Delaunay Graphs and their applications to Computer Graphics and Image Analysis

Lecture by

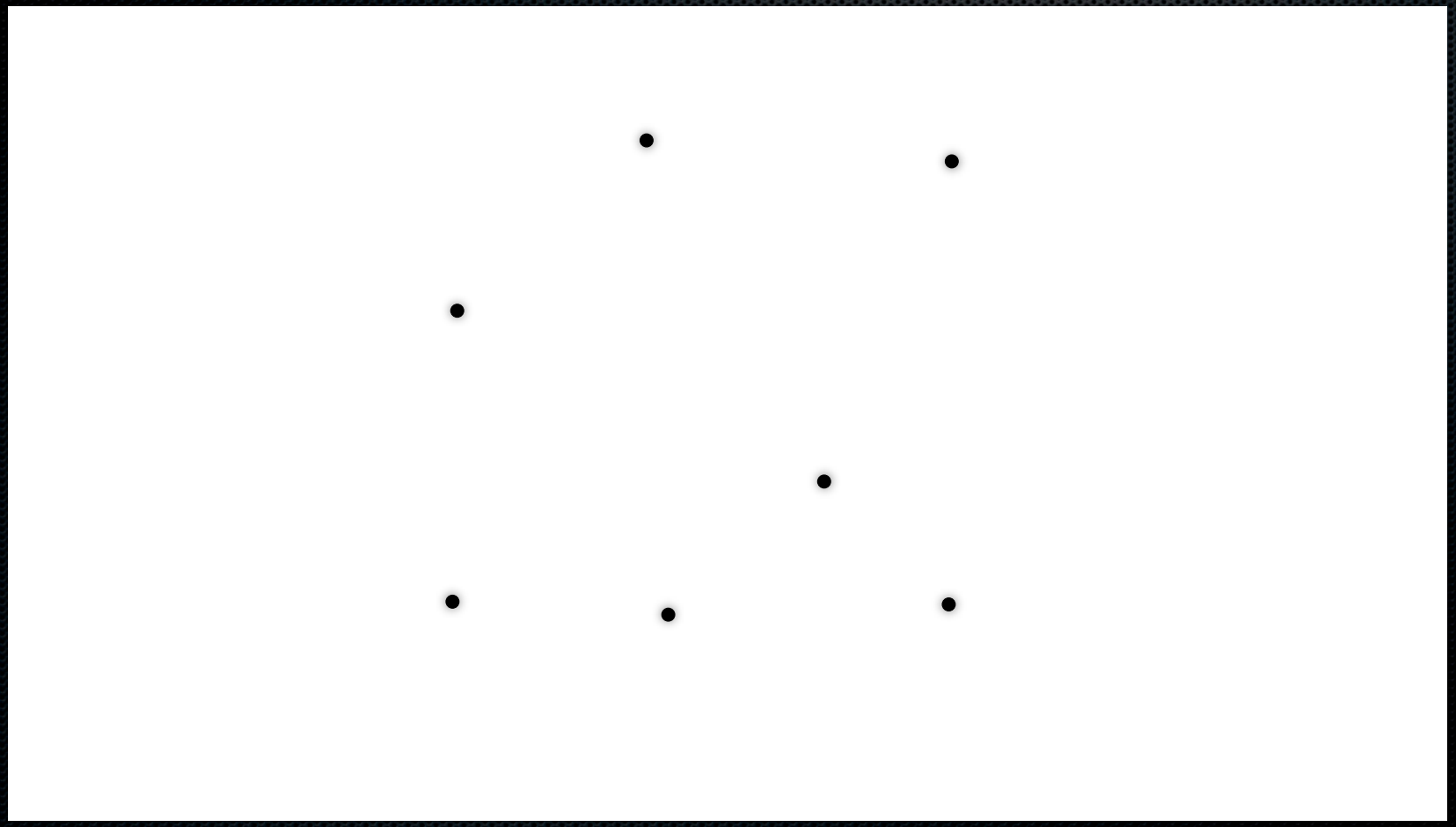
Associate Professor François Anton

Summer School on Graphs in Computer Graphics, Image and Signal Analysis

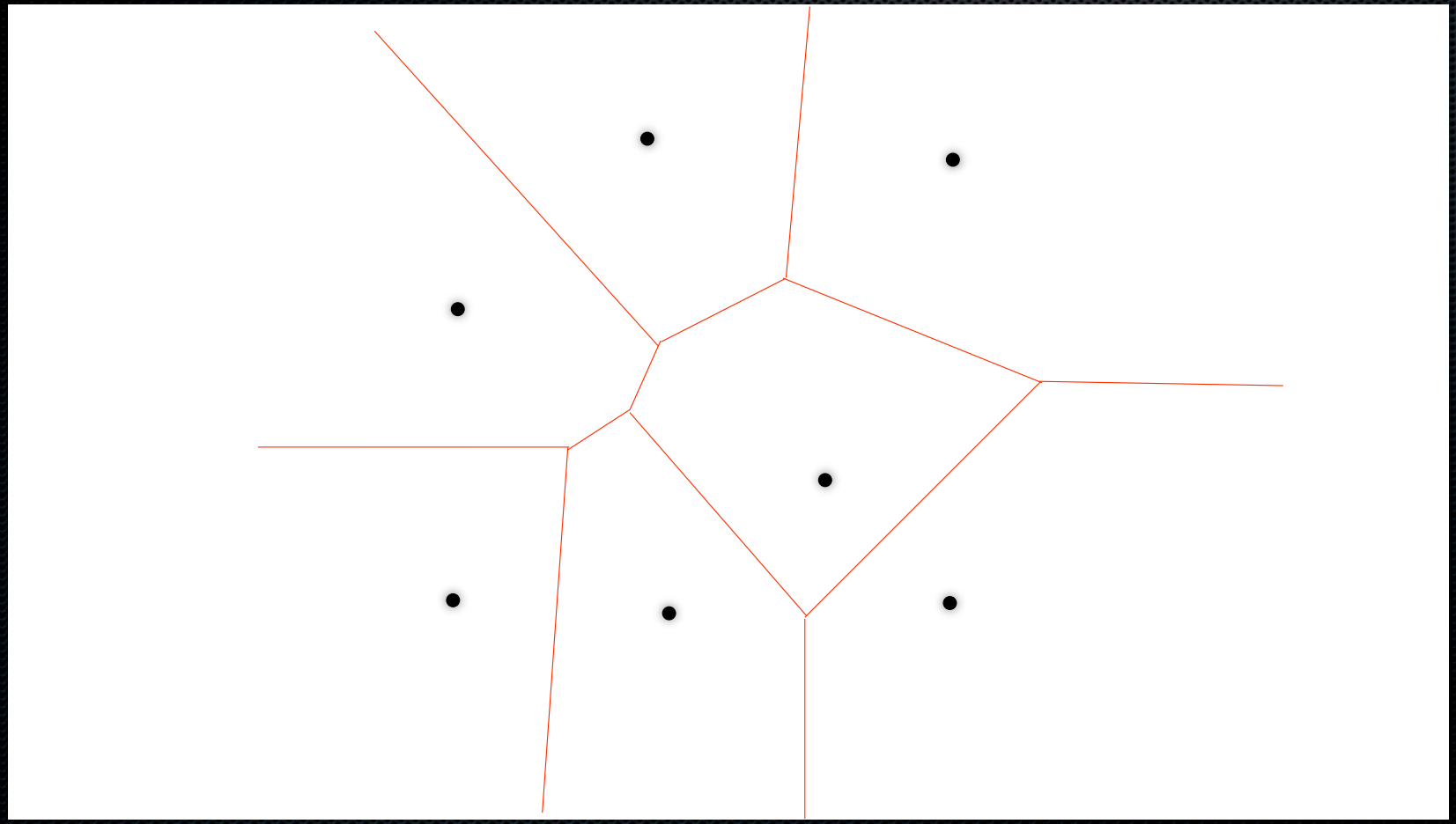
Rutsker, Bornholm, Denmark, August 2011

Delaunay graph and Voronoi/Dirichlet decomposition (points)

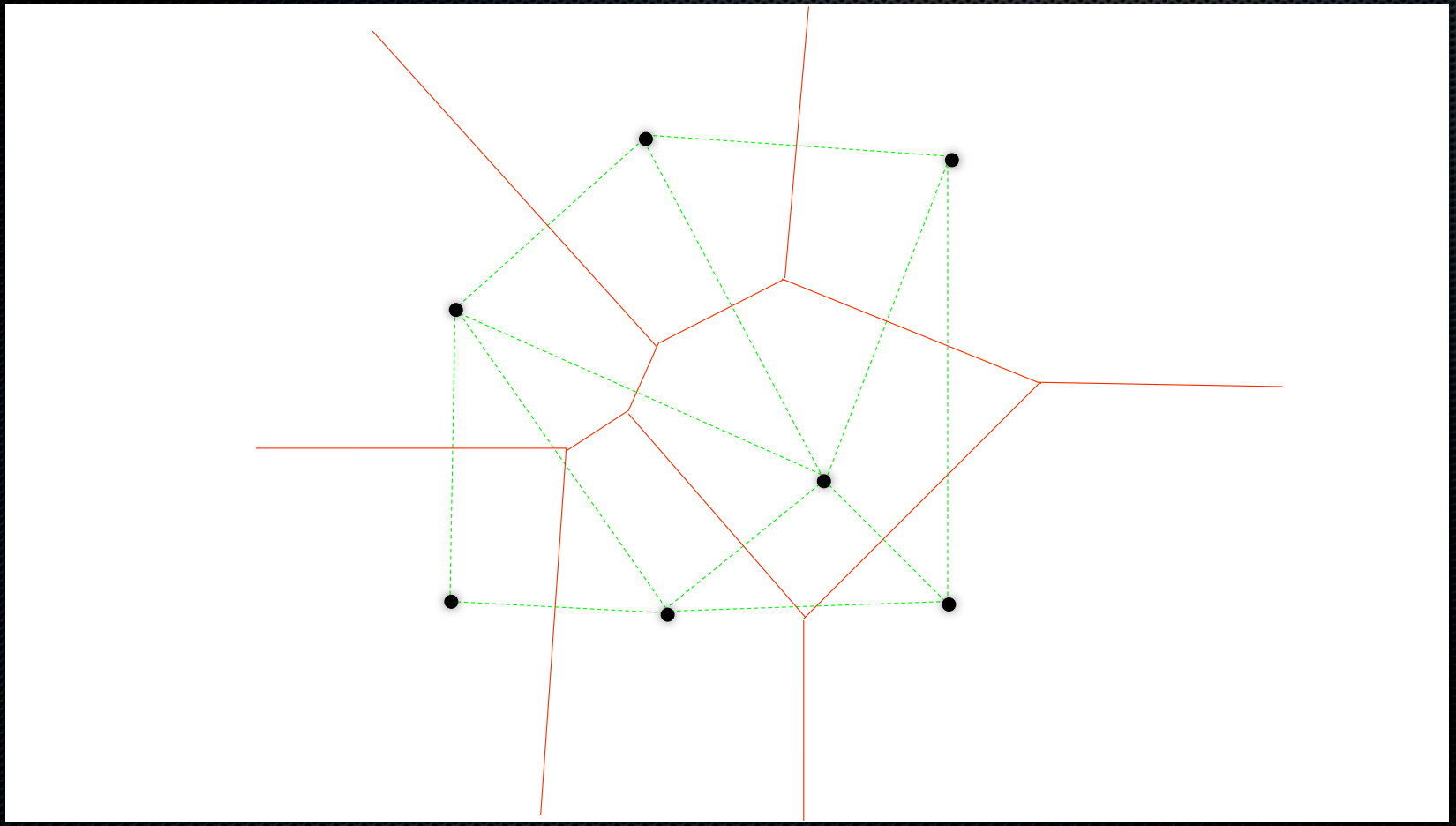
Delaunay graph and Voronoi/Dirichlet decomposition (points)



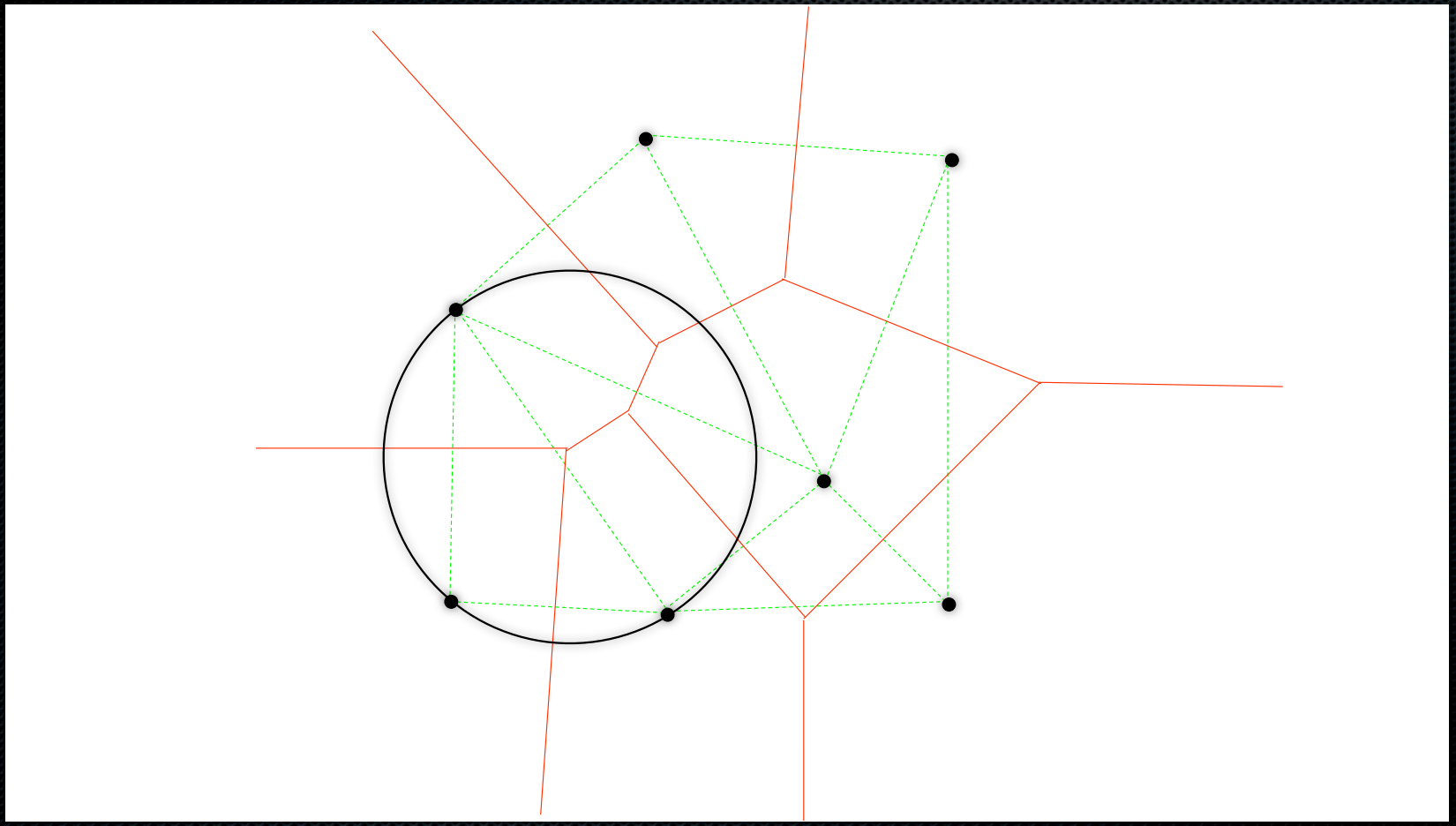
Delaunay graph and Voronoi/Dirichlet decomposition (points)



Delaunay graph and Voronoi/Dirichlet decomposition (points)



Delaunay graph and Voronoi/Dirichlet decomposition (points)



Voronoi diagrams of sites

Let M be a $n-$dimensional metric space, and δ denote the distance between points of M. Let $\mathcal{S} = \{s_1, ..., s_m\} \subset M$ be a set of $m \geq 2$ sites. The distance between $x \in M$ and a site s_i is $d(x, s_i) = \inf_{y \in s_i} \{\delta(x, y)\}$.

Definition

(Influence zone) For $s_i, s_j \in \mathcal{S}, s_i \neq s_j$, the *influence zone* $D(s_i, s_j)$ of s_i with respect to s_j is: $D(s_i, s_j) = \{x \in M | d(x, s_i) < d(x, s_j)\}$.

Definition

(Voronoi region) The *Voronoi region* $V(s_i, \mathcal{S})$ of $s_i \in \mathcal{S}$ with respect to the set $\mathcal{S}$ is: $V(s_i, \mathcal{S}) = \bigcap_{s_j \in \mathcal{S}, s_j \neq s_i} D(s_i, s_j)$.

Definition

(Voronoi diagram) The *Voronoi diagram* of $\mathcal{S}$ is the union $V(\mathcal{S}) = \bigcup_{s_i \in \mathcal{S}} \partial V(s_i, \mathcal{S})$ of all region boundaries.

Voronoi diagrams of sites

Let M be a $n-$dimensional metric space, and δ denote the distance between points of M. Let $\mathcal{S} = \{s_1, ..., s_m\} \subset M$ be a set of $m \geq 2$ sites. The distance between $x \in M$ and a site s_i is $d(x, s_i) = \inf_{y \in s_i} \{\delta(x, y)\}$.

Definition

(Influence zone) For $s_i, s_j \in \mathcal{S}, s_i \neq s_j$, the *influence zone* $D(s_i, s_j)$ of s_i with respect to s_j is: $D(s_i, s_j) = \{x \in M | d(x, s_i) < d(x, s_j)\}$.

Definition

(Voronoi region) The *Voronoi region* $V(s_i, \mathcal{S})$ of $s_i \in \mathcal{S}$ with respect to the set $\mathcal{S}$ is: $V(s_i, \mathcal{S}) = \bigcap_{s_j \in \mathcal{S}, s_j \neq s_i} D(s_i, s_j)$.

Definition

(Voronoi diagram) The *Voronoi diagram* of $\mathcal{S}$ is the union $V(\mathcal{S}) = \bigcup_{s_i \in \mathcal{S}} \partial V(s_i, \mathcal{S})$ of all region boundaries.

Voronoi diagrams of sites

Let M be a $n-$dimensional metric space, and δ denote the distance between points of M. Let $\mathcal{S} = \{s_1, ..., s_m\} \subset M$ be a set of $m \geq 2$ sites. The distance between $x \in M$ and a site s_i is $d(x, s_i) = \inf_{y \in s_i} \{\delta(x, y)\}$.

Definition

(Influence zone) For $s_i, s_j \in \mathcal{S}, s_i \neq s_j$, the *influence zone* $D(s_i, s_j)$ of s_i with respect to s_j is: $D(s_i, s_j) = \{x \in M | d(x, s_i) < d(x, s_j)\}$.

Definition

(Voronoi region) The *Voronoi region* $V(s_i, \mathcal{S})$ of $s_i \in \mathcal{S}$ with respect to the set $\mathcal{S}$ is: $V(s_i, \mathcal{S}) = \bigcap_{s_j \in \mathcal{S}, s_j \neq s_i} D(s_i, s_j)$.

Definition

(Voronoi diagram) The *Voronoi diagram* of $\mathcal{S}$ is the union $V(\mathcal{S}) = \bigcup_{s_i \in \mathcal{S}} \partial V(s_i, \mathcal{S})$ of all region boundaries.

Voronoi diagrams of sites

Let M be a $n-$dimensional metric space, and δ denote the distance between points of M. Let $\mathcal{S} = \{s_1, ..., s_m\} \subset M$ be a set of $m \geq 2$ sites. The distance between $x \in M$ and a site s_i is $d(x, s_i) = \inf_{y \in s_i} \{\delta(x, y)\}$.

Definition

(Influence zone) For $s_i, s_j \in \mathcal{S}, s_i \neq s_j$, the *influence zone* $D(s_i, s_j)$ of s_i with respect to s_j is: $D(s_i, s_j) = \{x \in M | d(x, s_i) < d(x, s_j)\}$.

Definition

(Voronoi region) The *Voronoi region* $V(s_i, \mathcal{S})$ of $s_i \in \mathcal{S}$ with respect to the set $\mathcal{S}$ is: $V(s_i, \mathcal{S}) = \bigcap_{s_j \in \mathcal{S}, s_j \neq s_i} D(s_i, s_j)$.

Definition

(Voronoi diagram) The *Voronoi diagram* of $\mathcal{S}$ is the union $V(\mathcal{S}) = \bigcup_{s_i \in \mathcal{S}} \partial V(s_i, \mathcal{S})$ of all region boundaries.

Delaunay graph of sites

Definition

(Delaunay graph) The *Delaunay graph* $DG(\mathcal{S})$ of $\mathcal{S}$ is the dual graph of $V(\mathcal{S})$ defined as follows:

- the set of vertices of $DG(\mathcal{S})$ is $\mathcal{S}$,
- for each $n-1-$dimensional facet of $V(\mathcal{S})$ that belongs to the common boundary of $V(s_i, \mathcal{S})$ and of $V(s_j, \mathcal{S})$ with $s_i, s_j \in \mathcal{S}$ and $s_i \neq s_j$, there is an edge of $DG(\mathcal{S})$ between s_i and s_j and reciprocally, and
- for each vertex of $V(\mathcal{S})$ that belongs to the common boundary of $V(s_{i_1}, \mathcal{S}), \ldots, V(s_{i_{n+2}}, \mathcal{S})$, with $\forall k \in \{1, ..., n+2\}, s_{i_k} \in \mathcal{S}$ all distinct, there exists a complete graph K_{n+2} between the $s_{i_k}, k \in \{1, ..., n+2\}$, and reciprocally.

Generalised Voronoi tessellation / Generalised Delaunay graph

Let S be the metric space in which we place ourselves (typically $\mathbb{R}^2$).

Definition

A mapping $\delta : S \times \mathcal{O} \longrightarrow \{0, 1\}$ defined by $(p, O_i) \mapsto \delta(p, O_i)$ such that:

$$\delta(p, O_i) = \begin{cases} 1, & \text{if } p \text{ is assigned to } O_i \\ 0, & \text{otherwise} \end{cases}$$

is called an assignment rule.

Generalised Voronoi tessellation / Generalised Delaunay graph

Under an assignment rule δ, we consider the set $v(O_i)$ of points assigned to O_i, and the set $e(O_i, O_j)$ of points assigned to both O_i and O_j with $i \neq j$.

Definition

A Voronoi tessellation is a set $\mathcal{V}(\mathcal{O}, \delta, S)$ such that the assignment rule δ satisfies the following two conditions:

- every point in S is assigned to at least one element of $\mathcal{O}$ i.e., $\forall p \in S, \sum_{i=1}^{n} \delta(p, O_i) \geq 1$;
- the set $e(O_i, O_j)$ pertains to the boundaries of $v(O_i)$ and of $v(O_i)$, i.e., $\forall \varepsilon > 0, \forall p \in e(O_i, O_j)$:
  - $N_\varepsilon(p) \cap [v(O_i) \setminus e(O_i, O_j)] \neq \emptyset$ and
  - $N_\varepsilon(p) \cap [S \setminus v(O_i)] \neq \emptyset$ and
  - $N_\varepsilon(p) \cap [v(O_j) \setminus e(O_i, O_j)] \neq 9\emptyset$ and
  - $N_\varepsilon(p) \cap [S \setminus v(O_j)] \neq \emptyset$.

