

Minimum-Error Triangulations for Sea Surface Reconstruction

Anna Arutyunova ✉

Institute for Computer Science,
Universität Bonn, Germany

Jan-Henrik Haunert

Institute of Geodesy and Geoinformation,
Universität Bonn, Germany

Jürgen Kusche

Institute of Geodesy and Geoinformation,
Universität Bonn, Germany

Philip Mayer ✉

Institute for Computer Science,
Universität Bonn, Germany

Heiko Röglin

Institute for Computer Science,
Universität Bonn, Germany

Anne Driemel

Hausdorff Center for Mathematics,
Universität Bonn, Germany

Herman Haverkort

Institute for Computer Science,
Universität Bonn, Germany

Elmar Langetepe

Institute for Computer Science,
Universität Bonn, Germany

Petra Mutzel

Institute for Computer Science,
Universität Bonn, Germany

Abstract

We apply state-of-the-art computational geometry methods to the problem of reconstructing a time-varying sea surface from tide gauge records. Our work builds on a recent article by Nitzke et al. (Computers & Geosciences, 157:104920, 2021) who have suggested to learn a triangulation D of a given set of tide gauge stations. The objective is to minimize the misfit of the piecewise linear surface induced by D to a reference surface that has been acquired with satellite altimetry. The authors restricted their search to k -order Delaunay (k -OD) triangulations and used an integer linear program in order to solve the resulting optimization problem.

In geometric terms, the input to our problem consists of two sets of points in \mathbb{R}^2 with elevations: a set \mathcal{S} that is to be triangulated, and a set \mathcal{R} of reference points. Intuitively, we define the error of a triangulation as the average vertical distance of a point in \mathcal{R} to the triangulated surface that is obtained by interpolating elevations of \mathcal{S} linearly in each triangle. Our goal is to find the triangulation of \mathcal{S} that has minimum error with respect to \mathcal{R} .

In our work, we prove that the minimum-error triangulation problem is NP-hard and cannot be approximated within any multiplicative factor in polynomial time unless $P = NP$. At the same time we show that the problem instances that occur in our application (considering sea level data from several hundreds of tide gauge stations worldwide) can be solved relatively fast using dynamic programming when restricted to k -OD triangulations for $k \leq 7$. In particular, instances for which the number of connected components of the so-called k -OD fixed-edge graph is small can be solved within few seconds.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Minimum-Error Triangulation, k -Order Delaunay Triangulations, Data dependent Triangulations, Sea Surface Reconstruction, fixed-Edge Graph

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.7

Related Version Full Version: <http://arxiv.org/abs/2203.07325>

Supplementary Material Software (Source Code and Information about the Data Acquisition):

<https://github.com/PhilipMayer94/dynamic-programming-for-min-error-triangulations>

archived at `swh:1:dir:a009b56de67b3679c496449aff63f6b343593a8d`



© Anna Arutyunova, Anne Driemel, Jan-Henrik Haunert, Herman Haverkort, Jürgen Kusche, Elmar Langetepe, Philip Mayer, Petra Mutzel, and Heiko Röglin; licensed under Creative Commons License CC-BY 4.0

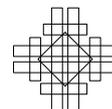
38th International Symposium on Computational Geometry (SoCG 2022).

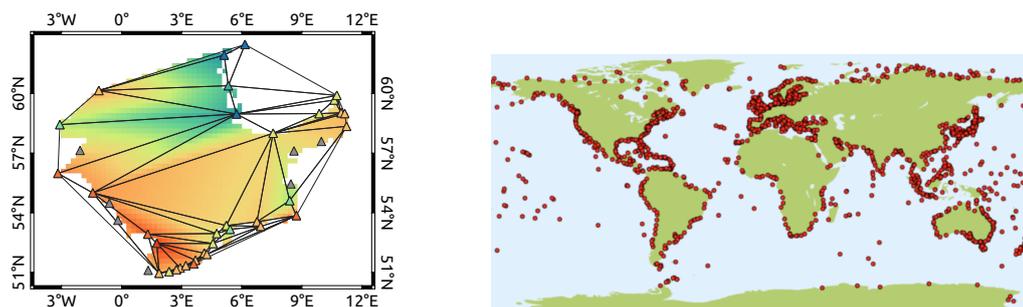
Editors: Xavier Goaoc and Michael Kerber; Article No. 7; pp. 7:1–7:18



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** Left: A minimum-error triangulation of the North Sea data (June 2010) with 34 tide gauge stations computed with the approach in [24]. Right: Locations of all tide gauge stations in the PSMSL database (www.psmsl.org/products/data_coverage).

Funding This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC-2047/1 – 390685813 and DFG grant RO 5439/1-1.

Acknowledgements We thank the anonymous reviewers for their insightful comments and suggestions.

1 Introduction

Reconstructing the sea level for the past is of paramount importance for understanding the influences of climate change. Two types of observational data are often used for this task: (1) data from tide gauge stations, which are usually located at the sea shore, and (2) gridded altimeter data acquired from satellites. The tide gauge data is available from the 18th century from stations that are sparsely distributed globally (e.g., the RLR database given by the PSMSL contains 1 548 stations). The gridded altimeter data, which has been acquired since 1993, admits much more accurate reconstructions of the sea surface for the last 29 years. We build on the work by Nitzke et al. [24], who suggested an approach for combining these two types of data using integer linear programming techniques. The approach is to learn a plausible triangulation of the tide gauge stations for an epoch E for which the altimeter data is available, and then use that triangulation to reconstruct the sea surface in another epoch, where gauge data is available, but no altimeter data. Given the gauge and altimeter data for E , the task is to compute a minimum-error triangulation of the gauge stations, that is, a triangulation that minimizes the sum of squared differences between the reference (altimeter) data and the piecewise linear surface defined with the triangulation.

For piecewise linear surfaces, Delaunay triangulations are often chosen, since they have many desirable properties. However, they are unique and so they do not have potential for optimization. On the other hand, computing a minimum-error triangulation among the set of all triangulations can lead to badly shaped triangles, which can cause large interpolation errors for epochs other than the training epoch. Therefore, Nitzke et al. [24] suggested computing a triangulation of minimum error among all k -order Delaunay (k -OD) triangulations [16]. A k -OD triangulation consists of triangles with up to k points inside each triangle’s circumcircle ($k = 0$ corresponds to Delaunay triangles). This creates room for optimization while ensuring (reasonably) well-shaped triangles. Moreover, restricting the solution to the set of k -order Delaunay triangulations has computational advantages. Nitzke et al. [24] modeled their approach as an integer linear program (ILP) and evaluated it on the North Sea dataset with up to 40 stations and $k \leq 3$, whose locations are projected on the

plane; see Figure 1. The evaluation showed that the k -OD minimum-error triangulation is substantially more effective than the method based on the Delaunay triangulation suggested in [25] for Sea Surface Anomaly reconstructions of up to 19 years back in time.

The aim of our work is to speed up the above approach using computational geometry in order to apply it to areas of global extent (instances with up to 800 tide gauge stations).

Our contribution.

- We first show that the minimum-error triangulation problem is NP-hard and that it is even NP-hard to approximate an optimal solution.
- We discuss an alternative optimization approach to the ILP-based one by Nitzke et al. [24]. Our approach is based on the dynamic programming (DP) algorithm by Silveira and Van Krefeld [28]. The runtime of the DP algorithm depends on the Delaunay order k ; since we are only interested in small orders, we are able to calculate minimum-error order- k Delaunay triangulations for the datasets given by the sea surface reconstruction problem.
- The algorithm's runtime depends on a subgraph of the Delaunay triangulation, which we call the order- k fixed-edge graph. It is known that for order 1 the fixed-edge graph is connected [16]. We investigate the fixed-edge graph for orders $k = 2, 3$. We show that for $k = 2$ no vertex can be isolated and give an example where the fixed-edge graph is not connected. For $k \geq 3$ we give an example where $\lfloor \frac{n}{6} \rfloor$ connected components are inside a face of the fixed-edge graph, which implies exponential runtime for the algorithm. This complements the observations by Silveira et al. given in [28].
- We perform experiments with different projections of the tide gauge dataset to analyze the structure of the fixed-edge graphs for a real-world dataset. Our experiments confirm the assumption by Silveira and Van Krefeld [28] that the DP algorithm can be used to solve practical problems for medium-sized datasets, if the order is small ($k \leq 7$).
- Lastly, we perform the reconstruction task that was given in [24] for the global dataset. Our evaluation shows that on the used global dataset with up to 800 stations the quality improves with growing k , which contrasts with the findings in [24] on the local North Sea dataset with about 40 stations, where $k = 2$ consistently delivered the best reconstructions.

