

On Visibility & Empty-Region Graphs

Sagi Katz & Ayellet Tal

Technion – Israel Institute of Technology

Abstract

Empty-Region graphs are well-studied in Computer Graphics, Geometric Modeling, Computational Geometry, as well as in Robotics and Computer Vision. The vertices of these graphs are points in space, and two vertices are connected by an arc if there exists an empty region of a certain shape and size between them. In most of the graphs discussed in the literature, the empty region is assumed to be a circle or the union/intersection of circles. In this paper we propose a new type of empty-region graphs—the γ -visibility graph. This graph can accommodate a variety of shapes of empty regions and may be defined in any dimension. Interestingly, we will show that commonly-used shapes are a special case of our graph. In this sense, our graph generalizes some empty-region graphs. Though this paper is mostly theoretical, it may have practical implication—the numerous applications that make use of empty-region graphs would be able to select the best shape that suits the problem at hand.

1. Introduction

In this paper we re-visit empty-region graphs. These graphs are aimed at structural analysis of point sets [1]. Intuitively, a vertex of an empty-region graph represents a point in space and an arc connects two vertices if there exists an empty region of a certain shape and size between their respective points. These graphs have applications in computer vision [2, 3, 4], machine learning [5, 6], computer graphics [7], pattern classification [8], geographic analysis [13], as well as in networking [9] and in Bioinformatics [10].

We establish a novel link between two concepts in computer graphics: visibility of point clouds and empty-region graphs. This is done by defining a new empty-region graph, the γ -visibility graph, which connects the two. Differently from previous works, our graph accommodates a variety of shapes of empty regions. Therefore, though this paper is theoretical, it may find various uses in graphics and in robotics, as specific shapes of empty-regions may better suit specific problems. We further prove that our graph generalizes some commonly-used empty-region graphs.

We start by a short description of visibility of point sets. Given a point set, considered to be a sample of a continuous surface, and a viewpoint, the goal is to determine the sub-set of visible points. More precisely, since points cannot occlude each other, we are basically seeking a sub-set that would be visible to the viewpoint, if the surface from which the set of points was sampled, was known. The traditional way to perform the task is to reconstruct the surface from which the points are sampled and then determine visibility on the reconstructed surface.

However, in [11] an operator was introduced that determines visibility directly on the set, skipping reconstruction. The operator performs two steps: In Step 1, a function maps every point in the set to an inverted domain. In Step 2, the convex hull of the transformed points and the viewpoint is calculated.

Points that reside on the convex hull of Step 2 turn out to be the pre-images of the visible points. In [12] the properties that should be satisfied by the function in Step 1 were identified and the operator was accordingly generalized to any function that satisfied these properties. This operator is termed the *Generalized Hidden Point Removal (GHPR) operator*.

We introduce in this paper a new graph structure, the γ -visibility graph. In this graph, two vertices are connected by an arc only if they are found to be visible to one another by the GHPR operator. This graph turns out to be an empty-region graph, as defined by [13]. Intuitively, this is so since indirectly, the GHPR operator “thresholds” the size of the empty regions between the viewpoint and the visible points. The shape of this region depends on the function applied in Step 1 and its size depends on the parameter γ of that function.

In contrast to most of the empty-region graphs proposed in the literature, in our case, the shape (template) region is not necessarily a union of circles or their intersection. Rather, it may take various shapes, which are determined by the function used in Step 1.

Our proposed γ -visibility graph has a couple of benefits. First, it generalizes empty-region graphs. The ability to define different shapes of empty regions makes it possible to match a specific shape to a specific application. For instance, a robot moving forward may not necessarily care about a circular empty region, but rather about a non-symmetric shape that emanates from the camera. Conversely, applications in communication may prefer circles. Second, we show that the Delaunay Triangulation is a special case of our γ -visibility graph.

The contributions of this paper are hence two-fold. First, we introduce a new and general graph structure, the γ -visibility graph and show how it provides a link between the class of visibility graphs to the class of empty-region graphs (Section 3). Second, we prove that the Delaunay triangulation is a special case of this graph (Section 4). This may have a couple of in-

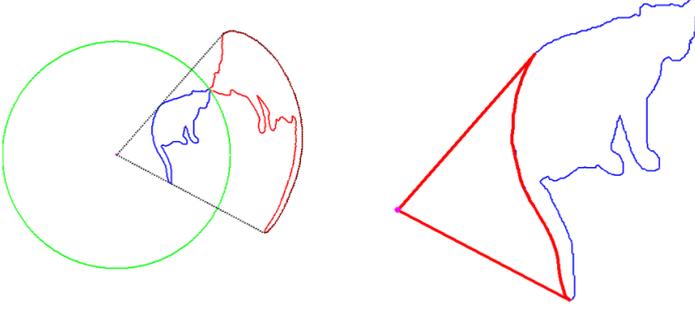


Figure 1: **The GHPR Operator.** Left: Transformation (in red) of a set of 2D points (in blue), with a viewpoint in magenta, using the f_{mirror} kernel. Right: back projection of the convex hull, which is the set of visible points.

interesting implications. On one hand, the relation between the structures leads to a novel algorithm for computing the Delaunay triangulation—one of the most useful geometric structures (Section 5). On the other hand, the Delaunay triangulation can be utilized to efficiently construct an approximation of the γ -visibility graph (Section 6).

2. Background

2.1. Visibility of point sets

Given a point cloud P and a viewpoint C , the goal is to determine the subset of P that would be visible to C , if the surface from which P was sampled was known. Since points cannot occlude each other (unless they accidentally fall on the same line from the viewpoint), the traditional way to solve the problem is to reconstruct the surface [14, 15] and determine visibility on the reconstructed surface.

In [11] an elegant operator (*HPR*) is proposed to determine visibility directly from the point set, without surface reconstruction or normal estimation. This operator was later generalized in [12] to the *GHPR* operator. Briefly, the operator consists of two steps:

1. **Point transformation:** A function maps every point $p_i \in P$ to an inverted domain \hat{P} . Assuming, without loss of generality, that the viewpoint is at the origin, the transformation is defined as

$$\hat{p}_i = F_f(p_i) = \begin{cases} \frac{p_i}{\|p_i\|} f(\|p_i\|), & p_i \neq 0 \\ 0 & p_i = 0 \end{cases} . \quad (1)$$

The kernel function $f(d)$ is a 1-dimensional continuous kernel function $f: \mathbb{R}^+ \rightarrow \mathbb{R}^+$ that, given the distance d of p_i from C , outputs an updated distance. f should satisfy three properties discussed in [12]: $f'(d) < 0$, $f(d) > 0$, and a condition regarding the parameter γ of the kernel.

