12. While the difference between the distance from $\mathbf{P}_1$ to its nearest neighbor of the point set and to the continuous approximation of the underlying surface can be neglected, for the point $\mathbf{P}_2$, which is in general more interesting than far points, the point-to-point distance computation causes unwanted wave-shaped distance field and aliasing effects close and beyond the slab boundary. To prevent possible discontinuities and inaccuracies in the distance field, we compute the distance to a piecewise-continuous approximation of the point set.

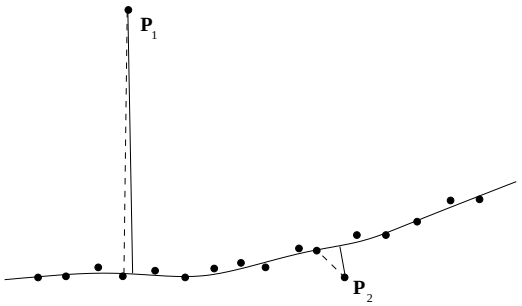

Figure 12: Calculating distance to a discrete (dashed lines) and continuous (solid lines) approximation of underlying surface.

In general, it is impossible to approximate the whole point set with one algebraic surface (with a fixed surface degree), keeping the average approximation distance within a user-specified threshold. Therefore, we segment the point set into several low-degree algebraic surface patches creating a piecewise-continuous approximation of the underlying point set.

For the segmentation a top-down, *region-growing*-based algorithm is used, see [VHB07]. A starting seed cluster consisting of points with similar principal curvatures is fitted with an implicit polynomial. If the fitting distance, i.e., the average distance of the seed points to the approximated surface, is smaller than a user-prescribed threshold T_s, the point cluster, or *segment*, denoted by S_i, and its corresponding approximated implicit surface by $\mathcal{I}(S_i)$, is accepted and extended by adding neighboring points $\mathbf{Q}_j$ to S_i satisfying a distance criterion $d(\mathbf{Q}_j, \mathcal{I}(S_i) < T_s$, $d(.,.)$ being the Euclidean distance. If no more points can be added to S_i, the procedure proceeds with next seed points until all points are processed.

In Figure 13, the segmentation of the slab close to Alaska is shown. The segmentation threshold was $T_s = 1.522$ km.

3.1 Distance function computation

After preprocessing and segmenting the point set the distance field calculation can be performed. For an arbitrary 3D point $\mathbf{P}$ its distance to a slab part is determined. Figure 14 shows different positions of the point $\mathbf{P}$ and different calculations to be performed depending on its position. If the point of interest lies above or below the slab (point $\mathbf{P}_i$), the distance to a quadric surface (or surfaces) is computed. If the point is located beyond the slab boundary (point $\mathbf{P}_o$), the point-curve distance is calculated.

The decision concerning the location of the point $\mathbf{P}$ is made based on its k-nearest neighbors from the point set. Usually, we choose $k = 6$. If the nearest or the second nearest neighbor is a boundary point, the

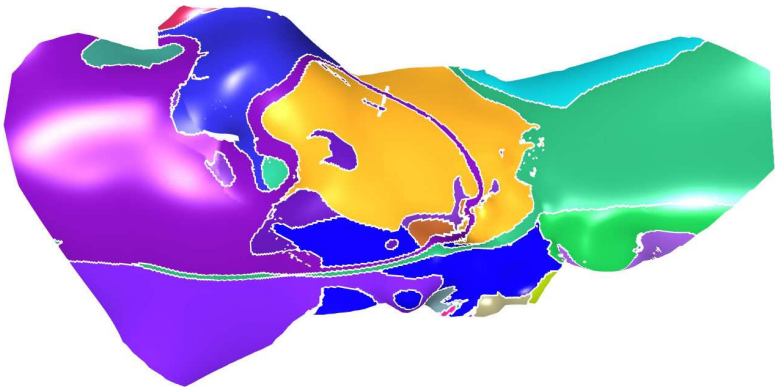

Figure 13: Segmentation of slab close to Alaska, *top* view.

distance from $\mathbf{P}$ to an approximation of the boundary is computed, see section 3.1.2. Otherwise, we perform a point-surface calculation, see section 3.1.1.