First part: Applications to Computer Graphics through Homotopy based Reconstruction

Homotopy Based Reconstruction from Acoustic Images

Ojaswa Sharma

DTU Informatics,
Technical University of Denmark, Denmark

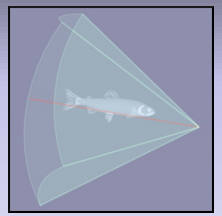
Supervisors

Associate Professor	Associate Professor
François Anton	**Niels Jørgen Christensen**
DTU Informatics	DTU Informatics

Examining Committee

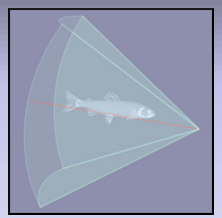
Professor	Professor	Associate Professor
Vagn Lundsgaards Hansen	**Jon Rokne**	**Joan Antoni Sellarès Chiva**
DTU mathematics	University of Calgary, Canada	University of Gorona, Spain

September 10, 2010

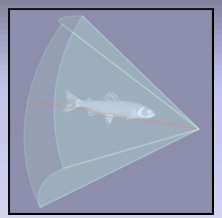


Outline

- Motivation
- Problem statement
- Contributions
- Segmentation of large volumetric images
- Object reconstruction from ordered set of intersections
- Object reconstruction from arbitrary cross sections
- Discussion

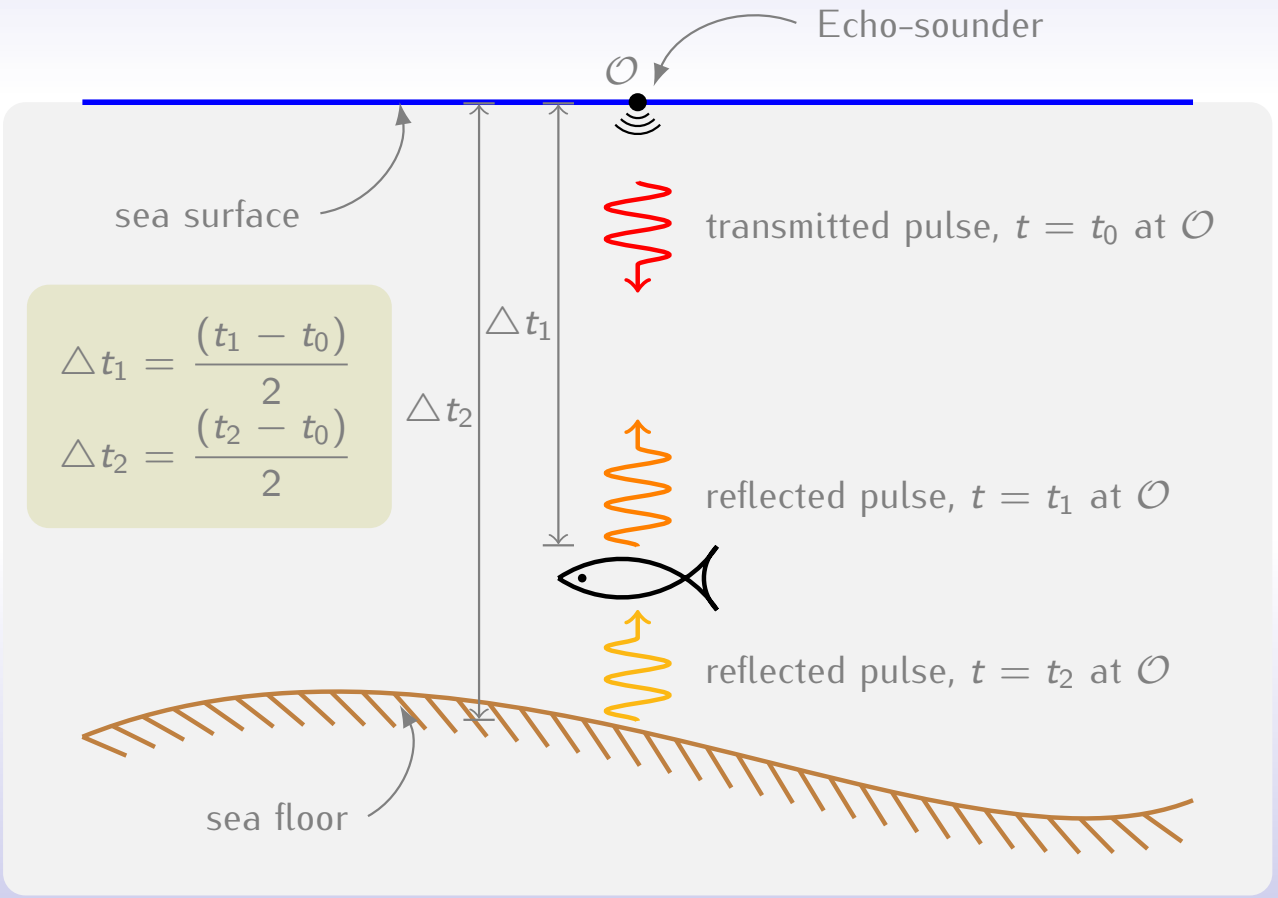


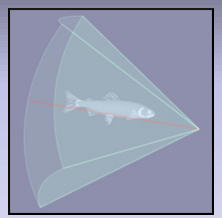
Holistic View



Reconstruction from Acoustic Signals

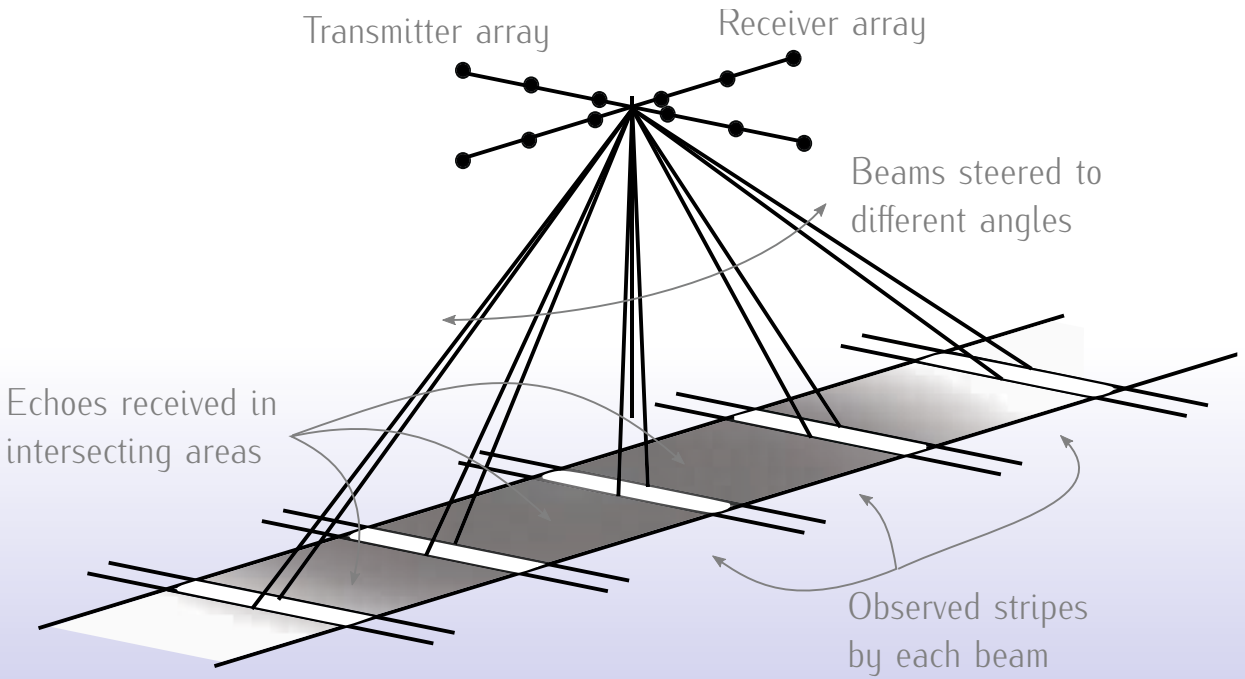
Acoustic waves probe inside the subject non-destructively. Echoes from the medium give an estimate of the relative position of the object below.

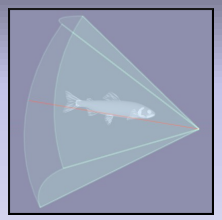




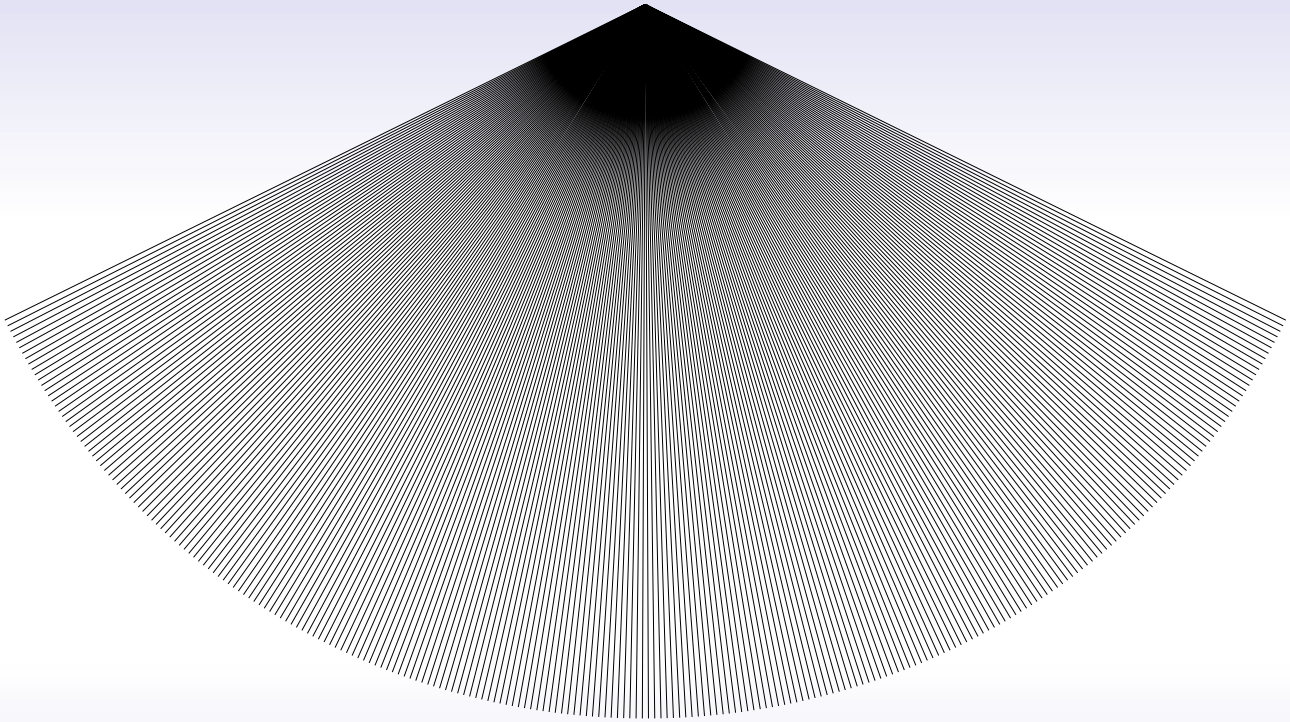
Reconstruction from Acoustic Signals

Acoustic waves probe inside the subject non-destructively. Echoes from the medium give an estimate of the relative position of the object below.

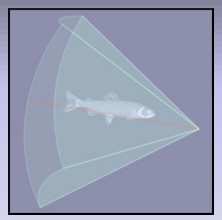




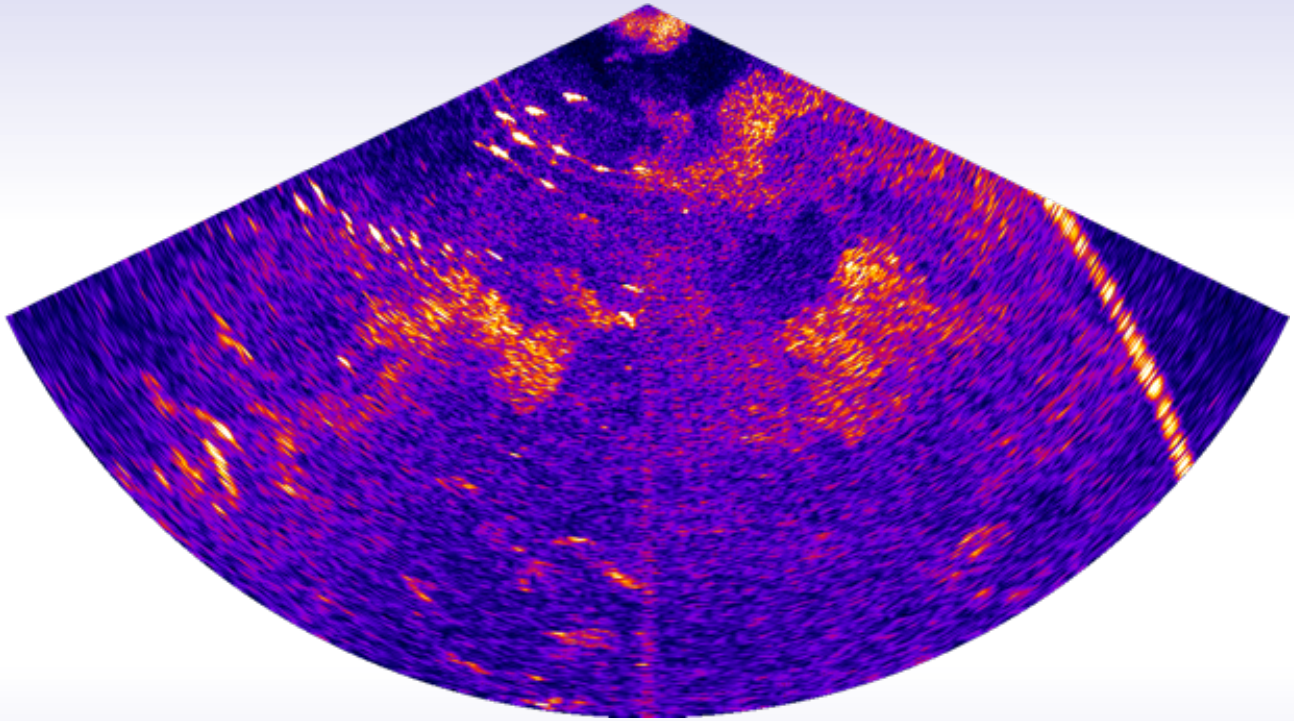
Acoustic Beam Geometry in $\mathbb{R}^2$



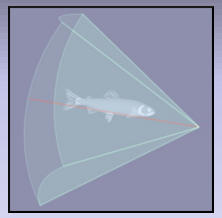
RESON 7128 [8], 396 kHz, 256 beams/fan, 127.5° operating sector



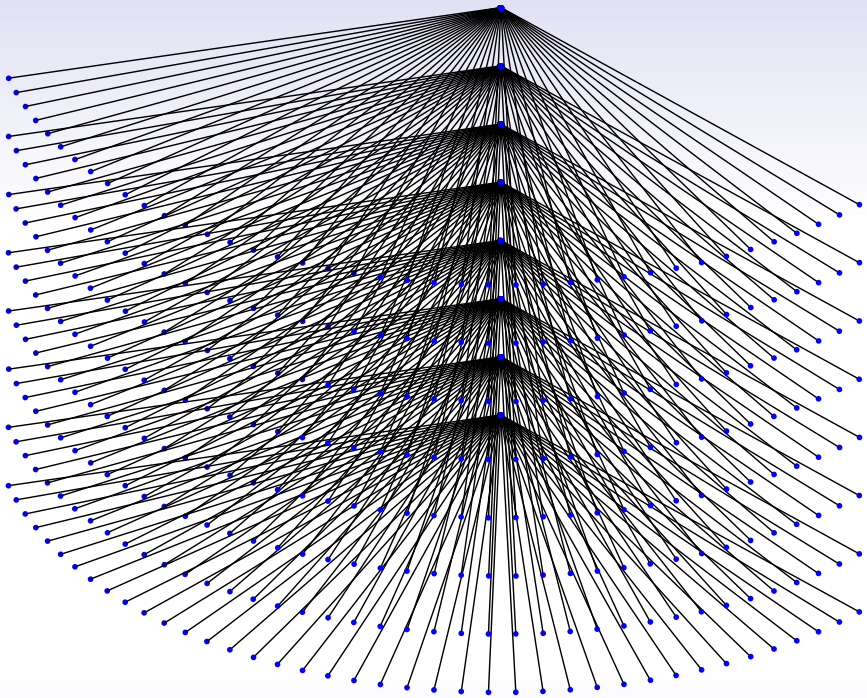
Acoustic Beam Geometry in $\mathbb{R}^2$



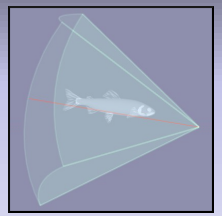
Intensity image, 686 $\times$ 1234



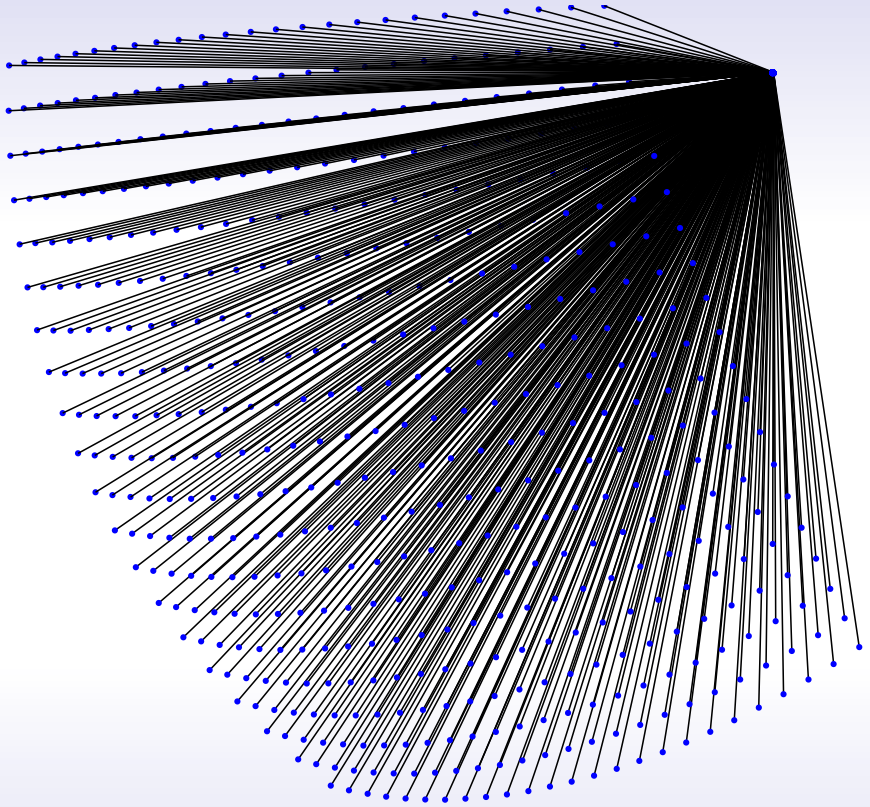
Acoustic Beam Geometry in $\mathbb{R}^3$



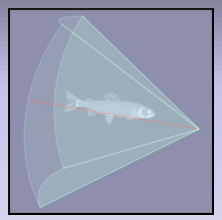
RESON 7128 [8], 396 kHz, 256 beams/fan, 127.5° operating sector



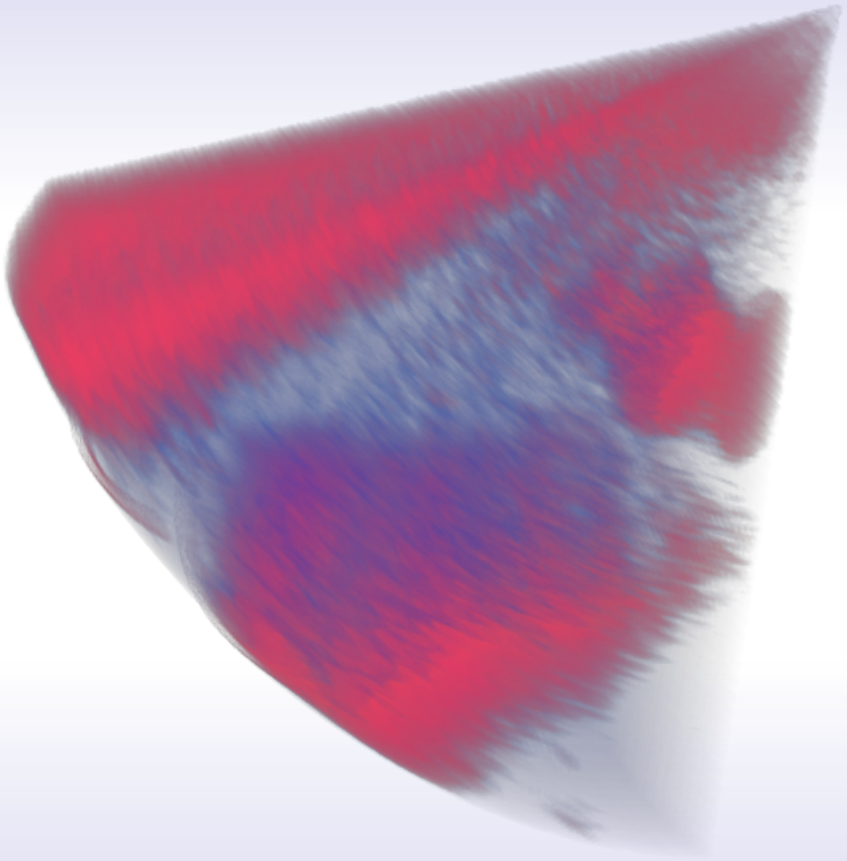
Acoustic Beam Geometry in $\mathbb{R}^3$



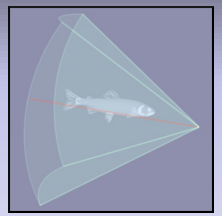
Simrad MS70 [14], 75-112 kHz, 500 beams (25×20), $60^\circ \times 45^\circ$ operating sector



Acoustic Beam Geometry in $\mathbb{R}^3$



Intensity volume

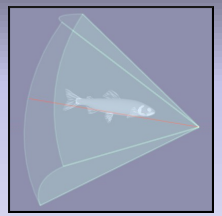


Problem Statement

The problem can be formally defined as

To reconstruct a surface representation of objects from sections of water volume scanned using an acoustic instrument (an echo-sounder).

Acoustic signals are one dimensional cross sections with higher dimensional objects (or lower dimensional objects embedded in higher dimensional space).



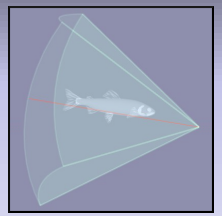
Problem Statement

The problem can be formally defined as

> To reconstruct a surface representation of objects from sections of water volume scanned using an acoustic instrument (an echo-sounder).
>
> Acoustic signals are one dimensional cross sections with higher dimensional objects (or lower dimensional objects embedded in higher dimensional space).

Main challenges in doing so are

- Noise



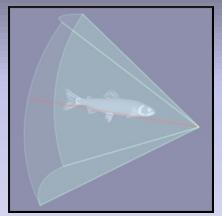
Problem Statement

The problem can be formally defined as

> To reconstruct a surface representation of objects from sections of water volume scanned using an acoustic instrument (an echo-sounder).
>
> Acoustic signals are one dimensional cross sections with higher dimensional objects (or lower dimensional objects embedded in higher dimensional space).

Main challenges in doing so are

- Noise
- Information extraction



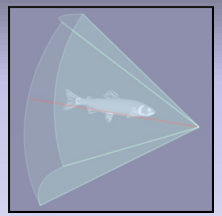
Problem Statement

The problem can be formally defined as

> To reconstruct a surface representation of objects from sections of water volume scanned using an acoustic instrument (an echo-sounder).
>
> Acoustic signals are one dimensional cross sections with higher dimensional objects (or lower dimensional objects embedded in higher dimensional space).

Main challenges in doing so are

- Noise
- Information extraction
- Huge amount of data



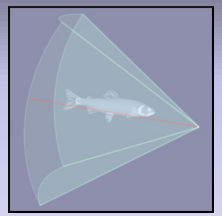
Problem Statement

The problem can be formally defined as

> To reconstruct a surface representation of objects from sections of water volume scanned using an acoustic instrument (an echo-sounder).
>
> Acoustic signals are one dimensional cross sections with higher dimensional objects (or lower dimensional objects embedded in higher dimensional space).

Main challenges in doing so are

- Noise *-Suppression, removal*
- Information extraction
- Huge amount of data



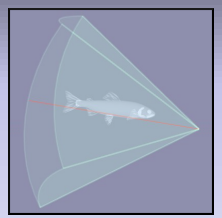
Problem Statement

The problem can be formally defined as

> To reconstruct a surface representation of objects from sections of water volume scanned using an acoustic instrument (an echo-sounder).
>
> Acoustic signals are one dimensional cross sections with higher dimensional objects (or lower dimensional objects embedded in higher dimensional space).

Main challenges in doing so are

- Noise — *-Suppression, removal*
- Information extraction — *-Topologically motivated algorithms*
- Huge amount of data



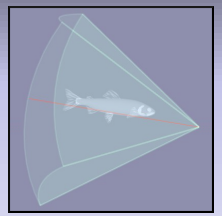
Problem Statement

The problem can be formally defined as

> To reconstruct a surface representation of objects from sections of water volume scanned using an acoustic instrument (an echo-sounder).
>
> Acoustic signals are one dimensional cross sections with higher dimensional objects (or lower dimensional objects embedded in higher dimensional space).

Main challenges in doing so are

- Noise — *-Suppression, removal*
- Information extraction — *-Topologically motivated algorithms*
- Huge amount of data — *-Parallel processing*

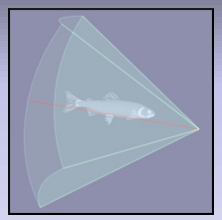


Roadmap

Reconstruction from ordered intersections

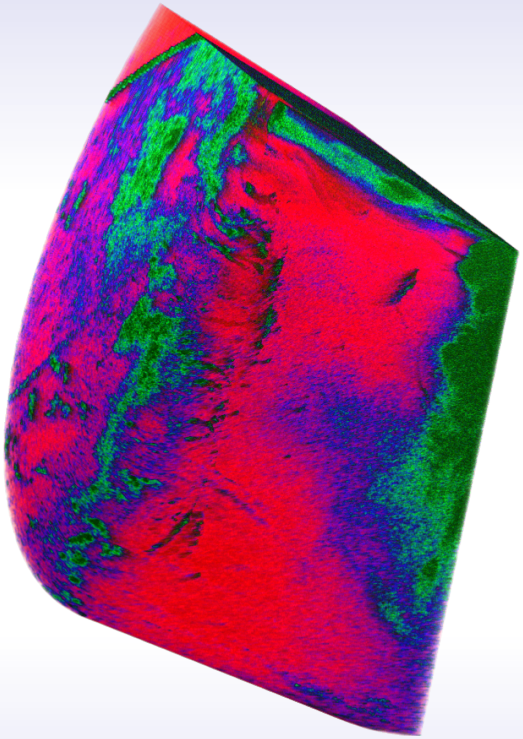
Homotopy based reconstruction from a set of ordered linear cross sections embedded in a higher dimensional space ($\mathbb{R}^2$ and $\mathbb{R}^3$).

- Continuous deformation of signals,
- Smooth,
- Monotonicity preserving, and
- Computationally less-expensive.