2. **Convex hull construction:** The convex hull of the transformed points and the viewpoint is calculated.

The main result of [11, 12] is that points that reside on the boundary of the convex hull of Step 2 are the pre-images of the visible points, as illustrated in Figure 1. (From now on we will refer to the boundary of the convex hull as the convex hull.)

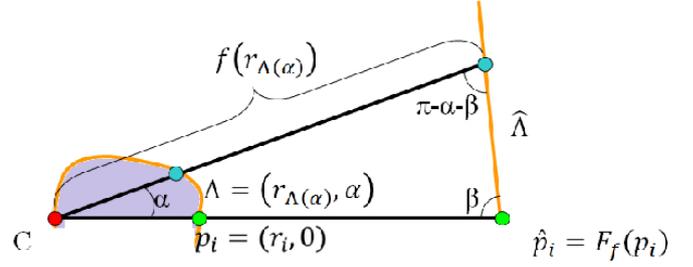


Figure 2: **Transformation setup & Empty region.** The curve Λ is transformed to the line $\hat{\Lambda}$ by F_f , where $f(d) = \gamma - d$. The empty region associated with p_i and bounded by Λ is in purple. The larger this area, the smaller β .

The underlying idea of this operator is that when the transformed points in \hat{P} reside on the convex hull, their pre-images in P are associated with large empty regions between them and the viewpoint. Thus, there is no need to find for each point the neighboring points that maximize the empty region size. This observation is important computationally and is the reason why the GHPR operator is so efficient. Without it, for every point, we would have to find the closest point(s), making the algorithm quadratic (and in 3D even cubic). Instead, all that needs to be done is to compute the convex hull and consider a point visible if its image is on the convex hull of the set of transformed points. Thus, the GHPR operator defines both the shape and the size of the empty region. The size of the empty region is adaptively determined by a point's neighbors.

Formally, given a kernel f and a point p_i , the Λ -curve is defined as a parametric curve that is the pre-image under F_f of a line passing through a transformed point $\hat{p}_i = F_f(p_i)$, as illustrated in Figure 2. In this figure, which uses polar coordinates, the line $\hat{\Lambda}$ creates an angle β with the x-axis. Any point on the Λ -curve, having a polar angle α , is transformed to a point on the line $\hat{\Lambda}$ with distance $f(r_{\Lambda(\alpha)})$ from C . If \hat{p}_i is on the convex hull, i.e. all the points in \hat{P} fall on one side of it, then its associated Λ -curve is empty of points from P . The larger the empty region, the smaller β . Therefore, the convex hull indirectly thresholds the size of the empty region, since for a point to be on the convex hull, its associated angle β should be $< \Pi$.

The kernel function f has a single parameter, γ . It controls the desirable size of the empty region, whereas f controls the shape of this region. Examples of kernel functions are $f_{inversion}(d) = d^\gamma$, $f_{mirror}(d) = \gamma - d$, and $f_{natural} = e^{-d\gamma}$.

Visibility graphs are graphs with nodes for each object, and arcs between objects that can see one another [16]. When the objects are points, the *point visibility graph* for a given point set P is defined as a graph G in which there is an undirected edge between two points (vertices) that are mutually visible to each other [17]. One can note that unless points are collinear, they cannot block each other. Therefore, this definition is strict and has only limited use in real-world problems where zero-dimensional points are rarely collinear. Some work has been done on proving theoretical properties for the point visibility graphs [18, 19]. Some variants of the problem were also explored, such as *point visibility graph within a simple polygon* [20,

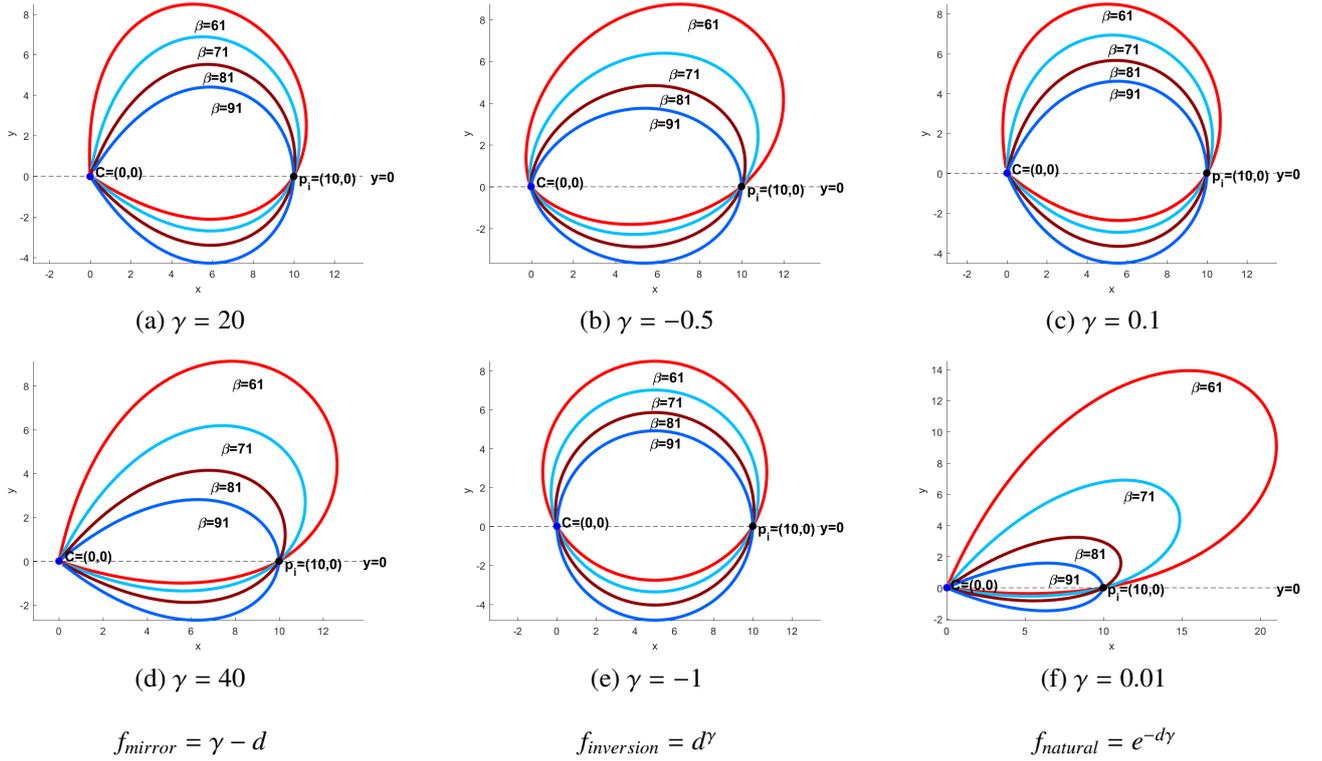


Figure 3: **Empty regions.** The empty regions (bounded by Λ -curves) for different kernels and parameter values, for a single point $p_i = (10, 0)$ and $C = (0, 0)$. Each graph contains several Λ -curves, each corresponds to a different value of β , the angle between the x-axis and the line that is the image of the Λ -curve (see Figure 2). It can be seen that the size of the empty region increases as β gets smaller.