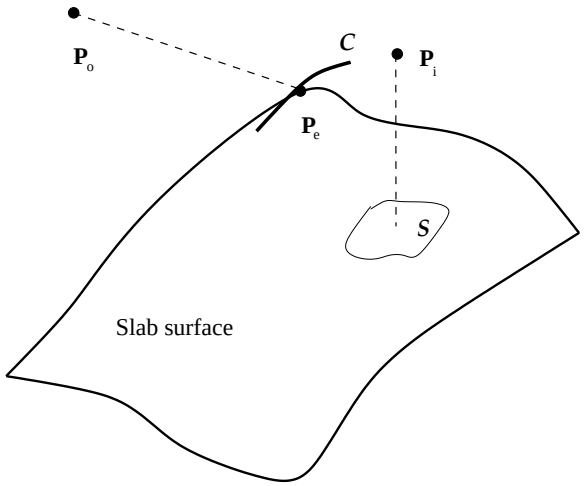

Figure 14: Computation of distance to surface or boundary of slab.

3.1.1 Point-surface computation

Let S_i be a segment with its corresponding quadric implicit polynomial represented as

$$\begin{aligned} \mathcal{I}(S_i) = \mathrm{F}(\mathbf{x}) = ax^2 + by^2 + cz^2 + 2fxy + 2gyz + 2hzx + 2px + 2qy + 2rz + d &= \\ \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} + 2\mathbf{B}^{\mathrm{T}}\mathbf{x} + d &= 0 \end{aligned}$$

where $\mathbf{A}$ is the 3x3 symmetric matrix $\mathbf{A} = \begin{pmatrix} a & f & h \\ f & b & g \\ h & g & c \end{pmatrix}$ and $\mathbf{B}$ is the vector $(p, q, r)^{\mathrm{T}}$.

It can be shown that the distance function of an arbitrary point $\mathbf{P}$ to the quadric surface is a polynomial of degree at most six. It is not acceptable to search numerically, in the worst case, all six solutions to find the nearest one. One can iteratively minimize the quadratic distance $d(\mathbf{x}) = \|\mathbf{P} - \mathbf{x}\|^2$ with a constraint $\mathrm{F}(\mathbf{x}) = 0$ using Lagrange multipliers, i.e.,

$$\begin{aligned} D(\lambda, \mathbf{x}) &= \|\mathbf{P} - \mathbf{x}\|^2 - \lambda \mathrm{F}(\mathbf{x}) \rightarrow \min \qquad \text{with} \\ \nabla D(\lambda, \mathbf{x}) &= 0. \end{aligned} \tag{4}$$

Equation (4) can be solved by using, for example, a globally convergent quasi-Newton method. Unfortunately, for fast convergence the Newton method requires suitable starting values. As the quadric surfaces have infinite area of definition, we use the original points to restrict the domain of the quadric surfaces. In Figure 15, the problem of an appropriate starting point on quadric surface is illustrated. Let $\mathbf{P}$ be an arbitrary point in space; we want to compute the distance from $\mathbf{P}$ to its nearest quadric surface $\mathrm{F}_n(\mathbf{x}) = 0$. First, k-nearest neighbors $\{\mathbf{Q}_1, \ldots, \mathbf{Q}_k\}$ from the point set to $\mathbf{P}$ are determined. For each neighbor $\mathbf{Q}_i$ its corresponding quadric surface is found. In order to find a good starting point, an intersection between the quadric surface and a line through $\mathbf{P}$ in direction $\mathbf{Q}_i - \mathbf{P}$ is computed (in Figure 15, *left*, the points $\mathbf{P}_1$ and $\mathbf{P}_2$). From a maximum of two possible intersection points we pick the one closer to $\mathbf{Q}_i$. If there are no intersection points (Figure 15, *right*), we compute the intersection between $\mathrm{F}_n(\mathbf{x})$ and a line passing through $\mathbf{Q}_i$ in the direction of $\mathbf{N}(\mathbf{Q}_i)$, $\mathbf{N}(\mathbf{Q}_i)$ being the estimated normal vector in point $\mathbf{Q}_i$. If this intersection computation yields no intersection points, we compute the intersection between $\mathbf{F}_n$ and a line passing through $\mathbf{Q}_i$ in the direction of $\nabla \mathrm{F}_n(\mathbf{Q})$. For our data sets it never happened that all three procedures yielded no intersection point.