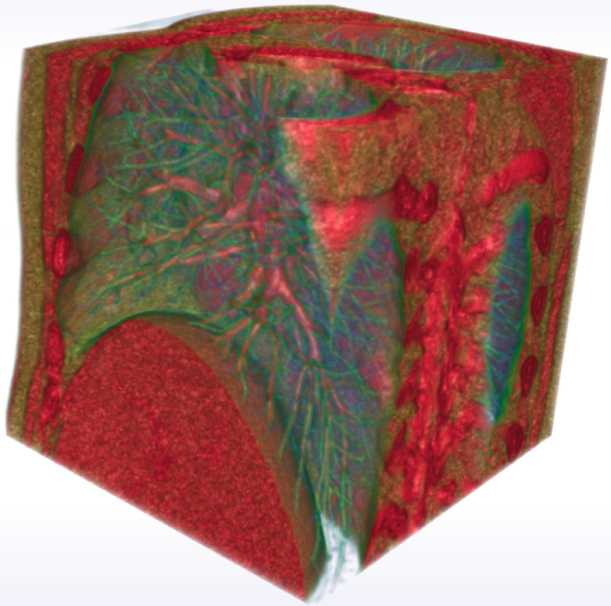


Volumetric Images

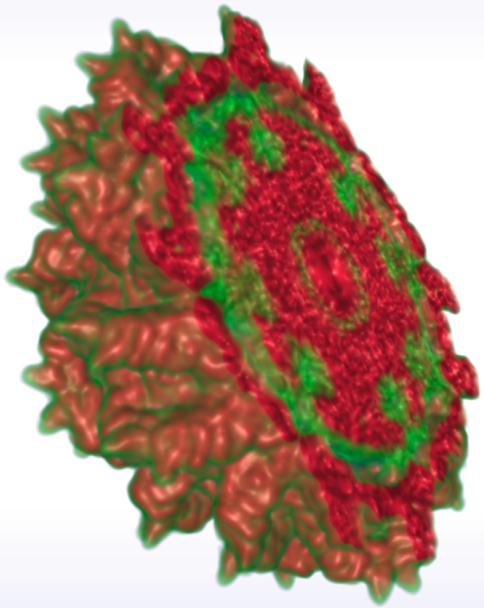
Large volumetric images are commonly acquired in many fields



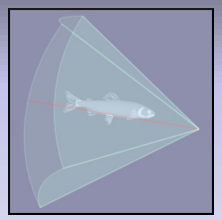
Acoustic image
Sonar
$686 \times 1234 \times 417$



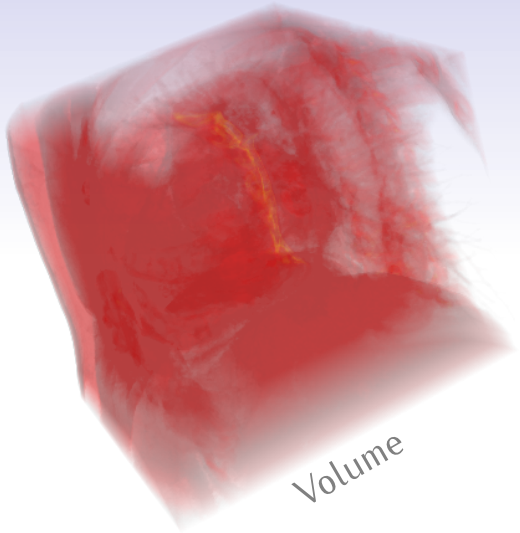
CT image
Thoracic cage
$512 \times 512 \times 512$

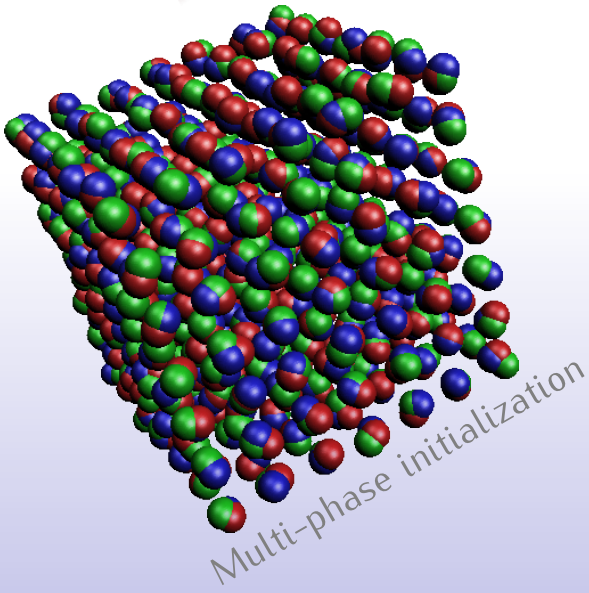


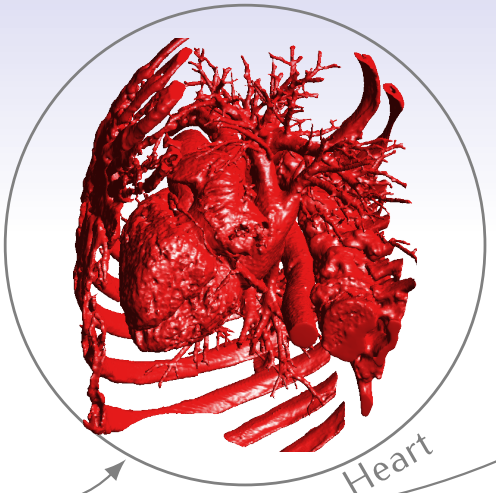
cryoEM image
PSV
$381 \times 381 \times 381$

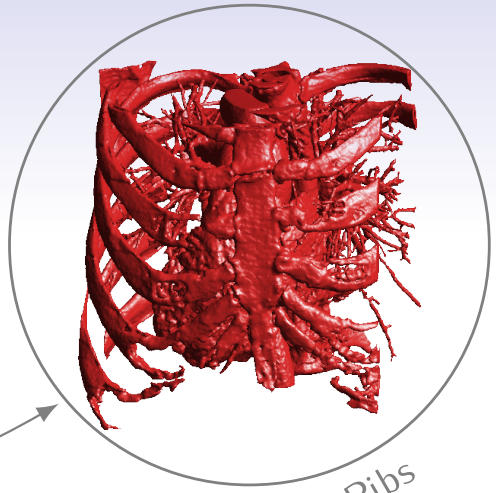


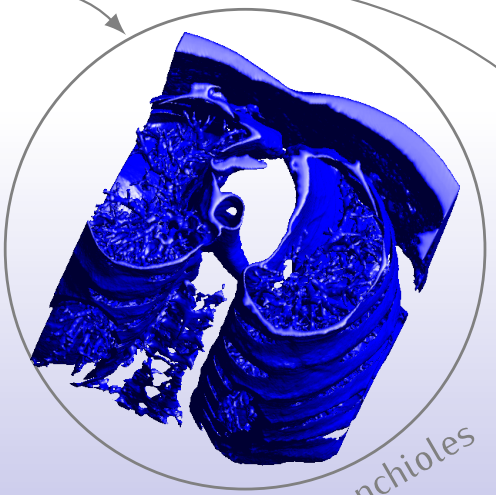
Results

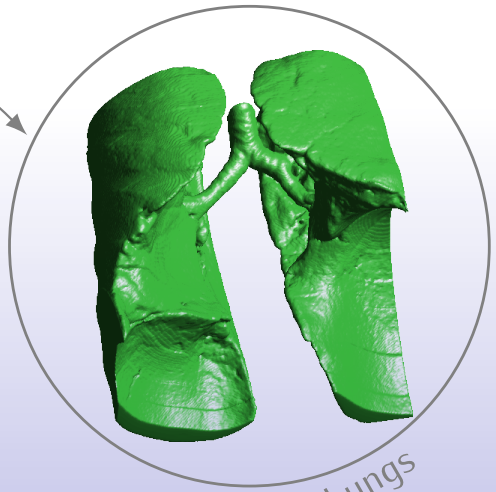

Volume

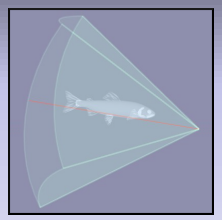

Multi-phase initialization


Heart

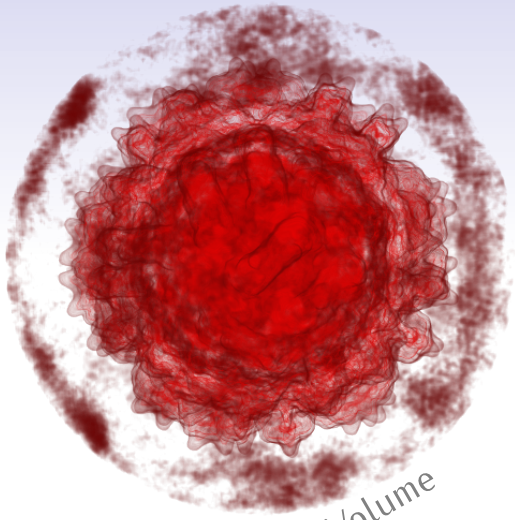

Ribs

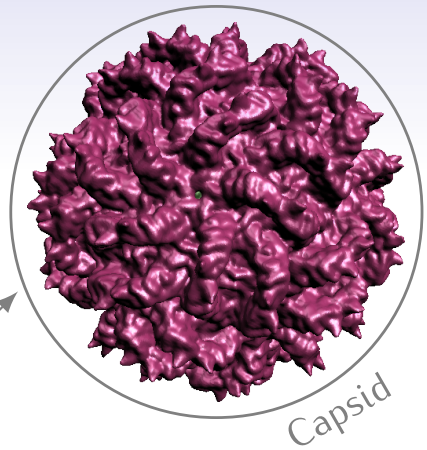

Bronchioles

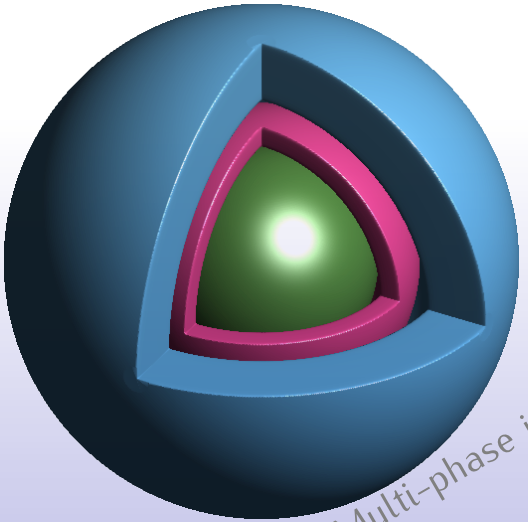

Lungs

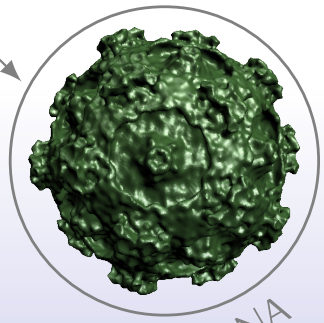


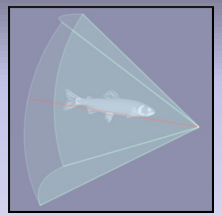
Results











Roadmap

Volume segmentation

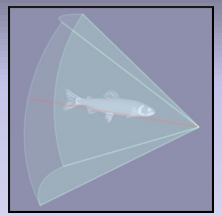
Level set method based streaming, smooth, and multi-phase segmentation.

Reconstruction from ordered intersections

Homotopy based reconstruction from a set of ordered linear cross sections embedded in a higher dimensional space ($\mathbb{R}^2$ and $\mathbb{R}^3$).

Reconstruction from arbitrary intersections

Homotopy based reconstruction from a set of arbitrarily placed linear cross sections embedded in a higher dimensional space.

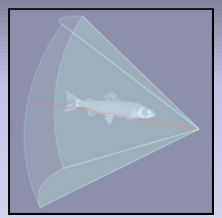


Roadmap

Reconstruction from arbitrary intersections

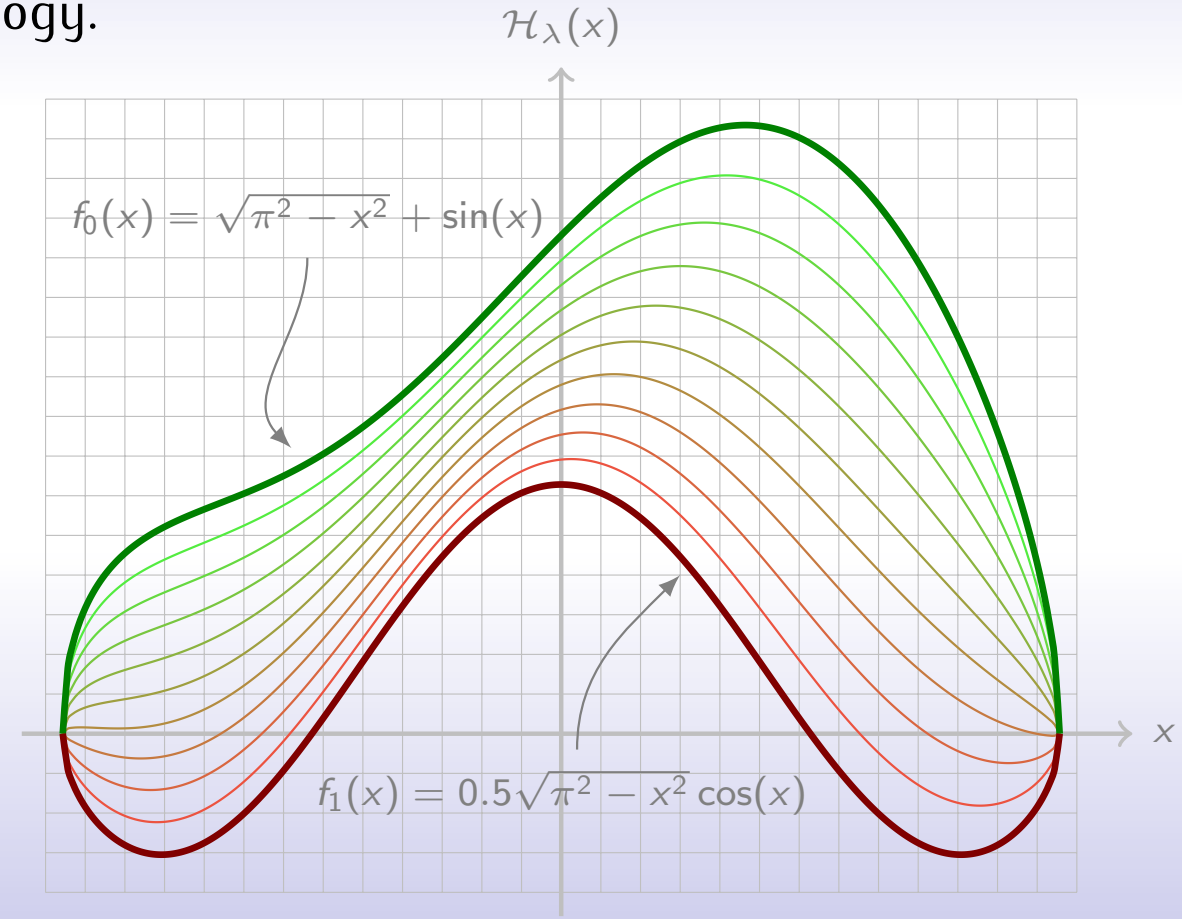
Homotopy based reconstruction from a set of arbitrarily placed linear cross sections embedded in a higher dimensional space.

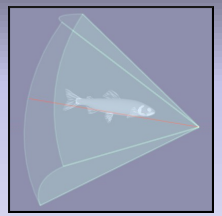
- A more general problem, and
- Edge based generic Barycentric coordinates.



Homotopy Continuation

Acoustic signals change continuously within and across frames. A homotopy or continuous deformation can effectively capture this change in morphology.





Homotopy Continuation

A **homotopy** is a family of continuous mappings $\mathcal{H} : X \times [0,1] \mapsto Y$.

Example

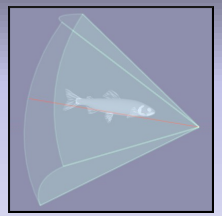
A *linear homotopy* can be written as

$$\mathcal{H}(\mathbf{x}, \lambda) = (1 - \lambda) f_0(\mathbf{x}) + \lambda f_1(\mathbf{x})$$

where $f_0(\mathbf{x})$ is the *initial map* and $f_1(\mathbf{x})$ is the *terminal map* of the homotopy.

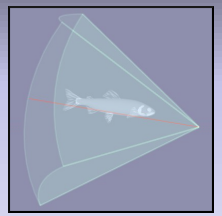
Use of continuous deformations in solving system of equations has resulted in stable algorithms for solutions [1]

- Predictor-corrector method, and
- Piecewise-linear method.



Homotopic Reconstruction

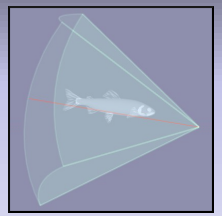
A homotopy allows us to continuously deform one function into another.



Homotopic Reconstruction

A homotopy allows us to continuously deform one function into another.

- If a function is attached to every acoustic signal, information between two consecutive signals can be reconstructed via a homotopy.



Homotopic Reconstruction

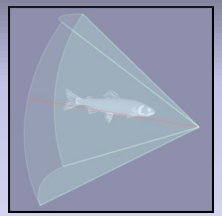
A homotopy allows us to continuously deform one function into another.

- If a function is attached to every acoustic signal, information between two consecutive signals can be reconstructed via a homotopy.

Beam function

Given a signal S,and its piecewise representation G with classes $\{C_i\}$, the characteristic function χ_k for a level C_k can be written as

$$\chi_k = \begin{cases} 1 & \text{if } G = C_k, \\ 0 & \text{otherwise.} \end{cases}$$



Homotopic Reconstruction

A homotopy allows us to continuously deform one function into another.

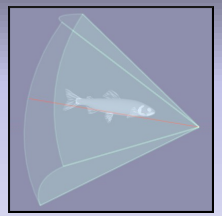
- If a function is attached to every acoustic signal, information between two consecutive signals can be reconstructed via a homotopy.

Beam function

Given a signal S,and its piecewise representation G with classes $\{C_i\}$, the characteristic function χ_k for a level C_k can be written as

$$\chi_k = \begin{cases} 1 & \text{if } G = C_k, \\ 0 & \text{otherwise.} \end{cases}$$

A suitable smooth function $f : \mathbb{R} \mapsto \mathbb{R}$ can be constructed and attached to a signal, given roots $r_i, i \in [0, p-1]$ of the characteristic function χ_k.



Homotopic Reconstruction

A homotopy allows us to continuously deform one function into another.

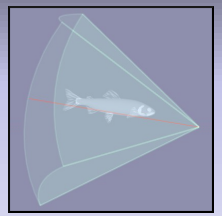
- If a function is attached to every acoustic signal, information between two consecutive signals can be reconstructed via a homotopy.
- An inherent ordering between acoustic beams allows for a parametrization between pairs of homotopies thus arising. This enables possibility of imposing smoothness at the boundary of the homotopies.

Homotopy field

Given a homotopy $\mathcal{H}_k(\mathbf{x}, \lambda)$ for a pair of beam functions $f_k(r)$ and $f_{k+1}(r)$, a set of homotopies give rise to a homotopy field

$$\mathbf{H}(r, \alpha) = \{\mathcal{H}_k(r, \lambda)\}, k = 0 \cdots n-1,$$

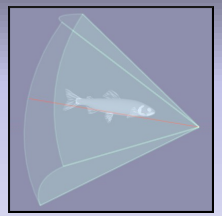
where $\alpha = g(\lambda)$ is the available parametrization due to geometric ordering of the signals.



Design of Homotopies

Based on this setup, a number of homotopies can be designed

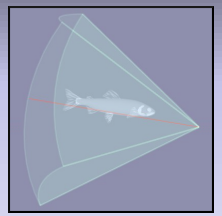
- Linear homotopy
  - ☹ C^0 w.r.t. the homotopy parameter(s),
  - ☹ Local, and
  - ☺ Shape preserving.



Design of Homotopies

Based on this setup, a number of homotopies can be designed

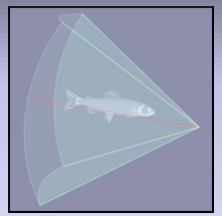
- Linear homotopy
  - ☹ C^0 w.r.t. the homotopy parameter(s),
  - ☹ Local, and
  - ☺ Shape preserving.
- Non-linear homotopy
  - ☺ $C^{\eta-1}$ w.r.t. the homotopy parameter(s),
  - ☹ Local, and
  - ☹ Staircase effect.



Design of Homotopies

Based on this setup, a number of homotopies can be designed

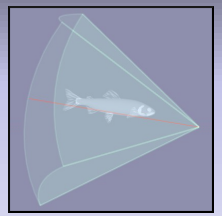
- Linear homotopy
  - ☹ C^0 w.r.t. the homotopy parameter(s),
  - ☹ Local, and
  - ☺ Shape preserving.
- Non-linear homotopy
  - ☺ $C^{\eta-1}$ w.r.t. the homotopy parameter(s),
  - ☹ Local, and
  - ☹ Staircase effect.
- Cubic spline homotopy
  - ☺ C^2 w.r.t. the homotopy parameter(s),
  - ☺ Global, and
  - ☹ Unnatural.



Design of Homotopies

Based on this setup, a number of homotopies can be designed

- Linear homotopy
  - ☹ C^0 w.r.t. the homotopy parameter(s),
  - ☹ Local, and
  - ☺ Shape preserving.
- Non-linear homotopy
  - ☺ $C^{\eta-1}$ w.r.t. the homotopy parameter(s),
  - ☹ Local, and
  - ☹ Staircase effect.
- Cubic spline homotopy
  - ☺ C^2 w.r.t. the homotopy parameter(s),
  - ☺ Global, and
  - ☹ Unnatural.
- Shape preserving homotopy.
  - ☺ C^2 in between the signals and C^1 at the signals,
  - ☺ Global, and
  - ☺ Shape preserving.

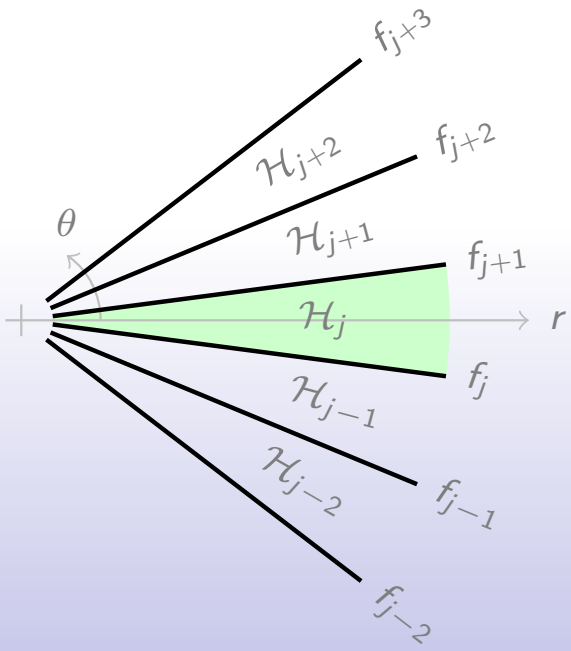


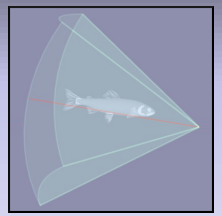
Linear Homotopy

$$\mathcal{H}_j(r,\lambda) = (1-\lambda)f_j(r) + \lambda f_{j+1}(r), \lambda \in [0,1].$$

The homotopy parameter λ is linked to the radial angle θ as

$$\lambda = \frac{\theta - \theta_j}{\theta_{j+1} - \theta_j}.$$





Linear Homotopy

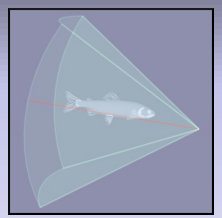
$$\mathcal{H}_j(r,\lambda) = (1-\lambda)f_j(r) + \lambda f_{j+1}(r), \lambda \in [0,1].$$

The homotopy parameter λ is linked to the radial angle θ as

$$\lambda = \frac{\theta - \theta_j}{\theta_{j+1} - \theta_j}.$$

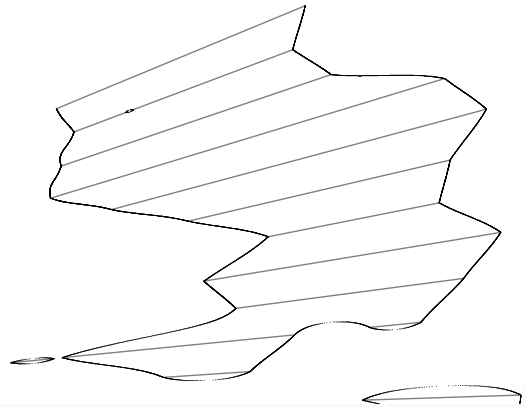
Proposition

$\mathbf{H} = \{\mathcal{H}_j\}$, *with* $\mathcal{H}_j$ *defined as above, results in a piecewise non-linear curve* $\mathbf{c} = \ker(H)$ *that is only* C^0 *in* θ.



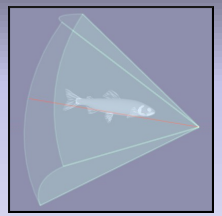
Linear Homotopy

$$\mathcal{H}_j(r,\lambda) = (1-\lambda)f_j(r) + \lambda f_{j+1}(r), \lambda \in [0,1].$$



Proposition

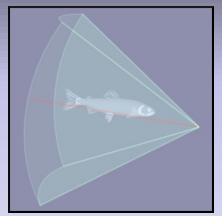
H $= \{\mathcal{H}_j\}$, *with* $\mathcal{H}_j$ *defined as above, results in a piecewise non-linear curve* **c** $= \ker(H)$ *that is only* C^0 *in* θ.



Non-linear Homotopy

$$\mathcal{H}_j(r, \lambda, \eta) = (1 - \lambda)^\eta f_j(r) + \lambda^\eta f_{j+1}(r), .$$

where $\lambda \in [0, 1]$, and $\eta \in \mathbb{R} > 1$.



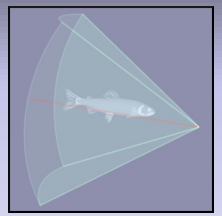
Non-linear Homotopy

$$\mathcal{H}_j(r, \lambda, \eta) = (1 - \lambda)^\eta f_j(r) + \lambda^\eta f_{j+1}(r), .$$

where $\lambda \in [0, 1]$, and $\eta \in \mathbb{R} > 1$.

Proposition

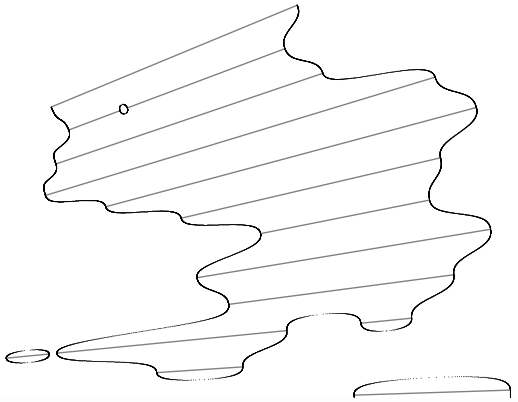
For $\eta > 1$, ker($\mathbf{H}$) generates at least a C^1 curve in θ for constant angular spacing of beams.