21].

In this paper we introduce a new class of graphs—the γ -visibility graph, which is induced from the GHPR operator.

2.2. Empty-region graphs

An empty-region graph is defined in \mathbb{R}^2 as follows [13]:

Definition 2.1. An Empty-Region Graph of a point set P , parameterized by a template region Ω , $ERG_\Omega(P) = (V, E)$, is a graph where $V = P$ and $\forall p_i, p_j \in V, p_i p_j \in E \Leftrightarrow \Omega(p_i, p_j) \cap P \setminus \{p_i, p_j\} = \emptyset$.

If the template region Ω is not symmetric, the graph is directed, and it is undirected otherwise.

Below, we briefly review some well-studied empty-region graphs. In [22], Toussaint introduces the *relative neighborhood graph* for which an arc exists between two points if they are close to each other at least as they are close to any other point. In a Gabriel graph [23] an arc exists between two points p and q , if there exists an empty ball with radius $\frac{d(p,q)}{2}$ that passes through both p and q . The Gabriel graph is contained in the Delaunay triangulation and contains the relative neighborhood graph.

In [1], Edelsbrunner et al. introduce the α -shapes. An edge of the alpha-shape is drawn between two points whenever there exists a generalized disk of radius $1/\alpha$ containing the entire point set and which has the property that the two points lie on its boundary.

Kirkpatrick and Radke introduce the β -skeleton, which is an undirected graph for which an edge exists between two points of P if and only if the β -neighborhood between the points is empty [24]. The β -neighborhood is the intersection or the union of disks. The β -neighborhood is generalized in [25].

Amenta et al. prove that for a specific value of β , the β -skeleton forms a polygonal reconstruction of an r -sampled smooth curve [26]. They also introduce the *Crust*, a subgraph of the Delaunay triangulation, which forms a reconstruction. The crust is an empty-region graph, where an edge between two points belongs to the Crust, if there exists a disk, empty of points, touching the two points. Finally, the θ -graph is a directed empty-region graph for which there exists an arc between p and q , if q is the nearest vertex to p within the cone it lies in [27].

3. Definition of the γ -visibility graph

In this section, we define a new type of graphs, the γ -visibility graph, which is defined over a set of points $P \in \mathbb{R}^n$. We show that this graph is an empty-region graph. However, unlike the graphs reviewed in Section 2.2, its template region may be of various shapes. This is demonstrated in Figure 3, where a variety of empty region templates are drawn.

Formally, given a set of points $P \in \mathbb{R}^n$, we define two types of γ -visibility graphs—a directed and an undirected graph, as follows. Let $GHPR(P, C)$ be the subset of points of P that are detected visible by the GHPR operator with the viewpoint at C .

Definition 3.1. A directed γ -visibility graph: $G(V, E)$ is a directed γ -visibility graph if its set of vertices is $V = P$ and, given two nodes $p_i, p_j \in P$, there exists an edge from p_i to p_j in E if $p_j \in \text{GHPR}(P, p_i)$, i.e. p_j is detected visible by the GHPR operator with a viewpoint at p_i .

According to [12], p_j is detected as visible to p_i (and thus, this edge exists), when there is an empty region, Ω_{ij} , associated with p_j for the viewpoint $C = p_i$ and that this empty region is sufficiently large (as defined in [12]). This is equivalent to the empty region in Definition 2.1.

The undirected version of the γ -visibility graph is defined as follows.

Definition 3.2. An undirected γ -visibility graph: $G(V, E)$ is an undirected γ -visibility graph if its set of vertices is $V = P$ and, given two nodes $p_i, p_j \in P$, there exists an edge $p_i p_j$ in E if $p_j \in \text{GHPR}(P, p_i)$ and $p_i \in \text{GHPR}(P, p_j)$.

Note that γ , which is the parameter for the GHPR operator, becomes a parameter for the γ -visibility graph.

Construction of the γ -visibility graph. To construct the directed γ -visibility graph, we first set $V = P$ and then run the GHPR operator for each point as a viewpoint. The result gives us the edges emanating from each of the points. The accumulation of the edges is the set of edges E of the graph we seek-after.

The complexity of the algorithm is $O(n \times \text{time}(\text{GHPR}))$, where $\text{time}(\text{GHPR})$ is the time to execute the GHPR operator. The latter is the time of computing the convex hull, which is $O(n \log n)$ in 2D and in 3D. Therefore, the algorithm costs $O(n^2 \log n)$ in 2D and in 3D.

4. γ -Visibility graphs & Delaunay triangulations

This section makes a couple of contributions. First, we prove the relation between the GHPR operator and the Delaunay triangulation / Voronoi diagram. Second, using this relation, we prove that the Delaunay triangulation is a special case of the γ -Visibility Graph. In the subsequent section we will use this fact to show that the visible shape, which is induced by GHPR operator (and will be formally defined below), in three dimensions forms a Delaunay triangulation in two dimensions.

While the γ -visibility graph can be constructed using any valid GHPR kernel f , we focus on the inversion kernel, due to its unique properties, which are described hereafter.

The inversion kernel: This kernel is defined, for a parameter $\gamma < 0$, as

$$f_{\text{inversion}}(d) = d^\gamma.$$

Recall that this kernel is applied within Equation 1. Figures 3(b),-(e) show the kernel's empty regions for different γ parameter values for a single point.

We start by proving below that for the inversion kernel with $\gamma = -1$, the Λ -curves are circles in \mathbb{R}^2 (spheres in \mathbb{R}^3). We will then use this lemma to prove the relation of our γ -visibility graph to the Delaunay Triangulation.

Lemma 4.1. The Λ -curves for the inversion kernel with parameter $\gamma = -1$ are spheres that pass through the viewpoint C .

