2012

Quadrilateral meshes with provable angle bounds

F. B. Atalay

S. Ramaswami

Dianna Xu
Bryn Mawr College, dxu@brynmawr.edu

Custom Citation
Atalay, F.B., Ramaswami, S. & Xu, D. Engineering with Computers (2012) 28: 31-56.

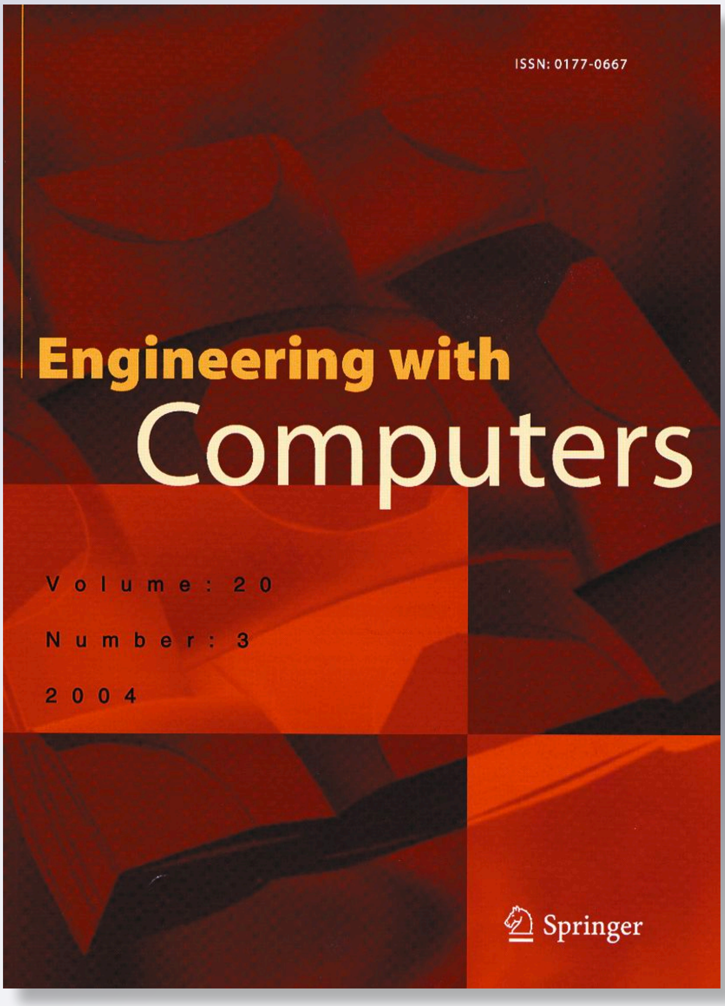
Quadrilateral meshes with provable angle bounds

F. Betul Atalay, Suneeta Ramaswami & Dianna Xu

Engineering with Computers
An International Journal for Simulation-Based Engineering

ISSN 0177-0667
Volume 28
Number 1

Engineering with Computers (2012)
28:31-56
DOI 10.1007/s00366-011-0215-0



Springer

Engineering with Computers (2012) 28:31–56
DOI 10.1007/s00366-011-0215-0

ORIGINAL ARTICLE

Quadrilateral meshes with provable angle bounds

F. Betul Atalay · Suneeta Ramaswami ·
Dianna Xu

Received: 25 May 2010 / Accepted: 1 March 2011 / Published online: 21 April 2011

Abstract In this paper, we present an algorithm that utilizes a quadtree data structure to construct a quadrilateral mesh for a simple polygonal region in which no newly created angle is smaller than 18.43°$(=\arctan(\frac{1}{3}))$ or greater than 171.86°$(=135^\circ + 2\arctan(\frac{1}{3}))$. This is the first known result, to the best of our knowledge, on a direct quadrilateral mesh generation algorithm with a provable guarantee on the angles.

Keywords Mesh-generation · Quadrilateral · Angle bounds · Quadtree

1 Introduction

The generation of quadrilateral meshes with provable guarantees on mesh quality poses several interesting open questions. While theoretical properties of triangle meshes are well understood [5, 8, 9, 10, 17, 18, 20, 21], much less is known about algorithms for *provably good* quadrilateral meshes. Analysts, however, prefer quadrilateral and hexahedral meshes for better solution quality in numerous applications [1, 3, 7, 15, 22]. This is because they have better convergence properties, and hence lower approximation errors, in finite element methods for solutions to systems of partial differential equations. Quadrilateral meshes also offer lower mesh complexity, and better directionality control for anisotropic meshing. For stable analytical results, however, it is critical to construct meshes with certain quality guarantees. Specifically, algorithms that construct well-shaped elements by providing bounds on minimum and maximum angles have much practical value. Techniques such as paving [6] work well in practice, but do not give provable angle guarantees. Circle-packing techniques have been used to construct quadrangulations with no angles larger than 120° for polygon interiors [4], but with no bound on smallest angle. An algorithm to construct linear-sized strictly convex quadrilateral meshes for arbitrary planar straight line graphs is given in [19], but without angle guarantees. It is possible to obtain a quadrilateral mesh with a minimum angle bound by converting a triangulation with bounded minimum angle [5, 11, 21], into a quadrilateral mesh (for example, by splitting every triangle into three quads [12]). However, the indirect approach of converting a triangulation into a quadrangulation is generally not preferred by practitioners as they give poorly shaped elements and lead to quadrilateral meshes that are larger in size than the triangle mesh. Our goal is to design a direct quadrilateral mesh generation algorithm that works well in practice while also providing a provable guarantee on the quality of the mesh.

Our contribution. In this paper, we present a new algorithm to generate quadrilateral meshes for simple polygonal

An extended abstract was presented at the 17th IMR [2]. A video and short paper based on this paper was presented at the 25th SoCG [13].

Suneeta Ramaswami's research was partially supported by NSF CCF0830589.

F. B. Atalay
Department of Computer Science, St. Joseph's University,
Philadelphia, PA, USA

S. Ramaswami (✉)
Department of Computer Science, Rutgers University,
Camden, NJ, USA
e-mail: rsuneeta@camden.rutgers.edu

D. Xu
Department of Computer Science, Bryn Mawr College,
Bryn Mawr, PA, USA

regions, possibly with holes, with a provable guarantee on the angle. We use quadtrees to show that no newly created angle in the quadrilateral mesh is smaller than $\arctan(\frac{1}{3}) = 18.43°$ or greater than $135° + 2\arctan(\frac{1}{3}) = 171.86°$. This is the first known direct quadrilateral mesh generation algorithm with a provable bound on the angle. (Quadtrees have been used to give triangular meshes without small angles for point sets and polygons in 2D [5], and octrees have been utilized to construct tetrahedral meshes with bounded aspect-ratio elements for polyhedra [18].)

In Sect. 2, we use quadtrees to construct a quadrilateral mesh for a point set in which the angles are bounded below by $45° - \arctan(\frac{1}{3}) = 26.57°$ and bounded above by $135° + \arctan(\frac{1}{3}) = 153.43°$. We then describe in Sect. 3 an algorithm that adapts the guaranteed-quality mesh of polygon vertices to polygon edges in order to construct a quadrilateral mesh for the interior of a simple polygon (possibly with holes) in which new angles (angles other than those determined by the input) are bounded below by $\arctan(\frac{1}{3}) = 18.43°$ and above by $135° + 2\arctan(\frac{1}{3}) = 171.86°$.

Throughout this paper, we use the shorter terms "quadrangulate" and "quadrangulation" instead of "quadrilateralize" and "quadrilateralization". We also sometimes use the word "quad" for quadrilateral. *Steiner points* are additional points, other than those provided by the input, inserted during the mesh generation process.

2 Point set mesh with bounded angles

We first describe an algorithm to construct a quadrilateral mesh with a minimum angle bound of 26.57° and a maximum angle bound of 153.43° for a given point set X. This algorithm will in turn be utilized in Sect. 3 to construct a quadrilateral mesh for the interior of an arbitrary simple polygon.

2.1 Construction of the quadtree

Given a point set X, we construct a quadtree for X with the following separation and balancing conditions. These conditions are similar to those in [5], but adapted to particular requirements for quadrilateral (rather than triangle) meshing.

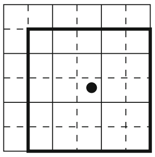
(A) Split a cell C (with side length of l) containing at least one point if it is crowded. A cell is crowded if one or more of the following conditions hold:

1. it contains more than one point from X;
2. one of the extended neighbors is split (an extended neighbor is a cell of same size sharing either a side or corner of C);
3. it contains a single point x with a nearest neighbor closer than $2\sqrt{2}l$ units away.

(B) When a crowded cell C is split, split those extended neighbors of C that share an edge or corner with a child of C containing an original point in X.

(C) The final quadtree is balanced so that the edge lengths of two adjacent cells differ at most by a factor of 2 (the neighbors each cell C with side length l have length $l/2$ or $2l$).

Observe that in a quadtree with the above separation and balancing conditions, a cell containing a point from X is guaranteed to be surrounded by eight empty cells of the same size. We refine the quadtree decomposition further to do the following: Split each of these eight empty quadtree cells into 2 × 2 cells and rebalance the quadtree. This converts the original 3 × 3 grid around every point $p \in X$ into a 6 × 6 grid. Furthermore, now p lies at the center of a 5 × 5 equal-sized grid (outlined in bold in figure) and is surrounded by 24 empty quadtree cells of the same size.



There are two reasons for this refinement step:

1. The final step of our algorithm to construct a quadrilateral mesh for X consists of warping a Steiner point in the mesh to an original point $p \in X$ (Sect. 2.4.3). This step is simplified considerably due to the refinement
2. The algorithm to construct a quadrilateral mesh with bounded minimum angle for non-acute polygons (Sect. 3) uses the 5 × 5 grid to quadrangulate the region near the polygon vertices.

Note that the 5 × 5 equal-sized grid is enough to guarantee our theoretical results and can be obtained without the 6 × 6 split first for a possible smaller-sized quadtree. The choice to subdivide all cells in the original 3 × 3 grid was for ease of implementation.

We construct a quadrilateral mesh with bounded minimum angle for X by placing Steiner points in the interior of the quadtree cells. The placement of the Steiner points is determined by identifying and applying templates to the quadtree decomposition. A leaf of the quadtree is an unsplit cell and we refer to these as *1-cells* in our discussion. A template is applied to each internal node of the quadtree.

2.2 The templates

A template is labeled by the number of children of a quadtree node that are 1-cells. Hence, we have six template configurations, for nodes with zero ($\mathcal{T}^{(0)}$), one ($\mathcal{T}^{(1)}$), two, three ($\mathcal{T}^{(3)}$) or four ($\mathcal{T}^{(4)}$) 1-cell children. Nodes with two 1-cell children have two layouts, $\mathcal{T}^{(2a)}$ and $\mathcal{T}^{(2b)}$.

Templates at the deepest level of subdivision. The templates at the deepest level of subdivision are shown in Fig. 1. Note that, all other possible configurations are symmetric to the depicted ones. In order to quadrangulate a template, first, a Steiner point is placed at the center of each quadtree cell. These points are denoted with full circles. We then place *extra* Steiner points, which are denoted by empty circles in the figure, for one of two reasons: (1) in $\mathcal{T}^{(1)}$, the top-left extra point and in $\mathcal{T}^{(2b)}$ the middle extra points are added to be able to quadrangulate *properly* within the template. (2) The remaining extra points are added in the 1-cells, halfway on the diagonal between the center Steiner point and the outer cell corner. The reason for adding the second type of Steiner points is that after an internal node is quadrangulated, it will provide a polygonal chain with an even number of points (we will call them *even-connector chains*) to which its neighbors can connect.

General templates. Our recursive algorithm applies templates to all internal nodes starting with the deepest ones. We generalize the templates to apply to an arbitrarily deep internal node as shown in Fig. 2. In general, when a template is applied to an internal node, its children which are not 1-cells will have already had templates applied to them, that is, each such child has been quadrangulated internally and it provides even-connector chains on all four sides. We can then connect the corresponding endpoints of the two neighboring chains to construct a polygon with guaranteed even number of vertices which can therefore be quadrangulated. We name this process "stitching", and it is illustrated by the cross-hatched regions in Fig. 2. In the figure, the processed internal nodes are depicted as black-boxes with even-connector chains at each side. Templates $\mathcal{T}^{(1)}$, $\mathcal{T}^{(2a)}$ and $\mathcal{T}^{(2b)}$ have three variations due to the possibility of a 1-cell being stitched with a 2-connector or a 4-connector chain along one or two of its sides. Similarly, $\mathcal{T}^{(3)}$ has two variations. Note that, in the bottom variations of $\mathcal{T}^{(2a)}$ and $\mathcal{T}^{(3)}$, the middle extra points are deleted to allow for a simpler quadrangulation (without adding any other Steiner points). Some of these templates can be simplified (for example, in $\mathcal{T}^{(2b)}$ the inner quad is redundant); however, these simplifications produce no improvement on the angle bound. Note also that the placement of the endpoint of a chain does not necessarily correspond to the exact location of the endpoint within the actual cell, due to the possible existence of type (ii) Steiner points.

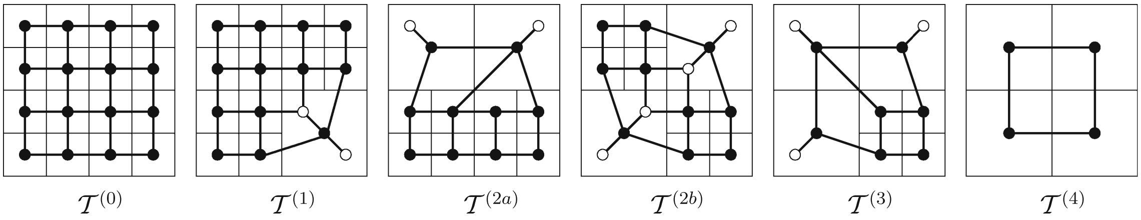


Fig. 1 Templates at the deepest level of the subdivision

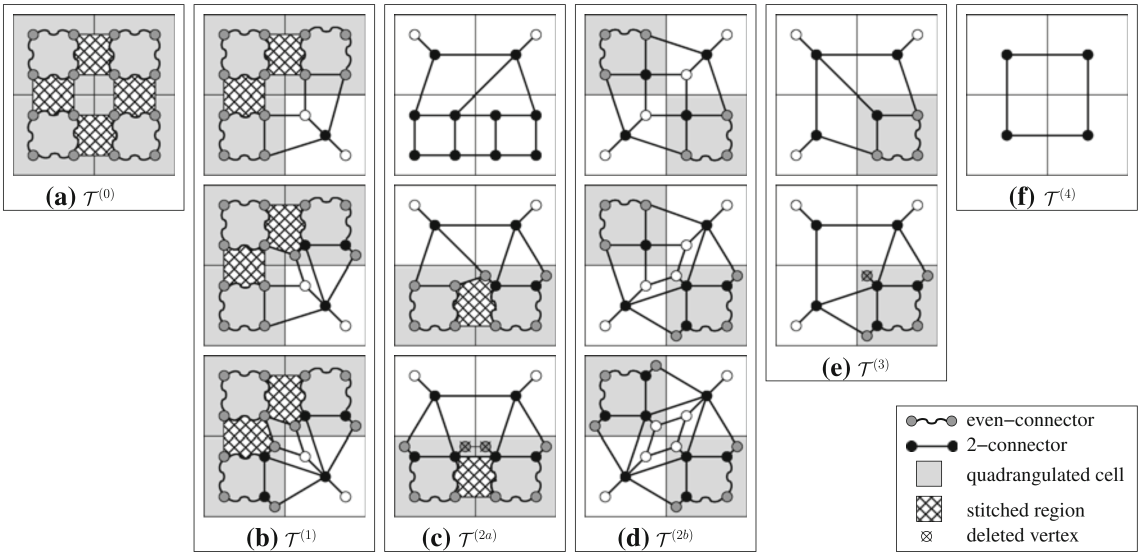
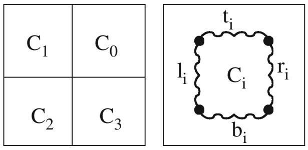


Fig. 2 General templates at arbitrary level of subdivision

Labeling the chains. The children quadrants of a cell are labeled as C_0, C_1, C_2, and C_3 in counterclockwise order starting from the northwest quadrant. The four chains surrounding a processed quadrant C_i are labeled as l_i, r_i, t_i and b_i for left, right, top and bottom chains, respectively.



2.3 The algorithm

The recursive procedure *applyTemplate* that applies a template to an internal node is presented in the code block given in Fig. 3. It is initially called with the root node of the quadtree. Note that the algorithm is presented only with respect to the depicted configurations of the templates. Symmetric configurations are handled similarly.