Non-linear Homotopy

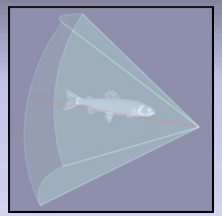
$$\mathcal{H}_j(r, \lambda, \eta) = (1 - \lambda)^\eta f_j(r) + \lambda^\eta f_{j+1}(r), .$$

where $\lambda \in [0, 1]$, and $\eta \in \mathbb{R} > 1$.



Proposition

*For $\eta > 1$, ker(**H**) generates at least a C^1 curve in θ for constant angular spacing of beams.*

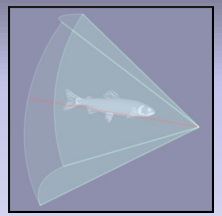


Cubic Spline Homotopy

For N radial beam functions $\{f_j(r)\}$, $j \in [0, N-1]$, additional smoothness constraints added to the linear homotopy

$$\lim_{\theta \to \theta_{j+1}^-} \frac{\partial \mathcal{H}_j(r,\theta)}{\partial \theta} = \lim_{\theta \to \theta_{j+1}^+} \frac{\partial \mathcal{H}_{j+1}(r,\theta)}{\partial \theta}, \; j \in [0, N-3], \text{and}$$

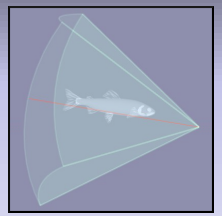
$$\lim_{\theta \to \theta_{j+1}^-} \frac{\partial^2 \mathcal{H}_j(r,\theta)}{\partial \theta^2} = \lim_{\theta \to \theta_{j+1}^+} \frac{\partial^2 \mathcal{H}_{j+1}(r,\theta)}{\partial \theta^2}, \; j \in [0, N-3].$$



Cubic Spline Homotopy

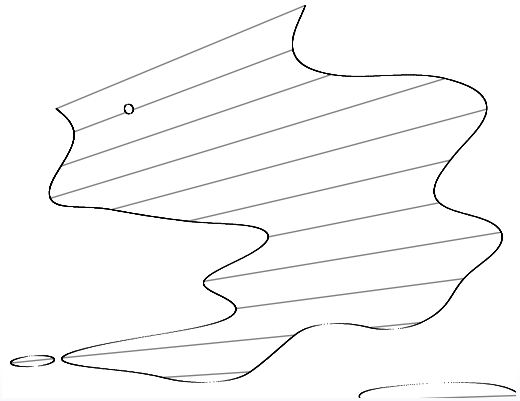
$$\mathcal{H}_j(r,\theta) = \sum_{i=0}^{3} g_{j,i}(r)(\theta - \theta_j)^i.$$

A linear system of equations has to be solved to deduce unknown functions $g_{j,i}(r)$ in terms of known beam functions $\{f_j\}$.

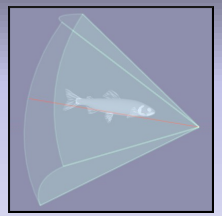


Cubic Spline Homotopy

$$\mathcal{H}_j(r,\theta) = \sum_{i=0}^{3} g_{j,i}(r)(\theta - \theta_j)^i.$$

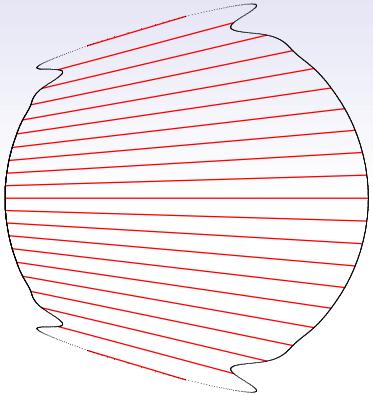


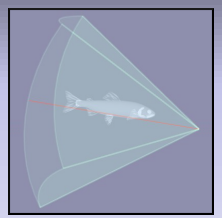
A linear system of equations has to be solved to deduce unknown functions $g_{j,i}(r)$ in terms of known beam functions $\{f_j\}$.



Shape Preserving Homotopy

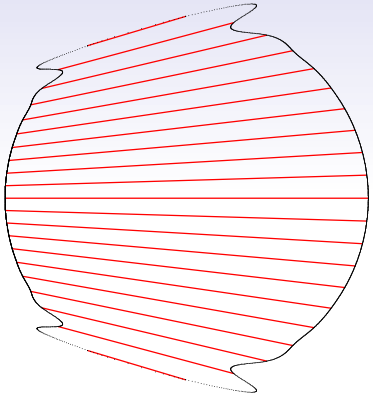
Monotonicity is not preserved with cubic splines





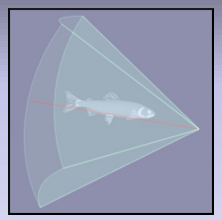
Shape Preserving Homotopy

Monotonicity is not preserved with cubic splines



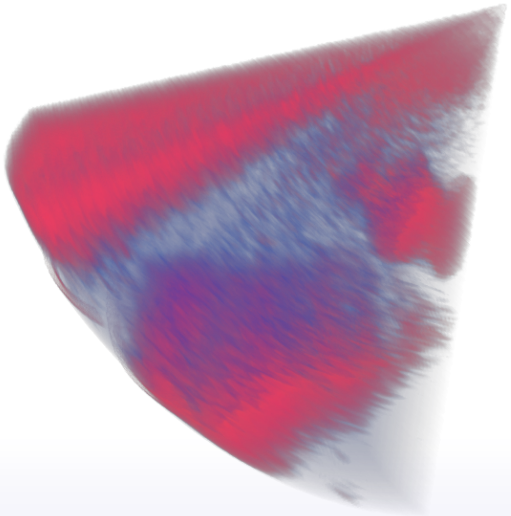
A shape preserving spline in terms of Hermite basis functions is

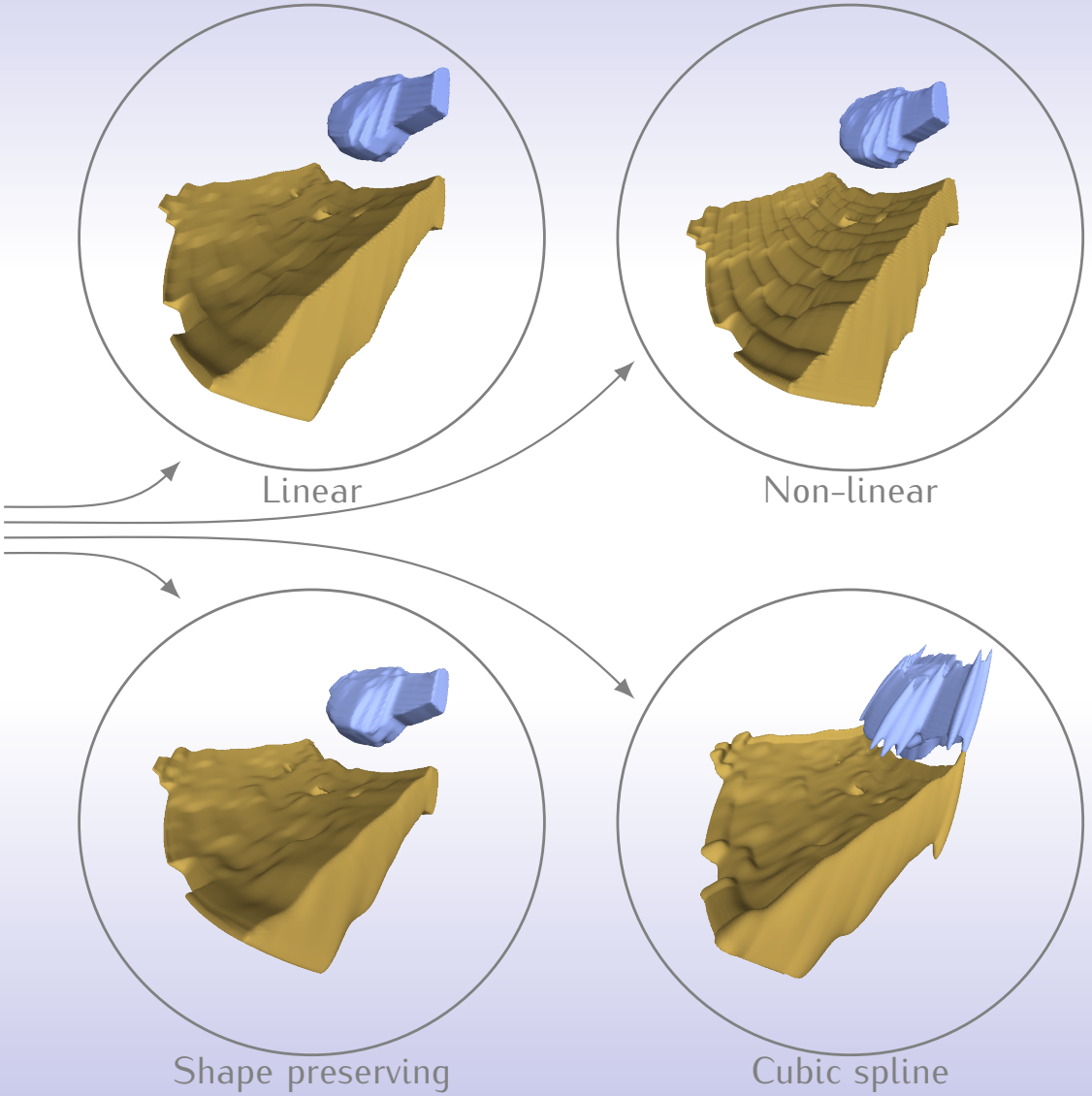
$$\begin{aligned}\mathcal{H}_j(r,\lambda) =& (1-3\lambda^2+2\lambda^3)f_j(r) + (3\lambda^2-2\lambda^3)f_{j+1}(r) \\ &+ (\lambda-2\lambda^2+\lambda^3)\left[\frac{\partial \mathcal{H}_j(r,\lambda)}{\partial\lambda}\right]_{\lambda=0} \\ &+ (\lambda^3-\lambda^2)\left[\frac{\partial \mathcal{H}_{j+1}(r,\lambda)}{\partial\lambda}\right]_{\lambda=0}.\end{aligned}$$

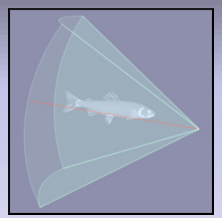


Results

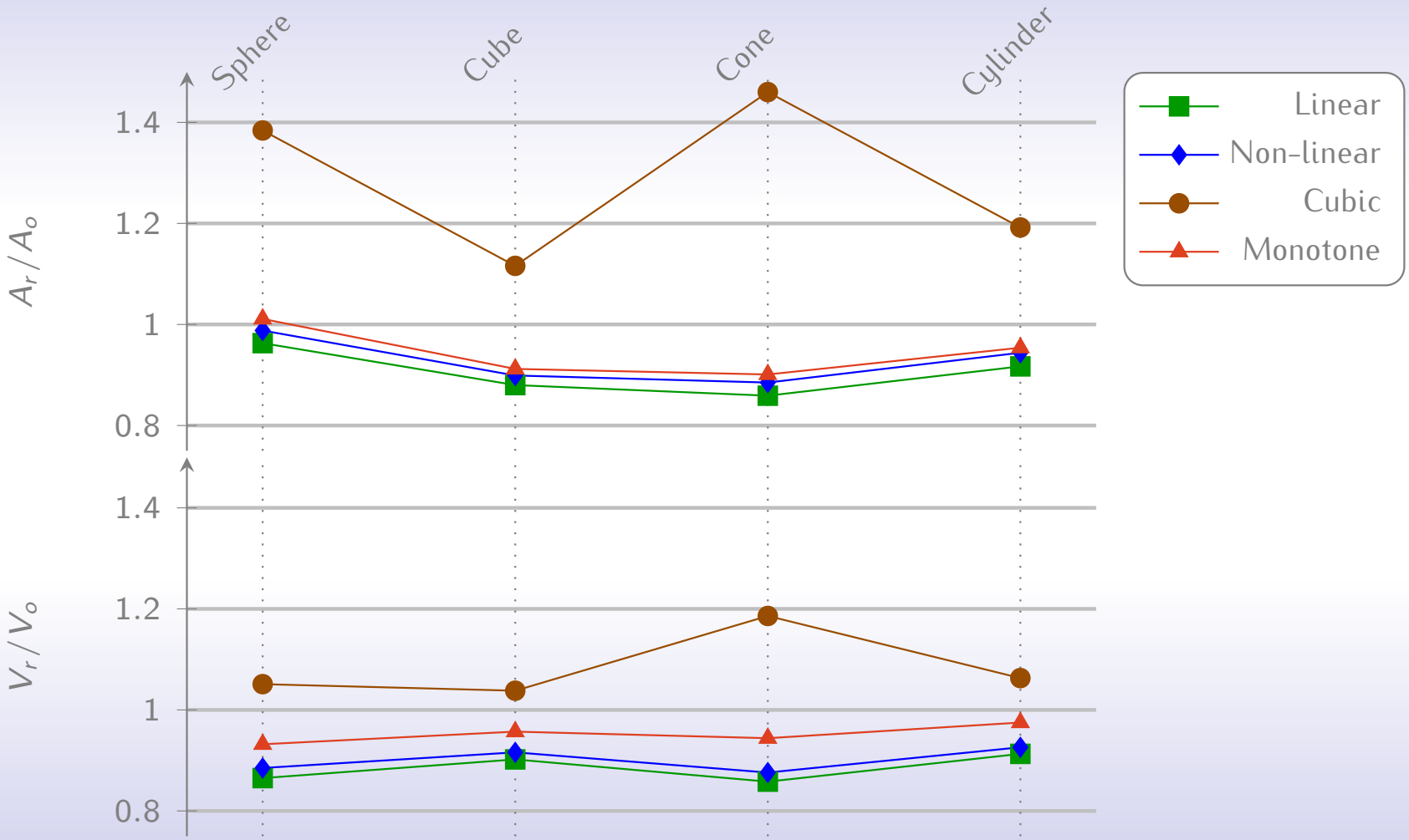
Volume

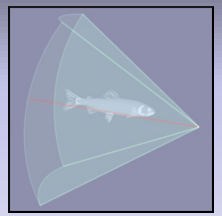






Reconstruction Accuracy





Roadmap

Volume segmentation

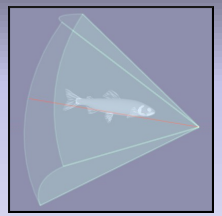
Level set method based streaming, smooth, and multi-phase segmentation.

Reconstruction from ordered intersections

Homotopy based reconstruction from a set of ordered linear cross sections embedded in a higher dimensional space ($\mathbb{R}^2$ and $\mathbb{R}^3$).

Reconstruction from arbitrary intersections

Homotopy based reconstruction from a set of arbitrarily placed linear cross sections embedded in a higher dimensional space.

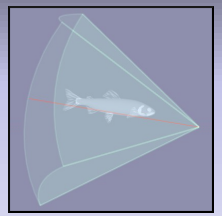


Reconstruction Problem in $\mathbb{R}^2$

Reconstruct object $\mathcal{O}$ intersected by a set of arbitrarily placed lines $\{\mathcal{L}_i\}, i \in [0, n-1]$ in a plane. The reconstruction $\mathcal{R}$ is such that

$$\mathcal{L}_i \bigcap \mathcal{O} = \mathcal{L}_i \bigcap \mathcal{R},$$

and $\mathcal{R}$ is similar to $\mathcal{O}$.

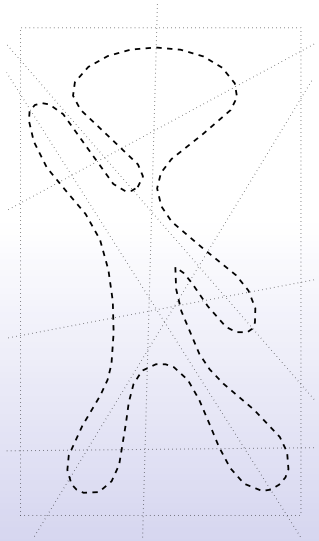


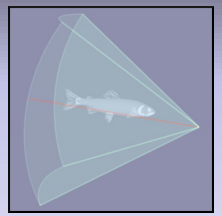
Reconstruction Problem in $\mathbb{R}^2$

Reconstruct object $\mathcal{O}$ intersected by a set of arbitrarily placed lines $\{\mathcal{L}_i\}, i \in [0, n-1]$ in a plane. The reconstruction $\mathcal{R}$ is such that

$$\mathcal{L}_i \bigcap \mathcal{O} = \mathcal{L}_i \bigcap \mathcal{R},$$

and $\mathcal{R}$ is similar to $\mathcal{O}$.



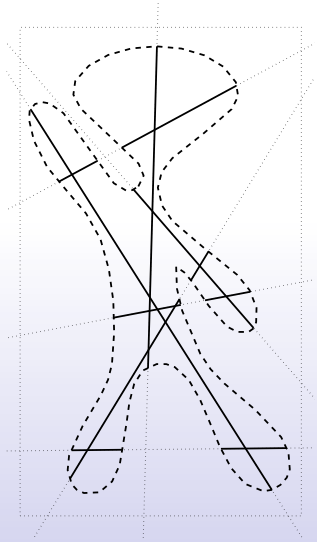


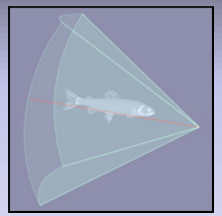
Reconstruction Problem in $\mathbb{R}^2$

Reconstruct object $\mathcal{O}$ intersected by a set of arbitrarily placed lines $\{\mathcal{L}_i\}, i \in [0, n-1]$ in a plane. The reconstruction $\mathcal{R}$ is such that

$$\mathcal{L}_i \bigcap \mathcal{O} = \mathcal{L}_i \bigcap \mathcal{R},$$

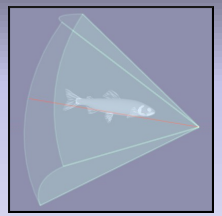
and $\mathcal{R}$ is similar to $\mathcal{O}$.





Multiple Variable Homotopy

A homotopy can be seen as a smooth transition from one map to another. We can extend this definition to multiple maps by defining a (linear) homotopy in multiple variables



Multiple Variable Homotopy

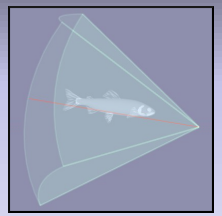
A homotopy can be seen as a smooth transition from one map to another. We can extend this definition to multiple maps by defining a (linear) homotopy in multiple variables

$$\mathcal{H}(\mathbf{x}, \lambda_0, \lambda_1, \cdots, \lambda_{n-1}) = \sum_{k=0}^{n-1} \lambda_k f_k(\mathbf{x}),$$

with

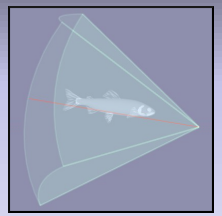
$$\sum_{t=0}^{n-1} \lambda_t = 1.$$

Here, $\{\lambda_k\}$ are n homotopy parameters.



Reconstruction Approach

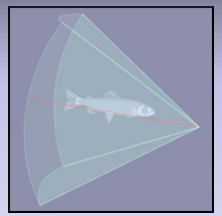
Starting with a set of cross sections $\{\mathcal{S}_{i,j}\}$ for lines $\{\mathcal{L}_i\}$ in a plane, we restrict reconstruction in the bounding box $\mathcal{B}_{box}$ of the cross sections.



Reconstruction Approach

Starting with a set of cross sections $\{\mathcal{S}_{i,j}\}$ for lines $\{\mathcal{L}_i\}$ in a plane, we restrict reconstruction in the bounding box $\mathcal{B}_{box}$ of the cross sections.

The set of lines $\{\mathcal{L}_i\}$ partition $\mathcal{B}_{box}$ into a set of convex polygons $\{\mathcal{G}_k\}, k \in [0, p-1]$.



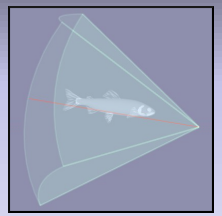
Reconstruction Approach

Starting with a set of cross sections $\{\mathcal{S}_{i,j}\}$ for lines $\{\mathcal{L}_i\}$ in a plane, we restrict reconstruction in the bounding box $\mathcal{B}_{box}$ of the cross sections.

The set of lines $\{\mathcal{L}_i\}$ partition $\mathcal{B}_{box}$ into a set of convex polygons $\{\mathcal{G}_k\}, k \in [0, p-1]$.

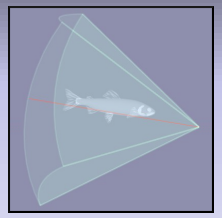
The reconstruction algorithm consists of assigning a homotopy $\mathcal{H}_k$ to every $\mathcal{G}_k$. As before, every line has an associated edge function with it. The reconstruction is then obtained as

$$\mathcal{R} = \bigcup_k \{(x, y) : \mathcal{H}_k = 0\}.$$



Homotopy Setup in a Polygon

The main challenge here is to setup the homotopy $\mathcal{H}_k$ over a polygon $\mathcal{G}_k$, given edge functions of the participating edges.

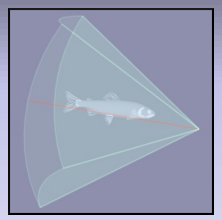


Homotopy Setup in a Polygon

The main challenge here is to setup the homotopy $\mathcal{H}_k$ over a polygon $\mathcal{G}_k$, given edge functions of the participating edges.

The problem can be cast as

> Given a convex polygon with associated smooth functions at its edges, is a consistent and plausible homotopy construction possible?
>
> Such a homotopy must also be at least C^1 at polygon boundaries, given that there is no natural parametrization available among the arbitrarily placed lines.



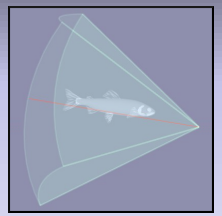
Homotopy Setup in a Polygon

The main challenge here is to setup the homotopy $\mathcal{H}_k$ over a polygon $\mathcal{G}_k$, given edge functions of the participating edges.

The problem can be cast as

> Given a convex polygon with associated smooth functions at its edges, is a consistent and plausible homotopy construction possible?
>
> Such a homotopy must also be at least C^1 at polygon boundaries, given that there is no natural parametrization available among the arbitrarily placed lines.

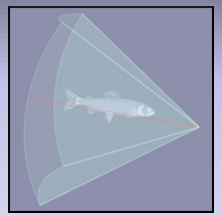
Edge based barycentric coordinates are defined in an attempt to answer this question!



Edge Based Barycentric Coordinates

Barycentric coordinates of a polygon can be considered as homotopy variables because of the two useful properties that they offer

- barycentric coordinates span a complete polygon, and
- partition of unity.



Edge Based Barycentric Coordinates

Barycentric coordinates of a polygon can be considered as homotopy variables because of the two useful properties that they offer

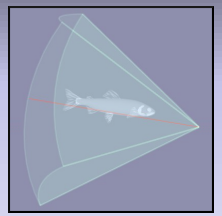
- barycentric coordinates span a complete polygon, and
- partition of unity.

Barycentric coordinates based on orthogonal distance

For an n sided polygon $\mathcal{G}_1$, the barycentric coordinate for an inside point p, corresponding to edge e_i can be written as

$$\lambda_i = \frac{1/\psi(h_i)}{\sum_{j=0}^{n-1} 1/\psi(h_j)}, i \in [0, n-1],$$

where, h_i is orthogonal distance from p to e_i, and $\psi : \mathbb{R} \mapsto \mathbb{R}$ is monotonically increasing.



Edge Based Barycentric Coordinates

Barycentric coordinates of a polygon can be considered as homotopy variables because of the two useful properties that they offer

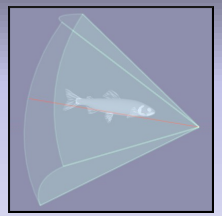
- barycentric coordinates span a complete polygon, and
- partition of unity.

Barycentric coordinates based on stolen areas in a Voronoi diagram

For an n sided polygon $\mathcal{G}_1$, the barycentric coordinate for an inside point p, corresponding to edge e_i can be written as

$$\lambda_i = \frac{\mathcal{A}_i}{\sum_{j=0}^{n-1} \mathcal{A}_j}, i \in [0, n-1],$$

where $\mathcal{A}_i$ is the stolen area for edge e_i and point p in the Voronoi diagram of all edges of $\mathcal{G}_1$ and p.

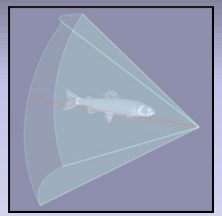


Homotopy Formulation

Homotopy $\mathcal{H}_k$ at p within $\mathcal{G}_k$ can be written as

$$\mathcal{H}_k(p) = \sum_{i=0}^{n_k-1} f_i\left(d_i(p)\right) \lambda_i(p),$$

where $d_i(p)$ is distance along edge e_i.



Homotopy Formulation

Homotopy $\mathcal{H}_k$ at p within $\mathcal{G}_k$ can be written as

$$\mathcal{H}_k(p) = \sum_{i=0}^{n_k-1} f_i\left(d_i(p)\right) \lambda_i(p),$$

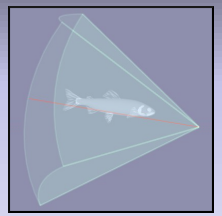
where $d_i(p)$ is distance along edge e_i.

Proposition

The curve $\mathcal{R} = \ker(\mathbf{H})$ generated by the homotopy field $\mathbf{H} = \bigcup \mathcal{H}_k$ is at least C^1 for the edge based barycentric coordinates.