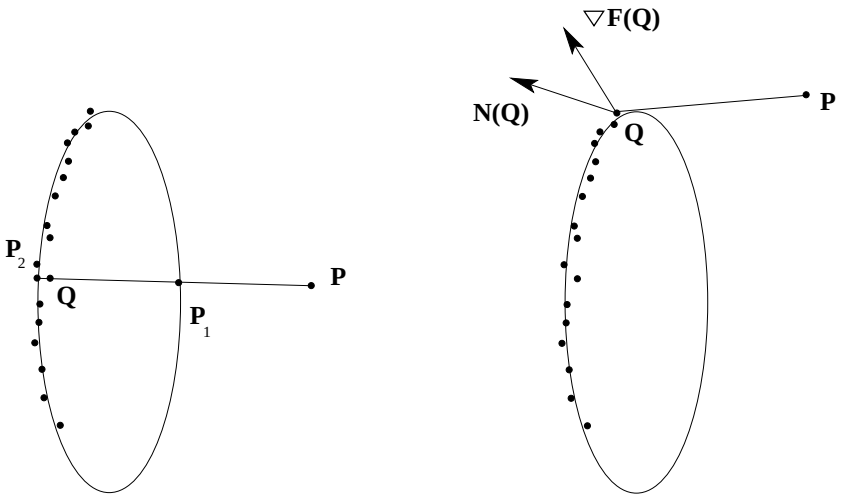

Figure 15: Finding the proper starting point on a quadric surface.

Having found a starting point we compute the distance from $\mathbf{P}$ to a quadric surface through Newton iterations. If the neighbors $\mathbf{Q}_i$ belong to different quadric surfaces, we calculate all distances and weight them according to the number of neighbors belonging to them. The final distance is computed as a weighted average of all distances.

3.1.2 Point-curve computation

If the nearest or the second nearest point of the point is a boundary point $\mathbf{P}_e$, we compute the distance of $\mathbf{P}$ to an approximation of the boundary rather than to a quadric surface.

First, the boundary in the vicinity of $\mathbf{P}_e$ is extended by recursive parsing through the neighborhoods of current boundary points and searching for further boundary points in their neighborhood. The procedure stops when the desired number of boundary points (denoted by $\mathcal{B} = \{\mathbf{B}_1, \ldots, \mathbf{B}_m\}$, $m = 10 - 30$) has been found. $\mathcal{B}$ is approximated by a best-fit line, and the points $\mathbf{B}_i$ are projected onto the line to acquire a suitable parameterization (we use centripetal parameterization). $\mathcal{B}$ is approximated by a quadratic Bézier or spline curve. The curve with the best fit is chosen and the distance from $\mathbf{P}$ to this curve is computed analytically (the distance function of a quadratic parametric curve being a cubic polynomial).