2.3.1 *Stitching chains*

Procedure *stitchChains* connects the four endpoints of two neighboring even-connector chains and quadrangulates the resulting polygon. Note that such a polygon is guaranteed to have even number of vertices on the boundary. The algorithm is illustrated in Fig. 4. Procedure *stitchChains* is only called if current template is of type $\mathcal{T}^{(0)}$, $\mathcal{T}^{(1)}$ or $\mathcal{T}^{(2a)}$. The action of this procedure is also illustrated by the crosshatched areas in Fig. 2a–c.

```
applyTemplate(QuadtreeNode N)
  templateType ← whichTemplate(N)
  for $C_i \in children(N)$
    if $C_i$ is not a 1-cell
      $(l_i, r_i, t_i, b_i) \leftarrow applyTemplate(C_i)$
    else
      construct $(l_i, r_i, t_i, b_i)$ for $C_i$.
  switch (templateType)
    case $\mathcal{T}^{(0)}$:
      $stitchChains(l_0, r_1)$, $stitchChains(b_1, t_2)$
      $stitchChains(r_2, l_3)$, $stitchChains(t_3, b_0)$
    case $\mathcal{T}^{(1)}$:
      $stitchChains(l_0, r_1)$, $stitchChains(b_1, t_2)$
      Place Steiner points and quadrangulate per Fig. 2(b).
    case $\mathcal{T}^{(2a)}$:
      $stitchChains(r_2, l_3)$
      Place Steiner points and quadrangulate per Fig. 2(c).
    case $\mathcal{T}^{(2b)}$:
      Place Steiner points and quadrangulate per Fig. 2(d).
    case $\mathcal{T}^{(3)}$:
      Place Steiner points and quadrangulate per Fig. 2(e).
    case $\mathcal{T}^{(4)}$:
      Place Steiner points and quadrangulate per Fig. 2(f).
  return $(l_1 + l_2, r_0 + r_3, t_1 + t_0, b_2 + b_3)$
```

Fig. 3 The procedure *applyTemplate* is used to quadrangulate node *N* recursively

```
stitchChains(Chain $ch_1$, Chain $ch_2$)
  switch (length($ch_1$), length($ch_2$))
    case $(2-2), (2-4), (4-2)$:
    Apply appropriate base case from Fig. 5.
    case $(2-6), (2-8))$:
    Apply appropriate base case from Fig. 6.
    default:
      $(f_1, s_1) \leftarrow getHalfChains(ch_1)$
      $(f_2, s_2) \leftarrow getHalfChains(ch_2)$
      $stitchChains(f_1, f_2)$
      $stitchChains(s_1, s_2)$
```

Fig. 4 The procedure *stitchChains* stitches two even-connector chains, one from each of the two neighbor cells sharing an edge

The quadrangulation process divides the chains into half chains, each of which spans the corresponding edge of a child quadrant. These half chains are then recursively stitched. Although the even-connector chains can be arbitrarily long, at the base case there are only four types of chains: chains with 2, 4, 6 or 8 connectors. Note that one of the chains being stitched at the base case is always a 2-connector chain; otherwise, the recursion would have further subdivided the chains. In other words, in the base case, one side consists of one or two 1-cells. In the case of only one 1-cell, it must have an extra point. In the case of two 1-cells, both do not have extra points. Across from a 1-cell, we can have one cell of the same size or two cells one size smaller, which may or may not have an extra point. Figs. 5 and 6 illustrate how the base-case chains are stitched (the stitching edges are dotted). Symmetric cases are not listed in the illustrations.

2.4 Angle bounds

2.4.1 *Minimum angle bounds*

We analyze the minimum angle bounds resulting from the application of *applyTemplate*, the base case of *stitchChains*, and the recursive step of *stitchChains*.

1. *General templates.* By construction, the minimum angle appears in templates $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2b)}$ and equals $45^\circ - \arctan(\frac{1}{3}) = 26.57^\circ$ (illustrated in Fig. 7).
2. *Stitching base case* The base cases of stitching generate the same minimum angle of $45^\circ - \arctan(\frac{1}{3}) = 26.57^\circ$ which can be found in Fig. 5(5).
3. *Stitching merging step* After the corresponding half-chains are stitched in the recursive step of *stitchChains*, a middle quad is formed by the four end points of the stitched half-chains. This middle quad gives a minimum angle of $2 \times \arctan(\frac{1}{4}) = 28.07^\circ$. See Fig. 7. Recall that these four points are by construction on the two diagonals that cross at the center of four quadtree quadrants. Furthermore, they are either at the center of the quadtree quadrant, or halfway down the diagonal from the center.

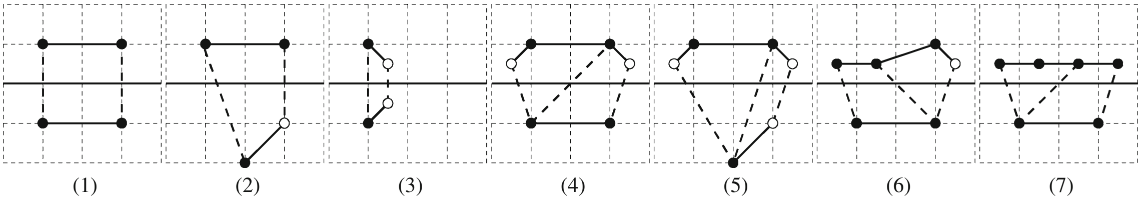
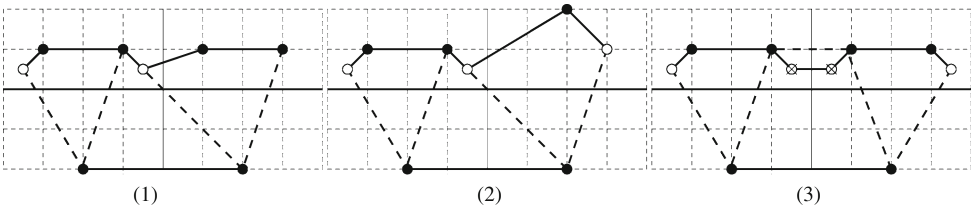


Fig. 5 Stitching 2-2 and 2-4 or 4-2 connector chains

Fig. 6 Stitching 2-6 and 2-8 connector chains



The worst-case configuration is illustrated in Fig. 7. This results from connecting any Fig. 5(5) connector chain with an inverted version of itself.

2.4.2 *Maximum angle bound*

Note that in the stitching cases illustrated by Figs. 5(7) and 6(1) and (2) as well as template $\mathcal{T}^{(2a)}$ (Fig. 1), there are degenerate quads with two edges on a straight line. In all cases, the vertex at the degenerate 180° angle is connected to a third vertex on the other side of the degenerate quad, by construction of our templates and stitching cases, as shown in Fig. 8. In the case of Fig. 8a, the degenerate vertex can be perturbed to the midpoint of its edge with the third vertex, thus reducing the 180° angle to $180 - 2\arctan(\frac{1}{2}) = 126.87°$. In the case of Fig. 8b and c, the degenerate vertex can be perturbed to its reflection in the adjacent cell. This results in reducing the 180° angle to $90 + \arctan(\frac{1}{3}) + \arctan(\frac{3}{5}) = 139.39°$ and increasing the other two angles to within the same bound.

To analyze the maximum angle bound given by the non-degenerate quads, we consider the quads generated by *applyTemplate*, the base case of *stitchChains*, and the recursive step of *stitchChains*.

1. ***General templates.*** By construction, the maximum angle appears in templates $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2b)}$ and equals $135 + \arctan(\frac{1}{3}) = 153.43°$ (illustrated in Fig. 7).
2. ***Stitching base case.*** The base cases of stitching generate the same maximum angle of $135 + \arctan(\frac{1}{3}) = 153.43°$ which can be found in Fig. 5(5).
3. ***Stitching merging step.*** The middle quad gives a maximum angle of $180 - 2\arctan(\frac{1}{4}) = 151.93°$. The worst-case configuration is illustrated in Fig. 7. This results from connecting any Fig. 5(5) connector chain with an inverted version of itself.

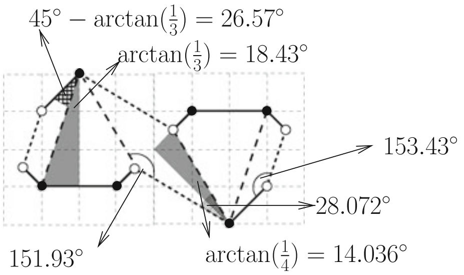

$45° - \arctan(\frac{1}{3}) = 26.57°$
$\arctan(\frac{1}{3}) = 18.43°$
153.43°
28.072°
151.93°
$\arctan(\frac{1}{4}) = 14.036°$

Fig. 7 Minimum and maximum angle bounds

2.4.3 *Warping to original points*

After the construction of the quadrilateral mesh using quadtree cell centers and extra points as Steiner points, we warp certain mesh vertices to the original points from the input point set X: See Fig. 9. Recall that the quadtree splitting rules of Sect. 2.1 ensure that the quadtree cell containing an original point $p \in X$ is surrounded by 24 empty quadtree cells of the same size. Moreover, the eight empty cells immediately surrounding p do not contain any extra points. Therefore, the warping step simply consists of translating the Steiner point in p's cell to p, along with all the incident edges. The worst-case minimum angle after the warping step is $2 \times (45 - \arctan(\frac{1}{3})) = 53.13°$. The worst-case maximum angle is $90 + \arctan(\frac{1}{3}) = 108.53°$.

In summary, we have shown the following result:

Theorem 1. *Given a quadtree decomposition with N quadtree cells satisfying the point set separation conditions for a point set X, applyTemplate constructs a mesh for X with at most 3N quadrilaterals in which every angle is at least* $45 - \arctan(\frac{1}{3}) = 26.57°$ *and at most* $135 + \arctan(\frac{1}{3}) = 153.43°$.

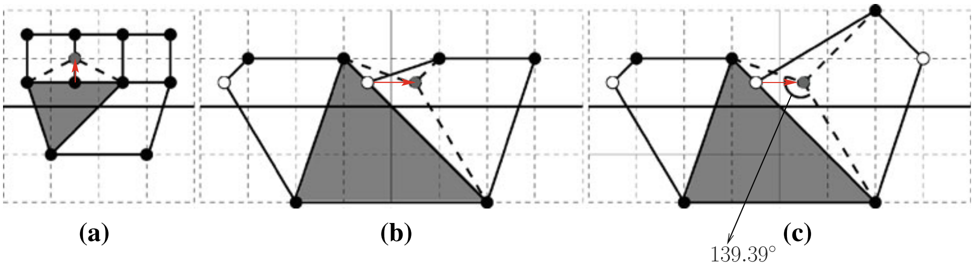


Fig. 8 Fixing degenerate quads

Observe that the value of N in the above theorem depends on the geometry of the point set as well as the size of the point set. Due to the point set separation conditions, which are derived from [5] and as was shown there, the size of the quadtree decomposition increases as the distance between the closest pair of points decreases. We have experimented on both randomly generated and real datasets of varying sizes. Results show that if n denotes the number of input points, N is approximately 80 $\times$ n at all times. See Fig. 10 for the result of our implementation on the 'Lake Superior' dataset.

3 Non-acute simple polygons

Given a simple polygon P, possibly with holes, with vertex set X, we give an algorithm to construct a quadrilateral mesh for P and its interior in which no new angle is larger than 18.43°. The basic idea behind the algorithm is to first construct a guaranteed-quality mesh for X as described in the previous section and then adapt this mesh to incorporate the edges of P. From now on, we use δP to refer to the polygon boundary, and P to refer to the union of the boundary as well as interior.

We start by describing in this section a provably good algorithm to construct a quadrilateral mesh with bounded minimum angle for a simple polygon P in which all interior angles are non-acute (i.e., $\geq 90°$). In Sect. 4, we describe how to handle acute angles and thus give an algorithm for general simple polygons.

Let P be a non-acute polygon with vertex set X and edges oriented counter-clockwise about the boundary. Let $\mathcal{QT}$ be a quadtree decomposition of X satisfying the point set separation conditions of Sect. 2.1 Let $\mathcal{Q}$ be a quadrilateral mesh for X with minimum angle 26.57, as guaranteed by Theorem 1. In this section, we describe a method to adapt $\mathcal{Q}$ to δP to create a *constrained* quadrilateral mesh for P. In a constrained quadrilateral mesh, we allow Steiner points to be inserted on δP as well, so that the union of the finite elements of the mesh is equal to P.

We start by describing in Sect. 3.1 an algorithm to adapt $\mathcal{Q}$ to include a single edge of P. In order to use this algorithm on all edges of P, $\mathcal{QT}$ must satisfy certain polygon edge separation conditions, which are discussed at the end of the section. The last remaining step to construct the final constrained mesh for P is to adapt the mesh to the regions around the vertices. This is described in Sect. 3.2.

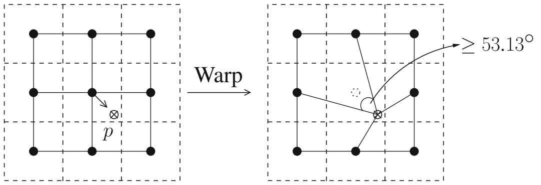


Fig. 9 Warping to the original point

3.1 Inserting an edge into $\mathcal{Q}$

Consider an edge $\vec{e} = (a, b)$ of P oriented from a to b, where $a, b \in X$. Assume that $\vec{e}$ makes an angle between $-$ 45° and 45° with the positive x axis (if not, orient the x axis so that this is the case). We say that a point lies "above" $\vec{e}$ if it lies in the open halfspace to the left of the oriented line through $\vec{e}$. We use $\vec{e}$ to define two chains of edges from $\mathcal{Q}$ and $\mathcal{QT}$, as described below:

(i) $\vec{e}$ intersects quadrilaterals of $\mathcal{Q}$. Edges of these quadrilaterals are used to define a chain of edges called the *quadrangulation chain* α associated with $\vec{e}$.

(ii) $\vec{e}$ intersects quadtree cells of $\mathcal{QT}$. The centers of these cells are used to define a chain of edges called the *quadtree chain* β associated with $\vec{e}$.

Quadrangulation chain (α) Let $q_1, q_2, \ldots, q_k$ be the quadrilaterals of $\mathcal{Q}$ having a non-empty intersection with $\vec{e}$, in left to right order as traversed from a to b (since the quadrilaterals are convex, each q_i is unique). Let E_i be the sequence of edges of q_i that lie entirely above $\vec{e}$. E_i may have 0, 1, or 2 edges. If E_i has two edges, they are listed in clockwise order about q_i. Then the *quadrangulation chain* α is defined as follows:

$$\alpha = E_1 \cdot E_2 \ldots \cdot E_k$$

where $\cdot$ represents edge concatenation. See Fig. 11 for an example of a quadrangulation chain, in which E_1 has 1

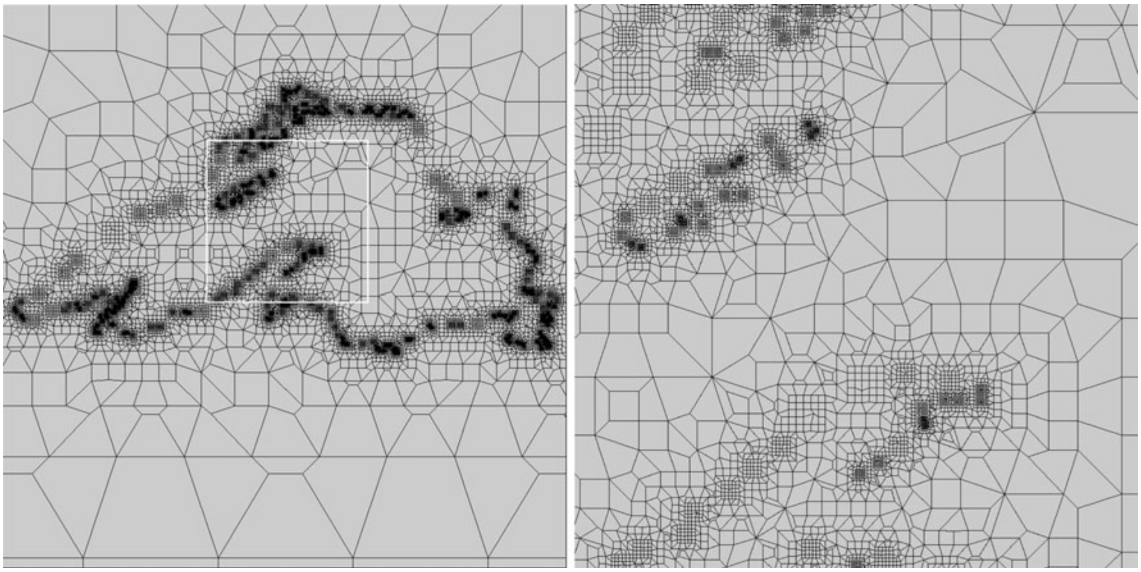


Fig. 10 Number of input points: 303. Minimum mesh angle: 26.57°. Number of quadtree cells: 20,845. Number of mesh vertices: 24,463. Number of mesh faces: 24,444. Mesh within *white square* is magnified on the right

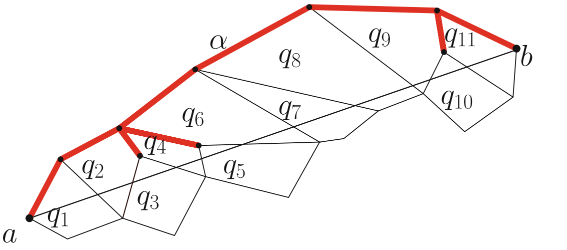


Fig. 11 Quadrangulation chain α, marked in *red* (bold)

edge, E_2 has 2 edges, and E_3 has 0 edges. Note that the same edge may repeat twice in α (the repetitions always appear consecutively) and such an edge is incident to q_i that has $|E_i| = 0$. For example, the quadrangulation chain in Fig. 11 has three repeating edges, which are incident to the quadrilaterals q_3, q_5 and q_{10}.

A vertex is said to belong to an edge if it is one of the endpoints of the edge. We say that $v \in \alpha$ if v is a vertex of $\mathcal{Q}$ and belongs to one of the edges of α. If we quadrangulate the region bounded by α and $\vec{e}$ by adding Steiner points either in the interior of the region or on $\vec{e}$ itself, the resulting quadrangulation is compatible with $\mathcal{Q}$ (since the edges of α are edges in $\mathcal{Q}$). However, in order to quadrangulate the region with the desired angle bounds, we need to know more about the geometry of α. The quadtree chain, described below, allows us to establish the required geometric properties for α.