Proof: Without loss of generality, suppose that C is located at the origin of the coordinate system. We begin by examining a given line in \mathbb{R}^2 , between two points \hat{p}_i, \hat{p}_j of the transformed set of points \hat{P} (after applying Equation 1). This line can be represented either in Cartesian coordinates as $(x, ax + b)$, or in polar coordinates as $(r(\theta), \theta) = (\frac{b}{\sin(\theta) - a \cos(\theta)}, \theta)$.

We would like to find the Λ -curve transformed to this line by F_f . For this, we need to apply an inverse transformation on the line. It is shown in [12] that the inverse transformation F_f^{-1} uses an inverse kernel, which in this case is $f^{-1}(d) = d^{\frac{1}{\gamma}} = d^{-1} = f(d)$. Applying this inverse kernel on the above line results with

$$(r(\theta), \theta) = \left(\frac{\sin(\theta) - a \cos(\theta)}{b}, \theta \right), \quad (2)$$

and in Cartesian coordinates:

$$(x, y) = \left(\frac{\sin(\theta) \cos(\theta) - a \cos^2(\theta)}{b}, \frac{\sin^2(\theta) - a \sin(\theta) \cos(\theta)}{b} \right).$$

This curve is a circle with radius

$$R = \frac{\sqrt{1 + a^2}}{2|b|}$$

centered at

$$(c_x, c_y) = \left(\frac{-a}{2b}, \frac{1}{2b} \right).$$

It remains to show that the circle passes through the origin. This can be shown by using Equation 2 and solving the following equation for θ :

$$r(\theta) = \frac{\sin(\theta) - a \cos(\theta)}{b} = 0.$$

The solution to this equation is $\theta = \arctan(a)$, for any given a and $b \neq 0$. $b = 0$ represents the case where $C = p_i$. In this case the circle passes through C since the line $(x, ax + b)$ becomes (x, ax) and passes through the origin, which is C . Thus, the radius of the circle is infinite and the circle is in fact a line connecting $C = p_i$ with p_j . \square

Figure 4(a) shows an example of the Λ -curves for $\gamma = -1$. For the rest of this paper, we will denote the Λ -curves for the case of $\gamma = -1$ as the Λ -circles.

The proof is very similar for \mathbb{R}^3 , where $ax + by + cz + d = 0$ is the plane that passes through 3 points (instead of the line in \mathbb{R}^2) and is transformed to a sphere by F_f^{-1} .

Thus far, we have shown that empty regions associated with pairs of point from P are shaped as circles. We now prove that the centers of these circles are vertices of the Voronoi Diagram of $P \cup C$ (Lemma 4.2) and show how they are related to the Delaunay triangulation (Lemmas 4.3–4.4).

Lemma 4.2. Given $P \in \mathbb{R}^2$ and C (where C may belong to P), the centers of the Λ -circles are the Voronoi vertices of a single Voronoi cell, the one that contains C , of the Voronoi diagram of $P \cup \{C\}$.

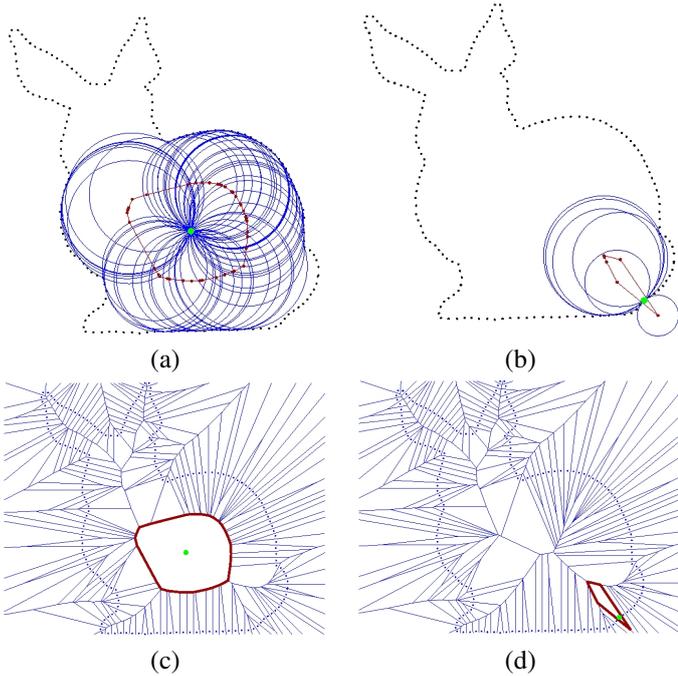


Figure 4: **Voronoi cell.** GHPR with $\gamma = -1$ calculates the Voronoi cell (in red) for C (in green). Each blue circle is centered at a Voronoi vertex. In the left column $C \notin P$, whereas in the right column, $C \in P$. The Voronoi cell is made up of centers of the Λ -circles (a)-(b). Points for which there are no circles (e.g., the ears) are invisible from C . For intuition, (c)-(d) show the full Voronoi diagram. It can be seen that the red region is indeed one cell of the full diagram.

Proof: According to Lemma 3.5 in [12], the regions Ω that are encapsulated in the Λ -curves are guaranteed to be empty (as they are back projections of edges of the convex hull). Moreover, the size of the empty region found by the GHPR operator and constrained by the shape of the Λ -curves, is maximized. Utilizing Theorem 7.4 from [28], these properties indicate that the centers of the Λ -circles are vertices of the Voronoi Diagram of $P \cup \{C\}$.

To explain this, according to the above theorem, a point is a vertex of the Voronoi diagram if and only if the largest empty circle that is centered at this point contains three or more sites on its boundary. In our case, it is shown in Lemma 3.3 in [12] that the Λ -curves that bound the largest empty region must pass through at least two points from P . In addition, we have shown that every Λ -curve must pass through C (Lemma 4.1).

These centers of circles form all the vertices of the Voronoi cell since C participates in all the Λ -circles. Therefore, connecting the centers of the circles forms the Voronoi cell of C . \square

Figure 4(a)-(b) shows (in red) the Voronoi cell, calculated for point C , located inside a bunny-shaped point set. The blue circles are the Λ -circles. Note that this lemma can be extended to higher dimensions.

In a dual manner, we show the relation of the GHPR operator to the Delaunay triangulation. Before doing that, we make a couple of definitions and illustrate them in Figure 5.

Definition 4.1. *Visible hull:* The visible hull is the back projection of the convex hull of $\hat{P} \cup C$.