The paper is organized as follows. First, we outline the formal definitions of the triangulation problem in Section 2. After that, we discuss related works in Section 3. In Section 4 we present our NP-hardness proof for the minimum-error triangulation problem. Section 5 presents the DP algorithm by Silveira et al. [28] and discusses our findings regarding the fixed-edge graphs. In Section 6 we provide the application of the DP algorithm to the sea surface reconstruction problem. Finally, we give our conclusion in Section 7.

2 The triangulation problem

Let $\mathcal{S} \subset \mathbb{R}^2$ be a set of n points and $f: \mathcal{S} \rightarrow \mathbb{R}$. We call \mathcal{S} the set of triangulation points and $f(s)$ the measurement value of $s \in \mathcal{S}$. Additionally, we are given a set $\mathcal{R} \subset \text{conv}(\mathcal{S})$ of m points and a function $h: \mathcal{R} \rightarrow \mathbb{R}$. We refer to \mathcal{R} as the set of reference points and to $h(r)$ as the reference value of $r \in \mathcal{R}$.

A triangulation D of \mathcal{S} is given by a maximal set of non-crossing straight-line edges between points in \mathcal{S} . We can extend the function f on the points in $\text{conv}(\mathcal{S})$ by linearly interpolating f in every triangle. In this way we obtain a piece-wise linear function $s_D: \text{conv}(\mathcal{S}) \rightarrow \mathbb{R}$. The *minimum-error triangulation problem* asks for a triangulation D of \mathcal{S} that minimizes the squared error between the reference values and the interpolation, i.e.,

$$\text{Err}_D(\mathcal{R}) = \sum_{r \in \mathcal{R}} (s_D(r) - h(r))^2.$$

For the dynamic programming algorithm used in our approach and discussed in Section 5, we transform the minimum-error triangulation problem to the *minimum triangle-weighted triangulation problem*. Let \mathfrak{T} be the set of all $O(n^3)$ possible triangles that may be used in any triangulation of \mathcal{S} . Then we can assign the weight

$$w_T(\mathcal{R}) = \sum_{r \in T} (s_T(r) - h(r))^2$$

to every triangle $T \in \mathfrak{T}$, where s_T is the linear interpolation given by the triangle T . If we assume that no reference point lies on any triangulation edge, we get

$$\text{Err}_D(\mathcal{R}) = \sum_{r \in \mathcal{R}} (s_D(r) - h(r))^2 = \sum_{T \in D} \sum_{r \in T} (s_T(r) - h(r))^2 = \sum_{T \in D} w_T(\mathcal{R}).$$

To get rid of the previous assumption we assign points that lie on an edge \overline{uv} only to the triangles left of \overrightarrow{uv} . Points coinciding with triangulation points can be ignored.

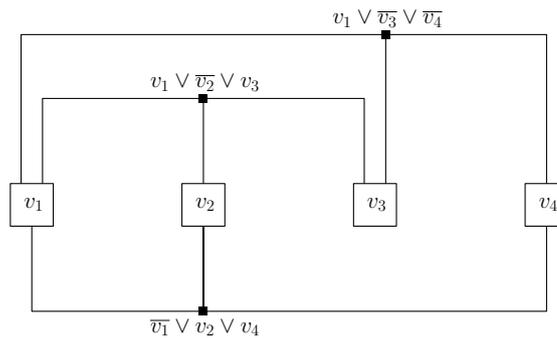
Using these weights our cost function becomes a decomposable measure as discussed by Bern and Eppstein in [6]. Broadly speaking, decomposable measures are all measures that easily allow computation using dynamic programming approaches for triangulations.

3 Related works

Sea level reconstruction. Conventional methods for sea level reconstruction use global base functions (empirical orthogonal base functions) which are *learned* within the altimeter decades [10]. Olivieri and Spada suggested the first triangulation-based reconstruction approach [25]. However, this approach does not use the altimeter data in any way and generates a Delaunay triangulation of the station data. Nevertheless, the resulting reconstruction of the sea surface was quite promising. The approach suggested by Nitzke et al. [24] marries the conventional thinking and the triangulation method. The authors proposed the use of data-dependent triangulations which were introduced in [12] by Dyn, Levin and Rippa. The particular focus of Nitzke et al. were the minimum-error triangulations. Since they also want to reconstruct the sea level in the pre-altimetry era, they formulate the reconstruction as a learning task and use higher-order Delaunay constraints, which were introduced in [16] by Gudmundsson, Hammar and van Kreveld, as regularizer.

Triangulating point sets. Triangulating point sets in the plane is a fundamental task of computational geometry. It is of high relevance for data interpolation and surface modeling tasks, where for every data point a data value (or height) is given in addition to the point's two coordinates. The Delaunay triangulation is most often applied as it optimizes several criteria and can be computed efficiently. In particular, it maximizes the minimum angle among all the angles of all the triangles. *Data-dependent triangulations* have been defined in [12] as triangulations that are computed under consideration of the data values. As optimization criteria the authors have considered (1) smoothness criteria, (2) criteria based on three-dimensional properties of the triangles, (3) variational criteria, and (4) the minimum-error criterion, which is optimized by the previously defined minimum-error triangulation.

There are many heuristics for computing data-dependent triangulations [3, 8, 12, 29], which are usually based on Lawson's edge flip algorithm [21]. For small instances, the problem can be solved to optimality based on integer linear programming [24]. There are multiple fixed-parameter-tractable algorithms using dynamic programming for the minimum-weight triangulation (MWT) problem [19, 9, 7, 4, 15] that can be adapted for decomposable measures [6]. Using problem specific structural properties the MWT problem has been solved for instances with up to 30 million points [17, 14].



■ **Figure 2** Embedding of the 3SAT formula $(\bar{v}_1 \vee v_2 \vee v_4) \wedge (v_1 \vee \bar{v}_2 \vee v_3) \wedge (v_1 \vee \bar{v}_3 \vee \bar{v}_4)$.

In [11, 27] heuristics and higher-order Delaunay constraints were used for terrain approximation. Using established techniques, exact polynomial-time algorithms can be obtained for restricted cases with higher-order Delaunay constraints [16, 28]. However, prior to our work, little was known about the complexity of computing or approximating minimum-error triangulations in the general case. For related problems some hardness results exist [2, 23].

4 Minimum-error triangulation is NP-hard

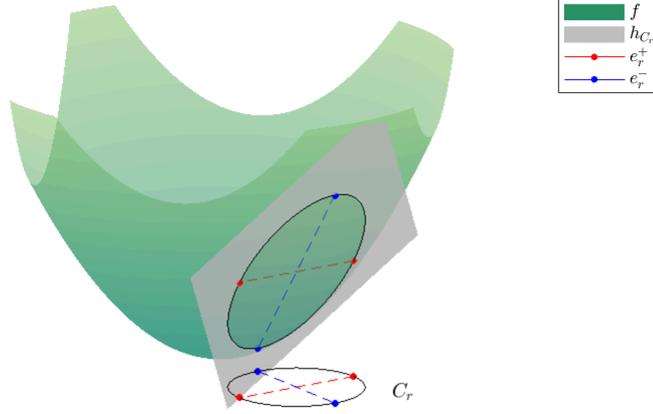
The *zero-error triangulation problem* asks for a triangulation D of \mathcal{S} with $s_D(r) = h(r)$ for all $r \in \mathcal{R}$, or equivalently $\text{Err}_D(\mathcal{R}) = 0$. We prove that this problem is NP-hard.