Quadtree chain (β). In the remainder of the paper, we frequently use the same symbol to refer to a quadtree cell as well as its center whenever the meaning is clear from the context. Given a cell center c, $N(c)$, $W(c)$, $S(c)$, and $E(c)$ denote, respectively, the set of north, west, south, and east neighbor cell centers of c (note that each set has at most two elements in it because of the balancing conditions for $\mathcal{QT}$).

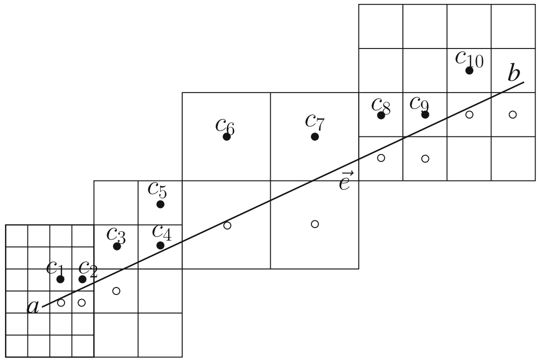


Fig. 12 Cell centers of quadtree chain β are shown as *filled circles*. *Unfilled circles* are cell centers that belong to C but not to β

Let C be the set of cell centers of quadtree cells in $\mathcal{QT}$ that are intersected by $\vec{e}$. C does not include the starting and ending cell centers, a and b. Let θ be the angle (in degrees) that $\vec{e}$ makes with the positive x axis. The *quadtree chain* β is defined as follows (refer to Fig. 12):

1. If $c \in C$ and c lies strictly above $\vec{e}$, then c belongs to β.
2. If $c \in C$ and c lies on or below $\vec{e}$, then $N(c) \subset \beta$. Note that the cell centers in $N(c)$ must lie above $\vec{e}$ under our assumption that $-45 \le \theta \le 45$.
3. If $c \in C$, c lies on or below $\vec{e}$, and $0 \le \theta \le 45$ (resp., $-45 \le \theta < 0$), then a cell center in $W(c)$ (resp., $E(c)$) belongs to β if it lies above $\vec{e}$.

In Fig. 12, $\beta = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}\}$. Centers c_2, c_3, c_7, and c_9 are in β because of conditions 1 and 2, c_4 is in β because of condition 1 only, c_1, c_6, and c_8 are in β because of condition 2 only, and c_5 is in β because of condition 3.

Let $\{c_1, c_2, \ldots, c_m\}$ be the cell centers in β in lexicographically sorted (by x, then y) order. Recall that $\mathcal{QT}$ is the quadtree decomposition of X from which $\mathcal{Q}$ was constructed. The overall approach to incorporating edge $\vec{e}$ into $\mathcal{Q}$ is summarized below:

(A) We first show that every c_i in the quadtree chain β belongs to the quadrangulation chain α.

(B) This fact allows us, in turn, to exploit the structure provided by $\mathcal{QT}$ and our algorithm from Sect. 2 to identify a small number of possible ways in which two consecutive points c_i and c_{i+1} of β can be connected along the chain α. We use α_i, $1 \leq i \leq m-1$, to refer to the subchain of α starting at c_i and ending at c_{i+1}. α_i may lie under $\overrightarrow{c_ic_{i+1}}$. In this case, we choose instead a chain of edges in $\mathcal{Q}$ lying above $\overrightarrow{c_ic_{i+1}}$ in order to simplify the final quadrangulation step in part (C) below.

(C) Finally, we quadrangulate the region bounded by $\vec{e}$ and α by breaking it into smaller sub-regions defined by perpendicular projections from c_i and c_{i+1} onto $\vec{e}$. The case analysis form part (B) is then used to prove a minimum angle guarantee of 18.43° for the quadrangulation of each subregion.

We first state and prove several lemmas required for steps (A)–(C).

Lemma 2. *Let* (u, v) *be an edge in* $\mathcal{Q}$ *and let* c_u (c_v) *be the quadtree cell containing* $u(v)$*. Then, for any quadtree cell* $c \in \mathcal{QT}, (u,v) \cap c = \emptyset$ *for all* $c \notin \{c_u, c_v\}$*.*

Proof. We prove the claim by showing that $(u,v) \subset c_u \cup c_v$. The procedures *applyTemplate* and *stitchChains* only add edges between points lying in the same quadtree cell, or quadtree cells that are edge or corner neighbors. Therefore, either $c_u = c_v$, in which case the claim is obviously true, or c_u and c_v are edge or corner neighbors. If c_u and c_v are corner neighbors, u and v lie on the line going through the common corner of c_u and c_v and containing the cell diagonals. This implies that $(u,v) \subset c_u \cup c_v$. Now consider the case when c_u and c_v are edge neighbors. Let e be the quadtree edge common to c_u and c_v. All edges in $\mathcal{Q}$ between c_u and c_v drawn by procedures *applyTemplate* and *stitchChains* intersect e (see Figs. 1, 2, 5, 6). This implies $(u,v) \subset c_u \cup c_v$. □

We make two observations below that will help us establish the relationship between the quadtree chain β and the quadrangulation chain α. These observations are about the edges of $\mathcal{Q}$ and follow directly from our *applyTemplate* algorithm to construct $\mathcal{Q}$.

Observation 1 An extra point a always has degree three. (Recall that extra points are vertices of $\mathcal{Q}$ that are not quadtree cell centers.) Furthermore, a always has one edge incident to its own cell center c_i and has two other edges that are incident to two points that lie along the line ℓ perpendicular to $\overline{ac_i}$ and passing through the corner of a's cell that is closest to a. In addition, these two edges must cross two distinct sides of a's cell. See Fig. 13a for an illustration. Observe that ℓ makes an angle of ±45° with the horizontal because of how extra points are chosen. Note that the gray points on ℓ may be extra points or cell centers lying in the edge neighbors of a's cell. Observe also that the intersection of a's cell with any quadrangulation edge incident on a lies entirely within the quadrant of c_i containing a.

Observation 2 Let X and Y denote two cells that are edge neighbors. Let x be an extra point or cell center in X and let y be an extra point or cell center in Y. An edge, $\overline{xy}$, if it exists, must lie entirely within the neighboring halves of X and Y. See Fig. 13b for an illustration.

Lemma 3. *Every cell center in the quadtree chain belongs to the quadrangulation chain.*

Proof. We want to establish that for $1 \leq i \leq m$, $c_i \in \alpha$. Suppose, for the sake of contradiction, that the claim is false and let c_i be the a cell center in the quadtree chain that does not belong to the quadrangulation chain.

Let $b_i \in S(c_i)$ be such that b_i lies on or below $\vec{e}$. If c_i has two south neighbors, pick one arbitrarily. See Fig. 14a. Note that, for any c_i on the quadtree chain, there is at least one south neighbor of c_i below $\vec{e}$ (and thus below α), except for one special case discussed at the end of the proof. Since c_i lies on the quadtree chain, we know that $\vec{e}$ intersects either c_i's cell, or b_i's cell (or both). Also, since c_i lies above α and b_i lies below α (by definition, there are no cell centers between α and $\vec{e}$), $\overline{c_ib_i}$ intersects α.

Let a and a' be the endpoints of an edge of α intersecting $\overline{c_ib_i}$, such that a is on or to the left and a' is on or to the right of $\overline{c_ib_i}$. Let A, A', C, and B be the quadtree cells

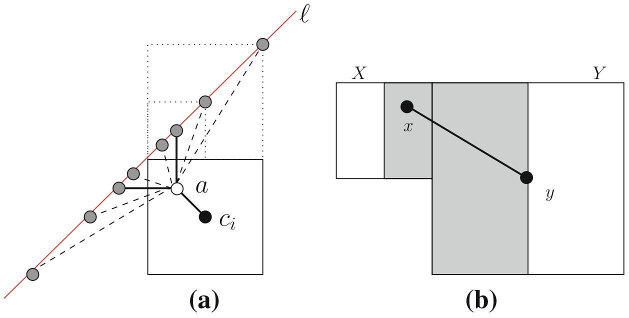


Fig. 13 **a** Observation 1: possible connections from extra point a. **b** Observation 2: if a point lying in the shaded half of cell X is connected to a point y in cell Y, y must lie in the shaded half of cell Y

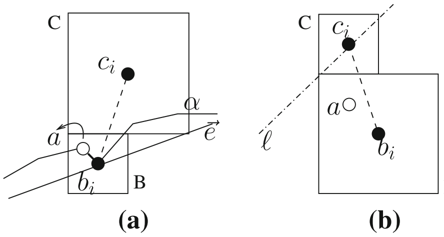


Fig. 14 Lemma 3, Case A

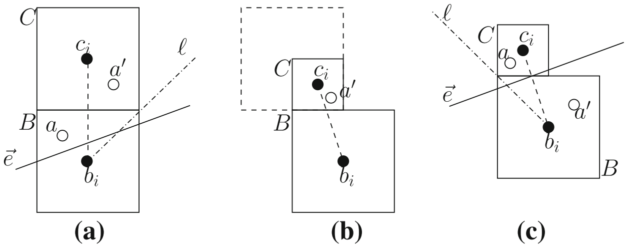


Fig. 15 Lemma 3, Case B

containing a, a', c_i, and b_i, respectively. It follows by definition that $\overline{c_ib_i} \subset C \cup B$. Also, since (a, a') is an edge of $\mathcal{Q}$, we know from Lemma 2 that $(a, a') \subset A \cup A'$. Therefore, because $\overline{c_ib_i}$ and (a, a') intersect, it follows that either $A \subset C \cup B$, or $A' \subset C \cup B$. This implies that $a \in C \cup B$, or $a' \in C \cup B$.

We have two cases, depending on whether C contains one of a or a'. We prove that in both cases, the existence of edge (a, a') leads to a contradiction.

Case A **Neither** a **nor** a' **lies in** C. In this case, at least one of a or a' must lie in B.

Case A.1 **Both** a **and** a' **lie in** B. This can only happen if one of a or a' coincides with b_i (since a and a' cannot both be extra points in B) and b_i is on $\vec{e}$. In that case, however, $\vec{e}$ and α intersect at b_i which contradicts the fact that (by definition) α lies entirely above $\vec{e}$. See Fig. 14a for an illustration of this case.

Case A.2 **Exactly one of** a **or** a' **lies in** B. Assume wlog that a lies in B. Note that since b_i lies below $\vec{e}$, a must be an extra point. See Fig. 14b. From Observation 1, we know that the intersection of $\overline{aa'}$ and cell B lies entirely within the quadrant of B containing a. This implies that $\overline{aa'}$ must intersect cell C, which contradicts Lemma 2 (because $a' \not\in C$). Note that we do not consider other placements of a because they lead to cases where $\overline{aa'}$ does not intersect $\overline{c_ib_i}$.

Case B **At least one of** a **or** a' **lies in** C.

Case B.1 a' **is an extra point in** C. In this case, a must be an extra point in B. Furthermore, from Observation 2, it must lie in the half of cell B that is adjacent to C. (Note that a cannot lie in any other cell, e.g. west neighbor of C, due to Observation 2.) If B is the same size or smaller than C, then $\overline{aa'}$, which crosses $\overline{c_ib_i}$, would violate Observation 1. See Fig. 15a for an illustration. Similarly, if B is larger than C, and C is aligned with the north east quadrant of B, $\overline{aa'}$ would violate Observation 1. If B is larger than C, and C is aligned with the north west quadrant of B as depicted in Fig. 15b, the placement of extra point a' indicates that the enclosing template of C is as shown by dotted lines. In this case, we reach a contradiction by observing that the alignment of B with the enclosing template of C is not possible in our template construction.

Case B.2 a **is an extra point in** C. In this case a' must be an extra point in B that lies in the half adjacent to C due to Observation 2. This case is depicted in Fig. 15c. However, in any possible size and alignment of B, existence of $\overline{aa'}$ that crosses $\overline{c_ib_i}$ violates Observation 1, leading to a contradiction.

Special case: As seen in Fig. 16, there is a case where both c_i and b_i are above $\vec{e}$. In this case, by construction, the edge (c_i, b_i) is an edge of $\mathcal{Q}$, and c_i, b_i, and $E(c_i)$ are three vertices of a quad intersected by $\vec{e}$. This implies that both c_i and b_i are cell centers on the quadrangulation chain. □

Lemma 4. *For* $1 \leq i \leq m-1$, c_i *and* c_{i+1} *are either edge or corner neighbors in* $\mathcal{QT}$.

Proof. The claim is a direct consequence of the definition of a quadtree chain. We distinguish two cases depending on whether cell c_i is intersected by segment $\vec{e}$. Recall that we use c_i to denote both the cell center and the cell itself.

Case A: **cell** c_i **is intersected by** $\vec{e}$. In this case, cell center c_i must lie above $\vec{e}$ since it belongs to $\mathcal{QT}$. We have the following cases for the cell center c_{i+1}.

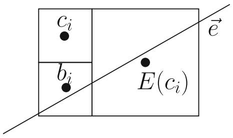


Fig. 16 Special case

Case A.1 **cell c_{i+1} is intersected by $\vec{e}$**. In this case, c_i and c_{i+1} are clearly neighbors due to the definition of QT.

Case A.2 **cell c_{i+1} is not intersected by $\vec{e}$**. In this case, c_i is (i) either the west neighbor of a cell b_i intersected by $\vec{e}$ with cell center below $\vec{e}$, or (ii) the north neighbor of b_i. Note that in either case b_i is the next cell intersected by $\vec{e}$ and that c_i and b_i must be neighbors in QT. In particular, c_i and b_i must be edge neighbors due to our assumption that $-45 \leq \theta \leq 45$. See Fig. 17(i) for the depiction of the case where c_i and b_i are edge neighbors and c_{i+1} is a west neighbor of b_i. Note that c_{i+1} lies immediately above c_i in this case. See Fig. 17(ii) for the cases in which c_i and b_i are edge neighbors and c_{i+1} is a north neighbor of b_i. In all cases, c_i and c_{i+1} are either edge or corner neighbors.

Case B **cell c_i is not intersected by $\vec{e}$**. In this case, c_i must either be a west neighbor or a north neighbor of a cell intersected by $\vec{e}$ with cell center b_i lying below $\vec{e}$.

Case B.1 c_i **is a west neighbor of** b_i. Then, we know that c_i lies above a cell, a_i, of the same size intersected by $\vec{e}$, and b_i is the east neighbor of both c_i and a_i. See Fig. 18. (This is the only case a west neighbor needs to be included in QT.) Due to our assumption that $-45 \leq \theta \leq 45$, the north neighbor(s) of b_i must be on QT. Clearly, the (left) north neighbor must be c_{i+1}, which is a corner neighbor of c_i.

Case B.2 c_i **is a north neighbor of** b_i. There are a number of possible cases depending on the relative sizes of c_i and b_i. Let a_N denote the cell incident to the lower right corner of c_i and sharing an edge with c_i in all the cases below. Let a_S be the south neighbor of a_N that is incident to the lower right corner of c_i.

(i) c_i **is the same size as** b_i. If a_S is below $\vec{e}$, a_N must lie above $\vec{e}$ since c_i is not intersected by $\vec{e}$ and θ cannot be larger than 45°. Hence, $c_{i+1} = a_N$. If a_S is above $\vec{e}$, $c_{i+1} = a_S$ because θ cannot be smaller than $-45°$, that is, any other cell center south of a_S cannot be above $\vec{e}$. See Fig. 19(i).

(ii) c_i **is half the size of** b_i. If c_i is the right north neighbor of b_i, the argument is identical to case (i). If c_i is the left north neighbor of b_i (see Fig. 19(ii)), a_N must lie above $\vec{e}$ (otherwise, $\theta > 45°$). Hence, $c_{i+1} = a_N$.

(iii) c_i **is twice the size of** b_i. If b_i is the right south neighbor of c_i, the argument is identical to case (i). Otherwise, b_i is the left south neighbor of c_i (see Fig. 19(iii)). In that case, either the right south neighbor of c_i is above $\vec{e}$, making it c_{i+1}, or it is below $\vec{e}$ and then c_{i+1} is one of a_N or a_S by an argument identical to case (i).

Lemma 5 *Let v_i be the vertical projection of c_i on $\vec{e}$. The segment $\overline{c_i v_i}$, $1 \leq i \leq m$, does not intersect α.*

Proof. Let b_i be a south neighbor of c_i such that b_i lies below α. Note that for any c_i on the quadtree chain, there is at least one south neighbor of c_i lying on or below $\vec{e}$ (and

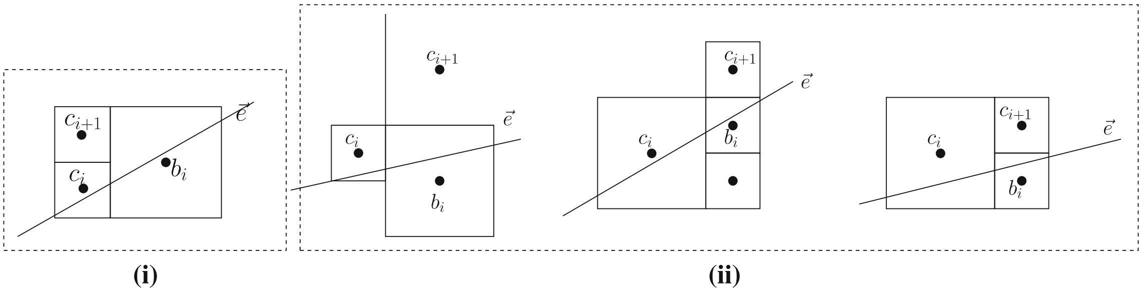
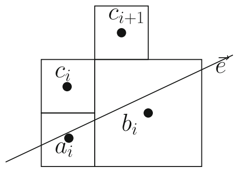


Fig. 17 Lemma 4 (i) Case A.2(i): c_{i+1} is directly above c_i. Note that c_{i+1} is included on QT since it is the west neighbor of b_i. (ii) Case A.2(ii)

Fig. 18 Lemma 4, Case B.1



hence below α), except for the one special case shown in Fig. 16. As explained in Lemma 3, (c_i, b_i) is part of the quadrangulation chain in this case. Note that the vertical projection from c_i coincides with (c_i, b_i) and thus the problem is reduced to proving that the vertical projection from b_i does not intersect α. This is covered in the subsequent regular cases.