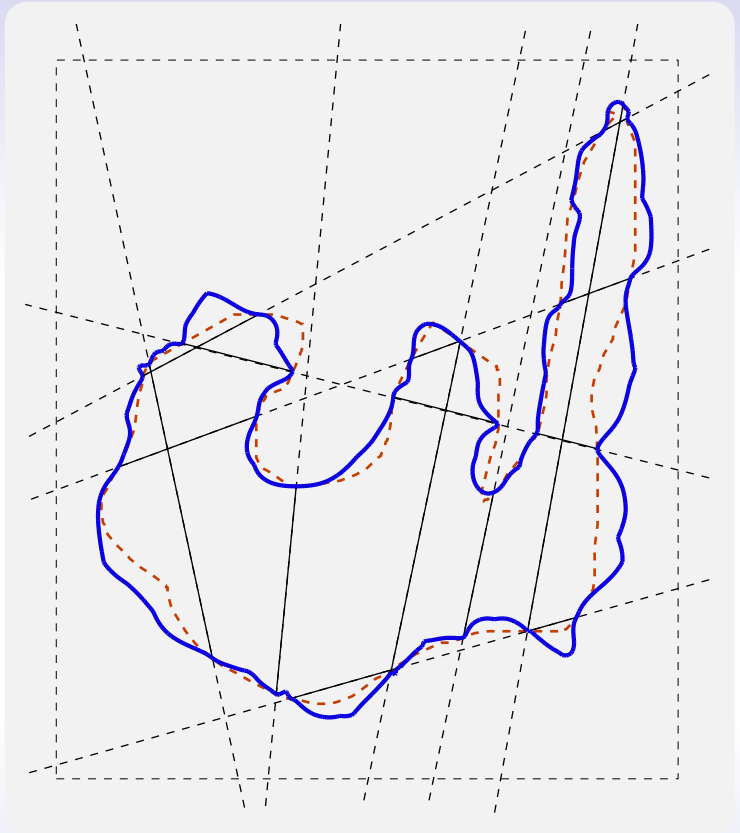
Corollary

At all the intersection points of the lines with boundary of the object, the reconstructed curve $\mathcal{R}$ is orthogonal to the intersecting lines.



Tangent Alignment Results

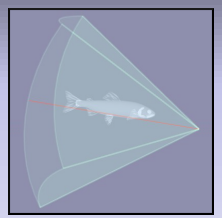
Objects and cutting lines



Reconstruction - Voronoi

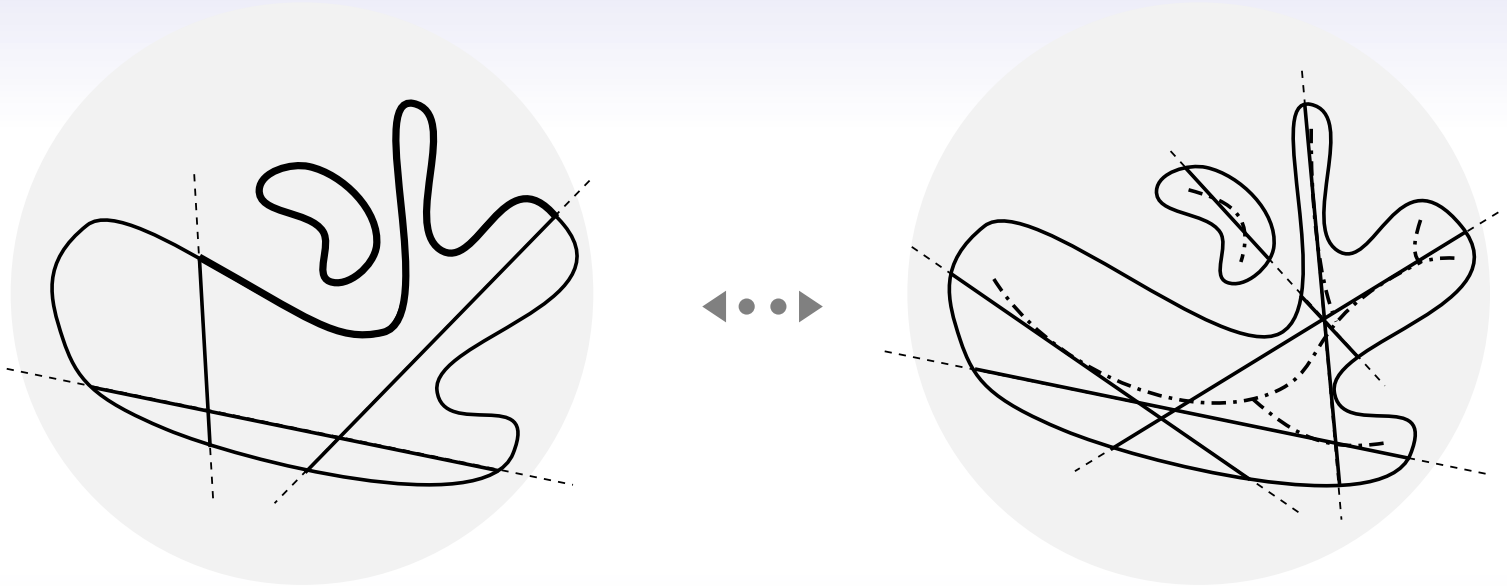
Non-linear ($\eta = 2$)

▶ Compare



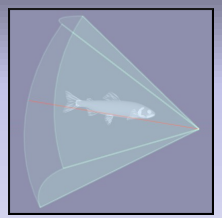
Sampling Condition

Reconstruction accuracy depends on the number of cutting lines and their placement.



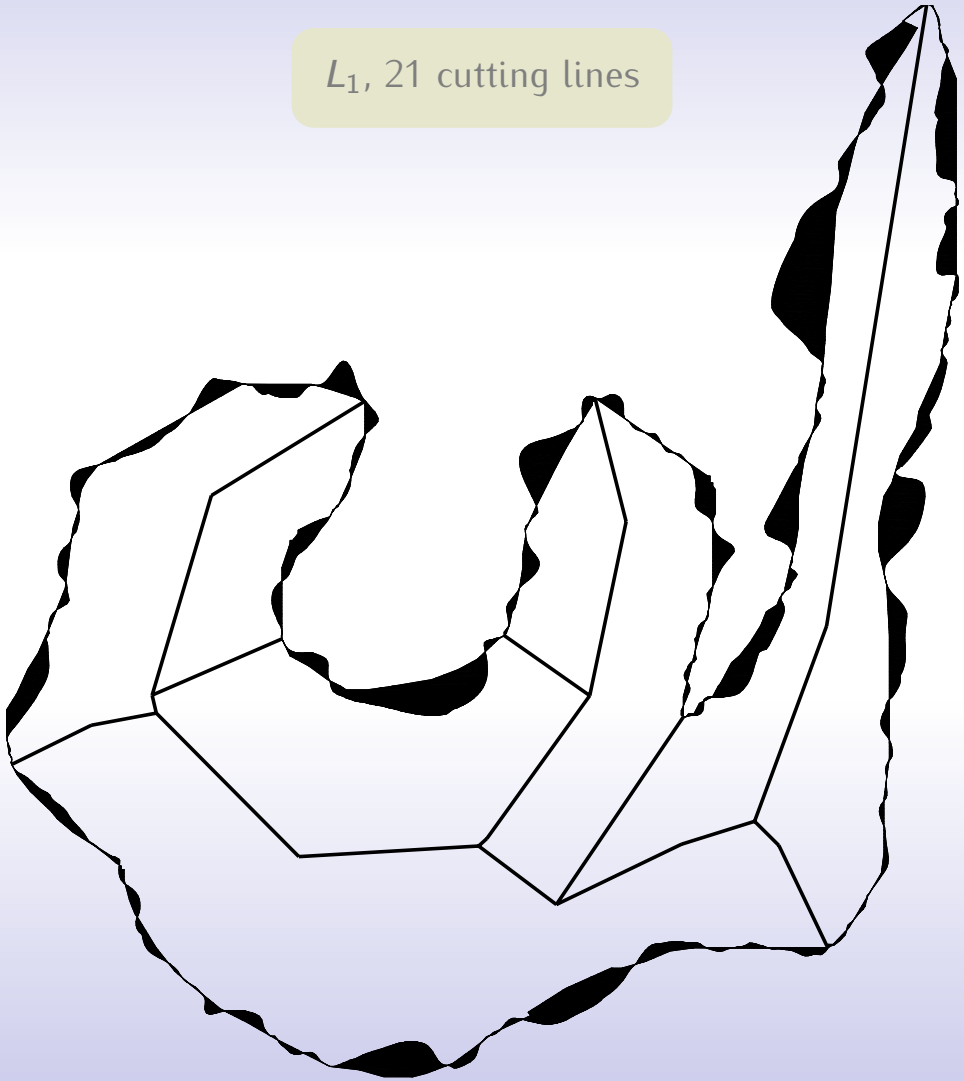
Tortuosity measure τ of a simply connected part of $\mathcal{R}$ in a region $\mathcal{G}_k$ indicates the straightness of the segment.

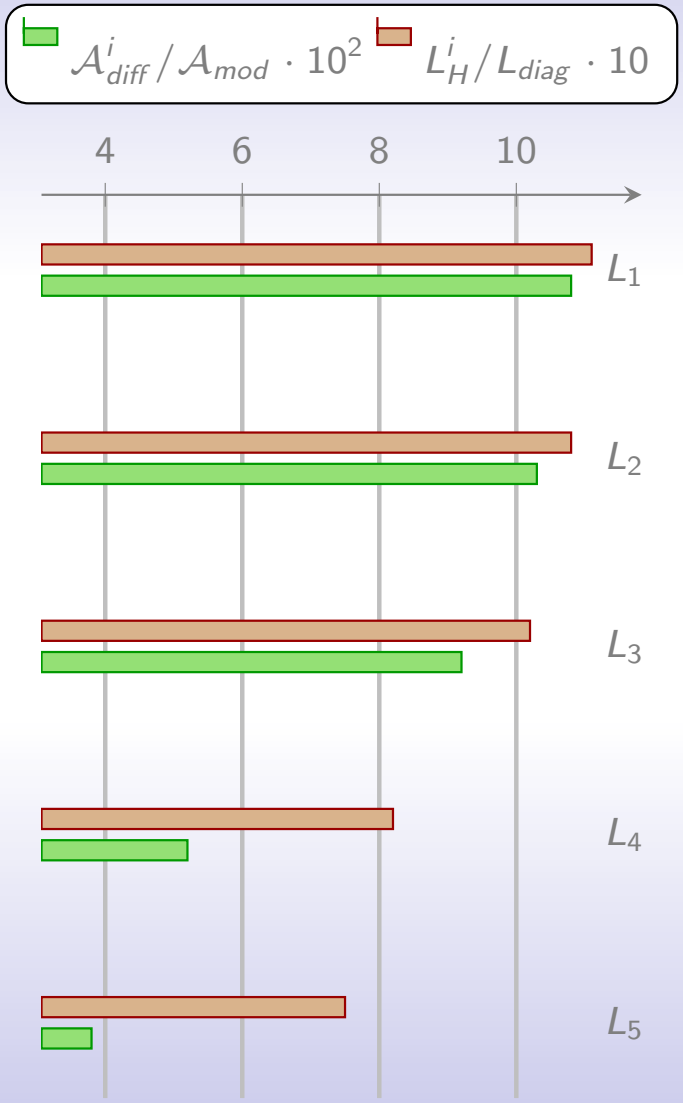
A lower τ ensures that salient features of $\mathcal{O}$ are sampled. One such sampling is done along the *medial axis* of $\mathcal{O}$.