► **Theorem 1.** *The zero-error triangulation problem is NP-hard. Thus the minimum-error triangulation problem cannot be approximated within any multiplicative factor in polynomial time unless $P=NP$.*

We prove this by a reduction from the planar 3SAT problem, which is NP-complete [22]. An instance of this problem can be embedded into the plane, where every clause is represented by a vertex and every variable by a box placed on the horizontal axis. A box is connected to a vertex via a rectilinear edge if the respective variable is contained in the clause. For an example, see Figure 2. Such an embedding is also used, for example, in [20].

For every instance of the planar 3SAT problem we construct an instance for the zero-error triangulation problem by replacing the boxes, vertices and edges of its rectilinear embedding in the plane by a set of triangulation points and reference points. For this purpose we handle each component of the 3SAT embedding individually. We construct the *variable gadgets* which replace the boxes, the *wire gadgets*, which replace the rectilinear edges and finally the *clause gadgets* and the *negation gadgets*, where the first replace the vertices and the second can be attached to variable gadgets to handle negated variables in a clause. The combination of these gadgets then constitutes an instance to the zero-error triangulation problem.

We ensure that there are two possible zero-error triangulations on the points belonging to a variable gadget and the attached negation gadgets and wire gadgets as follows. Points from \mathcal{S} together with their measurement value can be seen as points in \mathbb{R}^3 . We ensure that they lie on a paraboloid in \mathbb{R}^3 and exploit the properties of the paraboloid (its convexity and the correspondence of planes in \mathbb{R}^3 to circles in \mathbb{R}^2) to limit possible zero-error triangulations. Any such triangulation then corresponds to the assignment of value 0 (negative) or 1 (positive) to any variable. We claim that the instance can be triangulated with zero error if and only if the 3SAT instance is solvable.



■ **Figure 3** Example of a reference point r with coupled circle C_r and its positive/negative edges crossing at r . Lifting the red and blue points to \mathbb{R}^3 , with their measurement values as third coordinate, we see that these points lie on both the paraboloid and the plane containing $(r, h_{C_r}(r))$.

4.1 Notation and local properties

Our triangulation instance consists of a set of triangulation points with integral coordinates $\mathcal{S} \subset \mathbb{Z}^2$ and a set $\mathcal{R} \subset \text{conv}(\mathcal{S})$ of reference points. The measurement value of a point $p = (p_1, p_2) \in \mathcal{S}$ is given by $f(p) = p_1^2 + p_2^2$. In contrast, reference values are not determined by one single function. Instead we define a set of functions, one for every circle in \mathbb{R}^2 , and choose for every reference point one of these functions which determines the reference value of this point. Concretely, let C be a circle around a point $x = (x_1, x_2)$ with radius ρ . We denote with $I_C = \{y \in \mathbb{R}^2 \mid \|x - y\|_2 < \rho\}$ the *interior* of C and with $O_C = \mathbb{R}^2 \setminus (C \cup I_C)$ the *exterior* of C . Here $\|\cdot\|_2$ denotes the Euclidean norm. For a reference point $r = (r_1, r_2) \in \mathcal{R}$ we define the function

$$h_C(r) = 2x_1r_1 + 2x_2r_2 - x_1^2 - x_2^2 + \rho^2.$$

The function graph of f is the unit paraboloid $\{(p_1, p_2, p_1^2 + p_2^2) \mid (p_1, p_2) \in \mathbb{R}^2\}$ and the function graph of h_C is the plane containing the lifting of C onto the paraboloid (Figure 3).

Every point $r \in \mathcal{R}$ is then *coupled* to a circle, which we denote by C_r . It will be defined during the construction of the gadgets and determines the reference value $h(r) = h_{C_r}(r)$. Let an edge $e = \overline{st}$ denote the convex hull of two points (its vertices) $s, t \in \mathbb{R}^2$. For each $r \in \mathcal{R}$ we define a *positive edge* e_r^+ and a *negative edge* e_r^- both having triangulation points lying on C_r as endpoints and intersecting each other at r (i.e., $e_r^+ \cap e_r^- = \{r\}$). Figure 3 shows the whole construction. We say for a triangulation D that the *signal* at $r \in \mathcal{R}$ is *positive* if D contains edge e_r^+ and *negative* if it contains e_r^- , otherwise we call it *ambiguous*. Similarly for every set $M \subset \mathcal{R}$ we call D *positive* on M if the signal at all $r \in M$ is positive and *negative* on M if the signal at all $r \in M$ is negative. The error incurred by D on M is given by

$$\text{Err}_D(M) = \sum_{r \in M} (s_D(r) - h(r))^2.$$

A triangle T is the convex hull of three points $s, t, u \in \mathbb{R}^2$, which we call the vertices of T . We say that a triangle T is in D if all of its edges $\overline{st}, \overline{tu}, \overline{us}$ are in D and T does not contain further points from \mathcal{S} , i.e., $T \cap \mathcal{S} = \{s, t, u\}$. We say that $r \in \mathcal{R}$ is *represented with zero error* by T if $r \in T$ and the value at r of the linear interpolation of f on T equals $h(r)$.

► **Lemma 2.** *Let r be a point of \mathcal{R} and let $T \subset \mathbb{R}^2$ be a triangle with vertices s, t, u and $r \in \text{conv}(\{s, t, u\} \cap C_r)$. Then r is represented with zero error by T .*

If the 3SAT instance is satisfiable, we argue that there is a triangulation containing one of e_r^\pm for every reference point r . Lemma 2 states that such a triangulation has in fact zero error (see also Figure 3). To represent r with zero error in any other way, we need at least one triangulation point inside and one outside C_r . This follows from the convexity of f .

► **Lemma 3.** *Let $T \subset \mathbb{R}^2$ be a triangle with vertices s, t, u representing $r \in \mathcal{R}$ with zero error. If $r \notin \text{conv}(\{s, t, u\} \cap C_r)$, then $\{s, t, u\}$ has a non-empty intersection with I_{C_r} and O_{C_r} .*

We guarantee during the construction that only few triangulation points lie in I_{C_r} for each reference point r . With a concise case analysis we rule out that any of them can be used together with a point in O_{C_r} to form a triangle that represents r with zero error, which limits the choice to triangles containing one of e_r^\pm . This ensures that every zero-error triangulation yields a solution to the 3SAT instance.

Our triangulation instance contains a set of *mandatory edges* that we require to be part of any feasible triangulation of \mathcal{S} . Mandatory edges are not part of the zero-error triangulation problem as defined in Section 2, but they can be eliminated by an additional construction.

4.2 The gadgets

At the core of our reduction lies the design of the gadgets that constitute the triangulation instance. Before we dedicate ourselves to the more complicated gadgets we construct smaller elements called *bits* and *segments* which then are combined into the larger gadgets.

A *bit* at $r \in \mathbb{Z}^2$ occupies a small construction around the central point r , which is also the only reference point of this bit, and can be oriented either horizontally or vertically. We describe the horizontal bit. Point r is coupled to a circle C_r which is centered on r and has radius $\sqrt{2}$. The integer grid points on this circle, that is, the points $r + (\pm 1, \pm 1)$, are triangulation points. Moreover $r + (0, 1)$ and $r + (0, -1)$ are triangulation points, whereas $r + (-2, 0)$, $r + (-1, 0)$, $r + (1, 0)$ and $r + (2, 0)$ are *not*. Therefore, we call the latter points *forbidden*. Furthermore we define the positive and negative edge as

$$e_r^+ = \text{conv}(r + (-1, -1), r + (1, 1)), \quad e_r^- = \text{conv}(r + (-1, 1), r + (1, -1)).$$

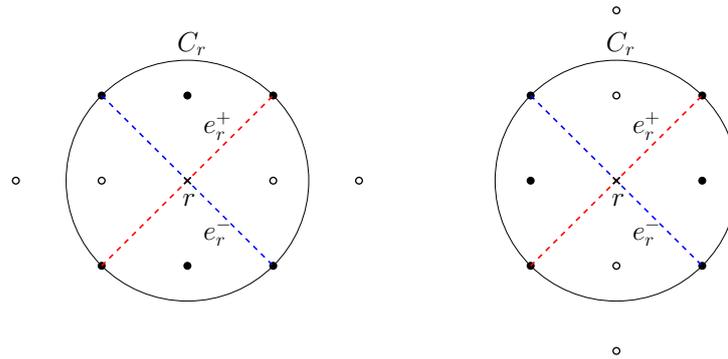
As $r + (\pm 1, \pm 1) \in C_r$, any triangle containing either e_r^+ or e_r^- represents r with zero error by Lemma 2. For the vertical bit we switch the definition of the positive and negative edge and rotate the whole construction by $\frac{\pi}{2}$. Figure 4 illustrates both constructions.