If $\vec{e}$ intersects the cell c_i but not b_i, the segment $\overline{c_i v_i}$ lies entirely in cell c_i. If $\vec{e}$ intersects b_i, the segment $\overline{c_i v_i}$ lies in the union of cells c_i and b_i. Suppose that $\overline{c_i v_i}$ is intersected by α. Lemma 2 implies that the quadrangulation edge intersecting $\overline{c_i v_i}$ must have one of its end points in cell c_i or cell b_i which in turn must be an extra point of that cell. Because of Observations 1 and 2 and the fact that b_i lies below $\vec{e}$, this extra point cannot come from cell c_i. This is because none of the edges incident on it can cross $\overline{c_i v_i}$. Hence, such an extra point must lie in cell b_i. We distinguish three cases depending on the relative sizes of the cells c_i and b_i.

Case 1 *b_i is the same size as c_i*. From Observations 1 and 2, an edge incident on a possible extra point a in b_i cannot cross $\overline{c_i v_i}$. See Fig. 20a for possible placements of a.

Case 2 *b_i is twice the size of c_i*. Possible placements of an extra point a is depicted in Fig. 20b. In case of the placement on the right, none of the edges incident on a can cross $\overline{c_i v_i}$ due to Observation 1. In case of the placement on the left, a cannot be on α since it must lie below $\overline{c_{i-1} c_i}$ for any possible c_{i-1}.

Case 3 *b_i is half the size of c_i*. Again due to Observation 1, among all possible placement of extra points in b_i respecting valid template alignments, there is no extra point with incident edges that can cross $\overline{c_i v_i}$. See Fig. 20c for possible placements of a.

□

Lemmas 3 and 5 imply that the edge sequence $(v_i, c_i) \cdot \alpha_i \cdot (c_{i+1}, v_{i+1}) \cdot (v_{i+1}, v_i)$ defines a simple polygon for all $1 \leq i \leq m - 1$. Call this polygon A_i (see Fig. 21). We now use Lemma 4 to prove that α_i is composed of at most four edges. This is done via a case analysis on the ways in which c_i and c_{i+1} are connected in $\mathcal{Q}$.

Lemma 6. *The number of edges in α_i is at most four.*

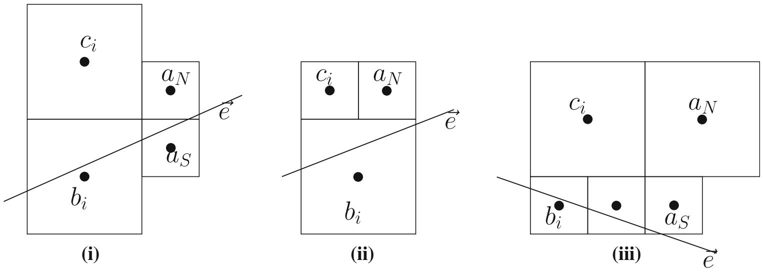
Proof. We know from Lemma 4 that c_i and c_{i+1} are either edge or corner neighbors in $\mathcal{QT}$. We consider each case separately. Our case analysis only depicts α_i with two or more edges (i.e., when c_i and c_{i+1} are not directly connected). Let s_i, $1 \leq i \leq m$ refer to the size of c_i's cell (by "size", we mean "side length").

Case 1 **c_i and c_{i+1} are edge neighbors**. In this case, the connectivity between c_i and c_{i+1} in $\mathcal{Q}$ may come from either the application of a template (*applyTemplate*) at some level of recursion, or the application of the stitching step (*stitchChains*) at some level of recursion. We consider different possibilities based on the ratio s_i:s_{i+1}, which may be 1:1, 1:2, or 2:1. Configurations for these cases are shown in Figs. 22, 23, and 24, respectively. Each of these figures indicates the minimum internal angle in A_i along α_i. Note that each of them is well above 18.43°. We depict only distinct α_i that differ in either the number of edges, or the angles at the vertices (that is, we do not show other, symmetric configurations that lead to the same α_i).

Case 2 **c_i and c_{i+1} are corner neighbors**. In this case, the connectivity between c_i and c_{i+1} in $\mathcal{Q}$ may come from the application of *applyTemplate*, the application of *stitchChains*, or through a *center quad*. The center quad is the quadrilateral formed at the center, i.e. the meeting point of the four quadrants, after a general template (ref. Fig. 2) is applied during the recursive step. Since c_i and c_{i+1} are corner neighbors, the ratio s_i:s_{i+1} can be 1:1, 1:2, 2:1, 1:4, or 4:1. We consider the case of center quads first, and then consider templates and stitchings.

Case 2.1 **α_i contains center quad edges**. Let s be the size of the cell adjacent to c_i as

Fig. 19 Lemma 4, Case B.2



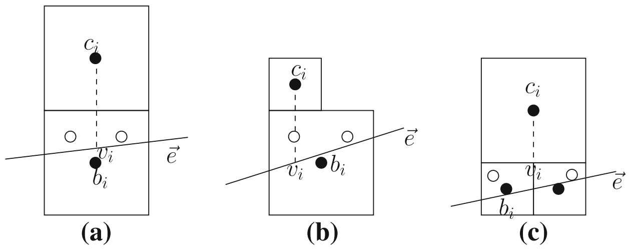


Fig. 20 Lemma 5: possible placements of extra point *a* are shown as *unfilled circles*. **a** Case 1, **b** Case 2 and **c** Case 3

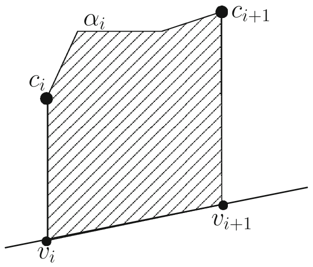


Fig. 21 Polygon A_i

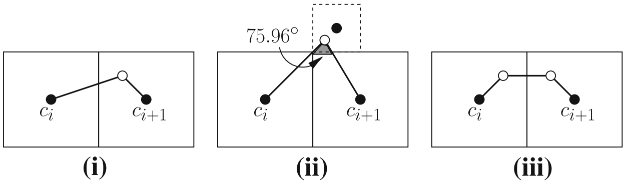


Fig. 22 Configurations for α_i when $s_i{:}s_{i+1}$ is 1:1. (*i*) and (*ii*) come from stitching base cases (Figs. 5, 6). (*iii*) occurs as a result of the stitching merge step

well as c_{i+1} and lying above $\overrightarrow{c_ic_{i+1}}$. We have the following possibilities for the ratio $s_i{:}s{:}s_{i+1}$, as determined by balancing conditions in $\mathcal{QT}$. Possible configurations for α_i when $s_i{:}s{:}s_{i+1} \equiv$ 1:1:1 are shown in Fig. 25. In Fig. 25(i), (iv), and (vi), the point in the cell adjacent to c_i and c_{i+1} may be either a cell center or an extra point of a larger cell. When $s_i{:}s{:}s_{i+1} \equiv 1{:}2{:}1$ or $s_i{:}s{:}s_{i+1} \equiv 1{:}\frac{1}{2}{:}1$, possible configurations of α_i are shown in Figs. 26 and 27, respectively.

When $s_i{:}s{:}s_{i+1} \equiv$ 1:1:2, possible configurations of α_i are shown in Fig. 28. For the case when $s_i{:}s{:}s_{i+1} \equiv$ 2:1:1, the α_i are obtained by reflections about the line $y = x$ of those in Case 2.1.4. Hence, the minimum internal angle shown in Fig. 28 holds here as well. Similarly, Fig. 29 depicts α_i when $s_i{:}s{:}s_{i+1} \equiv$ 1:2:2

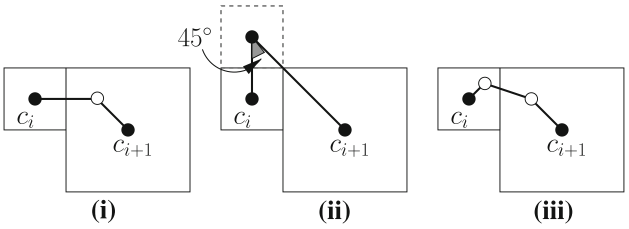


Fig. 23 Configurations for α_i when $s_i{:}s_{i+1}$ is 1:2. (*i*) and (*ii*) come from *applyTemplate* and stitching base cases, and (*iii*) occurs as a result of the stitching merge step

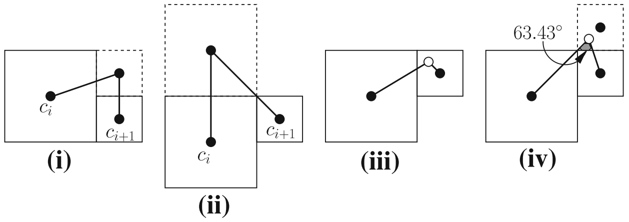


Fig. 24 Configurations for α_i when $s_i{:}s_{i+1}$ is 2:1. (*i*) and (*ii*) come from *applyTemplate* and stitching base cases, whereas (*iii*) and (*iv*) occur only in the stitching base cases

and the chains in this figure are reflections about the line $y = x$ of possible α_i when $s_i{:}s{:}s_{i+1} \equiv$ 2:2:1.

Finally, Fig. 30 shows possible configurations of α_i when $s_i{:}s{:}s_{i+1} \equiv$ 1:2:4. For the case when $s_i{:}s{:}s_{i+1} \equiv$ 4:2:1, the α_i are obtained by 180° rotations of those in Fig. 30.

Case 2.2 α_i **constructed by application of *applyTemplate* or *stitchChains***. All new configurations of α_i that occur by a template application, or a stitching step at some level of recursion are listed. By "new", we mean configurations that do not appear in Figs. 25, 26, 27, 28, 29 and 30. Note that when c_i and c_{i+1} are connected via templates or stitchings, $s_i{:}s_{i+1}$ is 1:1 (see Fig. 31), 1:2, (see Fig. 32) or 2:1 (reflections about the line of $y = x$ of the α_i in Fig. 32), but not 1:4 or 4:1. While Figs. 25, 26, 27, 28, 29, 30, 31 and 32 all depict c_i and c_{i+1} in the southwest and northeast quadrants, respectively, note that each of the α_i in these figures has a 90° rotational symmetry corresponding to c_i and c_{i+1} in the northwest and southeast quadrants. Clearly, this does not change the minimum internal angles indicated in those figures.

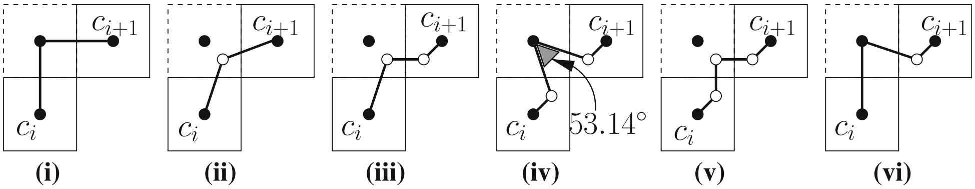


Fig. 25 s_i:s:$s_{i+1} \equiv$ 1:1:1

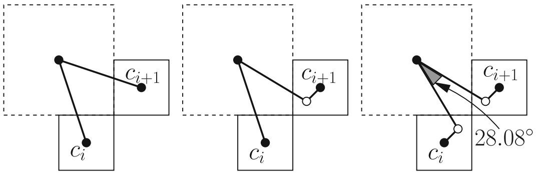


Fig. 26 s_i:s:$s_{i+1} \equiv$ 1:2:1

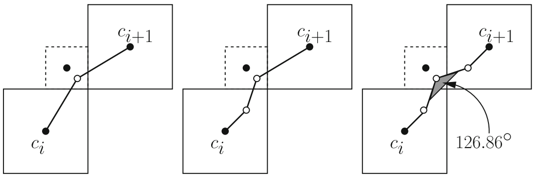


Fig. 27 s_i:s:$s_{i+1} \equiv 1{:}\frac{1}{2}{:}1$

It follows from the above case analysis that α_i has at most four edges. □

We now describe how to quadrangulate each polygonal region $A_i = (v_i, c_i) \cdot \alpha_i \cdot (c_{i+1}, v_{i+1}) \cdot (v_{i+1}, v_i)$ independently for $1 \leq i \leq m$. Before doing this, we first show that rather than using the vertical projections v_i and v_{i+1}, we may instead use perpendicular projections of c_i and c_{i+1} onto edge $\vec{e}$. This allows us to prove angle bounds for quadrangulating A_i that are independent of the angle that $\vec{e}$ makes with the horizontal (recall that this is between −45° and 45°).

Lemma 7. *Let δ_i be the signed angle (in degrees) between $\overrightarrow{c_i c_{i+1}}$ and the positive x-axis. Then* $\text{abs}(\delta_i) \in \{0, 18.43, 45, 71.57, 90\}$.

Proof. Since c_i and c_{i+1} are both cell centers in QT, and we know from Lemma 4 that they are edge or corner neighbors, it follows that there are a constant number of possibilities for δ_i: If c_i and c_{i+1} are edge neighbors with $s_i = s_{i+1}$, then δ_i is either 0 or 90°. If c_i and c_{i+1} are edge neighbors with $s_i \neq s_{i+1}$, then $\tan(\delta_i) = \frac{1}{3}$, i.e., $\text{abs}(\delta_i) = 18.43°$, or $\tan(\delta_i) = 3$, i.e., $\text{abs}(\delta_i) = 71.57°$. If c_i and c_{i+1} are corner neighbors, then $\text{abs}(\delta_i) = 45°$.

Let θ be the signed angle made by $\vec{e}$ with the positive x-axis. The value of θ determines the range of possibilities for δ_i. This is because of our definition of the quadtree chain, which specifies that either cell c_i (resp. c_{i+1}) has center above $\vec{e}$ and is intersected by $\vec{e}$, or it is the north/west neighbor of a cell intersected by $\vec{e}$ whose center lies below $\vec{e}$. Thus, for example, when $0 \leq \theta < 18.43°$, we must have $-45° \leq \delta_i < 90°$. That is, it is impossible for δ_i to equal −71.57° or −90° when the value of θ is small. The following table summarizes the possible values of δ_i for given ranges of θ:

Lemma 8. *Let p_i be the perpendicular projection of c_i on $\vec{e}$,and v_i the vertical projection of c_i on $\vec{e}$.Assume $\vec{e}$ makes an angle between −45° and 45° with the positive x axis. Then for all $1 \leq i < m$, c_{i+1} lies outside the triangle $\Delta(p_i c_i v_i)$.*

Proof. Observe that the relationship between θ and δ_i given in Table 1 implies that when $\theta \geq 0$, $\delta_i > -(90 - \theta)$, and when $\theta < 0$, $\delta_i > -90$. This implies that the segment $c_i c_{i+1}$ will never swing past the edges $c_i\ p_i$ or $c_i\ v_i$, and hence c_{i+1} lies strictly outside the triangle $\Delta(p_i c_i v_i)$. See Fig. 33 for an illustration of the case when $\theta > 0$. □

We know from Lemma 5 that for all $1 \leq i \leq m$, α_i does not intersect $\overline{c_i v_i}$ or $\overline{c_{i+1} v_{i+1}}$. Furthermore, we know from the proof of Lemma 6 that α_i lies above $\overrightarrow{c_i c_{i+1}}$ (that is, it does not intersect the region bounded by $c_i\ v_i\ v_{i+1} c_{i+1}$). Therefore, Lemma 8 implies that α_i does not intersect $\overline{c_i p_i}$ or $\overline{c_{i+1} p_{i+1}}$ either. We redefine polygon A_i to be $(p_i, c_i) \cdot \alpha_i \cdot (c_{i+1}, p_{i+1}) \cdot (p_{i+1}, p_i)$ (that is, it is defined by the perpendicular projections rather than the vertical ones).

Lemma 9. *Let $\phi_1 = \angle p_i c_i c_{i+1}$ and $\kappa_1 = \angle c_i c_{i+1} p_{i+1}$. Then* $\min\{\phi_1, \kappa_1\} \geq 18.43°$ $\max\{\phi_1, \kappa_1\} \leq 161.57°$.

Proof. Refer to Fig. 34a. Since $\phi_1 = 90 - \theta + \delta_i$ and $\kappa_1 = 90 + \theta - \delta_i$ (recall θ and δ_i are signed angles), and the fact $-71.57° \leq (\theta - \delta_i) \leq 71.57°$ (refer to Table 1), it follows that $\phi_1 \geq 18.43°$ and $\kappa_1 \geq 18.43°$. Since $\phi_1 + \kappa_1 = 180°$, it follows immediately that $\max\{\phi_1, \kappa_1\} \leq 161.57°$.

Lemma 10. *Suppose α_i has two edges, $c_i\ v$ and $v c_{i+1}$.Let $\phi_1 = \angle p_i c_i c_{i+1}, \kappa_1 = \angle c_i c_{i+1} p_{i+1}, \phi_2 = \angle c_{i+1} c_i v$,and $\kappa_2 = \angle c_i c_{i+1} v$. Then (i)* $\min\{\phi_1, \kappa_1\} > 18.43°$ *and (ii)* $\min\{\phi_1 + \phi_2, \kappa_1 + \kappa_2\} \geq 2 \times 18.43°$.