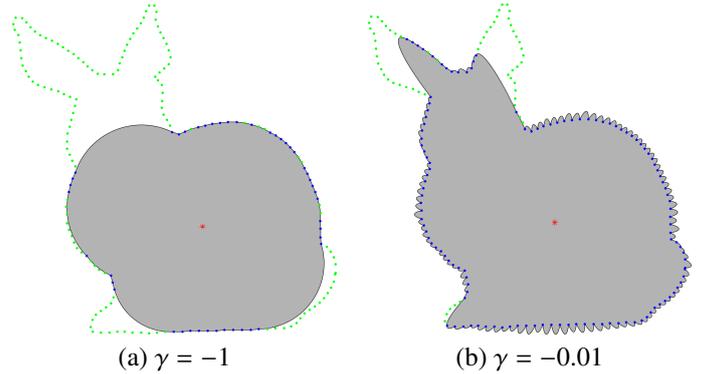


Figure 5: **Visible hull.** The visible hull that connects the blue points, bounds the gray region. The viewpoint is in red, the blue points on the boundary are visible, whereas the green points are not. Here, the inversion kernel is used.

Note that the visible hull is a union of Λ -curves, as defined by the GHPR operator. This is so since every edge of the convex hull is back projected to a Λ -curve, bounding an empty region, as defined in Figure 2.

Definition 4.2. *Visible shape:* The visible shape is a piecewise linear approximation of the visible hull, where points of P are connected by straight lines instead of by arcs.

Lemma 4.3. Given $P \in \mathbb{R}^2$ and C , every pair of points of P that are detected visible by the GHPR operator ($\gamma = -1$), whose transformed points are neighbors on the convex hull of $\hat{P} \cup \{C\}$, together with the viewpoint C , form a Delaunay triangle of the Delaunay triangulation of $P \cup \{C\}$.

Proof: According to Theorem 9.6 from [28], three points $p_i, p_k, p_l \in P$ are vertices of a face of the Delaunay graph of P if and only if the circle through the points contain no point of P in its interior. In our case, the Λ -curves form circles that pass through C and at least two points from P . Moreover, since the points are marked visible, the circles are empty. Therefore, the viewpoint C together with the two points from P form a Delaunay triangle. \square

The above lemma suggests that connecting the vertices of the visible shape to the viewpoint C forms a subset of the Delaunay triangulation of $P \cup \{C\}$. The following lemma proves that this set of triangles is the maximal subset of Delaunay triangles that include C as a vertex.

Lemma 4.4. The set of Delaunay triangles created by connecting the visible shape with C is the set of all the Delaunay triangles that contain C as a vertex.

Proof: Suppose, by contradiction, that there is an additional Delaunay triangle that contains C and does not belong to the GHPR triangles. Suppose, without loss of generality, that this triangle contains the points p_i, p_j and C . If p_i is inside the visible shape, it contradicts the fact that the visible shape is free of points from P . This is so since the visible shape is contained within the visible hull (as the arc is convex relatively to the viewpoint and is replaced by a straight edge) and the latter

is provenly empty of points. If p_i is outside the visible shape, then the edge connecting p_i with C must cross an edge of the visible hull, which contradicts the planarity of the Delaunay triangulation (Figure 6). \square

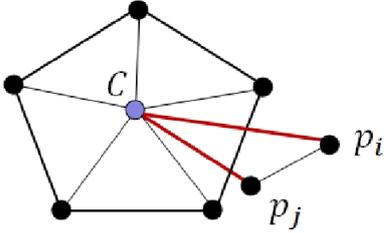


Figure 6: Connecting the visible shape (bold black lines) with C is the set of all Delaunay triangles that contain C as a vertex. If we add an additional triangle (red lines), it will cross the visible shape, which contradicts the planarity of the Delaunay Triangulation.

Figure 7 demonstrates the calculated Delaunay triangles, as described above. In particular, Figure 7(a) shows an example where $C \notin P$, whereas Figure 7(b) shows an example where $C \in P$.

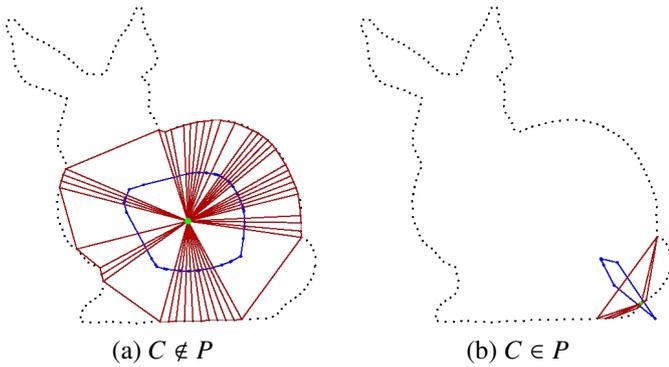


Figure 7: **Local Delaunay triangulation.** The visible set, as detected by the GHPR operator with $\gamma = -1$, can be used to calculate a local Delaunay triangulation of $P \cup \{C\}$. The Delaunay triangles that include C are marked in red, the Voronoi cell in blue and C in green.

5. Computing Delaunay triangulations by applying GHPR

After establishing the relation between the GHPR operator and the Delaunay triangulation, the natural question to ask is whether the GHPR operator can be a basis for a new algorithm for constructing the Delaunay triangulation of a set of points.

A naive algorithm may apply the GHPR operator once for each of the points of P (see Figure 8). One should note that it is possible to calculate the Delaunay triangulation and not just the Delaunay edges due to the fact that the visible shape, calculated by the GHPR operator, also radially sorts the neighboring points around a viewpoint.

However, this algorithm is not an efficient way of calculating the Delaunay triangulation, as applying the GHPR operator n times costs $O(n^2 \log n)$, whereas this task can be performed in $O(n \log n)$ (in 2D and 3D). This raises the question of whether

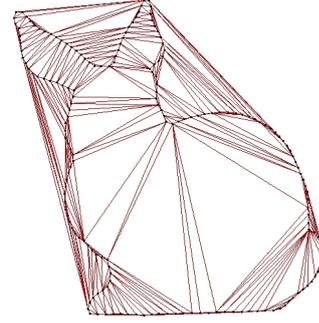


Figure 8: **Full Delaunay triangulation.** The full Delaunay triangulation is calculated in a naive manner, by applying the GHPR operator n times, where n is the number of points in P .

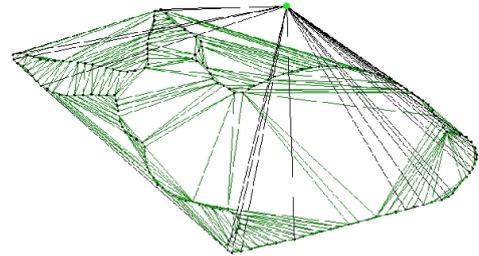


Figure 9: **Efficient computation of Delaunay triangulation.** A 2D Delaunay triangulation is calculated by applying the GHPR operator with \hat{C} positioned in \mathbb{R}^3 . Green and black edges represent the visible shape. Edges that do not include \hat{C} (in green) form the Delaunay triangulation.

the Delaunay triangulation can be efficiently computed using the GHPR operator.