► **Lemma 4.** *Suppose the instance contains a bit at r . If $\mathcal{S} \subset \mathbb{Z}^2$ and \mathcal{S} does not contain forbidden points of the bit, any triangulation D of \mathcal{S} with $\text{Err}_D(r) = 0$ contains one of e_r^\pm .*

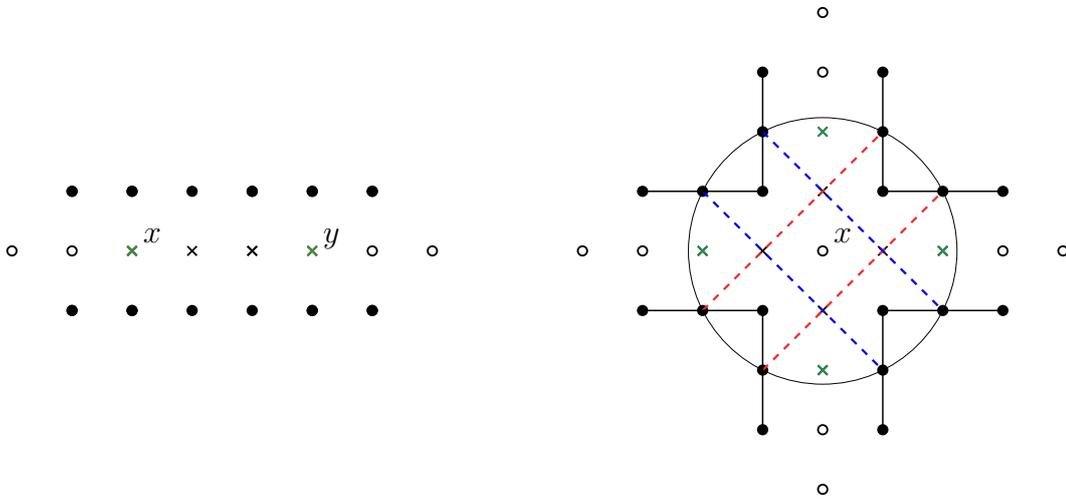
The next larger components are the *wire segment* and the *multiplier segment*, which we build from bits. They can be combined at specified reference points, which we call *anchor points*. These points are always reference points of bits.

A *wire segment* connects two points $x, y \in \mathbb{Z}^2$ lying on the same horizontal or vertical line. We place a horizontal or vertical bit on x, y and all integral points lying between these on the line connecting x and y . The anchor points of this segment are x, y .

A *multiplier segment* at a point $x \in \mathbb{Z}^2$ consist of two horizontal bits at $x \pm (2, 0)$ and two vertical bits at $x \pm (0, 2)$. These four points are simultaneously anchor points. Furthermore we add four inner reference points $x \pm (0, 1), x \pm (1, 0)$ whose coupled circle is of radius $\sqrt{5}$ and centered around x . So the circle contains the points $x + (\pm 2, \pm 1), x + (\pm 1, \pm 2)$. Figure 5



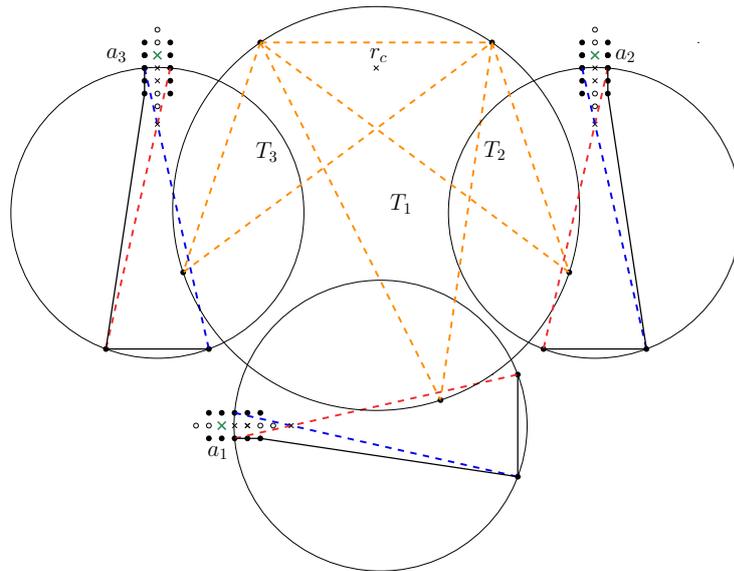
■ **Figure 4** The (horizontal/vertical) bit at r with the positive edge in red and the negative edge in blue. The black points are triangulation points and the white points are forbidden.



■ **Figure 5** Example of a horizontal wire segment on the left and a multiplier segment with mandatory edges on the right. The red or blue edges indicate the positive or negative edges of the crossing points, respectively. All white points and all reference points are forbidden. The green points are anchor points.

shows the wire segment and the multiplier segment including mandatory edges and the positive/negative edges of the inner reference points. To obtain the larger variable gadget and wire gadget we combine wire segments with multiplier segments. Two segments can be combined if they share a common anchor point. By the combination of two segments we mean the union of their reference points and triangulation points. A point is forbidden in the combination if it is forbidden in at least one of the segments. Thus it is not allowed to combine two segments if a triangulation point of one is forbidden in the other. The set of anchor points of the combination is defined as the symmetric difference of anchor point sets of both segments. This way we can combine arbitrarily many segments.

Remember that the *wire gadget* replaces the rectilinear edges of the 3SAT embedding, so it has to connect two points $x = (x_1, x_2), y = (y_1, y_2) \in \mathbb{Z}^2$. It consists of a multiplier segment placed on either (x_1, y_2) or (y_1, x_2) to form a corner, which is connected on two of its anchor points via two wire segments to both x and y . A *variable gadget* at $v \in \mathbb{Z}^2$ consists of ℓ multiplier segments at sufficiently large distance $\alpha \in \mathbb{Z}$, which we do not specify further. Here ℓ denotes the number of clauses. Concretely, we place a multiplier segment on each of



■ **Figure 6** The clause gadget, where the red/blue edges indicate the positive/negative edges of the crossing points. The triangles T_1, T_2, T_3 are orange and the anchor points a_1, a_2, a_3 green.

the points $v + (k\alpha, 0)$ with $0 \leq k \leq \ell - 1$ and connect them via horizontal wire segments at their anchor points. The multiplier segments ensure that the gadget can later be connected at its anchor points to multiple clause gadgets. We observe that the described combinations of segments for both gadgets are allowed and that they have the following crucial property.

► **Lemma 5.** *Suppose the instance contains a wire/variable gadget and let $\tilde{\mathcal{R}}$ be the reference points of this gadget. If $\mathcal{S} \subset \mathbb{Z}^2$ and \mathcal{S} does not contain forbidden points of the gadget, any triangulation D of \mathcal{S} with $\text{Err}_D(\tilde{\mathcal{R}}) = 0$ is either positive or negative on $\tilde{\mathcal{R}}$.*

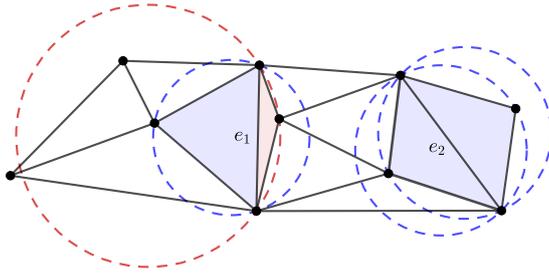
Now we define the *clause gadget* at a point $c \in \mathbb{Z}^2$, which combines three signals. To this end we add a reference point $r_c = c + (0, 11)$. Instead of a positive/negative edge it comes with three triangles T_1, T_2, T_3 whose vertices lie on C_{r_c} , each triangulating r_c with zero error. The clause gadget can be connected to other gadgets at three anchor points a_1, a_2, a_3 . With an additional construction we block the triangle T_i if the signal at a_i is positive for $i = 1, 2$ and T_3 if the signal at a_3 is negative. For the construction we refer to Figure 6 and [5].