Table 1 Range of values for θ and δ_i

Range of θ	Values of δ_i
$18.43° \leq \theta \leq 45°$	$-18.43° \leq \delta_i \leq 90°$
$0 < \theta < 18.43°$	$-45° \leq \delta_i \leq 71.57°$
$\theta = 0$	$-45° \leq \delta_i \leq 45°$
$-18.43° < \theta < 0$	$-71.57° \leq \delta_i \leq 45°$
$-45° \leq \theta \leq -18.43°$	$-71.57° \leq \delta_i \leq 18.43°$

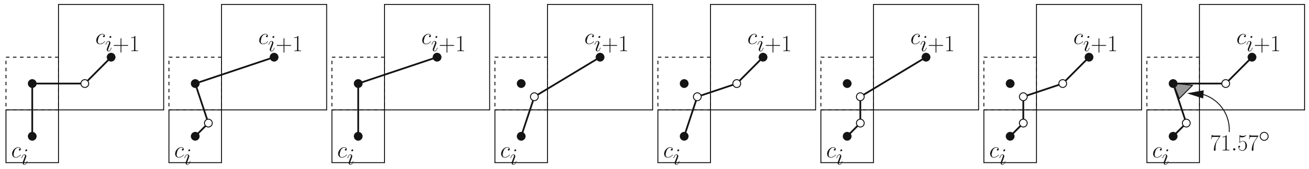
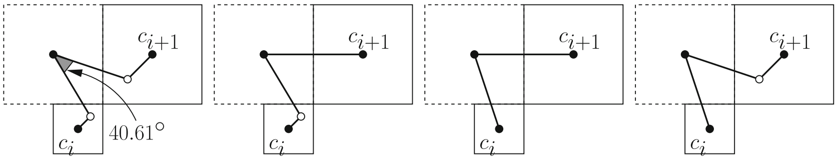


Fig. 28 $s_i{:}s{:}s_{i+1} \equiv$ 1:1:2

Fig. 29 $s_i{:}s{:}s_{i+1} \equiv$ 1:2:2



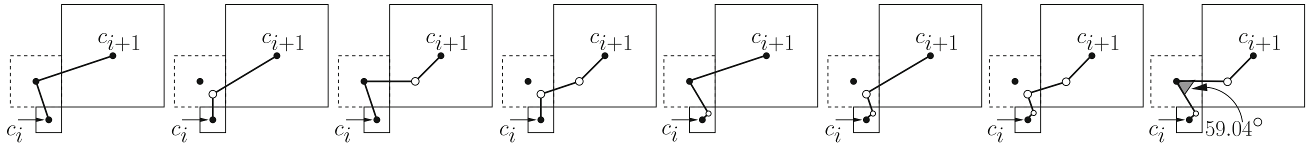


Fig. 30 $s_i{:}s{:}s_{i+1} \equiv$ 1:2:4

Fig. 31 $s_i{:}s_{i+1} \equiv$ 1:1

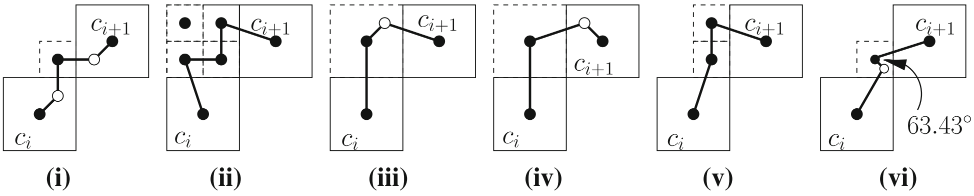
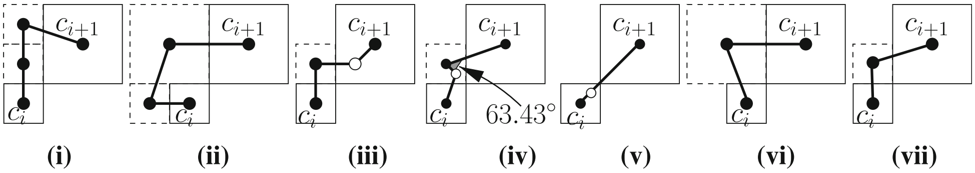


Fig. 32 $s_i{:}s_{i+1} \equiv$ 1:2



Proof.

(i) From Lemma 9 we know that $\min\{\phi_1, \kappa_1\} \geq 18.43°$. To see that it must be *strictly* greater, note that if $\min\{\phi_1, \kappa_1\} = 18.43°$, then $abs(\theta - \delta_i) = 71.57°$. From Lemma 7 and Table 1, it can be seen that $abs(\theta - \delta_i) = 71.57°$ when (a) $\theta = 18.43°$ and $\delta_i = 90°$, which is impossible because c_i and c_{i+1} are directly connected whenever $\delta_i = 90°$, or (b) $\theta = 0$ and $\delta_i = 71.57°$, which is also impossible because θ must be strictly >0 whenever $\delta_i = 71.57°$.

(ii) Refer to Fig. 34. First observe that if $\min\{\phi_2, \kappa_2\} \geq 18.43°$, then part (*i*) implies the claim. Hence assume that $\min\{\phi_2, \kappa_2\} < 18.43°$. The only configurations of α_i for which $\min\{\phi_2, \kappa_2\} < 18.43°$ are shown in Fig. 34c–e. In Fig. 34c, c_i and c_{i+1} are edge neighbors with $\min\{\phi_2, \kappa_2\} = 12.53° = \arctan(\frac{3}{5}) - \arctan(\frac{1}{3})$. In Fig. 34d and e, c_i and c_{i+1} are corner neighbors with $\min\{\phi_2, \kappa_2\} = 14.04° = \arctan(\frac{1}{4})$. Assume wlog that $\phi_1 + \phi_2 \leq \kappa_1 + \kappa_2$ and suppose, for the sake of contradiction, that $\phi_1 + \phi_2$

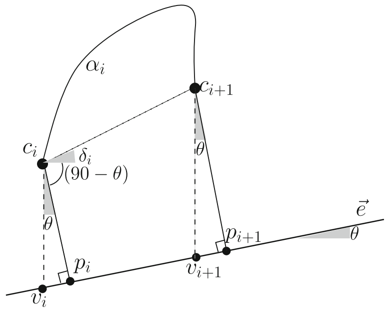


Fig. 33 c_{i+1} lies outside $\Delta(p_i c_i v_i)$

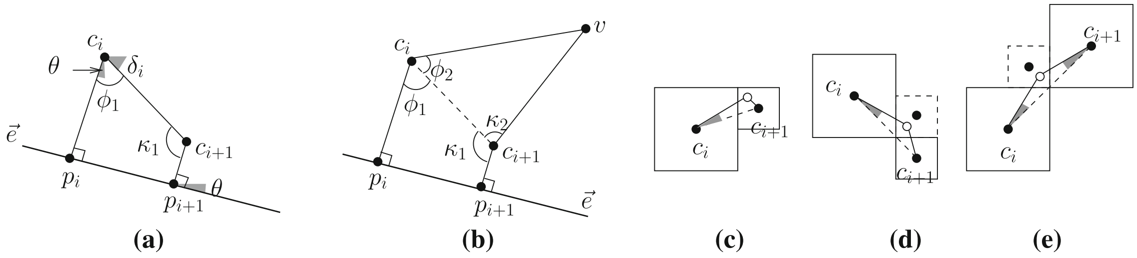


Fig. 34 **a** $\min\{\phi_1,\kappa_1\} \geq 18.43°$. **b** $\min\{\phi_1 + \phi_2, \kappa_1 + \kappa_2\} \geq 2 \times 18.43°$. **c** *Shaded angle* is 12.53°. **d**, **e** *Shaded angle* is 14.04°

$< 2 \times 18.43°$. If $\min\{\phi_2,\kappa_2\} = 12.53°$, we have $\mathrm{abs}(\delta_i) = 18.43°$ (see Fig. 34c). Therefore, $\phi_1 = 90 - \theta + \delta_i < 24.32°$. This inequality implies that $47.25° < \theta < 84.11°$, which violates our assumption about θ.

If $\min\{\phi_2,\kappa_2\} = 14.04°$, then $\mathrm{abs}(\delta_i) = 45°$ (see Fig. 34d, e). This implies $\phi_1 = 90 - \theta + \delta_i < 22.82°$. When $\delta_i = 45°$, this implies that $\theta > 112.18°$, which is impossible. When $\delta_i = -45°$, we have $\theta > 22.18°$, which violates the angle dependency shown in Table 1. This completes the proof of part (*ii*).

□

Lemma 11. *For* $1 \leq i \leq m - 1$, *the simple polygon* $A_i = (p_i, c_i) \cdot \alpha_i \cdot (c_{i+1}, p_{i+1}) \cdot (p_{i+1}, p_i)$ *can be quadrangulated with at most five quadrilaterals with a minimum angle of* $18.43°(=\arctan(\frac{1}{3}))$ *and maximum angle of* $171.86°$ $(=135° + 2\arctan(\frac{1}{3}))$.

Proof. From Lemma 6, we know that α_i has one, two, three, or four edges. We consider each case separately.

Case 1 α_i **has one edge**. In this case, A_i is already a quadrilateral. It follows from Lemma 9 that all angles of A_i are at least 18.43° and at most 161.57°.

Case 2 α_i **has two edges**. Let $c_i\ v$ and vc_{i+1} be the two edges of α_i. Let ϕ_1, κ_1, ϕ_2, and κ_2 be as in Fig. 34b. Let $\gamma = \angle c_i v c_{i+1}$. Observe that $\gamma \geq 26.57°$ because the edges of α_i come from $\mathcal{Q}$. The method to quadrangulate A_i depends on the angles ϕ_2 and κ_2. We show that in any case, A_i can be decomposed into three quadrilaterals.

- If $\min\{\phi_2,\kappa_2\} \geq 18.43°$, place a Steiner point s on $\overline{c_i c_{i+1}}$ such that the circle $\mathcal{C}$ centered at c_i with radius $c_i s$ intersects the edge $\overline{c_i p_i}$ (see Fig. 35a). Place another Steiner point at the perpendicular projection p of s onto $\vec{e}$. Connect s to c_i, c_{i+1}, and p to obtain a quadrangulation of A_i. We know from Lemma 10(*i*) that all angles in the resulting quadrangulation are strictly greater than 18.43°.

  Note that $\angle c_i s c_{i+1} = 180°$. We argue that s can be perturbed so that all angles in A_i are at most 171.86°. From Table 1 and the fact that $\phi_1 = 90 - \theta + \delta_i$, it follows that $\phi_1 \geq 26.57°$ *except* for the case when $-(26.57° - 18.43°) < \theta < 0$ *and* $\delta_i = -71.57°$. We have $\delta_i = -71.57°$ only when c_i and c_{i+1} are edge neighbors whose cell sizes have a 2:1 ratio and the larger cell is the north neighbor of the smaller one (see Fig. 35b). This configuration can only arise when $-18.43° < \theta < 0$. The only possible 2-edge connectivity between c_i and c_{i+1} under these conditions is illustrated in Fig. 23(i), 24(ii), or 24(iv). In each of these cases, the point p_i can be moved so that the edge $c_i p_i$ swings outward to increase ϕ_1 to 26.57°. See Fig. 35b. Therefore, in all cases we have $\phi_1 \geq 26.57°$. This implies that s can be moved along the circle $\mathcal{C}$ until $\angle p_i c_i s = 18.43°$ and $\angle s c_i c_{i+1} \geq 26.57° - 18.43°$, which in turn implies that $\angle c_i s c_{i+1} \leq 180° - (26.57° - 18.43°) = 171.86°$. The same upper bound on the remaining angles at s follows immediately.

- If $\min\{\phi_2,\kappa_2\} < 18.43°$, the placement of Steiner points depends on which of ϕ_2 and κ_2 is smaller than 18.43°. Recall from Lemma 10 that there are exactly three configuration of α_i for which $\min\{\phi_2,\kappa_2\} < 18.43°$ (ref Fig. 34c,d). We consider three cases:

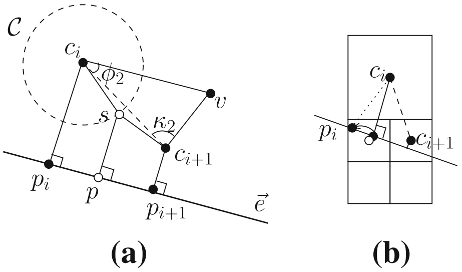


Fig. 35 α_i has two edges. **a** $\min\{\phi_2,\kappa_2\} \geq 18.43°$. **b** $\delta_i = -71.57°$

- $\phi_2 < 18.43°$ **and** $\kappa_2 \geq$ **18.43°**. In this case, either $\delta_i = 18.43°$ (Fig. 34(c)) or $\delta_i = -45°$ (Fig. 34d). In the case of the former, Table 1 implies that $-18.43° < \theta \leq 45°$, which in turn implies that $63.43 \leq \phi_1 \leq 90 + 2 \times 18.43$. In the case of the latter, we have $-45° \leq \theta \leq 18.43°$ and hence $26.57 \leq \phi_1 \leq 90$. We use these angle bounds for ϕ_1 to demonstrate a quadrangulation of A_i with a minimum angle bound of 18.43° and a maximum angle bound of 171.86°, as follows: Place a Steiner point s in A_i within the cone $c_i v c_{i+1}$ such that $\angle p_i c_i s = 18.43°$. The fact that $\angle s c_i v \geq 18.43$ follows immediately from Lemma 10. Also place a Steiner point at the perpendicular projection of s onto $\vec{e}$. Connect s to c_i, p, and c_{i+1}. See Fig. 36a. From the previously derived bounds, we know $26.57 \leq \phi_1 \leq 90 + 2 \times 18.43$, which implies $26.57 - 18.43 \leq \angle s c_i c_{i+1} \leq 90 + 18.43$ and hence $180 - (90 + 18.43) = 71.57 \leq \angle c_i s c_{i+1} \leq 180 - (26.57 - 18.43) = 171.86$. In addition, we have $\angle c_i s p = 180 - 18.43 = 161.57$. This implies $26.57 \leq \angle p s c_{i+1} \leq 126.86$ and $53.14 \leq \angle s c_{i+1} p_{i+1} \leq 153.43$. Finally, since $\kappa_2 \leq 63.43°$ in the configurations in Fig. 34c and d and $\kappa_1 \leq 153.43°$, it follows that $\angle s c_{i+1} v \leq 163.72°$.
- $\boldsymbol{\phi_2 \geq}$ **18.43° and** $\boldsymbol{\kappa_2 <}$ **18.43°**. This case is symmetric to the one above with $\delta_i = -18.43°$ or $\delta_i = 45°$. Carry out a procedure similar to the above case to obtain the same angle bounds.
- $\boldsymbol{\phi_2 <}$ **18.43° and** $\boldsymbol{\kappa_2 <}$ **18.43°**. The only configuration of α_i for which both ϕ_1 and ϕ_2 are less than 18.43° is shown in Fig. 34e. Observe that in this case, v can see $\vec{e}$. Let s be the perpendicular projection of v onto $\vec{e}$, unless $0 < \theta < -18.43°$, in which case let s be the vertical projection of v onto $\vec{e}$. Connect v to s to obtain a quadrangulation of A_i. See Fig. 36b. Since $26.57 \leq \phi_1 \leq 90$ in this case, minimum and maximum angle bounds on $\angle v c_i p_i$ and $\angle s v c_i$ follow immediately. Finally, $\angle c_{i+1} v s \geq 18.43$, we have $\angle p_{i+1} c_{i+1} v \leq 161.57$.

Case 3 α_i **has three edges**. The method used to quadrangulate A_i depends on whether the number of reflex internal vertices of α_i is zero or one (note that since α_i lies above $\overrightarrow{c_i c_{i+1}}$, it is not possible for both internal angles to be reflex):

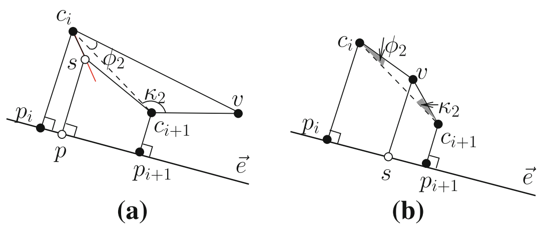


Fig. 36 α_i has two edges. **a** $\phi_2 < 18.43°$ and $\kappa_2 \geq 18.43°$. **b** $\phi_2 < 18.43°$ and $\kappa_2 < 18.43°$

- If the two internal angles along α_i are both convex, draw an edge between c_i and c_{i+1}, which quadrangulates A_i with two quadrilaterals. Some examples of such α_i can be seen in Figs. 23(iii) and 32(ii). In all such cases, Lemma 9 guarantees that all angles in the quad below $\overrightarrow{c_i c_{i+1}}$ is at least 18.43° and at most 161.57°. The quad above $\overrightarrow{c_i c_{i+1}}$ has a minimum angle of 26.57° and a maximum angle of 153.43°. See Fig. 37a.
- If one of the internal angles along α_i is reflex, c_i and c_{i+1} must be corner neighbors. Let r be the reflex vertex. r either lies on the segment $\overline{c_i c_{i+1}}$, or belongs to the quadtree cell $N(c_i)$ adjacent to c_i and c_{i+1} and lying above $\overrightarrow{c_i c_{i+1}}$. Several examples of the former appear in Figs. 25, 26, 27, 28, 29 and 30. For the latter, see Figs. 31(v) and (vi) and 32(iv).
If r lies on $\overline{c_i c_{i+1}}$, insert an edge from r to the perpendicular projection of r onto $\vec{e}$. See Fig. 37b. This decomposes A_i into a quadrilateral and a pentagon. Lemma 9 guarantees the minimum and maximum angle bounds for the quadrilateral. The pentagon can be decomposed into three quads by applying Case 2 (Figs. 35, 36a). Note that even though r may be an extra point rather than a cell center, Lemma 10 and the proven angle bounds in Case 2 are valid for this pentagon as well because r lies on $\overline{c_i c_{i+1}}$.
If r belongs to the quadtree cell $N(c_i)$, the edge from r to the perpendicular projection of r onto $\vec{e}$ may make one of the angles at r too small (this happens only when $-18.43° \leq \theta < 0$). In this case, use the vertical projection of r onto $\vec{e}$ (which lies between p_i and p_{i+1}) to decompose A_i into a quadrilateral and a pentagon. See Fig. 37c. The quadrilateral has the required angle bounds by construction.