4 Results

In Figure 16 all processing steps starting with the earthquake centers data and leading to the segmentation of the extended resampled point set are shown. Figure 16 **a)** shows pre-filtered earthquake centers, where several

outliers were removed. The slab has approximate dimensions of 700 km × 400 km × 180 km and contains 14,081 earthquake centers. Its thickness is circa 20 km. For better visualization, the point set was triangulated, see Figure 16 **b)**. In the first step, the point set was smoothed, see Figure 16 **c)** and its triangulation is shown in Figure 16 **d)**, using a neighborhood of 500 points and a cubic surface. For the extension, see Figure 16 **e)**, four additional boundary lines and two sets of contours were used, resulting in 100,114 points. Subsequently, the extended point set was resampled with approximation threshold $T_a = 4.0175$ km and a cubic surface, producing 69,580 points, see Figures 16 **f)** and 16 **g)** (triangulation of resampled point set). The segmentation of the resampled point set was performed using a segmentation threshold $T_s = 1.3$ km, see Figure 16 **h)**.

The visualization of the calculated distance field using 14 million gridded points is demonstrated in Figure 17. In the left figure, the zero set surface is rendered red and two additional section through the distance field are shown. In the right figure, an additional cross section through an offset surface to the zero set surface of the distance field is shown.

The most time-consuming step of the processing pipeline is the distance field computation. For simulation of a viscous flow, which requires distance calculation, a grid file with circa 14 million points was created. Distances to the points of this file were computed in about 12 minutes. This time also includes I/O operations, i.e., reading grid points from a file and writing distances to a file (together circa 1.2 GB), which took 30%–40% of the whole time. All other stages of the point set processing required only a few seconds.