If we could run the GHPR only once, the time complexity would be $O(n \log n)$, which is optimal. However, in order to be able to do it, two requirements need to be satisfied:

1. The GHPR operator with $\gamma = -1$ should be used, so that empty Λ -circles are created.
2. All the points in P should be visible from C , so that the visible shape will include all the points.

Can we locate C so that all the points in P are visible to it? The key idea is as follows. We define C in one dimension higher than the points. For example, when P is in \mathbb{R}^2 , any viewpoint $C \in \mathbb{R}^3$ with $z \neq 0$, will see all the points in P , so that all the points will belong to the visible shape, as required.

For instance, in Figure 4(a) only some of the points are visible to the viewpoints C . We now raise C above its 2D location to reside in \mathbb{R}^3 and denote this point \hat{C} . Then, the points would not be able to occlude one another and all the points would be marked visible to \hat{C} by the GHPR operator. In this case, the empty Λ -circles become empty Λ -spheres. The visible shape, excluding the triangles adjacent to \hat{C} , is the Delaunay triangulation, as illustrated in Figure 9.

Algorithm 1 summarizes the method described above. The subsequent lemmas prove the correctness of this algorithm. We begin by extending Lemma 4.3 to 3D and showing that connecting the 3D visible shape to \hat{C} results in 3D Delaunay tetrahedra. Then, we prove that the 3D Delaunay tetrahedra facets that do

not contain \hat{C} are in fact the 2D Delaunay triangulation we seek after.

Algorithm 1 Constructing the 2D Delaunay triangulation of P using the GHPR operator in 3D

Input: A point set $P \in \mathbb{R}^2$

Output: An edge set E_{DT} of the Delaunay triangulation

Algorithm:

$Q \leftarrow \{\emptyset\}$

$E_{DT} \leftarrow \{\emptyset\}$

$\hat{C} \leftarrow (c_x, c_y, c_z)$ where $c_z \neq 0$

for all $p_i = (p_{i,x}, p_{i,y}) \in P$ **do**

$Q \leftarrow Q \cup \{q_i = (p_{i,x}, p_{i,y}, 0)\}$

end for

// Applying the GHPR operator:

$\hat{Q} \leftarrow F_f(Q, \hat{C})$ with $\gamma = -1$

$CH \leftarrow \text{ConvexHull}(\hat{Q} \cup \hat{C})$

for all $\bar{i}\bar{j}$ s.t. $\bar{i}\bar{j} \in CH$, $q_i, q_j \neq \hat{C}$ **do**

$E_{DT} \leftarrow E_{DT} \cup \{\bar{i}\bar{j}\}$

end for

Lemma 5.1. Given $P \in \mathbb{R}^3$ and $C \in \mathbb{R}^3$, every three points of P whose transformed points form a triangle on the convex hull of $\hat{P} \cup \{C\}$, together with the viewpoint C , form a Delaunay tetrahedron of $P \cup \{C\}$.

Proof: In order to belong to the 3D Delaunay triangulation, a tetrahedron must be created by four points of the set, whose circumsphere is empty. It was already shown in Lemma 4.1 that the 3-dimensional empty regions form spheres that pass through C . The spheres are transformed by F_f to facets of the convex hull of $\hat{P} \cup C$. Therefore, in addition to C , the empty spheres also pass through at least three other points from P . \square

Lemma 5.2. Let $P_2 \in \mathbb{R}^2$ be a set of points and let $P_3 \in \mathbb{R}^3$ be its corresponding set of points, such that

$$P_3 = \{\hat{p}_i = (x_i, y_i, 0) | p_i = (x_i, y_i) \in P_2\}.$$

Also, let $\hat{C} \in \mathbb{R}^3$ with $c_z \neq 0$ be a viewpoint. The triangles, connecting points of P_2 , that have corresponding triangles on the convex hull of $\hat{P}_3 \cup \hat{C}$, form the 2D Delaunay triangulation of P_2 .

Proof: Lemma 5.1 already showed that connecting \hat{C} with triangles of P_3 that have corresponding triangles on the convex hull of $\hat{P}_3 \cup \hat{C}$, form 3D Delaunay tetrahedra. Now, it remains to show that the triangles that do not include \hat{C} of these 3D tetrahedra, form the 2D Delaunay triangulation of P_2 . For this, suppose that three points $p_i, p_j, p_k \in P_3$ with the addition of the point \hat{C} define a Delaunay tetrahedron and therefore, an empty sphere passes through these 4 points. Now, define a plane through the points p_i, p_j, p_k . The intersection of this plane with the empty sphere forms an empty circle. Therefore, there is an empty circle passing through the corresponding points in P_2 and these corresponding three points form a 2D Delaunay triangle.

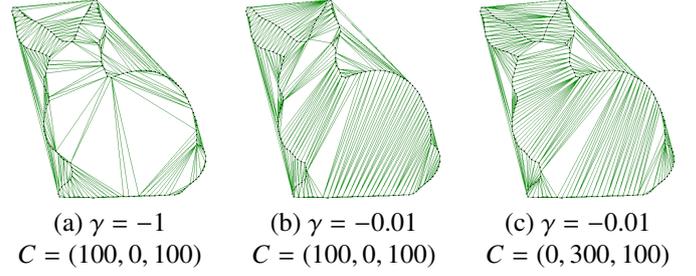


Figure 10: **Point set triangulations.** Using $\gamma = -1$ always produces a Delaunay triangulation. Using other values of γ produces triangulations that depend on the location of C .

Moreover, all the triangles of the 2D Delaunay triangulation of P_2 can be found in this way. This is so since this method covers the area bounded by the convex hull of P_2 with triangles. \square

It should be mentioned that one may produce a triangulation for a given point set using $\gamma \neq -1$. However, in this case the resulting triangulation will not be the Delaunay triangulation and the triangles created will depend on the location of C . In fact, as γ approaches 0, the triangles will become more and more “directed” toward the projection of the viewpoint, as show in Figure 10.

Note that although the connection between the Delaunay triangulation and the convex hull in a higher dimension is well-known [28] (using $z = x^2 + y^2$), the method described above (that uses a different transformation) gives it a new meaning in the context of visibility: The Delaunay triangulation can be thought of as being the visible shape, as seen from a viewpoint in a higher dimension.