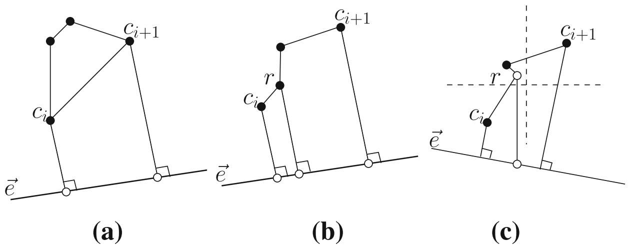


Fig. 37 α_i has three edges. **a** Two internal convex vertices. **b** One internal reflex vertex r lies on $\overline{c_i c_{i+1}}$. **c** One internal reflex vertex r belongs to quadtree cell $N(c_i)$

Since $\angle rc_{i+1}p_{i+1} > 26.57°$ (because we know $-18.43° \le \theta < 0$), it follows that we can apply the construction in Case 2 (refer Fig. 35a) to decompose the pentagon into three quads with the required minimum and maximum angle bounds.

Case 4 α_i **has four edges**. The α_i are classified according to the three internal vertices:

- If the three internal vertices consist of two reflex vertices separated by a convex vertex (e.g., Fig. 31(i)), the reflex vertices always lie on $\overline{c_i c_{i+1}}$. Insert edges from each reflex vertex to its perpendicular projection onto $\vec{e}$. This decomposes A_i into two quads and a pentagon. Then we can apply Case 2 to the pentagon. Lemmas 9 and 10 provide the required minimum and maximum angle bounds. Again, even though the reflex vertices are extra points, the lemmas still apply because they lie on $\overline{c_i c_{i+1}}$. See Fig. 38a.
- If the three internal vertices consist of two convex vertices separated by a reflex vertex (Fig. 31(ii)), decompose A_i into four quads as shown in Fig. 38b. The minimum and maximum angle bounds for the quads below $\overline{c_i c_{i+1}}$ follow from Lemma 9 and for the quads above $\overline{c_i c_{i+1}}$ from Theorem 1.

This completes the proof that A_i can be quadrangulated with at most five quadrilaterals with a minimum angle of 18.43° and maximum angle of 171.86°. □

3.1.1 Edge separation conditions for quadtree

Every edge $\vec{e}$ of the polygon P defines a chain of edges given by $\cup_{1 \le i \le m} \alpha_i$. From this chain, we obtain the polygons A_i, each of which is then quadrangulated as described above. In order to conduct this process independently for every edge of the polygon, we impose an *edge separation condition* on $\mathcal{QT}$. The edge separation condition requires that all quadrangulation chains $\cup_{1 \le i \le m} \alpha_i$ defined by the edges of the polygon be disjoint from each other. Recall that these chains do not start in the cell containing the segment endpoint, but rather in one adjacent to it. This allows quadrangulation chains to be separated completely, except in the 5 × 5 grid of cells around each polygon vertex. In the worst case, the edge separation condition requires that every cell intersected by a polygon edge be surrounded by a 3 × 3 grid of empty cells, but in practice, this requirement does not apply uniformly across the entire segment.

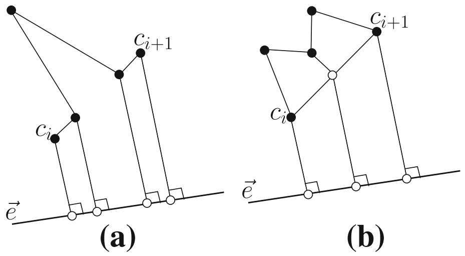


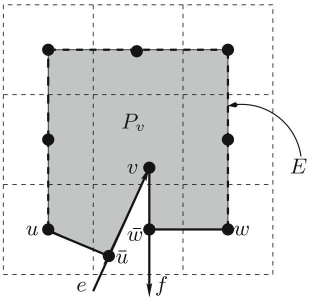
Fig. 38 α_i has four edges. **a** Two reflex vertices separated by a convex vertex. **b** Two convex vertices separated by a reflex vertex

3.2 Connecting quadtree chains around polygon vertices

For every edge of the polygon P, the quadtree chain starts and ends at a cell center within the 3 × 3 grid of quadtree cells that is guaranteed to exist around each of its endpoints.

Let v be a vertex of P and let e and f be the two edges incident on v, oriented counterclockwise (the interior of P lies to their left). Let u be the last quadtree chain vertex for edge e and let w be the first quadtree chain vertex for edge f. Note that u and w are both cell centers in the 3 × 3 grid around v; furthermore, the entire 3 × 3 grid does not contain extra points. Let $\bar{u}$ and $\bar{w}$ be the perpendicular projections of u and w onto e and f, respectively. Recall that the angle between edges e and f is at least 90°.

Let E be a sequence of edges connecting u to w in the 3 × 3 grid (shown dotted). The region around vertex v is meshed by quadrangulating the polygon defined by the edges $v\bar{u}, \bar{u}u, E, w\bar{w}, \bar{w}v$. Call this polygon P_v.



Lemma 12. *P_v can be decomposed into at most seven quadrilaterals with a minimum angle of* 18.43° *and maximum angle of* 171.86°.

Proof. The method used to quadrangulate P_v depends on the number of edges in E, which is between one and seven (inclusive).

Case 1 E **has one edge** When the number of edges in E is one, P_v is a pentagon. Note that the edges e and f must be angled strictly above the neighboring cell centers (illustrated as empty circles in Fig. 39); otherwise, E would contain more than one edge. One of u and w is the cell center of an edge neighbor to the center cell (where v resides) of the 3×3 grid while the other is a corner neighbor. Without loss of generality, Fig. 39 illustrates u as the cell center of the edge-neighboring cell. Now consider the triangle Δuvw, which always exists regardless of the location of v. Angle $\angle uvw$ depends on v's location and is at its minimum of 26.56° when v is at the exact upper left corner of the cell (see Fig. 39a). Since the angle between the edges e and f is at least 90°, there is always a ray $\vec{r}$ through v that intersects Δuvw and subtends angles of at least 26.57° with each of e and f.

Next, find the intersection t of $\vec{r}$ and the diagonal from u to the lower right corner of u's cell. Because the diagonal always intersects Δuvw regardless of the location of v, t always exists. Note the angle $\angle tuw$ is exactly 45°, and $45 + 18.43 < \angle \bar{u}ut < 135 + 18.43$ because edges e and f subtend a non-acute angle and must lie above the neighboring cell centers (drawn as empty circles in Fig. 39).

Finally, let q be the intersection between $\overline{vw}$ and the horizontal through t. Let $\bar{q}$ be the perpendicular projection of q onto f. Connect t to u, v, and q, and q to w and $\bar{q}$. The resulting quadrangulation of P_v into four quads is shown in Fig. 39a–c. As shown in Fig. 39d, under certain extreme placements of v and f, the angle $\angle vw\bar{w}$ becomes smaller than 18.43°. This situation can occur only when v lies in the upper right quadrant of its cell and f makes an angle greater than $90 - 18.43$ with the horizontal. In this case, we no longer extend the horizontal from t as far as vw, but stop sooner so that $\angle qw\bar{w} > 18.43°$. If the perpendicular from the new position of q to f does not lie on f, we change the quadrangulation as follows: Let $\bar{t}$ be the perpendicular projection of t onto e. Connect t to u, $\bar{t}$ and q, and q to w and v, as shown in Fig. 39d.

The minimum angle bound in the resulting quadrangulation of P_v follows from the facts that $\angle uwq \geq 18.43°$ and ray $\vec{r}$ subtends at least 26.57° with e as well as f. For the maximum angle bound, first observe that two of the four quads in the quadrangulation have a pair of angles that add up to 180°. These are the quads (u, w, q, t) and $(q, w, \bar{w}, \bar{q})$, or $(u, t, \bar{t}, \bar{u})$. Hence, the minimum angle bound immediately implies a maximum angle bound of $180 - 18.43 = 161.57°$ in these quads. In the cases when v is connected to t (Fig. 39a–c), we have $\angle vtq \leq 161.57°$ because $\angle tqv \geq 18.43°$. Furthermore, since $\angle vtq > 90°$, we have $\angle tq\bar{q} < 161.57°$ and $\angle utv < 135°$. As observed previously, $\angle \bar{u}ut < 135 + 18.43$. This proves the maximum angle bound in the case that v is connected to t. In the case when v is connected to q (Fig. 39d), note that $\angle v\bar{w}w > 90$, which implies that $\angle vqw < 161.57°$. In turn, $\angle v\bar{w}w < 161.57°$ since $\angle vqw > 90$. This proves maximum angle bounds for the quad $(v, q, w, \bar{w})$. For the quad $(\bar{t}, t, q, v)$, $\angle \bar{t}tq < 135°$ because $\angle ut\bar{t} > 90°$ for the placement of v implied by this case. Since the angle between e and f can be at most $180 + 26.57$, we know that $\angle \bar{t}vq < 90 + 26.57$. Finally, since $\angle tqw > 180 + 18.43$ and $\angle vqw > 90$, it follows that $\angle vqt < 180 - 18.43 = 161.57°$. Hence, we have a maximum angle bound of 161.57° in this case as well.

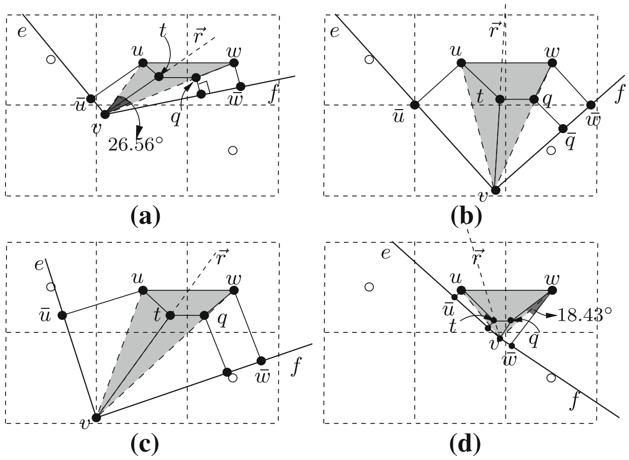
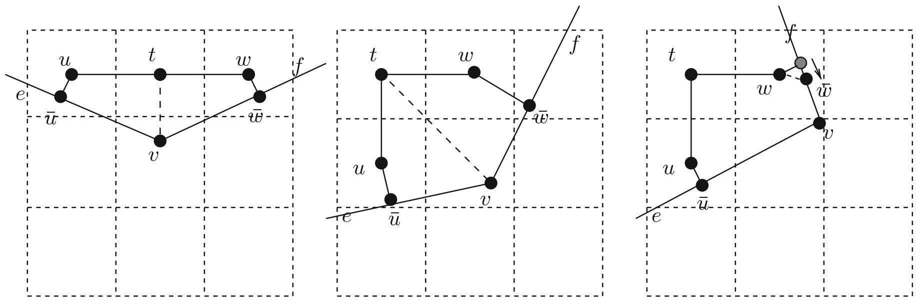


Fig. 39 Quadrangulating the corner regions. Number of edges between u and w is one

Case 2 E **has two edges** When the number of edges in E is two, P_v is decomposed into two quadrilaterals. There are two cases depending on the angles at u and w within P_v. If both $\angle \bar{u}ut \leq 171.86°$ and $\angle \bar{w}wt \leq 171.86°$, quadrangulate P_v by connecting v to t (see Fig. 40a–b). Since $\angle vtu$ and $\angle vtw$ are both at least 18.43° for any placement of v in its cell, it follows that $\angle vtu \leq 161.57°$. Furthermore, since $\angle \bar{u}ut \geq 90$, it follows that $18.43 \leq \angle \bar{u}vt \leq 161.57$. A symmetric argument proves the angle bounds for the quad $(t, w, \bar{w}, v)$.

Fig. 40 Quadrangulating the corner regions. Number of edges between u and w is two



(a) **(b)** **(c)**

If either $\angle \bar{u}ut > 171.86°$ or $\angle \bar{w}wt > 171.86°$, we move the perpendicular projection along its edge until the desired angle bounds are reached. Wlog, suppose $\angle \bar{w}wt > 171.86°$. Move $\bar{w}$ along f until $\angle \bar{w}wt = 171.86$. $\bar{w}$ is guaranteed to stay on f because $\angle vwt \le 135$. To see that the remaining angles (outside P_v) incident on w satisfy the angle bounds, observe that w must have a third edge from the quadrangulation of the quadtree chain for f. Since the angle between this edge and the edge tw is at least 18.43° by construction, it follows that after $\bar{w}$ has been moved, the angle made by this edge with $w\bar{w}$ is at most $360 - 18.43 - 171.86 = 169.71°$. Quadrangulate the new P_v as described in the previous paragraph, where the same angle bounds hold because $\angle v\bar{w}w > 90$.

Case 3 *E* **has three or more edges** When the number of edges in E is three or larger, connect v to the cell centers of its edge neighbors (with the exception of u and w). Furthermore, the cell centers connected to v are moved so that they are aligned vertically or horizontally with v. See Fig. 41. This decomposes P_v into quadrilaterals and one or two pentagons. The angle bounds for the quadrilaterals in the resulting quadrangulation follow immediately. Each pentagon is subdivided into three quads as follows (refer to Fig. 41a): Place a Steiner point s on ux and perturb it vertically such that $\angle usx = 171.86$. Connect s to u, x, and its vertical projection $\bar{s}$ onto e. For any placement of v in its cell, $\arctan(\frac{2}{3}) = 33.69 \le \angle uxt \le 90 - \arctan(\frac{1}{2}) = 63.43$ and $26.57 \le \angle tux \le 90 - \arctan(\frac{2}{3}) = 56.31$. Hence, since $\angle vxu \ge 26.57$, it follows that s can always be perturbed so that $\angle usx \le 171.86°$ and $\angle sxv \ge 18.43°$, which in turn imply the required bounds on $\angle xs\bar{s}$ and $\angle \bar{s}su$. We have $90 < \angle \bar{s}vx = \angle \bar{u}\bar{s}s < 161.57$, which implies $18.43 \angle v\bar{s}s < 90$. Finally, if $\angle su\bar{u} \le 171.86°$, we are done. If not, move $\bar{u}$ along e until $\angle su\bar{u} = 171.86°$. Since $\angle tux \ge 26.57$, the angle at u outside P_v is at most $360 - 171.86 - 26.57 = 161.57$.

□

Figure 42a shows quadrangulation chains $\cup_{1 \le i \le m}\alpha_i$ for some edges of a polygon (the entire polygon is shown in Fig. 46a), where the quadtree chain vertices are highlighted. Fig. 42b shows the chains along with the connections around the corners. In Fig. 42c, the region bounded by the polygon edge and its chain is then quadrangulated to incorporate the polygon edge into $\mathcal{Q}$.

3.3 Summary of algorithm

We summarize below the algorithm to quadrangulate the interior of a non-acute simple polygon P of n edges $e_1, e_2, \ldots, e_n$ and vertices $v_0, v_1, \ldots, v_{n-1}$, where $e_i = (v_{i-1}, v_i)$ (where $v_n = v_0$). The resulting quadrilaterals have a minimum angle bound of 18.43°:

```
Quadrangulate(P, n):
  $\mathcal{QT}$ ← Quadtree decomposition satisfying edge separation conditions for vertices of $P$
  $\mathcal{Q}$ ← Quadrangulation resulting from applyTemplate on $\mathcal{QT}$.
  for $e_i \in \{e_1, e_2, \ldots, e_n\}$
    $\alpha(e_i)$ ← Quadrangulation chain for $e_i$
    $\mathcal{Q}(e_i)$ ← Quadrangulation of region bounded by $e_i$ and $\alpha(e_i)$,
        as given by Lemma 11.
    $\mathcal{Q}(v_i)$ ← Quadrangulation of corner polygon $P_{v_i}$,
        as given by Lemma 12.
  $\mathcal{Q}'$ ← $\bigcup_{1 \le i \le n}\mathcal{Q}(e_i) \cup \bigcup_{1 \le i \le n}\mathcal{Q}(v_i)$

  return $\mathcal{Q}' \cup (\mathcal{Q} \cap (P - \mathcal{Q}'))$
```

Theorem 13. *Given a quadtree decomposition with N quadtree cells satisfying the edge separation condition for a simple polygon P, Quadrangulate(P, n) constructs a mesh for P with at most 5N quadrilaterals in which every angle is at least* $18.43°(=\arctan(\frac{1}{3}))$ *and at most* $171.86°(=135° + 2\arctan(\frac{1}{3}))$.