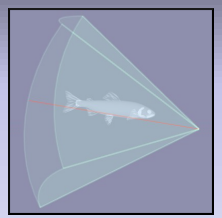


Reconstruction Accuracy

L_1, 21 cutting lines

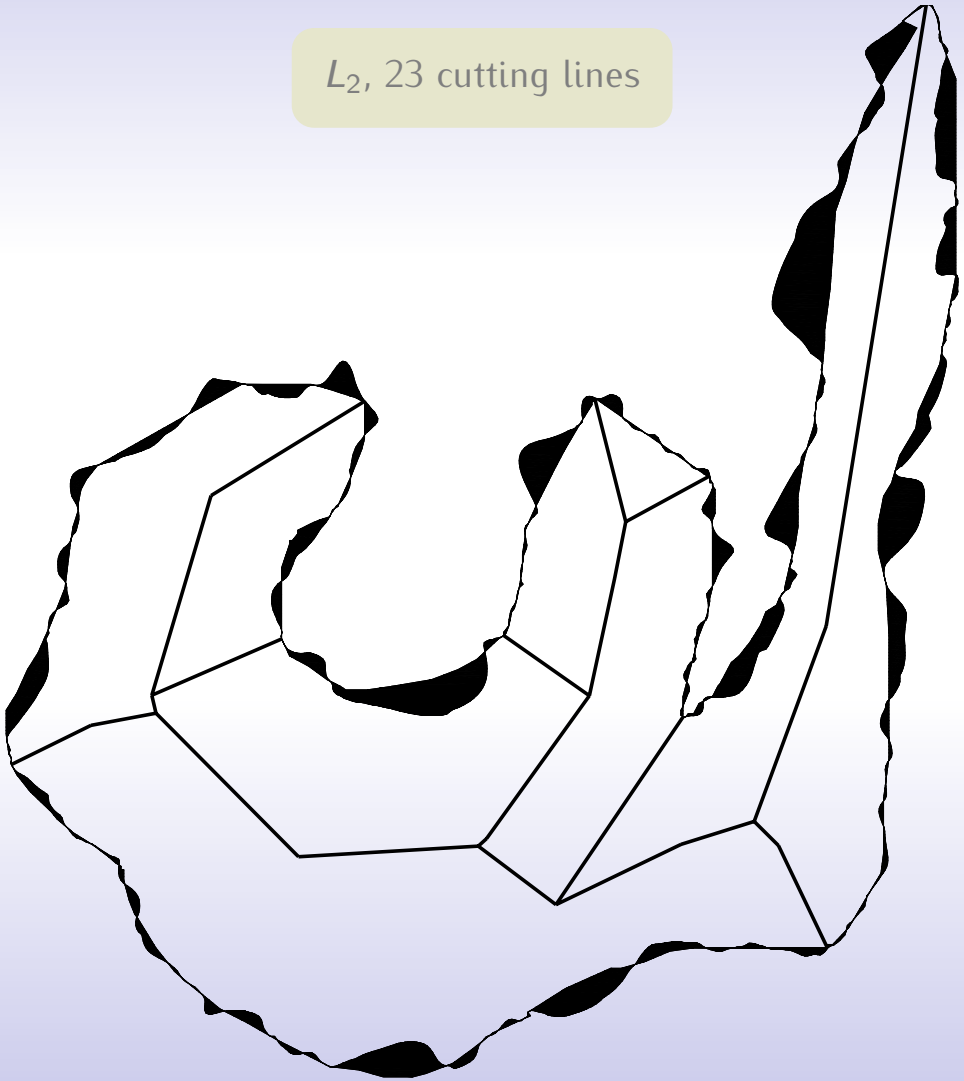


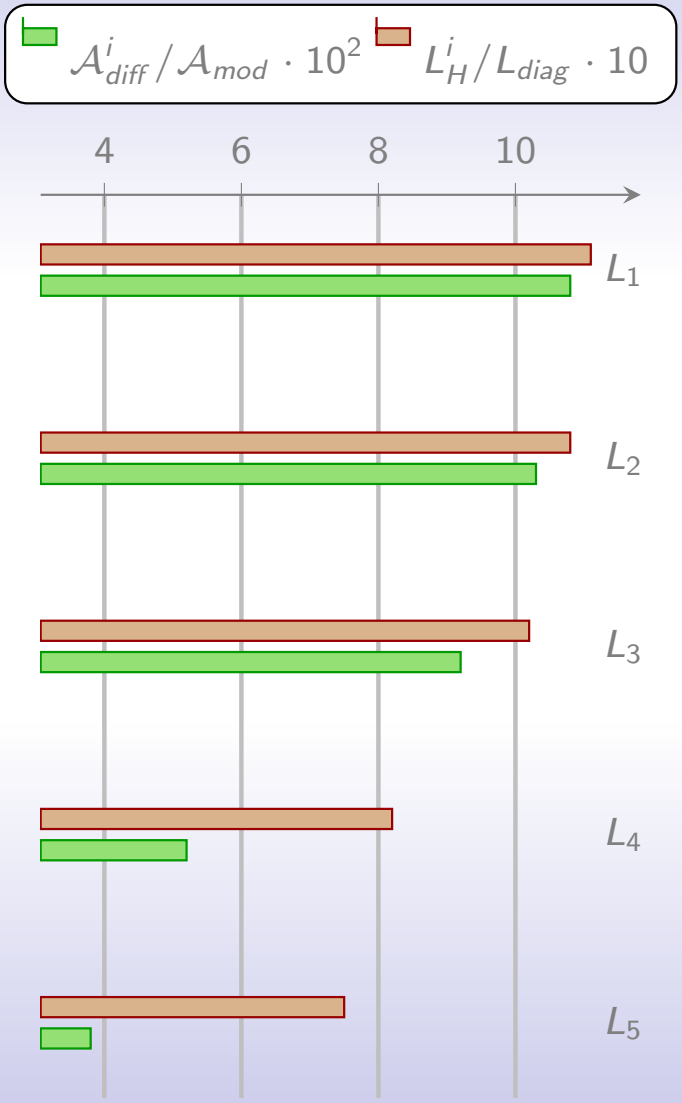


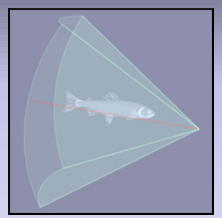


Reconstruction Accuracy

L_2, 23 cutting lines

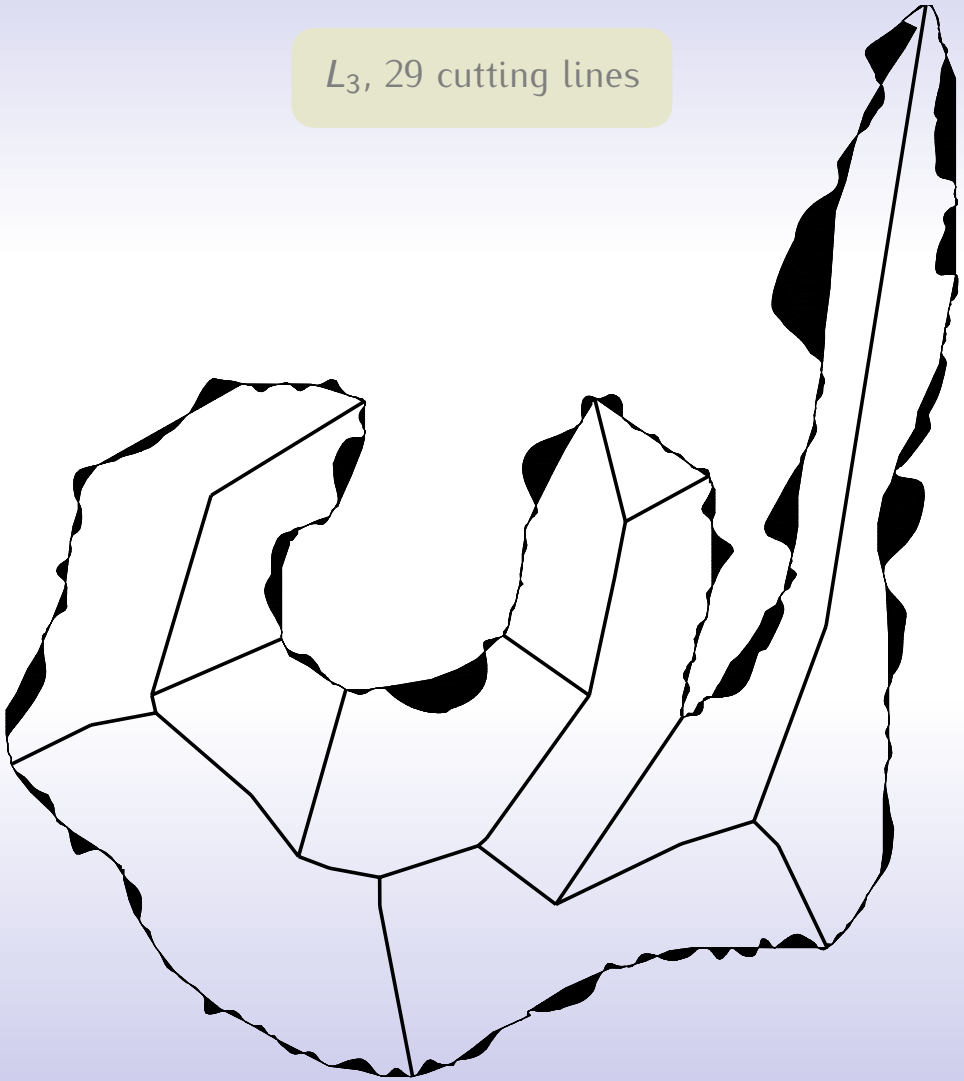


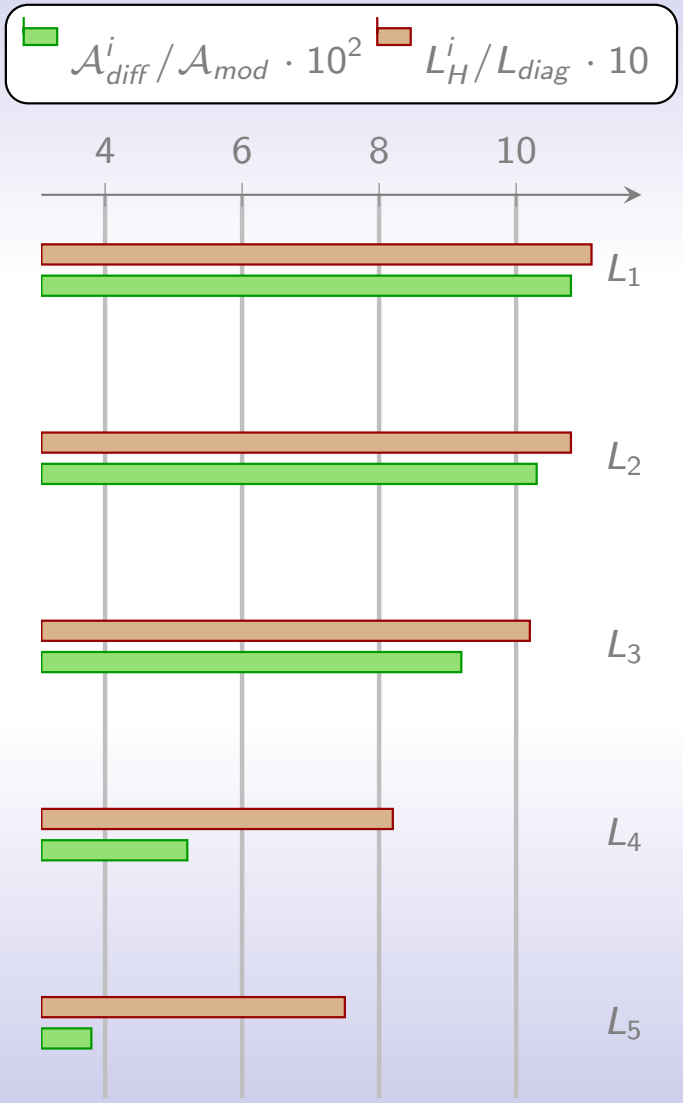


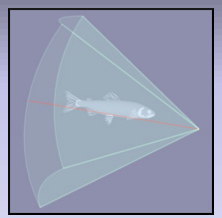


Reconstruction Accuracy

L_3, 29 cutting lines

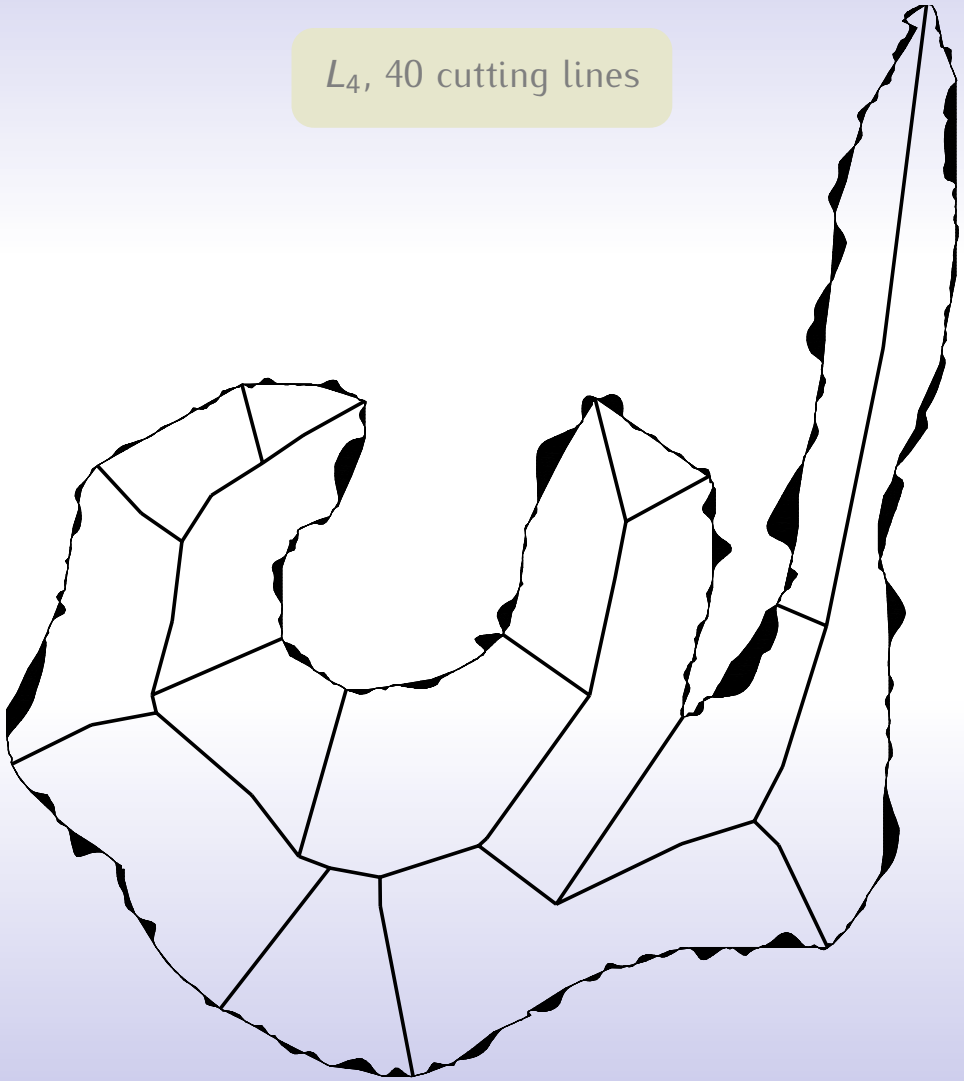


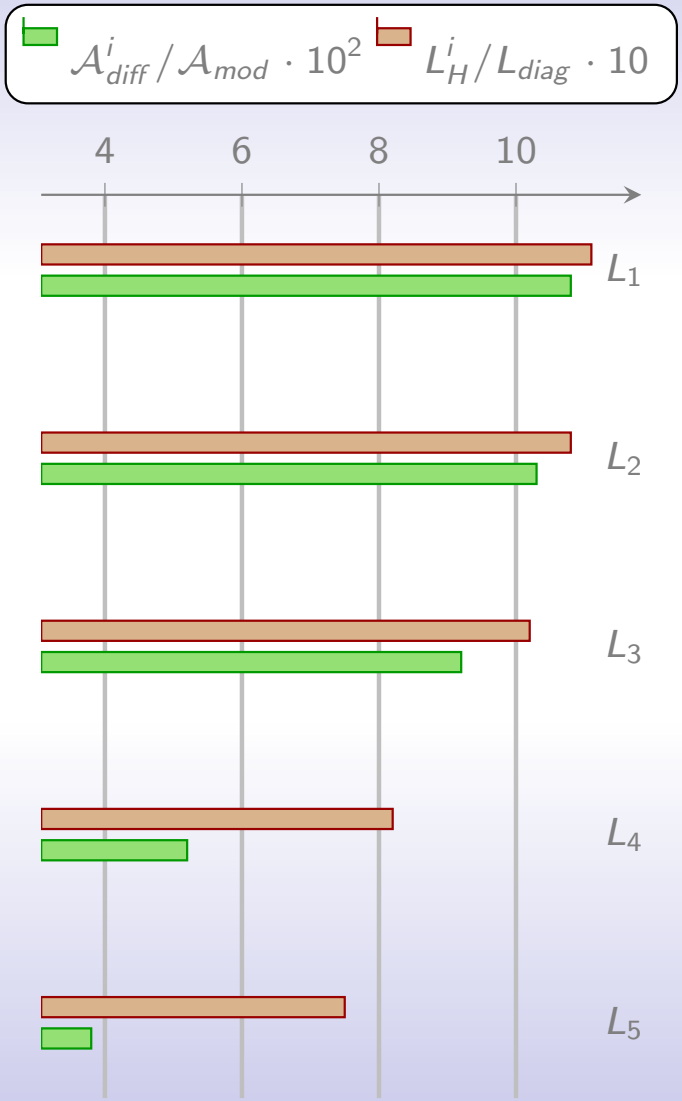


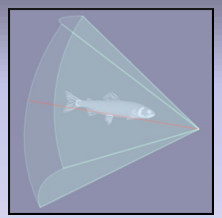


Reconstruction Accuracy

L_4, 40 cutting lines

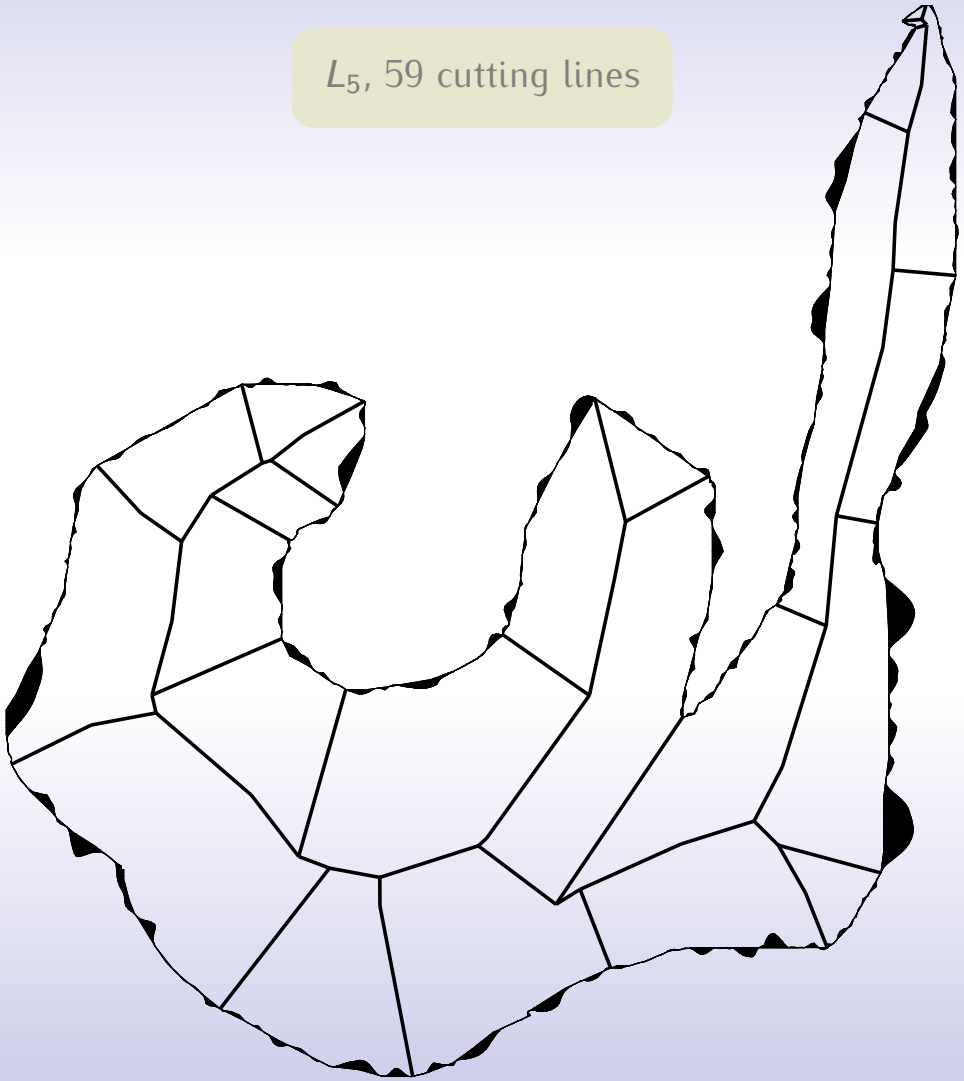


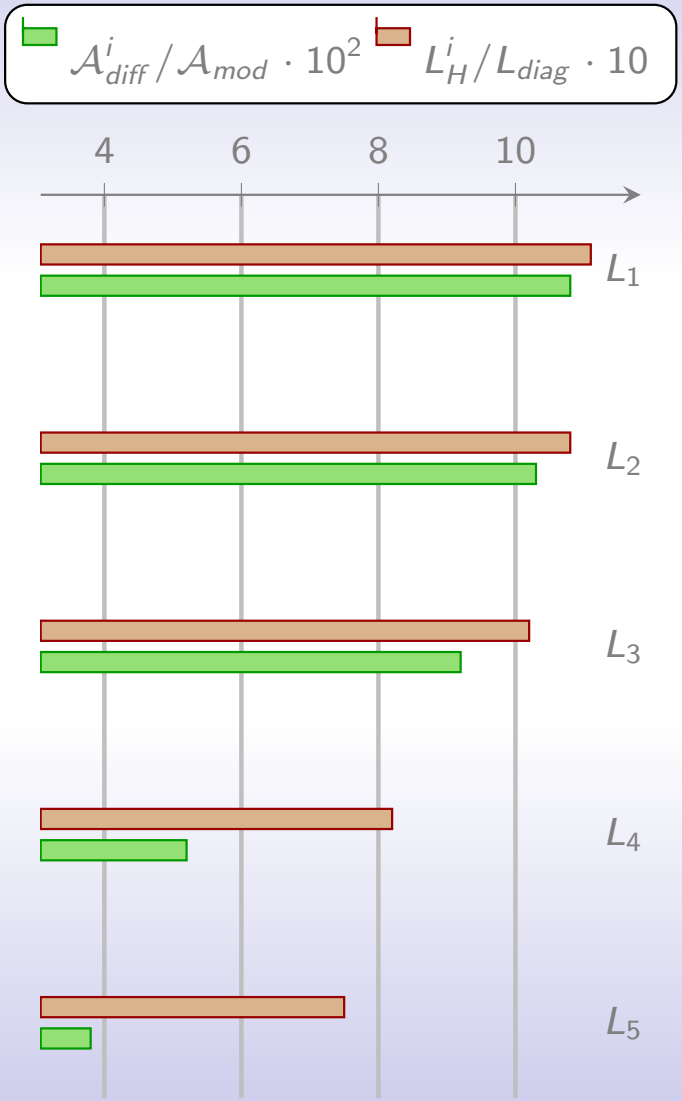


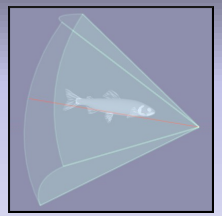


Reconstruction Accuracy

L_5, 59 cutting lines



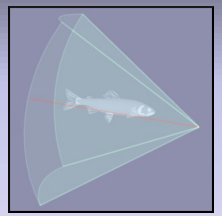




Significance

The discussed research work shows homotopy continuation as an effective mathematical tool to solve a class of reconstruction problems. Following are the salient points of this work

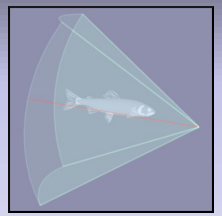
- Level set method used as an effective segmentation machinery. In that context, a fast and multi-phase GPU based solution is developed.



Significance

The discussed research work shows homotopy continuation as an effective mathematical tool to solve a class of reconstruction problems. Following are the salient points of this work

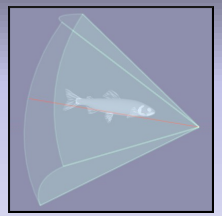
- Level set method used as an effective segmentation machinery. In that context, a fast and multi-phase GPU based solution is developed.
- Continuous deformations are applied to the problem of reconstruction from linear cross sections.



Significance

The discussed research work shows homotopy continuation as an effective mathematical tool to solve a class of reconstruction problems. Following are the salient points of this work

- Level set method used as an effective segmentation machinery. In that context, a fast and multi-phase GPU based solution is developed.
- Continuous deformations are applied to the problem of reconstruction from linear cross sections.
- A generalized and more difficult problem of reconstruction from arbitrarily placed cross sections is solved by means of multi-variable homotopy continuation. Generalized edge-based barycentric coordinates are suggested.



Significance

The discussed research work shows homotopy continuation as an effective mathematical tool to solve a class of reconstruction problems. Following are the salient points of this work

- Level set method used as an effective segmentation machinery. In that context, a fast and multi-phase GPU based solution is developed.
- Continuous deformations are applied to the problem of reconstruction from linear cross sections.
- A generalized and more difficult problem of reconstruction from arbitrarily placed cross sections is solved by means of multi-variable homotopy continuation. Generalized edge-based barycentric coordinates are suggested.
- Applications areas like fisheries, medical science, and computer vision in general are likely to benefit.

Second part: Applications to Computer Graphics through Delaunay graphs of spheres and quadrics

Exact computation of the Voronoi Diagram of spheres in 3D, its topology and its geometric invariants

François Anton Darka Mioc Marcelo Santos

Technical University of Denmark

University of New Brunswick

Outline

- Introduction
- Preliminaries (Voronoi diagrams of 3D spheres)
- Wu's algorithm
- The Voronoi vertex of four spheres
- Delaunay empty circumsphere predicate for spheres
- Conclusions

Motivation

- Generalized Voronoi diagrams, and especially the Voronoi diagram of spheres have not been explored sufficiently [8].
- With recent scientific discoveries in biology and chemistry, Voronoi diagrams of spheres have become more important for representing and analysing the molecular 3D structure and surface [11], the structure of the protein [7], etc.
- A limitation of approximative algorithms for the computation of the Voronoi diagram of spheres is that when approximate computations are performed on objects defined approximately (within some geometric tolerance), the propagation of the errors can be critical, especially if the final computation involves approximate intermediary computations.

Motivation

- Generalized Voronoi diagrams, and especially the Voronoi diagram of spheres have not been explored sufficiently [8].
- With recent scientific discoveries in biology and chemistry, Voronoi diagrams of spheres have become more important for representing and analysing the molecular 3D structure and surface [11], the structure of the protein [7], etc.
- A limitation of approximative algorithms for the computation of the Voronoi diagram of spheres is that when approximate computations are performed on objects defined approximately (within some geometric tolerance), the propagation of the errors can be critical, especially if the final computation involves approximate intermediary computations.

Motivation

- Generalized Voronoi diagrams, and especially the Voronoi diagram of spheres have not been explored sufficiently [8].
- With recent scientific discoveries in biology and chemistry, Voronoi diagrams of spheres have become more important for representing and analysing the molecular 3D structure and surface [11], the structure of the protein [7], etc.
- A limitation of approximative algorithms for the computation of the Voronoi diagram of spheres is that when approximate computations are performed on objects defined approximately (within some geometric tolerance), the propagation of the errors can be critical, especially if the final computation involves approximate intermediary computations.

Previous work (eastern world)

- Nishida and Sugihara [9] and Nishida et al. [10] extended the results in [2] by providing the topological structure of the Voronoi diagram of hyperspheres in $d-$dimensional space using low precision arithmetic ($2d + 4$ times longer bits for exact computation [9]).
- Kim et al. provide several important research contributions in the domain of the Voronoi diagrams of spheres including:
  - the computation of three-dimensional (3D) Voronoi diagrams [6];
  - Euclidean Voronoi diagram of 3D balls and its computation via tracing edges [5];
  - and the Euclidean Voronoi diagrams of 3D spheres and applications to protein structure analysis [7].

Previous work (eastern world)

- Nishida and Sugihara [9] and Nishida et al. [10] extended the results in [2] by providing the topological structure of the Voronoi diagram of hyperspheres in $d-$dimensional space using low precision arithmetic ($2d + 4$ times longer bits for exact computation [9]).
- Kim et al. provide several important research contributions in the domain of the Voronoi diagrams of spheres including:
  - the computation of three-dimensional (3D) Voronoi diagrams [6];
  - Euclidean Voronoi diagram of 3D balls and its computation via tracing edges [5];
  - and the Euclidean Voronoi diagrams of 3D spheres and applications to protein structure analysis [7].

Previous work (western world)

- Will [12] provides a method for the computation of additively weighted Voronoi Cells for applications in molecullar Biology, based on general methods for computing lower envelopes of algebraic surfaces [12]. In his algorithm he has to maintain three kinds of conflicts associated with the vertices, edge fragments and face fragments.
- Gavrilova early work on generalised Voronoi diagrams in her doctoral thesis [3] and subsequent work by Gavrilova and Rokne on topology updating of the kinematic Voronoi diagram of hyperspheres [2].
- Hanniel and Elber [4] provide an algorithm for computation of the Voronoi diagrams for planes, spheres and cylinders in $\mathbb{R}^3$ (lower envelope of the bisector surfaces similar to the algorithm of Will [12]).

Previous work (western world)

- Will [12] provides a method for the computation of additively weighted Voronoi Cells for applications in molecullar Biology, based on general methods for computing lower envelopes of algebraic surfaces [12]. In his algorithm he has to maintain three kinds of conflicts associated with the vertices, edge fragments and face fragments.
- Gavrilova early work on generalised Voronoi diagrams in her doctoral thesis [3] and subsequent work by Gavrilova and Rokne on topology updating of the kinematic Voronoi diagram of hyperspheres [2].
- Hanniel and Elber [4] provide an algorithm for computation of the Voronoi diagrams for planes, spheres and cylinders in $\mathbb{R}^3$ (lower envelope of the bisector surfaces similar to the algorithm of Will [12]).

Previous work (western world)

- Will [12] provides a method for the computation of additively weighted Voronoi Cells for applications in molecullar Biology, based on general methods for computing lower envelopes of algebraic surfaces [12]. In his algorithm he has to maintain three kinds of conflicts associated with the vertices, edge fragments and face fragments.
- Gavrilova early work on generalised Voronoi diagrams in her doctoral thesis [3] and subsequent work by Gavrilova and Rokne on topology updating of the kinematic Voronoi diagram of hyperspheres [2].
- Hanniel and Elber [4] provide an algorithm for computation of the Voronoi diagrams for planes, spheres and cylinders in $\mathbb{R}^3$ (lower envelope of the bisector surfaces similar to the algorithm of Will [12]).

Problem addressed in this research

- Current research efforts did not provide a symbolic computation method for the Voronoi Diagram and Delaunay graph (or quasi-triangulation) of spheres based on their geometric invariants.
- Anton and Mioc have provided an exact method for the computation of the Voronoi diagram of circles [1] using Gröbner basis and invariants.
- This paper provides a generalisation to the three-dimensional case using a much more powerful (working with differential polynomials) and tractable method: Wu's method [13].

Problem addressed in this research

- Current research efforts did not provide a symbolic computation method for the Voronoi Diagram and Delaunay graph (or quasi-triangulation) of spheres based on their geometric invariants.
- Anton and Mioc have provided an exact method for the computation of the Voronoi diagram of circles [1] using Gröbner basis and invariants.
- This paper provides a generalisation to the three-dimensional case using a much more powerful (working with differential polynomials) and tractable method: Wu's method [13].

Problem addressed in this research

- Current research efforts did not provide a symbolic computation method for the Voronoi Diagram and Delaunay graph (or quasi-triangulation) of spheres based on their geometric invariants.
- Anton and Mioc have provided an exact method for the computation of the Voronoi diagram of circles [1] using Gröbner basis and invariants.
- This paper provides a generalisation to the three-dimensional case using a much more powerful (working with differential polynomials) and tractable method: Wu's method [13].

Voronoi diagrams of spheres

Let $M = \mathbb{R}^3$, and δ denote the Euclidean distance between points. Let $\mathcal{S} = \{s_1, ..., s_m\} \subset M$ be a set of $m \geq 2$ spheres. The distance between $x \in M$ and a sphere s_i is $d(x, s_i) = \inf_{y \in s_i} \{\delta(x, y)\}$.

Definition

(Influence zone) For $s_i, s_j \in \mathcal{S}, s_i \neq s_j$, the *influence zone* $D(s_i, s_j)$ of s_i with respect to s_j is: $D(s_i, s_j) = \{x \in M | d(x, s_i) < d(x, s_j)\}$.

Definition

(Voronoi region) The *Voronoi region* $V(s_i, \mathcal{S})$ of $s_i \in \mathcal{S}$ with respect to the set $\mathcal{S}$ is: $V(s_i, \mathcal{S}) = \bigcap_{s_j \in \mathcal{S}, s_j \neq s_i} D(s_i, s_j)$.

Definition

(Voronoi diagram) The *Voronoi diagram* of $\mathcal{S}$ is the union $V(\mathcal{S}) = \bigcup_{s_i \in \mathcal{S}} \partial V(s_i, \mathcal{S})$ of all region boundaries.

Voronoi diagrams of spheres

Let $M = \mathbb{R}^3$, and δ denote the Euclidean distance between points. Let $\mathcal{S} = \{s_1, ..., s_m\} \subset M$ be a set of $m \geq 2$ spheres. The distance between $x \in M$ and a sphere s_i is $d(x, s_i) = \inf_{y \in s_i} \{\delta(x, y)\}$.

Definition

(Influence zone) For $s_i, s_j \in \mathcal{S}, s_i \neq s_j$, the *influence zone* $D(s_i, s_j)$ of s_i with respect to s_j is: $D(s_i, s_j) = \{x \in M | d(x, s_i) < d(x, s_j)\}$.

Definition

(Voronoi region) The *Voronoi region* $V(s_i, \mathcal{S})$ of $s_i \in \mathcal{S}$ with respect to the set $\mathcal{S}$ is: $V(s_i, \mathcal{S}) = \bigcap_{s_j \in \mathcal{S}, s_j \neq s_i} D(s_i, s_j)$.

Definition

(Voronoi diagram) The *Voronoi diagram* of $\mathcal{S}$ is the union $V(\mathcal{S}) = \bigcup_{s_i \in \mathcal{S}} \partial V(s_i, \mathcal{S})$ of all region boundaries.

Voronoi diagrams of spheres

Let $M = \mathbb{R}^3$, and δ denote the Euclidean distance between points. Let $\mathcal{S} = \{s_1, ..., s_m\} \subset M$ be a set of $m \geq 2$ spheres. The distance between $x \in M$ and a sphere s_i is $d(x, s_i) = \inf_{y \in s_i} \{\delta(x, y)\}$.

Definition

(Influence zone) For $s_i, s_j \in \mathcal{S}, s_i \neq s_j$, the *influence zone* $D(s_i, s_j)$ of s_i with respect to s_j is: $D(s_i, s_j) = \{x \in M | d(x, s_i) < d(x, s_j)\}$.

Definition

(Voronoi region) The *Voronoi region* $V(s_i, \mathcal{S})$ of $s_i \in \mathcal{S}$ with respect to the set $\mathcal{S}$ is: $V(s_i, \mathcal{S}) = \bigcap_{s_j \in \mathcal{S}, s_j \neq s_i} D(s_i, s_j)$.

Definition

(Voronoi diagram) The *Voronoi diagram* of $\mathcal{S}$ is the union $V(\mathcal{S}) = \bigcup_{s_i \in \mathcal{S}} \partial V(s_i, \mathcal{S})$ of all region boundaries.

Voronoi diagrams of spheres

Let $M = \mathbb{R}^3$, and δ denote the Euclidean distance between points. Let $\mathcal{S} = \{s_1, ..., s_m\} \subset M$ be a set of $m \geq 2$ spheres. The distance between $x \in M$ and a sphere s_i is $d(x, s_i) = \inf_{y \in s_i} \{\delta(x, y)\}$.

Definition

(Influence zone) For $s_i, s_j \in \mathcal{S}, s_i \neq s_j$, the *influence zone* $D(s_i, s_j)$ of s_i with respect to s_j is: $D(s_i, s_j) = \{x \in M | d(x, s_i) < d(x, s_j)\}$.

Definition

(Voronoi region) The *Voronoi region* $V(s_i, \mathcal{S})$ of $s_i \in \mathcal{S}$ with respect to the set $\mathcal{S}$ is: $V(s_i, \mathcal{S}) = \bigcap_{s_j \in \mathcal{S}, s_j \neq s_i} D(s_i, s_j)$.

Definition

(Voronoi diagram) The *Voronoi diagram* of $\mathcal{S}$ is the union $V(\mathcal{S}) = \bigcup_{s_i \in \mathcal{S}} \partial V(s_i, \mathcal{S})$ of all region boundaries.

Delaunay graph of spheres

Definition

(Delaunay graph) The *Delaunay graph* $DG(\mathcal{S})$ of $\mathcal{S}$ is the dual graph of $V(\mathcal{S})$ defined as follows:

- the set of vertices of $DG(\mathcal{S})$ is $\mathcal{S}$,
- for each $2-$dimensional facet of $V(\mathcal{S})$ that belongs to the common boundary of $V(s_i, \mathcal{S})$ and of $V(s_j, \mathcal{S})$ with $s_i, s_j \in \mathcal{S}$ and $s_i \neq s_j$, there is an edge of $DG(\mathcal{S})$ between s_i and s_j and reciprocally, and
- for each vertex of $V(\mathcal{S})$ that belongs to the common boundary of $V(s_{i_1}, \mathcal{S}), \ldots, V(s_{i_5}, \mathcal{S})$, with $\forall k \in \{1, \ldots, 5\}, s_{i_k} \in \mathcal{S}$ all distinct, there exists a complete graph K_5 between the $s_{i_k}, k \in \{1, \ldots, 5\}$, and reciprocally.

Definitions with polynomials

Definition

(from [13]) For any set of polynomials $\mathbb{P} \subset \mathcal{K}[\mathbb{X}]$, $Zero(\mathcal{P}) = \{x \in \mathcal{E}^n | P(x) = 0, \forall P \in \mathbb{P}\}$ is called a *variety*. For a set of polynomials $\mathbb{P}$ and a polynomial D, we define $Zero(\mathbb{P}/D) = Zero(\mathbb{P}) \setminus Zero(\{D\})$, called a *quasi-algebraic variety*.

Definition

(from [13]) Let $P \in k[x]$ be a polynomial. The *class* of P, denoted by $cls(P)$ is the c such that x_c is the largest variable that occurs in P. If $cls(P) = c$, then x_c is called the *leading variable* and denoted by $lvar(P)$, the highest degree monomial of P as a univariate polynomial in $lvar(P)$ is called the *leading monomial*, and its coefficient is called the *initial* of P.

Definitions with polynomials

Definition

(from [13]) For any set of polynomials $\mathbb{P} \subset \mathcal{K}[\mathbb{X}]$, $Zero(\mathcal{P}) = \{x \in \mathcal{E}^n | P(x) = 0, \forall P \in \mathbb{P}\}$ is called a *variety*. For a set of polynomials $\mathbb{P}$ and a polynomial D, we define $Zero(\mathbb{P}/D) = Zero(\mathbb{P}) \setminus Zero(\{D\})$, called a *quasi-algebraic variety*.

Definition

(from [13]) Let $P \in k[x]$ be a polynomial. The *class* of P, denoted by $cls(P)$ is the c such that x_c is the largest variable that occurs in P. If $cls(P) = c$, then x_c is called the *leading variable* and denoted by $lvar(P)$, the highest degree monomial of P as a univariate polynomial in $lvar(P)$ is called the *leading monomial*, and its coefficient is called the *initial* of P.