► **Lemma 6.** *Suppose the instance contains a clause gadget and let $\tilde{\mathcal{R}}$ be its reference points. If $\mathcal{S} \subset \mathbb{Z}^2$ and \mathcal{S} does not contain forbidden points of the gadget, any triangulation D of \mathcal{S} with $\text{Err}_D(\tilde{\mathcal{R}}) = 0$ must be negative on one of the anchor points a_1, a_2 or positive on a_3 .*

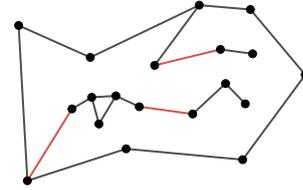
The last gadget, the *negation gadget*, is discussed in the full version [5]. It is constructed out of wires, multipliers and simplified clause gadgets. Finally, we replace the mandatory edges by an additional construction and argue that all gadgets keep their crucial properties. Using them we construct the zero-error triangulation instance and prove Theorem 1 in [5].

5 Higher-order Delaunay optimization

In the previous section we established that finding a minimum-error triangulation is NP-hard. Moreover, the experiments in [24] by Nitzke et al. suggest, that general minimum-error triangulations do not yield the most promising reconstructions of the sea surface. In their paper they used higher-order Delaunay (HOD) triangulations which allow a trade-off between a well shaped triangulation and a good approximation of the training dataset.



■ **Figure 7** A 2-OD triangulation; in blue the 1-OD and in red the 2-OD triangles; e_1 is a useful 2-OD edge and e_2 is a useful 1-OD edge.



■ **Figure 8** In black a (degenerate) polygon with connected components; in red one set H of connections.

In this section we summarize the algorithm given by Silveira et al. in [28]. Additionally, we extend upon their work by investigating the fixed-edge graphs in more detail.

We only consider point sets \mathcal{S} in general position, i.e., no four points lie on a circle and we denote the circle defined by three vertices $u, v, w \in \mathcal{S}$ by $C(u, v, w)$. A triangle $T_{u,v,w}$ is called an *order- k Delaunay (k -OD) triangle*, if $C(u, v, w)$ contains at most k points from \mathcal{S} in the interior. A triangulation is called *k -OD triangulation*, if all of its triangles have order k and an edge is called *useful k -OD edge*, if some k -OD triangulation of \mathcal{S} uses it; see Figure 7.

The minimum-error measure $\text{Err}_D(\mathcal{R})$ can be optimized using dynamic programming, since it is decomposable after pre-processing the triangle weights; see [6] for a formal definition. The well known DP algorithm that was independently proposed by Klincsek in [19] and Gilbert in [15] can be used to optimize polygon triangulations for decomposable measures in $O(n^3)$ time. In [28] the runtime of the DP algorithm is improved to $O(nk^2)$, if the algorithm only considers pre-processed k -OD edges and triangles instead of all possible ones.

Furthermore, Silveira et al. [28] extend the algorithm to the class of polygons P containing h connected components C_1, \dots, C_h ; see Figure 8. The algorithm performs an exhaustive search on a collection \mathcal{H} of sets of edges H , such that the planar graph $\bigcup_i C_i \cup P \cup H$ is connected for each $H \in \mathcal{H}$ and at least one H is used in the optimal triangulation. One of the main results in [28] is the existence of such a collection with size $O(k)^h$.

► **Theorem 7** (from [28]). *An optimal k -OD triangulation with respect to $\text{Err}_D(\mathcal{R})$ of a (degenerate) polygon with n boundary vertices and $h \geq 1$ components inside can be computed in $O(kn \log n) + O(k)^{h+2}n$ expected time.*

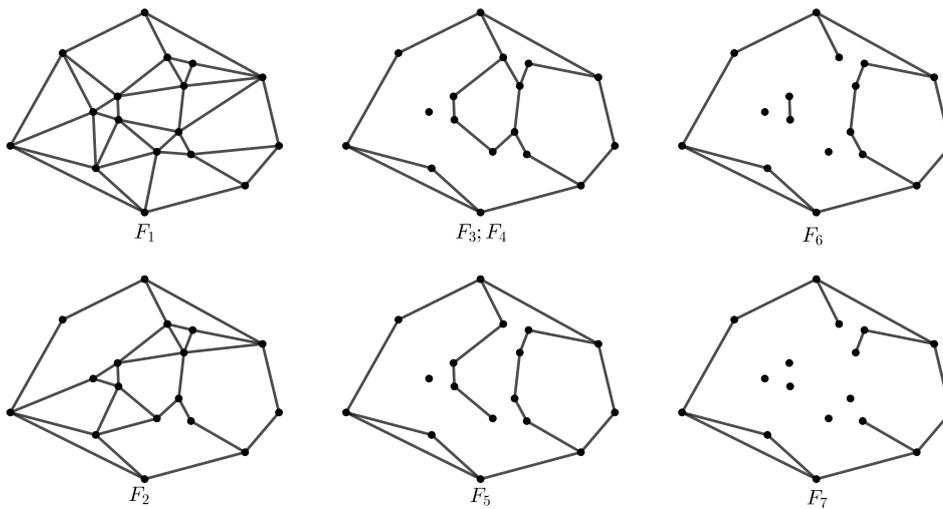
We can apply this algorithm to point sets by finding subgraphs F of the optimal triangulation [9, 28] and applying the DP algorithm to the faces of F .

5.1 The order- k fixed-edge graph

A subgraph that is naturally given by HOD constraints is the fixed-edge graph which was first discussed in [28]. The *order- k Delaunay (k -OD) fixed-edge graph* F_k of a pointset \mathcal{S} is given by all useful k -OD edges that are not intersected by any other useful k -OD edge.

► **Observation 8.** *Let \mathcal{S} be a set of n points. Let DT denote the Delaunay triangulation. We have $DT = F_0 \supset F_1 \supset F_2 \supset \dots \supset F_m = \dots = F_n \supset \text{conv}(\mathcal{S})$ for some $m \leq n$.*

In Figure 9 a sequence of fixed-edge graphs is illustrated. F_k decomposes the pointset into degenerate polygons P_1, \dots, P_m that may contain some connected components. An example is given in Figure 10. We can compute optimal solutions D_i for all P_i with the DP



■ **Figure 9** A sequence of fixed-edge graphs F_1, \dots, F_7 for an example point set.

algorithm. Since $\text{Err}_D(\mathcal{R})$ is decomposable, the optimal triangulation of \mathcal{S} is given by $\bigcup_i D_i$. Therefore, the runtime of the algorithm is dominated by the polygon with the maximum number of connected components c_{\max} . The application of Theorem 7 results in:

► **Corollary 9.** *An optimal k -OD triangulation of a point set \mathcal{S} with respect to $\text{Err}_D(\mathcal{R})$ can be computed in $O(kn \log n) + O(k)^{c_{\max}+2}n$ expected time.*

Next, we give some theoretical results with respect to the structure of F_2 and F_3 .

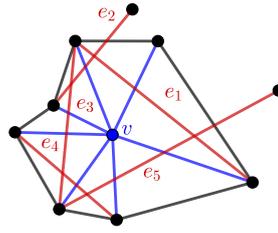
Let $v \in \mathcal{S}$ be a triangulation point. We call the graph N given by all edges of its incident Delaunay triangles its *Delaunay neighbourhood*, all of its incident edges in N its *connecting edges* and all other edges of N its *boundary edges*. A useful 2-OD edge that intersects a connecting edge is called *separation edge*; see Figure 11.