As a final note, we observe that the Delaunay triangulation is not an empty-region graph in the traditional sense (as in [13]), because it does not rely on a single empty region uniquely defined by two points p and q . To connect the Delaunay triangulation back to γ -visibility graphs, notice that each line through a mapped neighbor q is inverse mapped back to a circle whose interior could be viewed as one of many potential empty regions that may serve as witnesses of the edge between p and q . In fact, this interior (a disk) defines an empty region, which through inversion is mapped to a half-space with linear boundary.

6. Approximation of the γ -visibility graph using Delaunay Triangulation

In the previous section we showed how to utilize the GHPR operator in order to construct the Delaunay triangulation. This section proceeds to discuss the relationship between the γ -visibility graph (Definitions 3.1, 3.2) and the Delaunay triangulation. In particular, we do the “reverse” of the previous section—we show how to use the Delaunay triangulation in order to approximate the γ -visibility graph.

While the γ -visibility graph generally requires $O(n^2 \log n)$ time to construct in 2D and 3D (Section 3), we observe that the Delaunay triangulation gives rise to an $O(n \log n)$ algorithm for

approximating the γ -visibility graph using the inversion kernel and $\gamma \leq -1$.

Before describing the algorithm, we first prove that the γ -visibility graph, for $\gamma = -1$, is the Delaunay triangulation.

Lemma 6.1. *The γ -visibility graph, defined with the inversion kernel and $\gamma = -1$, is the Delaunay triangulation.*

Proof: We need to show that an edge belongs to the Delaunay triangulation of P , $p_i p_j \in E_{DT}$, if and only if it is an edge of the γ -visibility graph, $p_i p_j \in E_{GV}$. According to Definition 3.1, the existence of the edge $p_i p_j$ in the γ -visibility graph means that $p_j \in GHPR(P, p_i)$. According to Lemma 4.3, this means that $p_i p_j \in E_{DT}$. Moreover, if the edge belongs to the Delaunay triangulation, $p_i p_j \in E_{DT}$, then according to Lemma 4.4 it also belongs to the γ -visibility graph $p_i p_j \in E_{GV}$. \square

Utilizing Lemma 6.1, the algorithm, which is depicted in Algorithm 2, begins by calculating the Delaunay triangulation of P , which requires $O(n \log n)$ time. Then, for every point $p_i \in P$, its neighboring points are checked for visibility by transforming them using the inversion kernel (when $C = p_i$) and checking if they are on the convex hull. This operation is essentially similar to the regular GHPR operation, however, all the points that do not share an edge with p_i are assumed to be invisible and therefore, are not taken into account (as they are internal to the convex hull).

The latter operation requires $O(E)$, where E is the number of edges in the Delaunay triangulation. Therefore, the entire construction requires $O(E + n \log n)$. Using the Euler characteristic (which applies to planar graphs), this amounts to $O(n \log n)$.

Algorithm 2 An $O(n \log n)$ algorithm for constructing an approximated γ -visibility graph ($\gamma \leq -1$)

Input: A point set P

Output: An edge map E_{GV} of the approximated γ -visibility graph

Algorithm:

$E_{DT} \leftarrow \text{ComputeDelaunay}(P)$

$E_{GV} \leftarrow E_{DT}$

for all $p_i \in P$ **do**

$C \leftarrow p_i$

$\tilde{P} \leftarrow \{p_j | \overline{p_i p_j} \in E_{DT}\}$

$P_v \leftarrow GHPR(\tilde{P}, p_i)$

for all $\{j \text{ s.t. } p_j \notin P_v\}$ **do**

$E_{GV} \leftarrow E_{GV} \setminus \{\overline{p_i p_j}\}$

end for

end for

Figure 11 shows an approximated γ -visibility graph, for a 2D point set, versus an exact γ -visibility graph. It can be seen that for $\gamma = -1$, both methods result with the Delaunay triangulation. For smaller values of γ , the results are very similar, although not identical.

7. Conclusion

The contribution of this paper is mostly theoretical. It established a link between two problems: visibility of point clouds