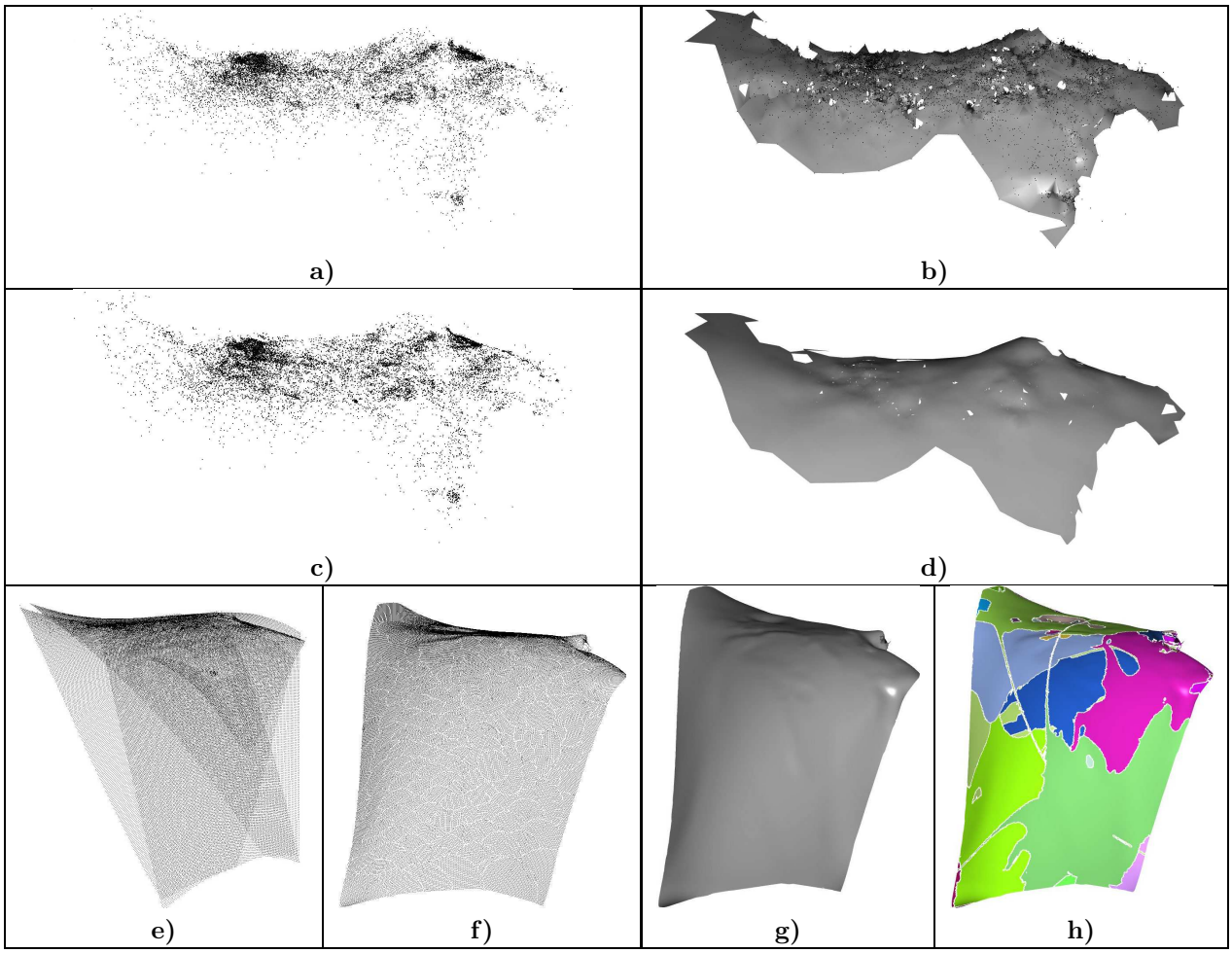

Figure 16: Stages of point set processing (digitized slab beneath Alaska).

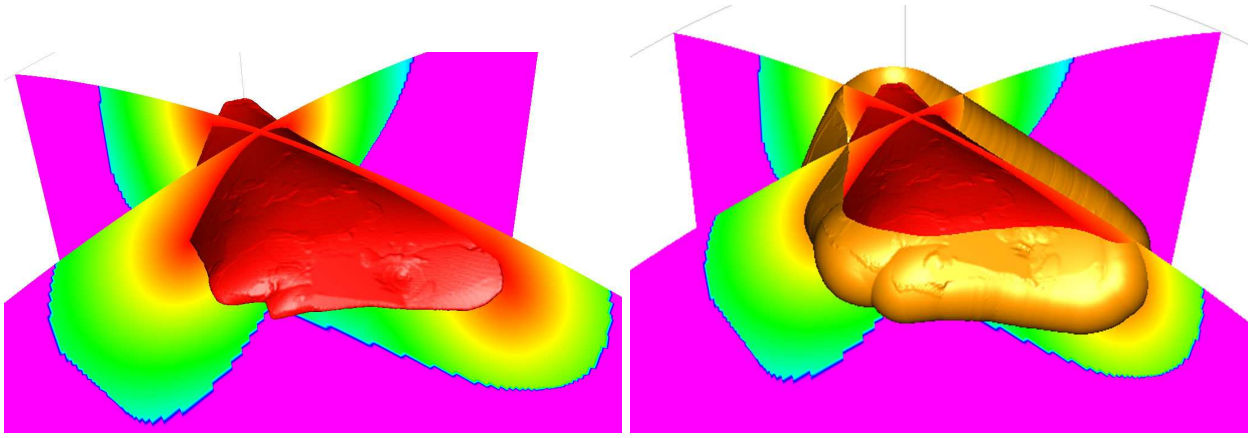

Figure 17: Visualization of distance field

5 Conclusions and Future Work

We presented a framework for computation of a distance field around a data set consisting of earthquake locations measured within subducted tectonic plates (slabs). Due to large measurement inaccuracies and the fact that earthquakes can be measured anywhere within a slab with thickness of several kilometers, extremely high noise level in the data requires us to perform special preprocessing prior to distance field computation. Our smoothing procedure uses large neighborhoods in order to find an approximation of the underlying surface for projection of noisy points onto this surface. We introduced a simple, powerful tool for extension of a slab beyond the earthquake centers using sparsely sampled contour and boundary lines. Subsequent resampling converts slab data into point sets with almost regular sampling and user-defined sampling density using a subdivision of the point set into MLS surfaces. Elliptic weight functions support continuous transitions between adjacent MLS patches. The boundary finding procedure provides a robust marking of boundary points, even for irregularly sampled point sets satisfying our prescribed sampling criteria (local feature size). The slab is subdivided into implicit surfaces, which provide an approximation of the underlying surface supporting the computation of a piecewise continuous distance field. For points outside the extended slab domain, an analytic point-curve distance calculation is used. Apart from the distance field computation, all other steps can be performed within a few seconds making possible iterative fine tuning of the slab shape and smoothness by the user.