► **Theorem 10.** *Let \mathcal{S} be a set of points. Then every vertex in F_2 is adjacent to at least one other vertex of \mathcal{S} .*

Proof. (Sketch; the complete proof is given in the full version of the paper [5]) It is sufficient to prove that for every vertex $v \in \mathcal{S}$ at least one connecting edge cannot be intersected by a separation edge. For the sake of contradiction we assume that there exists a set E of separation edges such that every connecting edge is intersected by at least one $e \in E$.



■ **Figure 10** The decomposition of a fixed-edge graph into polygons. We have $c_1 = 4$, $c_2 = 0$, $c_3 = 1$ and $c_4 = 1$ for the number of components in each polygon. Thus, we have $c_{\max} = 4$. Note that the component inside P_4 is not counted towards c_3 , but to c_4 .



■ **Figure 11** The Delaunay Neighbourhood of a point v and a cycle of separation edges given in red.

In a first step we can prove that at least one endpoint of any $e \in E$ must be part of the Delaunay neighbourhood of v . Additionally, we can show that no boundary edge \overline{uw} can be intersected by a separation edge for \overline{vu} and a separation edge for \overline{vw} . These observations imply that we can order the edges in E , such that for all i the separation edge e_i intersects e_{i-1} and e_{i+1} , i.e., the separation edges form a cycle as depicted in Figure 11.

Next, we show that every pair of consecutive separation edges $(\overline{u_i v_i}, \overline{u_{i+1} v_{i+1}})$ must satisfy a special property, i.e., it must hold that $u_{i+1} \in C(u_i, v_i, v_{i+1})$ and $v_{i+1} \in C(u_i, v_i, u_{i+1})$. Finally, we show that this is not possible which leads to a contradiction. ◀

It is well known [28, 16] that F_1 is connected ($c_{\max} = 0$). Silveira et al. stated in [28] that for $k > 1$ the value c_{\max} can be larger than 0. But their experiments do not yield any example for which F_2 is not connected. We complement the discussion by such an example. Additionally, we show for all $k \geq 3$ there are examples with $c_{\max} \in \Omega(n)$.

► **Observation 11.**

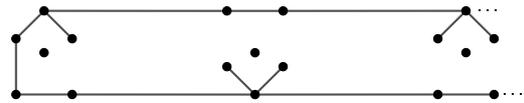
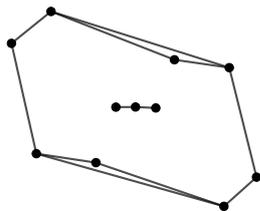
- *There exist point sets with $c_{\max} > 0$ for F_2 ; see Figure 12.*
- *For every n and $k \geq 3$ there are point sets of size n with $c_{\max} = \lfloor \frac{n}{6} \rfloor$ for F_k ; see Figure 13.*

Open question. Is there a constant d , such that F_2 has $c_{\max} \leq d$ for every point set?

Practical implications. Our results are interesting from a theoretical point of view, but the experiments in [28] with random point sets by Silveira et al. and also our own preliminary experiments indicate that for practical datasets c_{\max} is small for $k \leq 7$. Next, we confirm this assumption for the tide gauge dataset which is used for the sea surface reconstruction.

6 Experiments

We start this section by discussing the datasets. Next, we discuss the fixed-edge graphs of the tide gauge dataset. Afterwards, we provide the reconstruction process and our experimental setup. Finally, we present our results regarding the runtime and quality.



■ **Figure 12** An example with disconnected F_2 . ■ **Figure 13** An example with $c_{\max} = \frac{n}{6}$ for F_3 .

6.1 The datasets

The triangulation points for the minimum-error triangulation problem are given by the monthly tide-gauge time series from the Permanent Service for Mean Sea Level (PSMSL) [26], which is further discussed in [18]. We use the revised local reference (RLR) datasets. Furthermore, we remove some stations which do not have any values in our time-frame. This results in a dataset with 1502 stations, but not all of them record monthly. Thus, we only use between 513 and 804 different stations at once for a reconstruction.

As reference data \mathcal{R} we use the satellite altimeter datasets provided by the ESA Sea Level Climate Change Initiative (SLCCI), which are given in [13] and are further discussed in [1]. They are given as monthly gridded sea level anomalies with a spatial resolution of 0.25 degrees and are available for the timespan January 1993 to December 2015.

We assume that both datasets are given in radial coordinates. Since we focus on planar triangulations, we need to use a global map projection. We chose the Lambert azimuthal projection (LAP) which unfolds the sphere onto the plane starting at an anchor point (λ_0, ϕ_0) . For our experiments the LAP has one advantage: The projection results in significantly different distributions of the stations for sufficiently different anchor points (λ_0, ϕ_0) . This allows us to perform the fixed-edge graph experiments for a wide variety of point distributions.

It is important to note that the experiments in this paper focus on the runtime of the DP algorithm for a real world application. Thus, we only de-mean the tide gauge data as discussed in [24] and do not apply any additional corrections.

6.2 The fixed-edge graphs of the tide gauge set

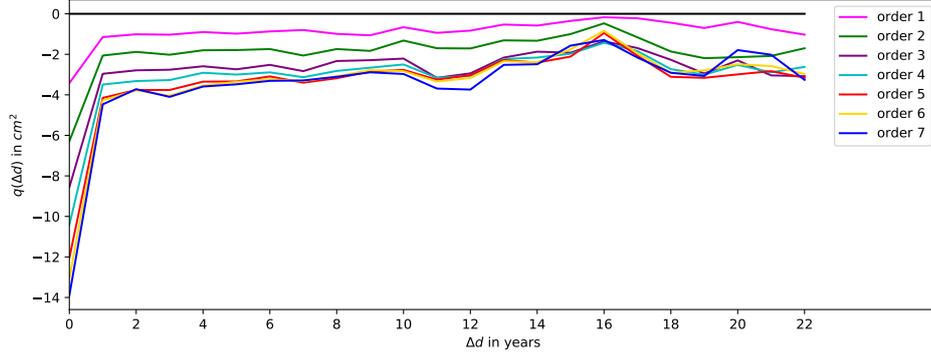
For our experiments with respect to the fixed-edge graphs we use the complete RLR dataset, i.e., all 1502 stations. We use the LAP with anchors (λ_0, ϕ_0) on an uniform 2-D 20×20 grid to generate 400 distributions of the dataset. In Table 1 the experiments are summarized. The values $\text{avg}_{c_{\max}}$ are given by the average value of c_{\max} over all samples. Additionally, we have min and max that depict the minimal and maximal value of c_{\max} for all samples. The results roughly coincide with the experiments performed on random point sets by Silveira et al. in [28] and our own preliminary experiments. The experiments suggest, that we can expect the DP algorithm to compute optimal solutions for $k \leq 7$ in reasonable time. Since Nitzke et al. suggest very small k for the reconstruction in [24], these experiments are promising.

6.3 Sea surface reconstruction

The reconstruction process can be summarized as follows: We learn a minimum-error triangulation D in some epoch i and then use it to reconstruct the sea surface at some other point in time j , by using the triangulation D with the height values of epoch j . Since not all tide gauge stations provide data for every epoch i , we need to consider the set G^{ij} which is given by all stations that have reasonable values for epoch i as well as for j . We denote the optimal triangulation using G^{ij} and the reference points A_i by D_M^{ij} . For comparison we use the Delaunay triangulation D_D^{ij} of the set G^{ij} which has already been successfully used for

■ **Table 1** The average of c_{\max} and the min/max value of c_{\max} for the projections of the RLR data.

	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	$k = 9$
$\text{avg}_{c_{\max}}$	0.00	0.00	0.45	1.20	2.05	3.68	7.11	15.88	33.16
min/max	0/0	0/0	0/2	0/3	1/5	2/12	3/18	6/38	11/82



■ **Figure 14** Averaged $q(\Delta d)$ of our approach w.r.t. the epoch difference Δd for different order k .

the sea surface reconstruction task in [25]. If we have altimeter data available for epoch j , we can evaluate the quality of our approximation. Overall the reconstruction for epoch j using i and order k can be performed as follows:

1. Compute the set G^{ij} and the k -OD triangles \mathfrak{T}^{ij} as described in [28].
2. Compute the weights $w_T(A_i)$ of all $T \in \mathfrak{T}^{ij}$ with respect to A_i as discussed in Section 2.
3. Compute the optimal k -OD triangulation D_M^{ij} with the DP algorithm given in Section 5 and also compute the Delaunay triangulation D_D^{ij} .
4. Evaluate the quality of the triangulations with respect to A_j .