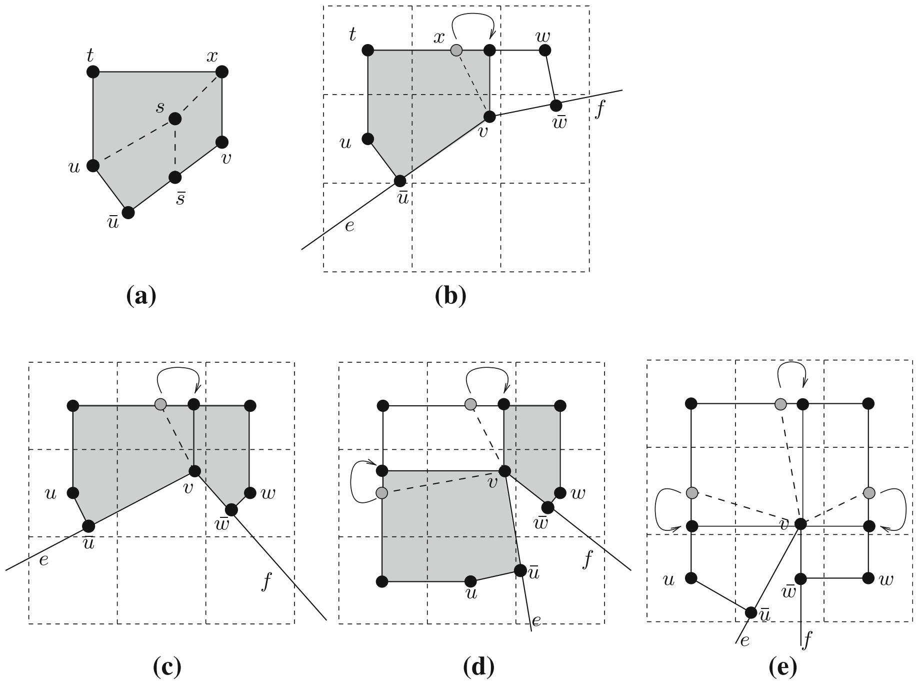


Fig. 41 Quadrangulating the corner regions. **a** Quadrangulating a pentagon. **b** One pentagon. **c**, **d** Two pentagons. **e** All quads

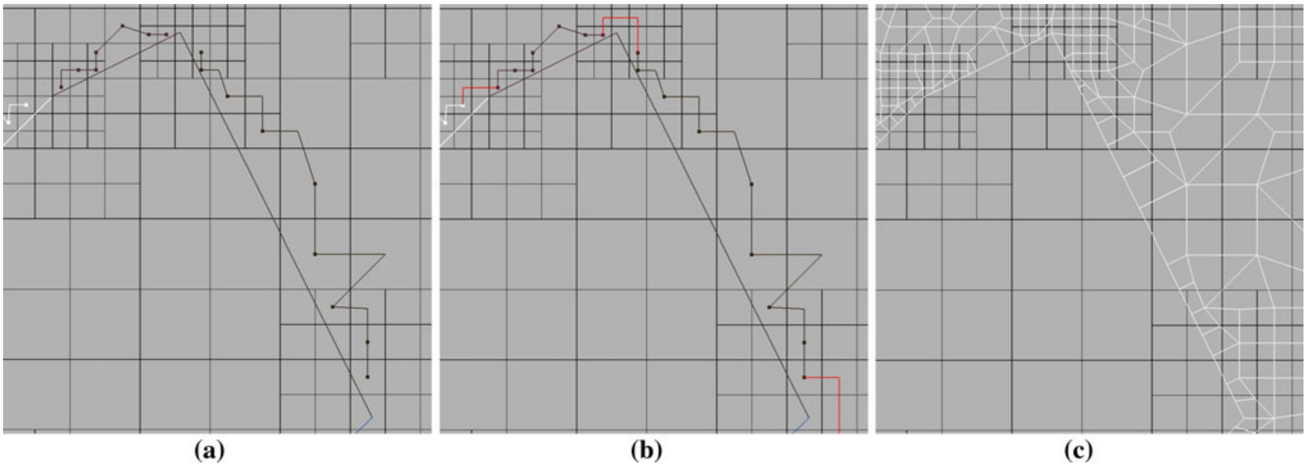


Fig. 42 **a** Quadrangulation chains for some polygon edges. **b** Connections around corners. **c** Using quadrangulation chains to incorporate edges into $\mathcal{Q}$

4 General simple polygons

We now describe how to adapt our algorithm to general simple polygons that may contain acute angles. Let P be a general simple polygon. The basic idea is to convert P into a polygon that contains only obtuse angles by "cutting off" the acute angle vertices by appropriately placing Steiner points. The modified non-acute polygon is then meshed by using the algorithm in Sect. 3 Finally, the cut pieces at the acute vertices are decomposed into quadrilaterals with the stated angle bounds. Further details are provided below.

Let a be an acute angle vertex of P. Let $\theta, 0 \leq \theta < 90$, be the angle at that vertex. Let v be a point on the angle bisector of a, and let p and q be the perpendicular projections of v onto the two edges incident at a (refer to Fig. 43). v is chosen so that the quadrangular region $apvq$ does not contain any other vertices of the polygon P. Place Steiner points at p, q, and v and draw the edges pv and vq. Perform this procedure at every acute vertex a of P, and cut the region $apvq$ from P. Let B be the polygon resulting from this procedure. Construct a quadrilateral mesh for B using the algorithm in Sect. 3 Observe that since pv and vq are edges of B, there might be Steiner points on that edge.

Let γ be the angle between pv and va, which is also the angle between qv and va, as shown in the figure. Note that γ must lie between 45° and 90° (because $\theta/2$ lies between 0 and 45). If pv and qv do not have any points on them after the quadrangulation of B, then we are done. So suppose now that pv has vertices $v_1, v_2, \ldots v_k$ (in Fig. 43, $k = 3$). Bisect the angle between pv and va, and let p_1 be the point of

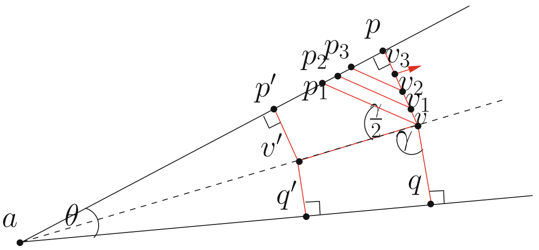


Fig. 43 Handling acute angles in *P*

intersection between this bisector and edge pa. vp_1 will be an edge of the quadrangulation. Draw edges v_ip_{i+1} parallel to vp_1, $1 \leq i \leq k-1$, as shown in the figure. Now choose a point v' on the bisector of angle at a such that the circle of tangency centered at v' touches edge ap at some point p' that is closer to a than is p_1. The point q' is the point of tangency on edge aq. Draw edges $p'v'$, $v'v$ and $v'q'$ to complete the quadrangulation. All angles in this quadrangulation (other than the one at a) are >22.5°. If there are points on the edge vq, carry out the symmetric procedure with those vertices.

We also show that all angles in this quadrangulation are at most 171.86°. First observe that because a is an acute angle, $90 < \angle p'v'v$, $\angle q'v'v < 135$ and $135 < \angle p'p_1v < (180 - 22.5) = 157.5$ (because $22.5 < \frac{\gamma}{2} < 45$). The same bounds follow for all the other angles incident at p_i and v_i, $1 \leq i \leq k$, with the exception of the 180° angle at v_k. Recall that the internal angle of p in B is 90°, but the quadrilateral mesh for B may contain a (non-boundary) edge incident on p (via one of the corner cases discussed in Sect. 3.2). However, the meshing algorithm in Section 3.2 guarantees that the resulting angles at p are at least 26.57°. This implies that we can perturb v_k along its non-boundary incident edge so that the angles at p are at least 18.43° and $\angle pv_kv_{k-1} < 180 - (26.57 - 18.43) = 171.86°$.

5 Mesh quality measures

We computed the following quality measures on two of our test datasets, the spiral polygon (Fig. 49) and the Lake

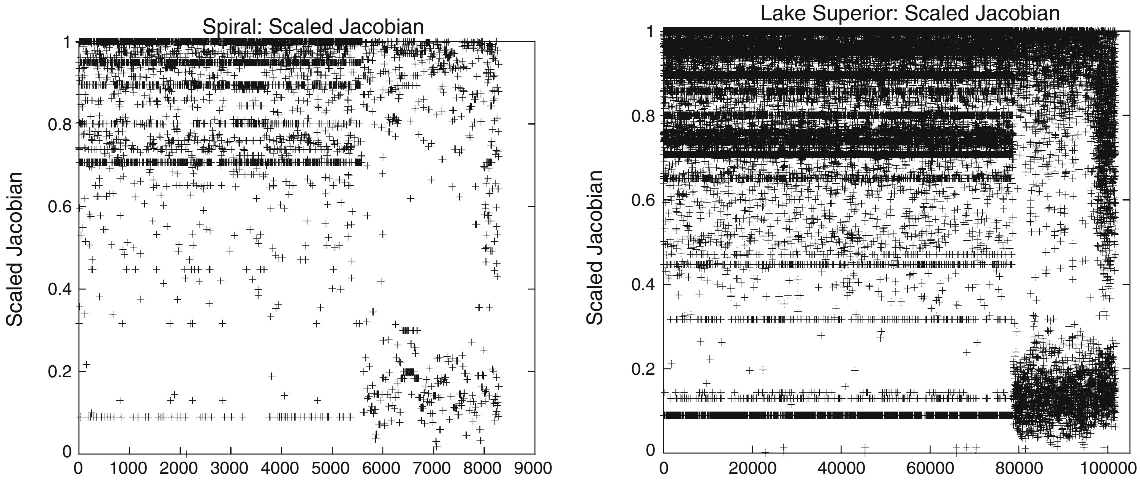


Fig. 44 Scattered plot of scaled Jacobians *left* spiral polygon, *right* lake superior polygon

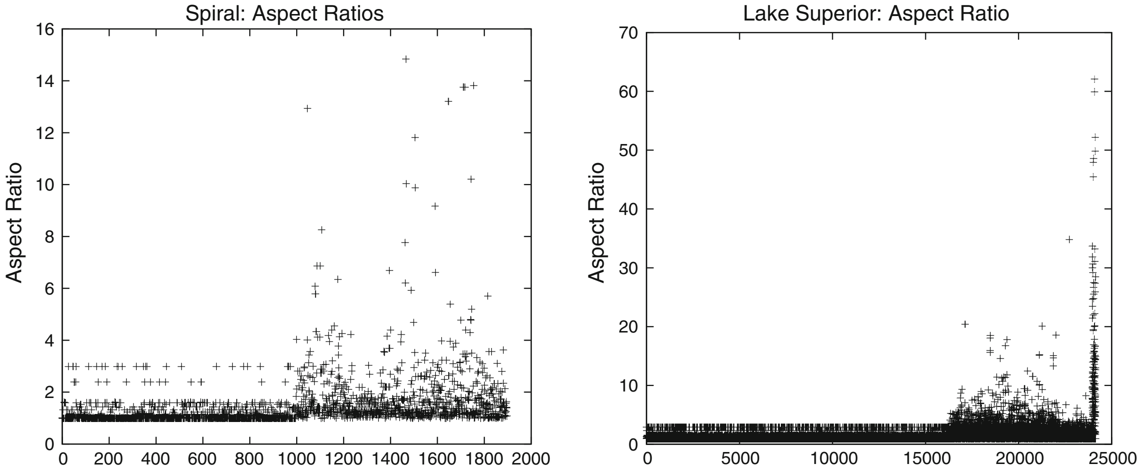


Fig. 45 Scattered plot of aspect ratios *left* spiral polygon, *right* lake superior polygon

Table 2 Numbers of vertices in the mesh with a given degree

Degree	2	3	4	5	6	7	8	9
Spiral	1	1,107	804	287	46	5	0	1
Lake Superior	19	11,623	10,445	4,154	657	100	25	3

Superior polygon (Fig. 50): (1) maximum and minimum angle in a quadrilateral, (2) maximum vertex degree, (3) scaled Jacobian (defined in [16]), and (4) aspect ratio (defined in [14]). We would like to point out that our current implementation simply uses a small perturbation to reduce 180° angles in the mesh. As a result, the implementation results on the maximum angle bound are higher than our provable bound of 171.86° (Theorem 13). For the spiral polygon, the minimum and maximum angles are 20.67° and 178.99°, respectively. For the Lake Superior polygon, they are 18.88° and 179.93°, respectively.

The maximum vertex degree in both our test polygons is 9. It occurs only when two $\mathcal{T}^{(2b)}$ cases appear next to each other around the same vertex. Note the redundant quad in the template which can easily be removed in a post-

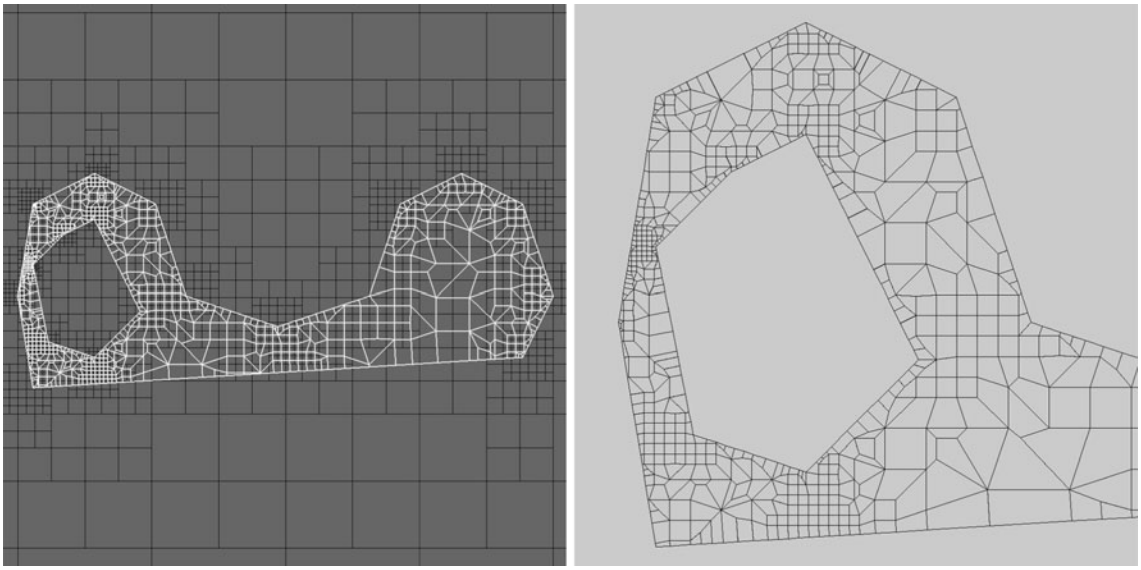


Fig. 46 Non-acute polygon with 19 edges. Minimum mesh angle: 24.26°. Number of quadtree cells: 1,399. Number of mesh vertices: 989. Number of mesh faces (quads): 841

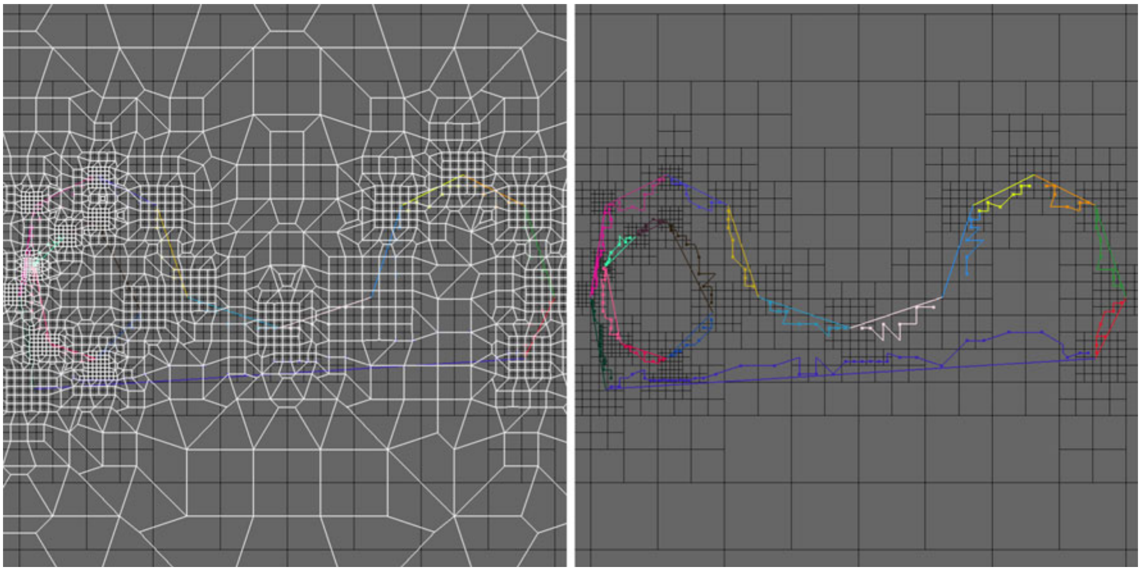


Fig. 47 *Left* Mesh given by point-set algorithm for polygon vertices. *Right* Quadrangulation chains

processing step, thereby removing all degree 9 vertices. Table 2 lists the number of vertices with each possible degree. Degree 2 arises at original corner vertices which did not get split. Whether a corner is split or not is decided by the corner meshing cases in Sect. 3.2.

We computed the scaled Jacobian for each vertex of each quadrilateral, and the scattered plots for these are given in Fig. 44. The minimum scaled Jacobian is 0.017576 in the spiral mesh and 0.001242 in the Lake Superior mesh.

We computed the aspect ratios for all the quadrilaterals in the two test meshes and the scattered plots of these aspect ratios are given in Fig. 45. The large aspect ratios in the Lake Superior mesh are due to the way we mesh the acute corners (Sect. 4). The largest aspect ratio in the spiral mesh is 14.83. The spiral mesh has no acute angles in the input polygon. We would like to point out that although there are some quadrilaterals with large aspect ratios, they are few in comparison with the total number of quadrilaterals in the mesh.