The choice of a seed point during resampling is done randomly. On or close to highly curved areas, which cannot be approximated well by a polynomial, this random seed point placement can result in a large number of MLS surfaces. We plan to use a more sophisticated approach based on a modification of the Voronoi diagram, driven by the curvatures, i.e., starting in areas with low Gaussian or mean curvature.

6 Acknowledgments

This work was supported by the German Research Foundation (DFG), which provided support to the first author via a post-doctoral research fellowship.

We gratefully acknowledge the support of the W.M. Keck Foundation provided to the UC Davis Center for Active Visualization in the Earth Sciences (CAVES), and thank the members of the Visualization and Computer Graphics Research Group at the Institute for Data Analysis and Visualization (IDAV) at the University of California, Davis.

References

[AB98] Amenta N., Bern M.: Surface reconstruction by voronoi filtering. In *SCG '98: Proceedings of the fourteenth annual symposium on Computational geometry* (1998), ACM Press, pp. 39–48.

[ABCO*01] Alexa M., Behr J., Cohen-Or D., Fleishman S., Levin D., Silva C. T.: Point set surfaces. In *Conference on Visualization* (2001), IEEE Computer Society, pp. 21–28.

[ACK01] Amenta N., Choi S., Kolluri R. K.: The power crust. In *6th ACM symposium on Solid Modeling and Applications* (2001), ACM Press, pp. 249–266.

[BGS03] Billen M. I., Gurnis M., Simons M.: Multiscale dynamics of the Tonga-Kermadec subduction zone. *Geophysical Journal International 153* (2003), 359–388.

[BO03] Belyaev A., Ohtake Y.: A comparison of mesh smoothing methods. In *Israel-Korea Bi-National Conference on Geometric Modeling and Computer Graphics* (2003), Tel Aviv University, pp. 83–87.

[Chi91] Chiao L.-Y.: *Membrane Deformation Rate and Geometry of Subducting Slabs*. Master's thesis, University of Washington, 1991.

[DMSB99] Desbrun M., Meyer M., Schröder P., Barr A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH '99* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 317–324.

[FDCO03] Fleishman S., Drori I., Cohen-Or D.: Bilateral mesh denoising. *ACM Transaction on Graphics 22*, 3 (2003), 950–953.

[GS98a] Gärtner B., Schönherr S.: *Smallest Enclosing Ellipses - An Exact and Generic Implementation*. Tech. Rep. B 98-05, Institut für Theoretische Informatik, ETH Zentrum Zürich / Institut für Informatik Freie Universität Berlin, 1998.

[GS98b] Gudmundsson Ó., Sambridge M.: A regionalized upper mantle (RUM) seismic model. *Journal of Geophysical Research 103* (1998), 7121–7136.

[IB77] Isacks B. L., Barazangi M.: Geometry of Benioff zones: lateral segmentation and downwards bending of the subducted lithosphere. *Island Arcs, Deep Sea Trenches and Back-arc Basins 1* (1977), 99–114.

[JB10] Jadamec M. A., Billen M. I.: Reconciling surface plate motions with rapid three- dimensional mantle flow around a slab edge. *Nature 465* (2010), 338341.

[JDD03] Jones T. R., Durand F., Desbrun M.: Non-iterative, feature-preserving mesh smoothing. *ACM Transaction on Graphics 22*, 3 (2003), 943–949.

[KCVS98] Kobbelt L., Campagna S., Vorsatz J., Seidel H.-P.: Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH'98* (1998), ACM Press, pp. 105–114.