For the evaluation we compute the *empirical variance* of a triangulation

$$\sigma_{ij}^2(D) = \frac{1}{n-1} \sum_{T \in D} \sum_{a \in A_j, a \in T} (s_T(a) - h_j(a))^2,$$

where n is the number of altimeter points in $\text{conv}(D)$. Note that this is exactly the average minimum error. Additionally, we define the *variance reduction* of a reconstruction by

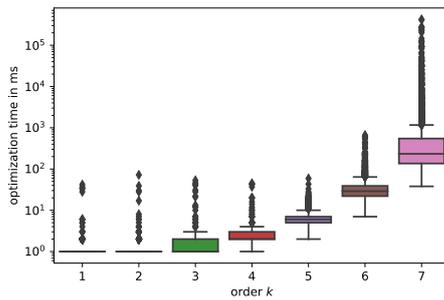
$$\Delta\sigma_{ij}^2 = \sigma_{ij}^2(D_M^{ij}) - \sigma_{ij}^2(D_D^{ij}).$$

Next, we can group together reconstructions for epochs i, j and i', j' where $|i - j| = |i' - j'|$. This allows us to define the *average variance reduction* of a temporal difference Δd by

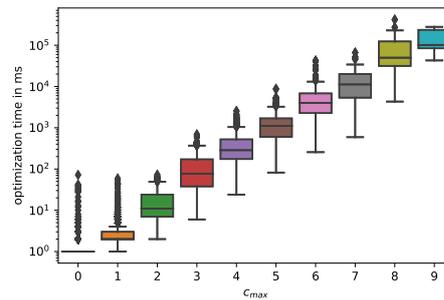
$$q(\Delta d) = \frac{1}{|\mathcal{D}(\Delta d)|} \sum_{(i,j) \in \mathcal{D}(\Delta d)} \Delta\sigma_{ij}^2.$$

The set $\mathcal{D}(\Delta d)$ is given by all tuples (i, j) with $|i - j| = \Delta d$. Using the temporal difference, we can investigate how far back in time our optimized triangulation outperforms the Delaunay triangulation (DT). Nitzke et al.[24] noticed that q has a seasonal behaviour, i.e., q has local maxima every 12 month. Thus, we only use datasets with $j = i \pm 12l$ with $l \in \mathbb{N}$ for the reconstruction. A more in depth discussion of the evaluation methods can be found in [24].

Reconstruction quality. For all of the experiments we choose an LAP anchored in the Atlantic Ocean, namely $(-40, 16)$. We compute all possible reconstructions for epochs i and j with $i \geq j$ for the orders $k \leq 7$, i.e., we use every epoch i for training and validate the learned triangulation on all possible epochs j with $j = i - 12l$. Next, we group them with respect to Δd . In Figure 14 the $q(\Delta d)$ values are depicted. Recall that our approach performs better than the DT, if $q(\Delta d) < 0$. It should be mentioned, however, that for $\Delta d \geq 18$ the quality of the experiments deteriorates, since only few samples span this epoch difference.



■ **Figure 15** Optimization time depending on the order.



■ **Figure 16** Optimization time depending on c_{\max} .

Note that the variance reductions for $\Delta d = 0$ are far better than for larger Δd , since the reconstruction epoch is the same as the training epoch. The variance reductions for order 1 and order 2 are smoother, but also worse than the ones for higher orders. For $\Delta d > 10$ the variance reductions for the orders 3–6 are very similar and even order 7 is comparable. The aforementioned orders also share local extrema at $\Delta d = 10, 11, 18, 20$. For order 7 the extrema become more pronounced which leads to better minima but also to worse maxima. Note that calculating the empirical variances $\sigma_{ij}^2(D_D^{ij})$ for all epochs yields values between 80cm^2 and 120cm^2 . Hence, for example, an absolute variance reduction of 2cm^2 roughly coincides with a relative variance reduction of 2%.

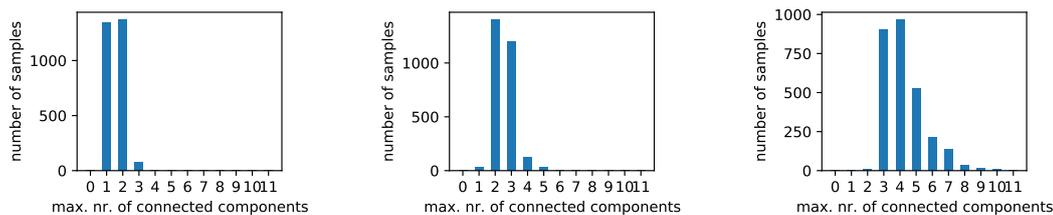
The overall variance reduction gets better for higher orders. This is contrary to the results by Nitzke et al. [24], who suggested $k = 1, 2$ for the reconstruction. This difference may have geometric reasons, i.e., the points in the North Sea dataset used in [24] more or less trace a polygon without inner points and our global datasets have a more arbitrary distribution. Moreover, the LAP distorts distances as well as angles which may also contribute to the different results for the local and global datasets.

Runtime. For the experiments we used a machine with an *AMD Ryzen 5 3600 6-Core Processor* clocked at *4.4 GHz* and *16 GB RAM*. We did not implement the geometric pre-processing as discussed in [16]. Our pre-processing has roughly cubic runtime (3–4 seconds per reconstruction). For larger orders k we expect the optimization to dominate the runtime.

The optimization time with respect to the order is given in Figure 15. Note that the optimization time for $k \leq 5$ is at most 30ms. For $k = 6$ the average runtime is still low with roughly 50ms. For $k = 7$ most datasets can be optimized in a few seconds, but some need around 20 minutes for the optimization and five datasets reach a cut-off time of one hour.

The box-plot in Figure 16 depicts the runtime with respect to the number of connected components c_{\max} . The logarithmic scaling nicely illustrates the exponential increase. If we also consider the distribution of c_{\max} for the different datasets and orders, we can easily connect the two box-plots. For $k \leq 4$ all of the datasets have $c_{\max} \leq 2$. Thus, the maximal runtime for orders $k \leq 4$ matches the worst runtime for $c_{\max} \leq 2$. For orders $k = 5, 6, 7$ the c_{\max} distributions are illustrated in Figure 17. Note that for $k = 5$ and $k = 6$ most datasets still have $c_{\max} \leq 2$ which results in the very low average runtime. For $k = 7$ the distribution starts to shift towards higher c_{\max} which results in the higher average runtime.

In summary, our experiments show that for our datasets we can compute k -OD min-error triangulations for $k \leq 6$ and also for $k = 7$ except for a few samples in reasonable time.



■ **Figure 17** The c_{\max} distribution of the reconstruction datasets for orders $k = 5, 6, 7$.

7 Conclusion

We prove that it is NP-hard to approximate an optimal solution to the minimum-error triangulation problem. Our results also imply the inapproximability of the following generalization: minimizing the distance between s_D and h on \mathcal{R} for any metric on \mathbb{R}^m , especially the L_p -metric $(\sum_{r \in \mathcal{R}} |s_D(r) - h(r)|^p)^{1/p}$ for $p \in [1, \infty)$ and the L_∞ -metric $\max_{r \in \mathcal{R}} |s_D(r) - h(r)|$. Additionally, we apply the dynamic programming algorithm by Silveira et al. [28] to minimum-error triangulations and extend their experiments, regarding the fixed edges to a real world dataset. We further investigate the fixed-edge graphs for order $k = 2$ and give a worst-case example for $k = 3$. Finally, we perform the dynamic sea surface reconstruction similar to Nitzke et al. in [24] for significantly larger datasets using a new algorithmic approach.