Triangular sets

Definition

(from [13]) A polynomial P_1 has higher *ordering* than a polynomial P_2, denoted as $P_2 \prec P_1$, if either $cls(P_1) > cls(P_2)$, or $c = cls(P_1) = cls(P_2)$ and $deg(P_1, x_c) > deg(P_2, x_c)$. If none of two polynomials has higher ordering than the other, they are said to have the same rank, denoted as $P_1 \sim P_2$.

Definition

(from [13]) A polynomial Q is *reduced* with respect to P, if $cls(P) = c > 0$ and $deg(Q, x_c) < deg(P, x_c)$. A sequence of non-zero polynomials $\mathcal{A} : A_1, A_2, \cdots, A_r$ is a *triangular set* if either $r = 1$ or $cls(A_1) < cls(A_2) < \cdots < cls(A_r)$. A triangular set is called an *ascending chain*, or simply a chain, if A_j is reduced with respect to A_i for $i < j$. For a chain $\mathcal{A}$, we denote $\mathbb{I}_{\mathcal{A}}$ as the product of the initials of the polynomials in $\mathcal{A}$.

Triangular sets

Definition

(from [13]) A polynomial P_1 has higher *ordering* than a polynomial P_2, denoted as $P_2 \prec P_1$, if either $cls(P_1) > cls(P_2)$, or $c = cls(P_1) = cls(P_2)$ and $deg(P_1, x_c) > deg(P_2, x_c)$. If none of two polynomials has higher ordering than the other, they are said to have the same rank, denoted as $P_1 \sim P_2$.

Definition

(from [13]) A polynomial Q is *reduced* with respect to P, if $cls(P) = c > 0$ and $deg(Q, x_c) < deg(P, x_c)$. A sequence of non-zero polynomials $\mathcal{A} : A_1, A_2, \cdots, A_r$ is a *triangular set* if either $r = 1$ or $cls(A_1) < cls(A_2) < \cdots < cls(A_r)$. A triangular set is called an *ascending chain*, or simply a chain, if A_j is reduced with respect to A_i for $i < j$. For a chain $\mathcal{A}$, we denote $\mathbb{I}_{\mathcal{A}}$ as the product of the initials of the polynomials in $\mathcal{A}$.

Triangular sets

Definition

(from [13]) A polynomial P_1 has higher *ordering* than a polynomial P_2, denoted as $P_2 \prec P_1$, if either $cls(P_1) > cls(P_2)$, or $c = cls(P_1) = cls(P_2)$ and $deg(P_1, x_c) > deg(P_2, x_c)$. If none of two polynomials has higher ordering than the other, they are said to have the same rank, denoted as $P_1 \sim P_2$.

Definition

(from [13]) A polynomial Q is *reduced* with respect to P, if $cls(P) = c > 0$ and $deg(Q, x_c) < deg(P, x_c)$. A sequence of non-zero polynomials $\mathcal{A} : A_1, A_2, \cdots, A_r$ is a *triangular set* if either $r = 1$ or $cls(A_1) < cls(A_2) < \cdots < cls(A_r)$. A triangular set is called an *ascending chain*, or simply a chain, if A_j is reduced with respect to A_i for $i < j$. For a chain $\mathcal{A}$, we denote $\mathbb{I}_{\mathcal{A}}$ as the product of the initials of the polynomials in $\mathcal{A}$.

Basic sets

Lemma

(from [13]) A sequence of (ascending) chains steadily lower in ordering is finite.

Definition

(from [13]) A *basic set* of a polynomial set $\mathbb{P}$ is any chain of lowest ordering contained in $\mathbb{P}$. A polynomial Q is called *reduced* with respect to a chain $\mathcal{A}$ if Q is reduced with respect to all the polynomials in $\mathcal{A}$.

Lemma

(from [13]) Let $\mathcal{A}$ be a basic set of a polynomial set $\mathbb{P}$. If P is reduced with respect to $\mathcal{A}$, then a basic set of $\mathbb{P} \cup P$ is of lower ordering than that of $\mathbb{P}$.

Basic sets

Lemma

(from [13]) A sequence of (ascending) chains steadily lower in ordering is finite.

Definition

(from [13]) A *basic set* of a polynomial set $\mathbb{P}$ is any chain of lowest ordering contained in $\mathbb{P}$. A polynomial Q is called *reduced* with respect to a chain $\mathcal{A}$ if Q is reduced with respect to all the polynomials in $\mathcal{A}$.

Lemma

(from [13]) Let $\mathcal{A}$ be a basic set of a polynomial set $\mathbb{P}$. If P is reduced with respect to $\mathcal{A}$, then a basic set of $\mathbb{P} \cup P$ is of lower ordering than that of $\mathbb{P}$.

Basic sets

Lemma

(from [13]) A sequence of (ascending) chains steadily lower in ordering is finite.

Definition

(from [13]) A *basic set* of a polynomial set $\mathbb{P}$ is any chain of lowest ordering contained in $\mathbb{P}$. A polynomial Q is called *reduced* with respect to a chain $\mathcal{A}$ if Q is reduced with respect to all the polynomials in $\mathcal{A}$.

Lemma

(from [13]) Let $\mathcal{A}$ be a basic set of a polynomial set $\mathbb{P}$. If P is reduced with respect to $\mathcal{A}$, then a basic set of $\mathbb{P} \cup P$ is of lower ordering than that of $\mathbb{P}$.

Euclidean division of polynomials

Let F and G be non-zero polynomials with $c = cls(F)$ and $I = init(F)$.

- Either G is reduced with respect to F, or $deg(G, x_c) \geq deg(F, x_c)$, and then it is possible to divide G by F as univariate polynomials in x_c.
- Indeed, let $k = deg(G, x_c) - deg(F, x_c)$, $k' = deg(G, x_c)$, and I' be the coefficient of $x_c^{k'}$ in G, then $deg\left(IG - I'x_c^k F\right) < k'$.
- Therefore in a finite number of steps $s \leq k + 1$, we get that $I^s G = QF + R$ where Q and R are polynomials in $\mathcal{K}[\mathbb{X}]$ with R reduced with respect to F. R is uniquely determined and called the *reminder* of G with respect to F and denoted as $R = prem(G, F)$.
- *Division formula*: $JG = \sum_i Q_i A_i + R$, where R is reduced with respect to $\mathcal{A}$, and R is called the reminder of G with respect to $\mathcal{A}$, and denoted as $R = prem(G, \mathcal{A})$.

Euclidean division of polynomials

Let F and G be non-zero polynomials with $c = cls(F)$ and $I = init(F)$.

- Either G is reduced with respect to F, or $deg(G, x_c) \geq deg(F, x_c)$, and then it is possible to divide G by F as univariate polynomials in x_c.
- Indeed, let $k = deg(G, x_c) - deg(F, x_c)$, $k' = deg(G, x_c)$, and I' be the coefficient of $x_c^{k'}$ in G, then $deg\left(IG - I'x_c^k F\right) < k'$.
- Therefore in a finite number of steps $s \leq k + 1$, we get that $I^s G = QF + R$ where Q and R are polynomials in $\mathcal{K}[\mathbb{X}]$ with R reduced with respect to F. R is uniquely determined and called the *reminder* of G with respect to F and denoted as $R = prem(G, F)$.
- *Division formula*: $JG = \sum_i Q_i A_i + R$, where R is reduced with respect to $\mathcal{A}$, and R is called the reminder of G with respect to $\mathcal{A}$, and denoted as $R = prem(G, \mathcal{A})$.

Euclidean division of polynomials

Let F and G be non-zero polynomials with $c = cls(F)$ and $I = init(F)$.

- Either G is reduced with respect to F, or $deg(G, x_c) \geq deg(F, x_c)$, and then it is possible to divide G by F as univariate polynomials in x_c.
- Indeed, let $k = deg(G, x_c) - deg(F, x_c)$, $k' = deg(G, x_c)$, and I' be the coefficient of $x_c^{k'}$ in G, then $deg\left(IG - I'x_c^k F\right) < k'$.
- Therefore in a finite number of steps $s \leq k + 1$, we get that $I^s G = QF + R$ where Q and R are polynomials in $\mathcal{K}[\mathbb{X}]$ with R reduced with respect to F. R is uniquely determined and called the *reminder* of G with respect to F and denoted as $R = prem(G, F)$.
- *Division formula*: $JG = \sum_i Q_i A_i + R$, where R is reduced with respect to $\mathcal{A}$, and R is called the reminder of G with respect to $\mathcal{A}$, and denoted as $R = prem(G, \mathcal{A})$.

Euclidean division of polynomials

Let F and G be non-zero polynomials with $c = cls(F)$ and $I = init(F)$.

- Either G is reduced with respect to F, or $deg(G, x_c) \geq deg(F, x_c)$, and then it is possible to divide G by F as univariate polynomials in x_c.
- Indeed, let $k = deg(G, x_c) - deg(F, x_c)$, $k' = deg(G, x_c)$, and I' be the coefficient of $x_c^{k'}$ in G, then $deg\left(IG - I'x_c^k F\right) < k'$.
- Therefore in a finite number of steps $s \leq k + 1$, we get that $I^s G = QF + R$ where Q and R are polynomials in $\mathcal{K}[\mathbb{X}]$ with R reduced with respect to F. R is uniquely determined and called the *reminder* of G with respect to F and denoted as $R = prem(G, F)$.
- *Division formula*: $JG = \sum_i Q_i A_i + R$, where R is reduced with respect to $\mathcal{A}$, and R is called the reminder of G with respect to $\mathcal{A}$, and denoted as $R = prem(G, \mathcal{A})$.

Euclidean division of polynomials

Let F and G be non-zero polynomials with $c = cls(F)$ and $I = init(F)$.

- Either G is reduced with respect to F, or $deg(G, x_c) \geq deg(F, x_c)$, and then it is possible to divide G by F as univariate polynomials in x_c.
- Indeed, let $k = deg(G, x_c) - deg(F, x_c)$, $k' = deg(G, x_c)$, and I' be the coefficient of $x_c^{k'}$ in G, then $deg\left(IG - I'x_c^k F\right) < k'$.
- Therefore in a finite number of steps $s \leq k + 1$, we get that $I^s G = QF + R$ where Q and R are polynomials in $\mathcal{K}[\mathbb{X}]$ with R reduced with respect to F. R is uniquely determined and called the *reminder* of G with respect to F and denoted as $R = prem(G, F)$.
- *Division formula*: $JG = \sum_i Q_i A_i + R$, where R is reduced with respect to $\mathcal{A}$, and R is called the reminder of G with respect to $\mathcal{A}$, and denoted as $R = prem(G, \mathcal{A})$ [13].

Wu's algorithm

Starting with a polynomial set $\mathbb{P}_0 = \mathbb{P}$, one should:

- select a basis $\mathcal{B}_0$ of $\mathbb{P}_0$,
- compute the set $\mathbb{R}_0$ of non-zero reminders of polynomials of $\mathbb{P}_0 \setminus \mathcal{B}_0$ with respect to $\mathcal{B}_0$,
- then, let $\mathbb{P}_1 = \mathbb{P}_0 \cup \mathbb{R}_0$ and one should compute a basis set $\mathbb{B}_1$ in $\mathbb{P}_1$,
- then, one should compute the set $\mathbb{R}_1$ of non-zero reminders of polynomials of $\mathbb{P}_1 \setminus \mathcal{B}_1$ with respect to $\mathcal{B}_1$.
- By Lemma 3.8, $\mathcal{B}_1$ is of lower ordering than $\mathcal{B}_0$. Therefore, such a process has a finite number of steps, and the final result is a basic set $\mathcal{B}_m = \mathcal{C}$, such that the corresponding set of non-zero reminders $\mathbb{R}_m$ is the empty set and $prem(\mathbb{P}, \mathcal{C}) = \{0\}$. Thus, for each chain that can be obtained in such a way, $Zero(\mathbb{P}) \subseteq Zero(\mathcal{C})$.

Wu's algorithm

Starting with a polynomial set $\mathbb{P}_0 = \mathbb{P}$, one should:

- select a basis $\mathcal{B}_0$ of $\mathbb{P}_0$,
- compute the set $\mathbb{R}_0$ of non-zero reminders of polynomials of $\mathbb{P}_0 \setminus \mathcal{B}_0$ with respect to $\mathcal{B}_0$,
- then, let $\mathbb{P}_1 = \mathbb{P}_0 \cup \mathbb{R}_0$ and one should compute a basis set $\mathbb{B}_1$ in $\mathbb{P}_1$,
- then, one should compute the set $\mathbb{R}_1$ of non-zero reminders of polynomials of $\mathbb{P}_1 \setminus \mathcal{B}_1$ with respect to $\mathcal{B}_1$.
- By Lemma 3.8, $\mathcal{B}_1$ is of lower ordering than $\mathcal{B}_0$. Therefore, such a process has a finite number of steps, and the final result is a basic set $\mathcal{B}_m = \mathcal{C}$, such that the corresponding set of non-zero reminders $\mathbb{R}_m$ is the empty set and $prem(\mathbb{P}, \mathcal{C}) = \{0\}$. Thus, for each chain that can be obtained in such a way, $Zero(\mathbb{P}) \subseteq Zero(\mathcal{C})$.

Wu's algorithm

Starting with a polynomial set $\mathbb{P}_0 = \mathbb{P}$, one should:

- select a basis $\mathcal{B}_0$ of $\mathbb{P}_0$,
- compute the set $\mathbb{R}_0$ of non-zero reminders of polynomials of $\mathbb{P}_0 \setminus \mathcal{B}_0$ with respect to $\mathcal{B}_0$,
- then, let $\mathbb{P}_1 = \mathbb{P}_0 \cup \mathbb{R}_0$ and one should compute a basis set $\mathbb{B}_1$ in $\mathbb{P}_1$,
- then, one should compute the set $\mathbb{R}_1$ of non-zero reminders of polynomials of $\mathbb{P}_1 \setminus \mathcal{B}_1$ with respect to $\mathcal{B}_1$.
- By Lemma 3.8, $\mathcal{B}_1$ is of lower ordering than $\mathcal{B}_0$. Therefore, such a process has a finite number of steps, and the final result is a basic set $\mathcal{B}_m = \mathcal{C}$, such that the corresponding set of non-zero reminders $\mathbb{R}_m$ is the empty set and $prem(\mathbb{P}, \mathcal{C}) = \{0\}$. Thus, for each chain that can be obtained in such a way, $Zero(\mathbb{P}) \subseteq Zero(\mathcal{C})$.

Wu's algorithm

Starting with a polynomial set $\mathbb{P}_0 = \mathbb{P}$, one should:

- select a basis $\mathcal{B}_0$ of $\mathbb{P}_0$,
- compute the set $\mathbb{R}_0$ of non-zero reminders of polynomials of $\mathbb{P}_0 \setminus \mathcal{B}_0$ with respect to $\mathcal{B}_0$,
- then, let $\mathbb{P}_1 = \mathbb{P}_0 \cup \mathbb{R}_0$ and one should compute a basis set $\mathbb{B}_1$ in $\mathbb{P}_1$,
- then, one should compute the set $\mathbb{R}_1$ of non-zero reminders of polynomials of $\mathbb{P}_1 \setminus \mathcal{B}_1$ with respect to $\mathcal{B}_1$.
- By Lemma 3.8, $\mathcal{B}_1$ is of lower ordering than $\mathcal{B}_0$. Therefore, such a process has a finite number of steps, and the final result is a basic set $\mathcal{B}_m = \mathcal{C}$, such that the corresponding set of non-zero reminders $\mathbb{R}_m$ is the empty set and $prem(\mathbb{P}, \mathcal{C}) = \{0\}$. Thus, for each chain that can be obtained in such a way, $Zero(\mathbb{P}) \subseteq Zero(\mathcal{C})$.

Wu's algorithm

Starting with a polynomial set $\mathbb{P}_0 = \mathbb{P}$, one should:

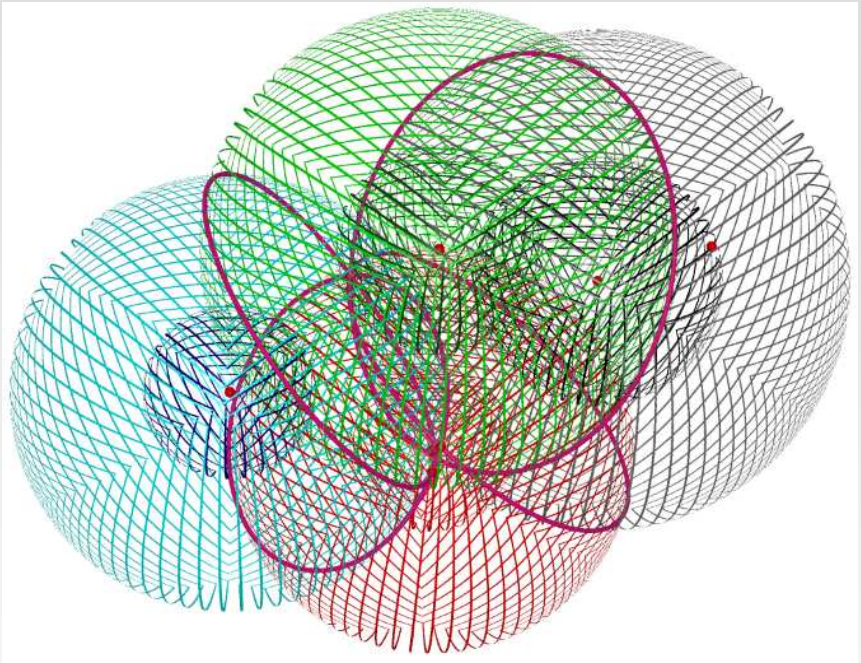
- select a basis $\mathcal{B}_0$ of $\mathbb{P}_0$,
- compute the set $\mathbb{R}_0$ of non-zero reminders of polynomials of $\mathbb{P}_0 \setminus \mathcal{B}_0$ with respect to $\mathcal{B}_0$,
- then, let $\mathbb{P}_1 = \mathbb{P}_0 \cup \mathbb{R}_0$ and one should compute a basis set $\mathbb{B}_1$ in $\mathbb{P}_1$,
- then, one should compute the set $\mathbb{R}_1$ of non-zero reminders of polynomials of $\mathbb{P}_1 \setminus \mathcal{B}_1$ with respect to $\mathcal{B}_1$.
- By Lemma 3.8, $\mathcal{B}_1$ is of lower ordering than $\mathcal{B}_0$. Therefore, such a process has a finite number of steps, and the final result is a basic set $\mathcal{B}_m = \mathcal{C}$, such that the corresponding set of non-zero reminders $\mathbb{R}_m$ is the empty set and $prem(\mathbb{P}, \mathcal{C}) = \{0\}$. Thus, for each chain that can be obtained in such a way, $Zero(\mathbb{P}) \subseteq Zero(\mathcal{C})$.

The Voronoi vertex of four spheres

A Voronoi vertex is a solution of one of the following system $\mathcal{I}$ of polynomial equations:

$\mathcal{I}$:

$$\begin{cases} (x-a)^2+(y-b)^2+(z-c)^2-(r\pm v)^2=0 \\ (x-d)^2+(y-e)^2+(z-f)^2-(s\pm v)^2=0 \\ (x-g)^2+(y-h)^2+(z-i)^2-(t\pm v)^2=0 \\ (x-j)^2+(y-k)^2+(z-l)^2-(u\pm v)^2=0 \end{cases}$$



Rewritting the Voronoi vertex

A Voronoi vertex is a solution of one of the following system $\mathcal{II}$ of polynomial equations:

$\mathcal{II}$:

$$\begin{cases} x^2+y^2+z^2-(v)^2 \\ (x-a')^2+(y-b')^2+(z-c')^2-(s'\pm v)^2 \\ (x-d')^2+(y-e')^2+(z-f')^2-(t'\pm v)^2 \\ (x-g')^2+(y-h')^2+(z-i')^2-(u'\pm v)^2 \end{cases}$$

where $(a',b',c')=(d,e,f)-(a,b,c)$,
$(d',e',f')=(g,h,i)-(a,b,c)$,
$(g',h',i')=(j,k,l)-(a,b,c)$, $s'=s-r$,
$t'=t-r$ and $u'=u-r$.

$$\begin{cases} x^2+y^2+z^2-(v)^2 \\ -2a'x-2b'y-2c'z-2s'v+(a'^2+b'^2+c'^2-s'^2) \\ -2d'x-2e'y-2f'z-2t'v+(d'^2+e'^2+f'^2-t'^2) \\ -2g'x-2h'y-2i'z-2u'v+(g'^2+h'^2+i'^2-u'^2) \end{cases}$$

$\mathbb{P}_0=\{p_1,p_2,p_3\}$
$\mathcal{A}_0=\{p_1\}$.
The reminders of the division by $\mathcal{A}_0$ are polynomials in v, x, and y. The first reminder that has the lowest ordering is $r_{00}=prem(p_2,p_1)$.

It can be added to $\mathcal{A}_0$. However, $r_{02}=prem(s_1,p_1)$ cannot be added to $\mathcal{A}_0$, since they have the same ordering as r_{00} (same class and same degree).
$\mathcal{A}_1=\mathcal{A}_0\bigcup\{r_{10}=prem(r_{01},r00)\}$.
The other non-zero reminder at this class level is another polynomial in v and x: $r_{11}=prem(r_{02},r_{00})$.
$\mathcal{A}_2=\mathcal{A}_1\bigcup\{r_{20}=prem(r_{11},r_{10})\}$, which is a quadratic polynomial in v. Thus, our basic set is
$\mathcal{C}: C_1,C_2,C_3,C_4$, where

$$\begin{cases} C_1=J\cdot v^2+K\cdot v+L \\ C_2=A\cdot x+H\cdot v+I \\ C_3=-A\cdot y+E\cdot v+F \\ C_4=A\cdot z+B\cdot v+C \end{cases}$$

Rewritting the Voronoi vertex

A Voronoi vertex is a solution of one of the following system $\mathcal{II}$ of polynomial equations:

$\mathcal{II}$:

$$\begin{cases} x^2+y^2+z^2-(v)^2 \\ (x-a')^2+(y-b')^2+(z-c')^2-(s'\pm v)^2 \\ (x-d')^2+(y-e')^2+(z-f')^2-(t'\pm v)^2 \\ (x-g')^2+(y-h')^2+(z-i')^2-(u'\pm v)^2 \end{cases}$$

where $(a',b',c')=(d,e,f)-(a,b,c)$,
$(d',e',f')=(g,h,i)-(a,b,c)$,
$(g',h',i')=(j,k,l)-(a,b,c)$, $s'=s-r$,
$t'=t-r$ and $u'=u-r$.

$$\begin{cases} x^2+y^2+z^2-(v)^2 \\ -2a'x-2b'y-2c'z-2s'v+(a'^2+b'^2+c'^2-s'^2) \\ -2d'x-2e'y-2f'z-2t'v+(d'^2+e'^2+f'^2-t'^2) \\ -2g'x-2h'y-2i'z-2u'v+(g'^2+h'^2+i'^2-u'^2) \end{cases}$$

$\mathbb{P}_0=\{p_1,p_2,p_3\}$
$\mathcal{A}_0=\{p_1\}$.
The reminders of the division by $\mathcal{A}_0$ are polynomials in v, x, and y. The first reminder that has the lowest ordering is $r_{00}=prem(p_2,p_1)$.

It can be added to $\mathcal{A}_0$. However, $r_{02}=prem(s_1,p_1)$ cannot be added to $\mathcal{A}_0$, since they have the same ordering as r_{00} (same class and same degree).
$\mathcal{A}_1=\mathcal{A}_0\bigcup\{r_{10}=prem(r_{01},r00)\}$.
The other non-zero reminder at this class level is another polynomial in v and x: $r_{11}=prem(r_{02},r_{00})$.
$\mathcal{A}_2=\mathcal{A}_1\bigcup\{r_{20}=prem(r_{11},r_{10})\}$, which is a quadratic polynomial in v. Thus, our basic set is
$\mathcal{C}: C_1,C_2,C_3,C_4$, where

$$\begin{cases} C_1=J\cdot v^2+K\cdot v+L \\ C_2=A\cdot x+H\cdot v+I \\ C_3=-A\cdot y+E\cdot v+F \\ C_4=A\cdot z+B\cdot v+C \end{cases}$$

Rewritting the Voronoi vertex

A Voronoi vertex is a solution of one of the following system $\mathcal{II}$ of polynomial equations:

$\mathcal{II}$:

$$\begin{cases} x^2+y^2+z^2-(v)^2 \\ (x-a')^2+(y-b')^2+(z-c')^2-(s'\pm v)^2 \\ (x-d')^2+(y-e')^2+(z-f')^2-(t'\pm v)^2 \\ (x-g')^2+(y-h')^2+(z-i')^2-(u'\pm v)^2 \end{cases}$$

where $(a',b',c')=(d,e,f)-(a,b,c)$,
$(d',e',f')=(g,h,i)-(a,b,c)$,
$(g',h',i')=(j,k,l)-(a,b,c)$, $s'=s-r$,
$t'=t-r$ and $u'=u-r$.

$$\begin{cases} x^2+y^2+z^2-(v)^2 \\ -2a'x-2b'y-2c'z-2s'v+(a'^2+b'^2+c'^2-s'^2) \\ -2d'x-2e'y-2f'z-2t'v+(d'^2+e'^2+f'^2-t'^2) \\ -2g'x-2h'y-2i'z-2u'v+(g'^2+h'^2+i'^2-u'^2) \end{cases}$$

$\mathbb{P}_0=\{p_1,p_2,p_3\}$

$\mathcal{A}_0=\{p_1\}$.

The reminders of the division by $\mathcal{A}_0$ are polynomials in v, x, and y. The first reminder that has the lowest ordering is $r_{00}=prem(p_2,p_1)$.

It can be added to $\mathcal{A}_0$. However, $r_{01}=prem(p_3,p_1)$ and $r_{02}=prem(s_1,p_1)$ cannot be added to $\mathcal{A}_0$, since they have the same ordering as r_{00} (same class and same degree).

$\mathcal{A}_1=\mathcal{A}_0\bigcup\{r_{10}=prem(r_{01},r00)\}$.