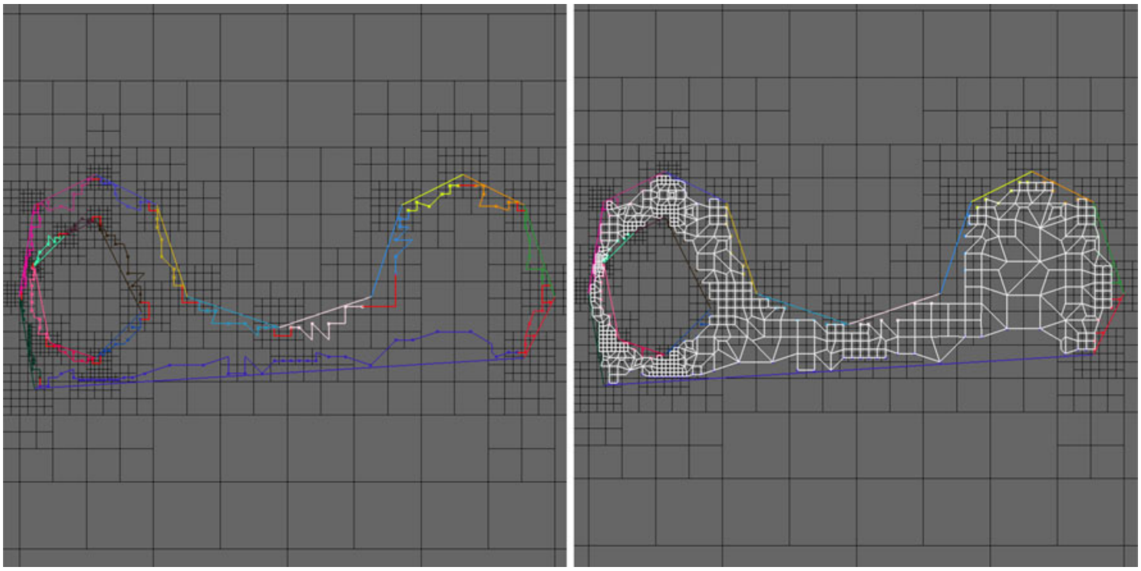


Fig. 48 *Left* Quadrangulation chains and corner chains. *Right* Partial mesh of polygon interior prior to quadrangulating edge or corner regions

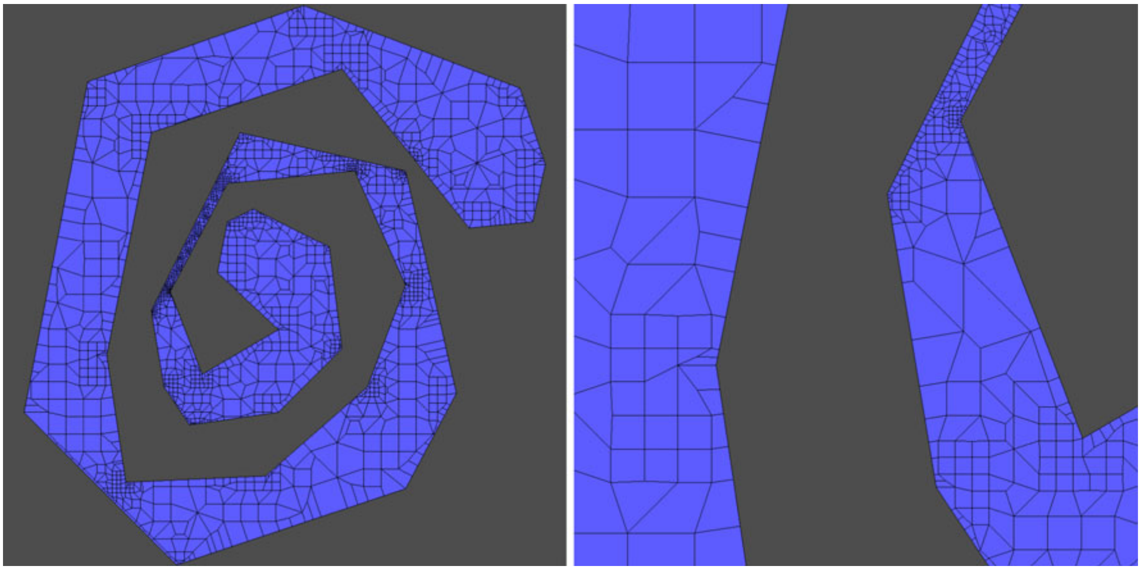


Fig. 49 Spiral polygon with 33 edges. Minimum mesh angle: 20.67°. Number of quadtree cells: 2,623. Number of mesh vertices: 2,213. Number of mesh faces (quads): 1,859

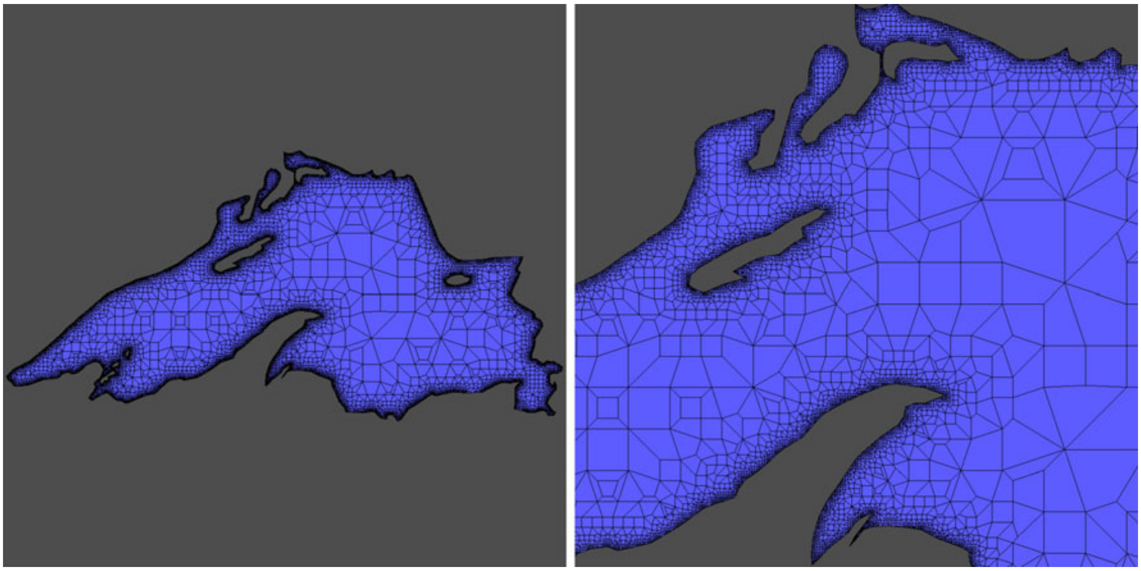


Fig. 50 Lake Superior polygon with 303 edges. Minimum mesh angle: 18.88°. Number of quadtree cells: 33,925. Number of mesh vertices: 27,026. Number of mesh faces (quads): 24,130

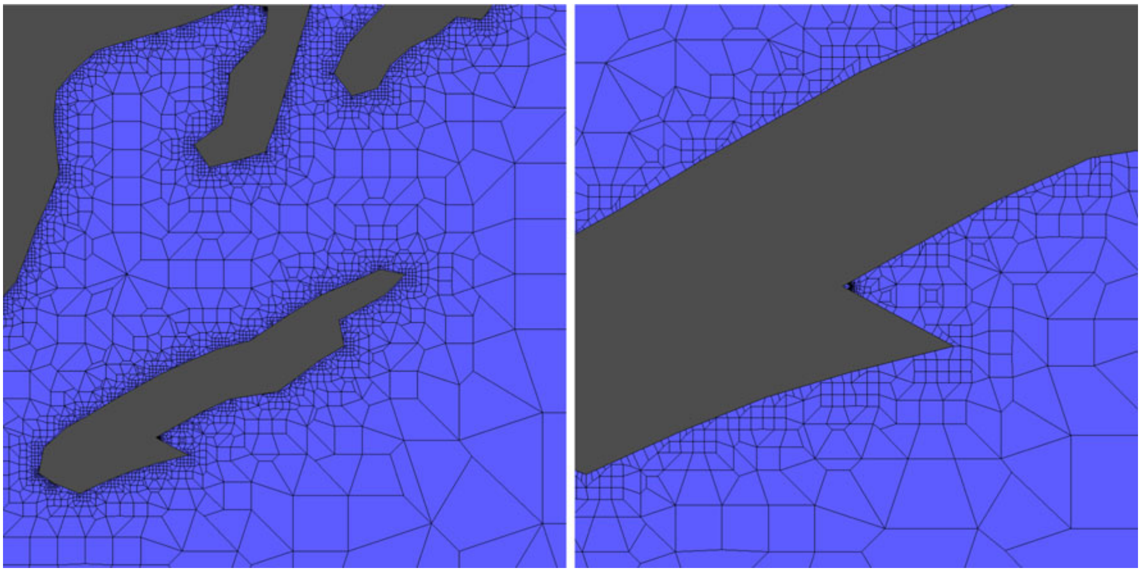


Fig. 51 Lake Superior polygon: zoomed in view

6 Conclusion

Sample meshes generated by our algorithm are shown in Figs. 46, 47, 48, 49, 50, 51, 52 and 53. Figures 46, 47 and 48 show the results of our algorithm on a non-acute polygon with a single hole. Figure 49 shows results on a spiral non-acute polygon without holes. Finally, Figs. 50, 51, 52 and 53 show results on the classic Lake Superior polygon, which is an acute polygon with two holes. Observe that in these examples the ratio of the number of quadrilaterals to the number of quadtree cells is less than one.

This paper presents the first known direct method to generate a quadrilateral mesh for the interior of a simple polygon (possibly with holes) in which every new angle in the mesh is guaranteed to be at least 18.43° and at most 171.86°. The main open question resulting from this work is its extension to polygon interior as well as exterior. While our algorithm itself is applicable to the interior or

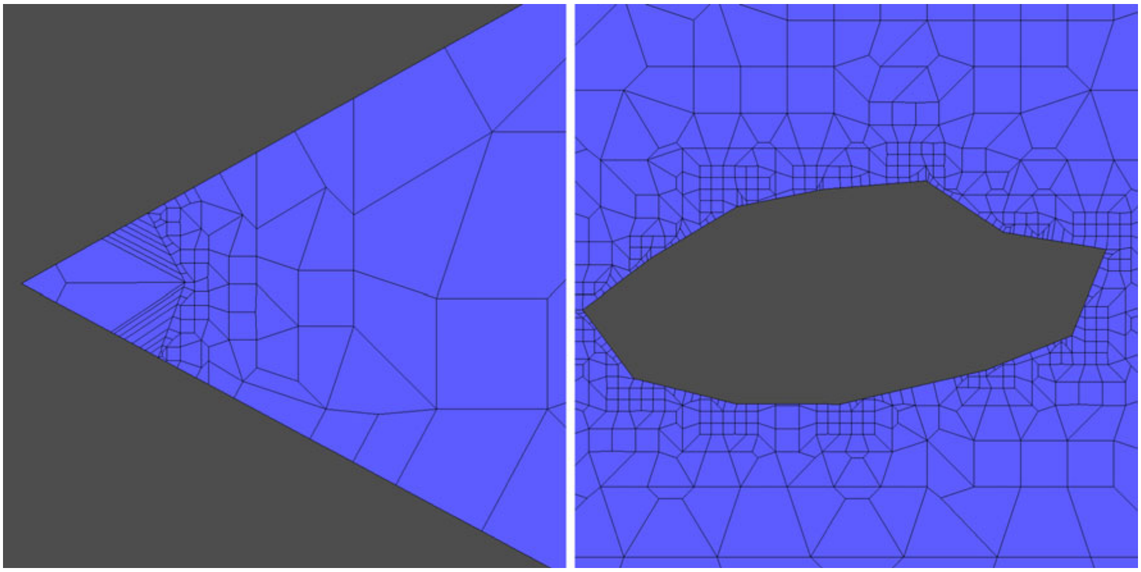


Fig. 52 Lake Superior polygon: zoomed in view

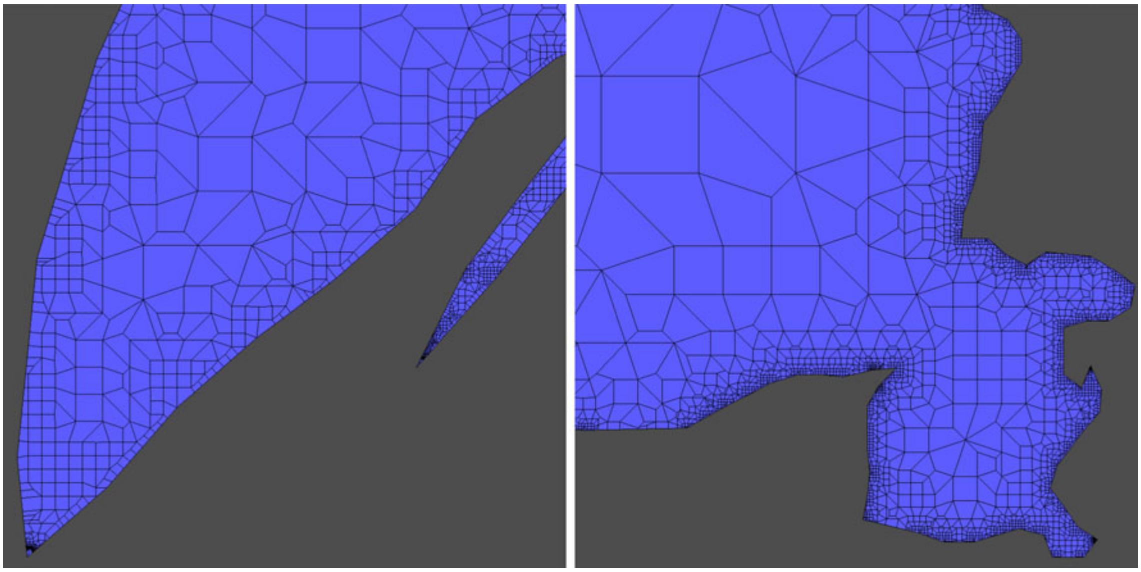


Fig. 53 Lake Superior polygon: zoomed in view

the exterior of the polygon, the difficulty of adapting it to both lies in resolving mesh compatibility at the boundary without propagating the changes throughout the mesh. We are currently investigating alternative strategies to mesh the region bounded by quadtree chains on both sides of each polygon edge.

Acknowledgments The authors would like to thank anonymous reviewers for helpful comments that served to significantly improve the presentation in the paper.

References

1. Allman DJ (1988) A quadrilateral finite element including vertex rotations for plane elasticity analysis. Int J Numer Methods Eng 26:717–730
2. Atalay FB, Ramaswami S, Xu D (2008) Quadrilateral meshes with bounded minimum angle. In: Proceedings of the 17th International Meshing Roundtable. Springer, Heidelberg, pp 73–91
3. Benzley S, Perry E, Merkley K, Clark B, Sjaardema G (1995) A comparison of all hexahedral and all tetrahedral finite element meshes for elastic and elastic-plastic analysis. In: Proceedings of the 4th international meshing roundtable, pp 179–192

4. Bern M, Eppstein D (1997) Quadrilateral meshing by circle packing. In: 6th IMR, pp 7–19
5. Bern M, Eppstein D, Gilbert J (1994) Provably good mesh generation. J Comp Sys Sci 48:384–409
6. Blacker T, Stephenson M (1991) Paving: a new approach to automated quadrilateral mesh generation. Int J Numer Methods Eng 32(4):811–847
7. Brauer, JR (eds) (1993) What every engineer should know about finite element analysis, 2nd edition. Marcel-Dekker, NY
8. Cheng S-W, Dey T, Ramos E, Ray T (2004) Quality meshing for polyhedra with small angles. In: Proceedings 20th annual symposium on computational geometry
9. Chew LP (1993) Guaranteed-quality mesh generation for curved surfaces. In: Proceedings of the 9th ACM symposium on computational geometry, pp 274–280
10. Chew LP (1997) Guaranteed-quality delaunay meshing in 3d. In: Proceedings of the 13th ACM symposium on computational geometry, pp 391–393
11. Edelsbrunner H (1987) Algorithms in combinatorial geometry. Springer, Berlin
12. Everett H, Lenhart W, Overmars M, Shermer T, Urrutia J (1992) Strictly convex quadrilaterilizations of polygons. In: Proceedings 4th Canadian conference on computational geometry, pp 77–82
13. Hine S, Atalay FB, Xu D, Ramaswami S (2009) Video: quadrilateral meshes with bounded minimum angle. In: Proceedings 25th ACM symposium on computational geometry (SoCG 2009), pp 90–91
14. Robinson J (1987) Cre method of element testing and the jacobian shape parameters. Eng Comput 4:113–118
15. Johnston BP, Sullivan JM, Kwasnik A (1991) Automatic conversion of triangular finite meshes to quadrilateral elements. Int J Numer Methods Eng 31(1):67–84
16. Knupp Patrick M (2000) Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantities. Part i—a framework for surface mesh optimization. Int J Numer Methods Eng 48:401–420
17. Miller GL, Talmor D, Teng S-H, Walkington N (1995) A delaunay based numerical method for three dimensions: Generation, formulation, and partition. In: Proceedings of the 27th ACM symposium on the theory of computing, pp 683–692
18. Mitchell S, Vavasis S (1992) Quality mesh generation in three dimensions. In: Proceedings of the 8th annual symposium on computational geometry, pp 212–221
19. Ramaswami S, Siqueira M, Sundaram T, Gallier J, Gee J (2005) Constrained quadrilateral meshes of bounded size. Int J Comput Geom Appl 15(1):55–98 (invited to special issue of selected papers from the 12th IMR)
20. Shewchuk J (2002) Constrained Delaunay tetrahedralizations and provably good boundary recovery. In: Proceedings of the 11th international meshing roundtable, pp 193–204
21. Shewchuk JR (1996) Triangle: engineering a 2D quality mesh generator and Delaunay triangulator. In: Applied computational geometry: towards geometric engineering. LNCS, vol 1148
22. Zienkiewicz OC, Taylor RL (1989) The finite element method. McGraw-Hill, NY