A future line of research is the extension of the dynamic programming algorithm to datasets on the sphere, i.e., spherical triangulations. This would allow a more realistic reconstruction of the global dynamic sea surface. A combination with ILP techniques will be a further step [14]. It would also be interesting to include multiple datasets for the learning of the reconstruction triangulation. We believe that our work will open the door for the application of optimal triangulation approaches to the problem of multi-decadal global sea level reconstructions from tide gauge data. In addition, with the growing amount of satellite and in-situ ocean sensors (buoys, Argo floats, ...) we see potential for a more widespread application of triangulation methods in generating gridded ocean data products.

References

- 1 M. Ablain, A. Cazenave, G. Larnicol, M. Balmaseda, P. Cipollini, Y. Faugère, M. J. Fernandes, O. Henry, J. A. Johannessen, P. Knudsen, O. Andersen, J. Legeais, B. Meyssignac, N. Picot, M. Roca, S. Rudenko, M. G. Scharffenberg, D. Stammer, G. Timms, and J. Benveniste. Improved sea level record over the satellite altimetry era (1993–2010) from the Climate Change Initiative project. *Ocean Science*, 11(1):67–82, 2015. doi:10.5194/os-11-67-2015.
- 2 Pankaj K. Agarwal and Subhash Suri. Surface approximation and geometric partitions. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '94, pages 24–33, USA, 1994. Society for Industrial and Applied Mathematics.
- 3 Lyuba Alboul, Gertjan Kloosterman, Cornelis Traas, and Ruud van Damme. Best data-dependent triangulations. *Journal of Computational and Applied Mathematics*, 119(1):1–12, 2000. doi:10.1016/S0377-0427(00)00368-X.
- 4 Efthymios Anagnostou and Derek Corneil. Polynomial-time instances of the minimum weight triangulation problem. *Computational Geometry*, 3(5):247–259, 1993. doi:10.1016/0925-7721(93)90016-Y.
- 5 Anna Arutyunova, Anne Driemel, Jan-Henrik Haunert, Herman Haverkort, Jürgen Kusche, Elmar Langetepe, Philip Mayer, Petra Mutzel, and Heiko Röglin. Minimum-error triangulations for sea surface reconstruction. 2022. arXiv:2203.07325v1.

- 6 Marshall Bern and David Eppstein. Mesh generation and optimal triangulation. In *Computing in Euclidean Geometry*, 1992. doi:10.1142/9789814355858_0002.
- 7 Magdalene Borgelt, Christian Borgelt, and Christos Levcopoulos. Fixed parameter algorithms for the minimum weight triangulation problem. *Int. J. Comput. Geometry Appl.*, 18:185–220, June 2008. doi:10.1142/S0218195908002581.
- 8 Jeffrey L. Brown. Vertex based data dependent triangulations. *Computer Aided Geometric Design*, 8(3):239–251, 1991. doi:10.1016/0167-8396(91)90008-Y.
- 9 Siu-Wing Cheng, Mordecai J. Golin, and Jeffrey Tsang. Expected case analysis of {221}-skeletons with applications to the construction of minimum-weight triangulations. Master's thesis, Hong Kong University of Science and Technology, 1995.
- 10 John A. Church, Neil J. White, Richard Coleman, Kurt Lambeck, and Jerry X. Mitrovica. Estimates of the Regional Distribution of Sea Level Rise over the 1950–2000 Period. *Journal of Climate*, 17(13):2609–2625, July 2004. doi:10.1175/1520-0442(2004)017<2609:EOTRDO>2.0.CO;2.
- 11 Thierry de Kok, Marc van Kreveld, and Maarten Löffler. Generating realistic terrains with higher-order Delaunay triangulations. *Computational Geometry*, 36(1):52–65, 2007. Special Issue on the 21st European Workshop on Computational Geometry. doi:10.1016/j.comgeo.2005.09.005.
- 12 Nira Dyn, David Levin, and Samuel Rippa. Data Dependent Triangulations for Piecewise Linear Interpolation. *IMA Journal of Numerical Analysis*, 10(1):137–154, January 1990. doi:10.1093/imanum/10.1.137.
- 13 ESA. Sea level CCI ECV dataset: Time series of gridded sea level anomalies(sla), 2021. European Space Agency (ESA). URL: <https://catalogue.ceda.ac.uk/uuid/142052b9dc754f6da47a631e35ec4609>.
- 14 S.P. Fekete, A. Haas, Y. Lieder, E. Niehs, M. Perk, V. Sack, and C. Scheffer. On hard instances of the minimum-weight triangulation problem. In *36th European Workshop on Computational Geometry (EuroCG 2020)*, March 2020.
- 15 P. Gilbert. New results on planar triangulations. Master's thesis, University of Illinois, Coordinated Science Lab, Urbana, IL, USA, 1979.
- 16 Joachim Gudmundsson, Mikael Hammar, and Marc van Kreveld. Higher order Delaunay triangulations. *Computational Geometry*, 23(1):85–98, 2002. doi:10.1016/S0925-7721(01)00027-X.
- 17 Andreas Haas. Solving large-scale minimum-weight triangulation instances to provable optimality. In Bettina Speckmann and Csaba D. Tóth, editors, *34th International Symposium on Computational Geometry, SoCG 2018*, volume 99 of *LIPICs*, pages 44:1–44:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.SoCG.2018.44.
- 18 Simon J. Holgate, Andrew Matthews, Philip L. Woodworth, Lesley J. Rickards, Mark E. Tamisiea, Elizabeth Bradshaw, Peter R. Foden, Kathleen M. Gordon, Svetlana Jevrejeva, and Jeff Pugh. New Data Systems and Products at the Permanent Service for Mean Sea Level. *Journal of Coastal Research*, 29(3):493–504, December 2012. doi:10.2112/JCOASTRES-D-12-00175.1.
- 19 G.T. Klincsek. Minimal triangulations of polygonal domains. In Peter L. Hammer, editor, *Combinatorics 79*, volume 9 of *Annals of Discrete Mathematics*, pages 121–123. Elsevier, 1980. doi:10.1016/S0167-5060(08)70044-X.
- 20 Donald E. Knuth and Arvind Raghunathan. The problem of compatible representatives. *SIAM Journal on Discrete Mathematics*, 5(3):422–427, 1992.
- 21 Charles L. Lawson. Software for C^1 surface interpolation. In John R. Rice, editor, *Mathematical Software*, pages 161–194. Academic Press, 1977. doi:10.1016/B978-0-12-587260-7.50011-X.
- 22 David Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11:329–343, 1982.
- 23 Wolfgang Mulzer and Günter Rote. Minimum-weight triangulation is NP-hard. *Journal of the ACM*, 55(2):1–29, May 2008. doi:10.1145/1346330.1346336.

7:18 Minimum-Error Triangulations for Sea Surface Reconstruction

- 24 Alina Nitzke, Benjamin Niedermann, Luciana Fenoglio-Marc, Jürgen Kusche, and Jan-Henrik Haunert. Reconstructing the dynamic sea surface from tide gauge records using optimal data-dependent triangulations. *Computers & Geosciences*, 157:104920, 2021. doi:10.1016/j.cageo.2021.104920.
- 25 Marco Olivieri and Giorgio Spada. Spatial sea-level reconstruction in the Baltic Sea and in the Pacific Ocean from tide gauges observations. *Annals of Geophysics*, 59(3), 2016. doi:10.4401/ag-6966.
- 26 Permanent service for mean sea level (PSMSL), 2021. Retrieved 19 Apr 2021 from <http://www.psmsl.org/data/obtaining/>.
- 27 Natalia Rodríguez and Rodrigo I. Silveira. Implementing data-dependent triangulations with higher order Delaunay triangulations. *ISPRS International Journal of Geo-Information*, 6(12), 2017. doi:10.3390/ijgi6120390.
- 28 Rodrigo I. Silveira and Marc van Kreveld. Optimal higher order Delaunay triangulations of polygons. *Computational Geometry*, 42(8):803–813, 2009. Special Issue on the 23rd European Workshop on Computational Geometry. doi:10.1016/j.comgeo.2008.02.006.
- 29 Kai Wang, Chor-Pang Lo, George A. Brook, and Hamid R. Arabnia. Comparison of existing triangulation methods for regularly and irregularly spaced height fields. *International Journal of Geographical Information Science*, 15(8):743–762, 2001. doi:10.1080/13658810110074492.