Approx. γ -visibility graph **Exact γ -visibility graph**

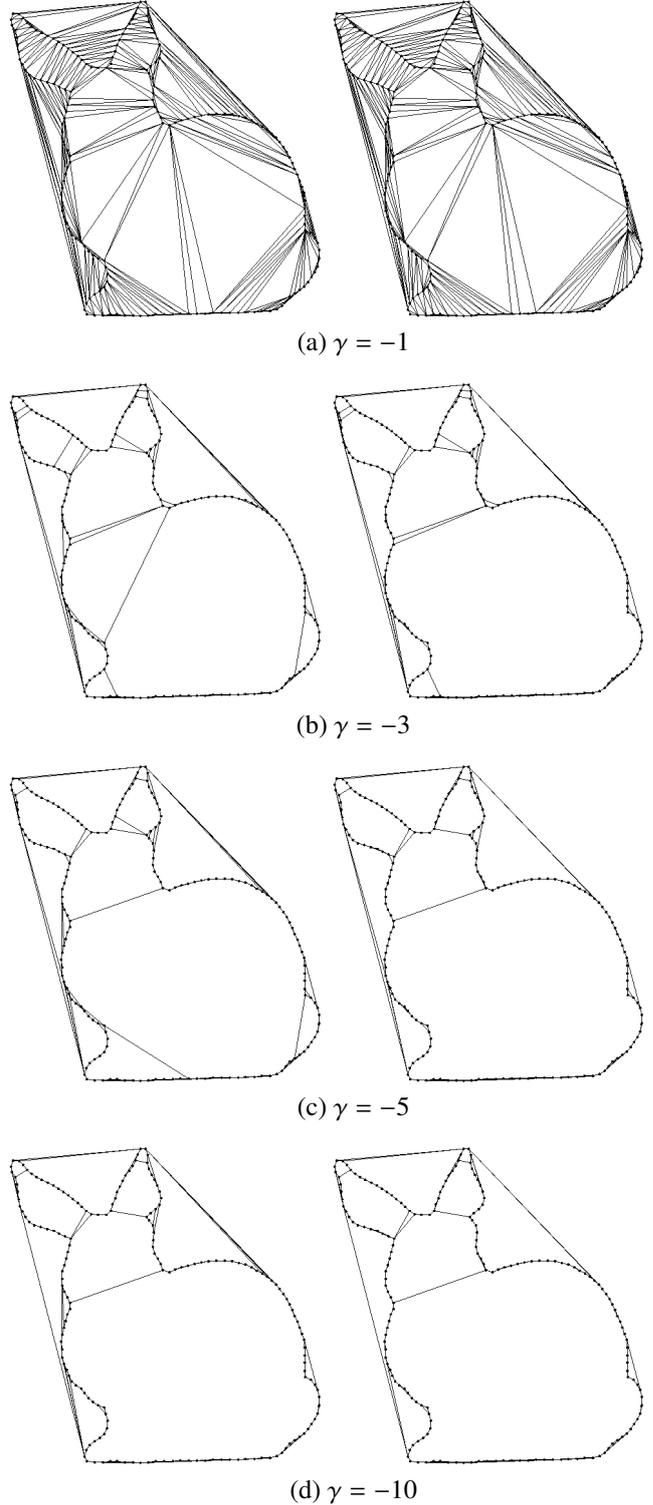


Figure 11: **The approximated γ -visibility graph.** This figure compares the approximation to the exact γ -visibility graph for various values of γ . The results show that the graphs are similar, though small changes exist. As γ decreases, we are left with fewer edges of the graph, where only the vertices that are most visible one to another are connected. It is interesting to note that the result of the approximation provides hints for segmenting an object, given a sample of its boundary.

and empty-region graphs. This is done by introducing a new type of graphs, the γ -visibility graph.

Unlike previous empty-region graphs, this is a general graph, whose empty-regions can be shaped in a variety of forms and in any dimension. This has the potential to benefit the numerous applications that make use of empty-region graphs, by better choosing the template that is most suitable to the problem.

Finally, we have proved that the Delaunay triangulation is a special case of our graph. This leads to a novel optimal algorithm for constructing the Delaunay triangulation, by applying the GHPR operator. Conversely, we have shown how the Delaunay triangulation can be utilized for approximating the γ -visibility graph.

A natural direction for future research is investigating applications that may benefit from different shapes of empty-region graphs. As mentioned before, a robot may rather utilize a non-circular shape of the kernel, such as $f_{natural}$, which emanates from the camera, whereas more circular shapes may be preferable in clustering.

References

- [1] Edelsbrunner H, Kirkpatrick D, Seidel R. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory* 1983;29(4):551–8.
- [2] Bouchaffra D. Mapping dynamic bayesian networks to α -shapes: Application to human faces identification across ages. *IEEE Trans Neural Netw Learning Syst* 2012;23(8):1229–41.
- [3] Finlayson GD, Schaefer G. Convex and non-convex illuminant constraints for dichromatic colour constancy. In: *CVPR*. 2001, p. 598–604.
- [4] Varytimidis C, Rapantzikos K, Avrithis Y. W α sh: Weighted α -shapes for local feature detection. In: *European Conference on Computer Vision (ECCV)*. 2012,.
- [5] Correa CD, Lindstrom P. Locally-scaled spectral clustering using empty region graphs. In: *International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD)*. ACM. ISBN 978-1-4503-1462-6; 2012, p. 1330–8.
- [6] Zhang W, King I. Locating support vectors via β -skeleton technique. In: *International Conference on Neural Information Processing (ICONIP)*. 2002, p. 1423–7.
- [7] Correa C, Lindstrom P. Towards robust topology of sparsely sampled data. *IEEE Transactions on Visualization and Computer Graphics* 2011;17(12):1852–61.
- [8] Devroye L. On the inequality of cover and hart in nearest neighbor discrimination. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1981;3:75–8.
- [9] Manvendy B, Satyajayant M, X. G. Distributed topology control in wireless ad hoc networks using β -skeleton. In: *Workshop on High Performance Switching and Routing (HPSR)*. 2005, p. 371–5.
- [10] Zhou W, Yan H. Alpha shape and delaunay triangulation in studies of protein-related interactions. *Briefings in Bioinformatics* 2014;15(1):54–64.
- [11] Katz S, Tal A, Basri R. Direct visibility of point sets. *ACM Transactions on Graphics (TOG)* 2007;24.
- [12] Katz S, Tal A. On the visibility of point clouds. In: *Int. Conference on Computer Vision (ICCV)*. 2015,.
- [13] Cardinal J, Collette S, Langerman S. Empty region graphs. *Computational geometry theory appl* 2009;42(3):183–95.
- [14] Berger M, Tagliasacchi A, Seversky LM, Alliez P, Levine JA, Sharf A, et al. State of the art in surface reconstruction from point clouds. In: *Eurographics STAR*. 2014,.
- [15] Kazhdan M, Hoppe H. Screened poisson surface reconstruction. *ACM Transactions on Graphics* 2013,.
- [16] Goodman JE, O'Rourke J, editors. *Handbook of discrete and computational geometry*. Boca Raton, FL, USA: CRC Press, Inc.; 1997.
- [17] Ghosh SK, Goswami PP. Unsolved problems in visibility graphs of points, segments and polygons. *CoRR* 2010,.
- [18] Kára J, Pór A, Wood DR. On the chromatic number of the visibility graph of a set of points in the plane. *Discrete Computational Geometry* 2005;34(3):497–506.
- [19] Matousek J. Blocking visibility for points in general position. *Discrete Computational Geometry* 2009;42(2):219–23.
- [20] Hershberger J. Finding the visibility graph of a simple polygon in time proportional to its size. In: *The third annual symposium on computational geometry*. SCG; New York, NY, USA: ACM; 1987, p. 11–20.
- [21] Ben-Moshe B, Hall-Holt O, Katz MJ, Mitchell J. Computing the visibility graph of points within a polygon. In: *The twentieth annual symposium on Computational geometry*. SCG; New York, NY, USA: ACM. ISBN 1-58113-885-7; 2004, p. 27–35.
- [22] Toussaint GT. The relative neighbourhood graph of a finite planar set. *Pattern Recognition* 1980;12:261–8.
- [23] Gabriel RK, Sokal RR. A new statistical approach to geographic variation analysis. *Systematic Zoology* 1969;18(3):259–78.
- [24] Kirkpatrick DG, Radke JD. *Framework for Computational Morphology*. Elsevier Science Publishers; 1985.
- [25] Veltkamp RC. The gamma-neighborhood graph. *Computational Geometry* 1991;1:227–46.
- [26] Amenta N, Bern M, Eppstein D. The crust and the beta-skeleton: Combinatorial curve reconstruction. In: *Graphical Models and Image Processing*. 1998, p. 125–35.
- [27] Keil J, Gutwin C. Classes of graphs which approximate the complete euclidean graph. *Discrete & Computational Geometry* 1992;7:13–28.
- [28] de Berg M, van Kreveld M, Overmars M, Schwarzkopf O. *Computational geometry: algorithms and applications*. NJ, USA: Springer-Verlag New York, Inc.; 1997.