[LW05] Lee K.-W., Wang W.-P.: Feature-preserving mesh denoising via bilateral normal filtering. In *9th International Conference on Computer Aided Design and Computer Graphics* (2005), IEEE Computer Society, pp. 275–280.

[MGK06] Miller M. S., Gorbatov A., Kennett B. L. N.: Three-dimensional visualization of a near-vertical slab tear beneath the southern Mariana arc. *Geochemistry, Geophysics, Geosystems 7* (2006), 6012–.

[NMHJ96] Nothard S., McKenzie D., Haines J., Jackson J.: Gaussian curvature and the relationship between the shape and the deformation of the Tonga slab. *Geophysical Journal International 127* (1996), 311–327.

[OBB01] Ohtake Y., Belyaev A. G., Bogaevski I.: Mesh regularization and adaptive smoothing. *Computer-Aided Design 33*, 11 (2001), 789–800.

[RH02] Ratchkovzki N. A., Hansen R. A.: New evidence for segmentation of the Alaska subduction zone. *Bulletin of the Seismological Society of America 92* (2002), 1754–1765.

[Rom03] Romanowicz B.: Global mantle tomography: Progress status in the past 10 years. *Annual Review of Earth and Planetary Sciences 31* (2003), 303–328.

[SA06] Syracuse E. M., Abers G. A.: Global compilation of variations in slab depth beneath arc volcanoes and implications. *Geochemistry, Geophysics, Geosystems 7* (2006).

[SPP00] Sipkin S. A., Person W. J., Presgrave B. W.: Earthquake bulletins and catalogs at the usgs national earthquake information center. In *Iris News Letter*, vol. 1. 2000, pp. 2–4.

[SRML07a] Sun X., Rosin P., Martin R., Langbein F.: Fast and effective feature-preserving mesh denoising. *IEEE Transactions on Visualization and Computer Graphics 13*, 5 (2007), 925–938.

[SRML07b] Sun X., Rosin P. L., Martin R. R., Langbein F. C.: Random walks for mesh denoising. In *Symposium on Solid and Physical Modeling* (2007), ACM, pp. 11–22.

[Tac00] Tackley P. J.: Mantle convection and plate tectonics: Toward an integrated physical and chemical theory. *Science 288* (2000), 2002–2007.

[Tau95] Taubin G.: A signal processing approach to fair surface design. In *ACM SIGGRAPH* (1995), ACM Press, pp. 351–358.

[VB07] Vančo M., Brunnett G.: Geometric preprocessing of noisy point sets: an experimental study. *Computing 79, Special issue on Geometric Modeling* (2007).

[VBS99] Vančo M., Brunnett G., Schreiber T.: A hashing strategy for efficient k-nearest neighbors computation. In *International Conference on Computer Graphics* (1999), IEEE Computer Society, pp. 120–127.

[vdHESN91] van der Hist R., Engdahl R., Spakman W., Nolet G.: Tomographic imaging of subducted lithosphere below northwest Pacific island arcs. *Nature 353* (1991), 37–43.

[VHB07] Vančo M., Hamann B., Brunnett G.: Surface Reconstruction from Unorganized Point Data With Quadrics. *accepted for Computer Graphics Forum* (2007).

[VR84] Vassiliou M. S. B. H. H., Raefsky A.: The distribution of earthquakes with depth and stress in subducting slabs. *Journal of Geodynamics 1* (1984), 11–28.

[Wel91] Welzl E.: Smallest enclosing disks (balls and ellipsoids). In *New Results and New Trends in Computer Science*, LNCS. Springer, 1991.

[wZ90] wei Zhoua H.: Observations on earthquake stress axes and seismic morphology of deep slabs. *Geophysical Journal International 103*, 2 (1990), 377–401.

[ZTS*03] Zhang H., Thurber C., Shelly D., Ide S., Beroza G. C., Hasegawa A.: High-resolution subducting slab structure beneath northern Honshu, Japan, revealed by double-difference tomography. *AGU Fall Meeting Abstracts 32*, 4 (2003), 361–364.