The other non-zero reminder at this class level is another polynomial in v and x: $r_{11}=prem(r_{02},r_{00})$.

$\mathcal{A}_2=\mathcal{A}_1\bigcup\{r_{20}=prem(r_{11},r_{10})\}$, which is a quadratic polynomial in v. Thus, our basic set is

$\mathcal{C}: C_1,C_2,C_3,C_4$, where

$$\begin{cases} C_1=J\cdot v^2+K\cdot v+L \\ C_2=A\cdot x+H\cdot v+I \\ C_3=-A\cdot y+E\cdot v+F \\ C_4=A\cdot z+B\cdot v+C \end{cases}$$

Rewritting the Voronoi vertex

A Voronoi vertex is a solution of one of the following system $\mathcal{II}$ of polynomial equations:

$\mathcal{II}$:

$$\begin{cases} x^2+y^2+z^2-(v)^2 \\ (x-a')^2+(y-b')^2+(z-c')^2-(s'\pm v)^2 \\ (x-d')^2+(y-e')^2+(z-f')^2-(t'\pm v)^2 \\ (x-g')^2+(y-h')^2+(z-i')^2-(u'\pm v)^2 \end{cases}$$

where $(a',b',c')=(d,e,f)-(a,b,c)$,
$(d',e',f')=(g,h,i)-(a,b,c)$,
$(g',h',i')=(j,k,l)-(a,b,c)$, $s'=s-r$,
$t'=t-r$ and $u'=u-r$.

$$\begin{cases} x^2+y^2+z^2-(v)^2 \\ -2a'x-2b'y-2c'z-2s'v+(a'^2+b'^2+c'^2-s'^2) \\ -2d'x-2e'y-2f'z-2t'v+(d'^2+e'^2+f'^2-t'^2) \\ -2g'x-2h'y-2i'z-2u'v+(g'^2+h'^2+i'^2-u'^2) \end{cases}$$

$\mathbb{P}_0=\{p_1,p_2,p_3\}$
$\mathcal{A}_0=\{p_1\}$.
The reminders of the division by $\mathcal{A}_0$ are polynomials in v, x, and y. The first reminder that has the lowest ordering is $r_{00}=prem(p_2,p_1)$.

It can be added to $\mathcal{A}_0$. However, $r_{01}=prem(p_3,p_1)$ and $r_{02}=prem(s_1,p_1)$ cannot be added to $\mathcal{A}_0$, since they have the same ordering as r_{00} (same class and same degree).
$\mathcal{A}_1=\mathcal{A}_0\bigcup\{r_{10}=prem(r_{01},r00)\}$.
The other non-zero reminder at this class level is another polynomial in v and x: $r_{11}=prem(r_{02},r_{00})$.
$\mathcal{A}_2=\mathcal{A}_1\bigcup\{r_{20}=prem(r_{11},r_{10})\}$, which is a quadratic polynomial in v. Thus, our basic set is
$\mathcal{C}: C_1,C_2,C_3,C_4$, where

$$\begin{cases} C_1=J\cdot v^2+K\cdot v+L \\ C_2=A\cdot x+H\cdot v+I \\ C_3=-A\cdot y+E\cdot v+F \\ C_4=A\cdot z+B\cdot v+G \end{cases}$$

Rewritting the Voronoi vertex

A Voronoi vertex is a solution of one of the following system $\mathcal{II}$ of polynomial equations:

$\mathcal{II}$:

$$\begin{cases} x^2+y^2+z^2-(v)^2 \\ (x-a')^2+(y-b')^2+(z-c')^2-(s'\pm v)^2 \\ (x-d')^2+(y-e')^2+(z-f')^2-(t'\pm v)^2 \\ (x-g')^2+(y-h')^2+(z-i')^2-(u'\pm v)^2 \end{cases}$$

where $(a',b',c')=(d,e,f)-(a,b,c)$,
$(d',e',f')=(g,h,i)-(a,b,c)$,
$(g',h',i')=(j,k,l)-(a,b,c)$, $s'=s-r$,
$t'=t-r$ and $u'=u-r$.

$$\begin{cases} x^2+y^2+z^2-(v)^2 \\ -2a'x-2b'y-2c'z-2s'v+(a'^2+b'^2+c'^2-s'^2) \\ -2d'x-2e'y-2f'z-2t'v+(d'^2+e'^2+f'^2-t'^2) \\ -2g'x-2h'y-2i'z-2u'v+(g'^2+h'^2+i'^2-u'^2) \end{cases}$$

$\mathbb{P}_0=\{p_1,p_2,p_3\}$
$\mathcal{A}_0=\{p_1\}$.
The reminders of the division by $\mathcal{A}_0$ are polynomials in v, x, and y. The first reminder that has the lowest ordering is $r_{00}=prem(p_2,p_1)$.

It can be added to $\mathcal{A}_0$. However, $r_{01}=prem(p_3,p_1)$ and $r_{02}=prem(s_1,p_1)$ cannot be added to $\mathcal{A}_0$, since they have the same ordering as r_{00} (same class and same degree).

$\mathcal{A}_1=\mathcal{A}_0\bigcup\{r_{10}=prem(r_{01},r00)\}$.
The other non-zero reminder at this class level is another polynomial in v and x: $r_{11}=prem(r_{02},r_{00})$.
$\mathcal{A}_2=\mathcal{A}_1\bigcup\{r_{20}=prem(r_{11},r_{10})\}$, which is a quadratic polynomial in v. Thus, our basic set is
$\mathcal{C}: C_1,C_2,C_3,C_4$, where

$$\begin{cases} C_1=J\cdot v^2+K\cdot v+L \\ C_2=A\cdot x+H\cdot v+I \\ C_3=-A\cdot y+E\cdot v+F \\ C_4=A\cdot z+B\cdot v+G \end{cases}$$

Rewritting the Voronoi vertex

A Voronoi vertex is a solution of one of the following system $\mathcal{II}$ of polynomial equations:

$$\mathcal{II}: \begin{cases} x^2+y^2+z^2-(v)^2 \\ (x-a')^2+(y-b')^2+(z-c')^2-(s'\pm v)^2 \\ (x-d')^2+(y-e')^2+(z-f')^2-(t'\pm v)^2 \\ (x-g')^2+(y-h')^2+(z-i')^2-(u'\pm v)^2 \end{cases}$$

where $(a',b',c')=(d,e,f)-(a,b,c)$, $(d',e',f')=(g,h,i)-(a,b,c)$, $(g',h',i')=(j,k,l)-(a,b,c)$, $s'=s-r$, $t'=t-r$ and $u'=u-r$.

$$\begin{cases} x^2+y^2+z^2-(v)^2 \\ -2a'x-2b'y-2c'z-2s'v+(a'^2+b'^2+c'^2-s'^2) \\ -2d'x-2e'y-2f'z-2t'v+(d'^2+e'^2+f'^2-t'^2) \\ -2g'x-2h'y-2i'z-2u'v+(g'^2+h'^2+i'^2-u'^2) \end{cases}$$

$\mathbb{P}_0=\{p_1,p_2,p_3\}$

$\mathcal{A}_0=\{p_1\}$.

The reminders of the division by $\mathcal{A}_0$ are polynomials in v, x, and y. The first reminder that has the lowest ordering is $r_{00}=prem(p_2,p_1)$.

It can be added to $\mathcal{A}_0$. However, $r_{01}=prem(p_3,p_1)$ and $r_{02}=prem(s_1,p_1)$ cannot be added to $\mathcal{A}_0$, since they have the same ordering as r_{00} (same class and same degree).

$\mathcal{A}_1=\mathcal{A}_0\bigcup\{r_{10}=prem(r_{01},r00)\}$.

The other non-zero reminder at this class level is another polynomial in v and x: $r_{11}=prem(r_{02},r_{00})$.

$\mathcal{A}_2=\mathcal{A}_1\bigcup\{r_{20}=prem(r_{11},r_{10})\}$, which is a quadratic polynomial in v. Thus, our basic set is

$\mathcal{C}: C_1,C_2,C_3,C_4$, where

$$\begin{cases} C_1=J\cdot v^2+K\cdot v+L \\ C_2=A\cdot x+H\cdot v+I \\ C_3=-A\cdot y+E\cdot v+F \\ C_4=A\cdot z+B\cdot v+G \end{cases}$$

Rewritting the Voronoi vertex

A Voronoi vertex is a solution of one of the following system $\mathcal{II}$ of polynomial equations:

$\mathcal{II}$:

$$\begin{cases} x^2+y^2+z^2-(v)^2 \\ (x-a')^2+(y-b')^2+(z-c')^2-(s'\pm v)^2 \\ (x-d')^2+(y-e')^2+(z-f')^2-(t'\pm v)^2 \\ (x-g')^2+(y-h')^2+(z-i')^2-(u'\pm v)^2 \end{cases}$$

where $(a',b',c')=(d,e,f)-(a,b,c)$,
$(d',e',f')=(g,h,i)-(a,b,c)$,
$(g',h',i')=(j,k,l)-(a,b,c)$, $s'=s-r$,
$t'=t-r$ and $u'=u-r$.

$$\begin{cases} x^2+y^2+z^2-(v)^2 \\ -2a'x-2b'y-2c'z-2s'v+(a'^2+b'^2+c'^2-s'^2) \\ -2d'x-2e'y-2f'z-2t'v+(d'^2+e'^2+f'^2-t'^2) \\ -2g'x-2h'y-2i'z-2u'v+(g'^2+h'^2+i'^2-u'^2) \end{cases}$$

$\mathbb{P}_0=\{p_1,p_2,p_3\}$
$\mathcal{A}_0=\{p_1\}$.
The reminders of the division by $\mathcal{A}_0$ are polynomials in v, x, and y. The first reminder that has the lowest ordering is $r_{00}=prem(p_2,p_1)$.

It can be added to $\mathcal{A}_0$. However, $r_{01}=prem(p_3,p_1)$ and $r_{02}=prem(s_1,p_1)$ cannot be added to $\mathcal{A}_0$, since they have the same ordering as r_{00} (same class and same degree).
$\mathcal{A}_1=\mathcal{A}_0\bigcup\{r_{10}=prem(r_{01},r00)\}$.
The other non-zero reminder at this class level is another polynomial in v and x: $r_{11}=prem(r_{02},r_{00})$.
$\mathcal{A}_2=\mathcal{A}_1\bigcup\{r_{20}=prem(r_{11},r_{10})\}$, which is a quadratic polynomial in v. Thus, our basic set is
$\mathcal{C}: C_1,C_2,C_3,C_4$, where

$$\begin{cases} C_1=J\cdot v^2+K\cdot v+L \\ C_2=A\cdot x+H\cdot v+I \\ C_3=-A\cdot y+E\cdot v+F \\ C_4=A\cdot z+B\cdot v+G \end{cases}$$

Rewritting the Voronoi vertex

A Voronoi vertex is a solution of one of the following system $\mathcal{II}$ of polynomial equations:

$\mathcal{II}$:

$$\begin{cases} x^2+y^2+z^2-(v)^2 \\ (x-a')^2+(y-b')^2+(z-c')^2-(s'\pm v)^2 \\ (x-d')^2+(y-e')^2+(z-f')^2-(t'\pm v)^2 \\ (x-g')^2+(y-h')^2+(z-i')^2-(u'\pm v)^2 \end{cases}$$

where $(a',b',c')=(d,e,f)-(a,b,c)$,
$(d',e',f')=(g,h,i)-(a,b,c)$,
$(g',h',i')=(j,k,l)-(a,b,c)$, $s'=s-r$,
$t'=t-r$ and $u'=u-r$.

$$\begin{cases} x^2+y^2+z^2-(v)^2 \\ -2a'x-2b'y-2c'z-2s'v+(a'^2+b'^2+c'^2-s'^2) \\ -2d'x-2e'y-2f'z-2t'v+(d'^2+e'^2+f'^2-t'^2) \\ -2g'x-2h'y-2i'z-2u'v+(g'^2+h'^2+i'^2-u'^2) \end{cases}$$

$\mathbb{P}_0=\{p_1,p_2,p_3\}$
$\mathcal{A}_0=\{p_1\}$.
The reminders of the division by $\mathcal{A}_0$ are polynomials in v, x, and y. The first reminder that has the lowest ordering is $r_{00}=prem(p_2,p_1)$.

It can be added to $\mathcal{A}_0$. However, $r_{01}=prem(p_3,p_1)$ and $r_{02}=prem(s_1,p_1)$ cannot be added to $\mathcal{A}_0$, since they have the same ordering as r_{00} (same class and same degree).
$\mathcal{A}_1=\mathcal{A}_0\bigcup\{r_{10}=prem(r_{01},r00)\}$.
The other non-zero reminder at this class level is another polynomial in v and x: $r_{11}=prem(r_{02},r_{00})$.
$\mathcal{A}_2=\mathcal{A}_1\bigcup\{r_{20}=prem(r_{11},r_{10})\}$, which is a quadratic polynomial in v. Thus, our basic set is
$\mathcal{C}$: C_1,C_2,C_3,C_4, where

$$\begin{cases} C_1=J\cdot v^2+K\cdot v+L \\ C_2=A\cdot x+H\cdot v+I \\ C_3=-A\cdot y+E\cdot v+F \\ C_4=A\cdot z+B\cdot v+C. \end{cases}$$

Simplification using invariants

Using geometric invariants represents a very important simplification: we have rewritten in a quadratic univariate polynomial in v of the form $J \cdot v^2 + K \cdot v + L$:

- the term in v^2, that had 224 monomials in the parameters $a, b, c, d, e, f, g, h, i, s, t, u$ into a term J that has only 4 monomials in the simple invariants,
- the term in v, that had 1080 monomials in the parameters $a, b, c, d, e, f, g, h, i, s, t, u$ into a term K that has only 3 monomials in the simple invariants,
- the constant term, that had 2276 monomials in the parameters $a, b, c, d, e, f, g, h, i, s, t, u$ into a term L that has only 3 monomials in the simple invariants

The univariate polynomial of the ascending chain has thus been simplified from a polynomial containing 3580 terms into a polynomial containing only 10 terms using invariants!

Simplification using invariants

Using geometric invariants represents a very important simplification: we have rewritten in a quadratic univariate polynomial in v of the form $J \cdot v^2 + K \cdot v + L$:

- the term in v^2, that had 224 monomials in the parameters $a, b, c, d, e, f, g, h, i, s, t, u$ into a term J that has only 4 monomials in the simple invariants,
- the term in v, that had 1080 monomials in the parameters $a, b, c, d, e, f, g, h, i, s, t, u$ into a term K that has only 3 monomials in the simple invariants,
- the constant term, that had 2276 monomials in the parameters $a, b, c, d, e, f, g, h, i, s, t, u$ into a term L that has only 3 monomials in the simple invariants

The univariate polynomial of the ascending chain has thus been simplified from a polynomial containing 3580 terms into a polynomial containing only 10 terms using invariants!

Simplification using invariants

Using geometric invariants represents a very important simplification: we have rewritten in a quadratic univariate polynomial in v of the form $J \cdot v^2 + K \cdot v + L$:

- the term in v^2, that had 224 monomials in the parameters $a, b, c, d, e, f, g, h, i, s, t, u$ into a term J that has only 4 monomials in the simple invariants,
- the term in v, that had 1080 monomials in the parameters $a, b, c, d, e, f, g, h, i, s, t, u$ into a term K that has only 3 monomials in the simple invariants,
- the constant term, that had 2276 monomials in the parameters $a, b, c, d, e, f, g, h, i, s, t, u$ into a term L that has only 3 monomials in the simple invariants

The univariate polynomial of the ascending chain has thus been simplified from a polynomial containing 3580 terms into a polynomial containing only 10 terms using invariants!

Simplification using invariants

Using geometric invariants represents a very important simplification: we have rewritten in a quadratic univariate polynomial in v of the form $J \cdot v^2 + K \cdot v + L$:

- the term in v^2, that had 224 monomials in the parameters $a, b, c, d, e, f, g, h, i, s, t, u$ into a term J that has only 4 monomials in the simple invariants,
- the term in v, that had 1080 monomials in the parameters $a, b, c, d, e, f, g, h, i, s, t, u$ into a term K that has only 3 monomials in the simple invariants,
- the constant term, that had 2276 monomials in the parameters $a, b, c, d, e, f, g, h, i, s, t, u$ into a term L that has only 3 monomials in the simple invariants

The univariate polynomial of the ascending chain has thus been simplified from a polynomial containing 3580 terms into a polynomial containing only 10 terms using invariants!

Simplification using invariants

Using geometric invariants represents a very important simplification: we have rewritten in a quadratic univariate polynomial in v of the form $J \cdot v^2 + K \cdot v + L$:

- the term in v^2, that had 224 monomials in the parameters $a, b, c, d, e, f, g, h, i, s, t, u$ into a term J that has only 4 monomials in the simple invariants,
- the term in v, that had 1080 monomials in the parameters $a, b, c, d, e, f, g, h, i, s, t, u$ into a term K that has only 3 monomials in the simple invariants,
- the constant term, that had 2276 monomials in the parameters $a, b, c, d, e, f, g, h, i, s, t, u$ into a term L that has only 3 monomials in the simple invariants

The univariate polynomial of the ascending chain has thus been simplified from a polynomial containing 3580 terms into a polynomial containing only 10 terms using invariants!

Algebraically Independent invariants and dependent invariants

The offset variable can be computed by solving the quadratic equation $J \cdot v^2 + K \cdot v + L = 0$, which has no solution if $K^2 < 4JL$, one solution $v = \frac{K}{2J}$ if $K^2 = 4JL$, and two solutions $v = \frac{K \pm \sqrt{K^2 - 4JL}}{2J}$ if $K^2 > 4JL$. With expansions only:

$$A = -\begin{vmatrix} a' & b' & c' \\ d' & e' & f' \\ g' & h' & i' \end{vmatrix}$$

$$B = 2\begin{vmatrix} a' & b' & -s' \\ d' & e' & -t' \\ g' & h' & -u' \end{vmatrix}$$

$$C = \begin{vmatrix} a' & b' & a'^2 + b'^2 + c'^2 - s'^2 \\ d' & e' & d'^2 + e'^2 + f'^2 - t'^2 \\ g' & h' & g'^2 + h'^2 + i'^2 - u'^2 \end{vmatrix}$$

$$E = 2\begin{vmatrix} a' & c' & -s' \\ d' & f' & -t' \\ g' & i' & -u' \end{vmatrix}$$

$$F = \begin{vmatrix} a' & c' & a'^2 + b'^2 + c'^2 - s'^2 \\ d' & f' & d'^2 + e'^2 + f'^2 - t'^2 \\ g' & i' & g'^2 + h'^2 + i'^2 - u'^2 \end{vmatrix}$$

$$H = 2\begin{vmatrix} b' & c' & -s' \\ e' & f' & -t' \\ h' & i' & -u' \end{vmatrix}$$

$$I = \begin{vmatrix} b' & c' & a'^2 + b'^2 + c'^2 - s'^2 \\ e' & f' & d'^2 + e'^2 + f'^2 - t'^2 \\ h' & i' & g'^2 + h'^2 + i'^2 - u'^2 \end{vmatrix}$$

Algebraically Independent invariants and dependent invariants

The offset variable can be computed by solving the quadratic equation $J \cdot v^2 + K \cdot v + L = 0$, which has no solution if $K^2 < 4JL$, one solution $v = \frac{K}{2J}$ if $K^2 = 4JL$, and two solutions $v = \frac{K \pm \sqrt{K^2 - 4JL}}{2J}$ if $K^2 > 4JL$. With expansions only:

$$A = -\begin{vmatrix} a' & b' & c' \\ d' & e' & f' \\ g' & h' & i' \end{vmatrix}$$

$$B = 2\begin{vmatrix} a' & b' & -s' \\ d' & e' & -t' \\ g' & h' & -u' \end{vmatrix}$$

$$C = \begin{vmatrix} a' & b' & a'^2 + b'^2 + c'^2 - s'^2 \\ d' & e' & d'^2 + e'^2 + f'^2 - t'^2 \\ g' & h' & g'^2 + h'^2 + i'^2 - u'^2 \end{vmatrix}$$

$$E = 2\begin{vmatrix} a' & c' & -s' \\ d' & f' & -t' \\ g' & i' & -u' \end{vmatrix}$$

$$F = \begin{vmatrix} a' & c' & a'^2 + b'^2 + c'^2 - s'^2 \\ d' & f' & d'^2 + e'^2 + f'^2 - t'^2 \\ g' & i' & g'^2 + h'^2 + i'^2 - u'^2 \end{vmatrix}$$

$$H = 2\begin{vmatrix} b' & c' & -s' \\ e' & f' & -t' \\ h' & i' & -u' \end{vmatrix}$$

$$I = \begin{vmatrix} b' & c' & a'^2 + b'^2 + c'^2 - s'^2 \\ e' & f' & d'^2 + e'^2 + f'^2 - t'^2 \\ h' & i' & g'^2 + h'^2 + i'^2 - u'^2 \end{vmatrix}$$

Algebraically Independent invariants and dependent invariants

The offset variable can be computed by solving the quadratic equation $J \cdot v^2 + K \cdot v + L = 0$, which has no solution if $K^2 < 4JL$, one solution $v = \frac{K}{2J}$ if $K^2 = 4JL$, and two solutions $v = \frac{K \pm \sqrt{K^2 - 4JL}}{2J}$ if $K^2 > 4JL$. With expansions only:

$$A = -\begin{vmatrix} a' & b' & c' \\ d' & e' & f' \\ g' & h' & i' \end{vmatrix}$$

$$B = 2\begin{vmatrix} a' & b' & -s' \\ d' & e' & -t' \\ g' & h' & -u' \end{vmatrix}$$

$$C = \begin{vmatrix} a' & b' & a'^2 + b'^2 + c'^2 - s'^2 \\ d' & e' & d'^2 + e'^2 + f'^2 - t'^2 \\ g' & h' & g'^2 + h'^2 + i'^2 - u'^2 \end{vmatrix}$$

$$E = 2\begin{vmatrix} a' & c' & -s' \\ d' & f' & -t' \\ g' & i' & -u' \end{vmatrix}$$

$$F = \begin{vmatrix} a' & c' & a'^2 + b'^2 + c'^2 - s'^2 \\ d' & f' & d'^2 + e'^2 + f'^2 - t'^2 \\ g' & i' & g'^2 + h'^2 + i'^2 - u'^2 \end{vmatrix}$$

$$H = 2\begin{vmatrix} b' & c' & -s' \\ e' & f' & -t' \\ h' & i' & -u' \end{vmatrix}$$

$$I = \begin{vmatrix} b' & c' & a'^2 + b'^2 + c'^2 - s'^2 \\ e' & f' & d'^2 + e'^2 + f'^2 - t'^2 \\ h' & i' & g'^2 + h'^2 + i'^2 - u'^2 \end{vmatrix}$$

Algebraically Independent invariants and dependent invariants

The offset variable can be computed by solving the quadratic equation $J \cdot v^2 + K \cdot v + L = 0$, which has no solution if $K^2 < 4JL$, one solution $v = \frac{K}{2J}$ if $K^2 = 4JL$, and two solutions $v = \frac{K \pm \sqrt{K^2 - 4JL}}{2J}$ if $K^2 > 4JL$. With expansions only:

$$A = -\begin{vmatrix} a' & b' & c' \\ d' & e' & f' \\ g' & h' & i' \end{vmatrix}$$

$$B = 2\begin{vmatrix} a' & b' & -s' \\ d' & e' & -t' \\ g' & h' & -u' \end{vmatrix}$$

$$C = \begin{vmatrix} a' & b' & a'^2 + b'^2 + c'^2 - s'^2 \\ d' & e' & d'^2 + e'^2 + f'^2 - t'^2 \\ g' & h' & g'^2 + h'^2 + i'^2 - u'^2 \end{vmatrix}$$

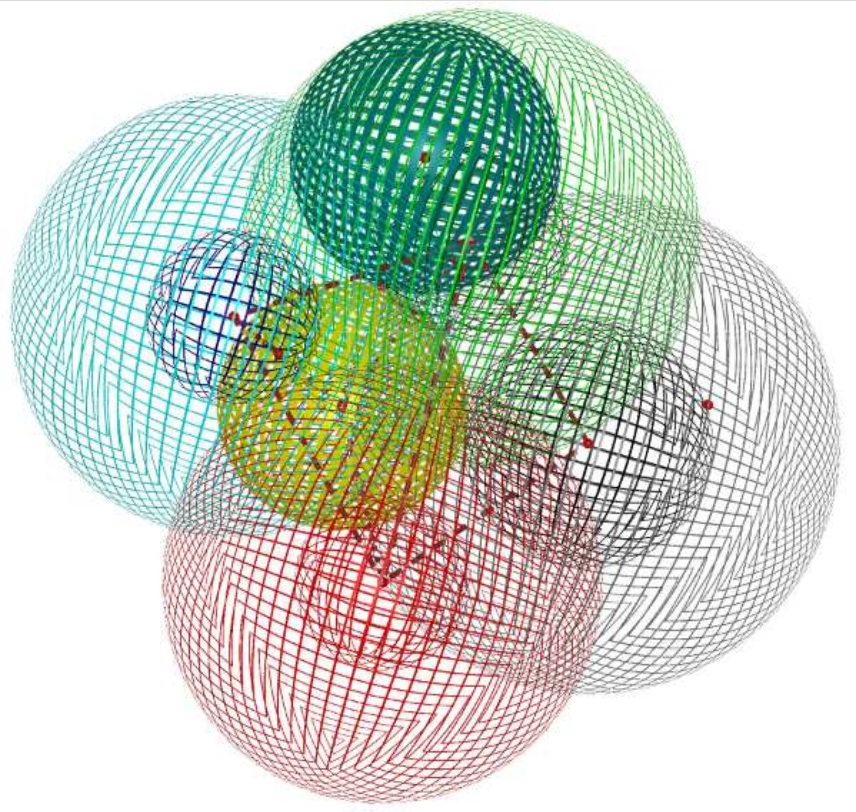
$$E = 2\begin{vmatrix} a' & c' & -s' \\ d' & f' & -t' \\ g' & i' & -u' \end{vmatrix}$$

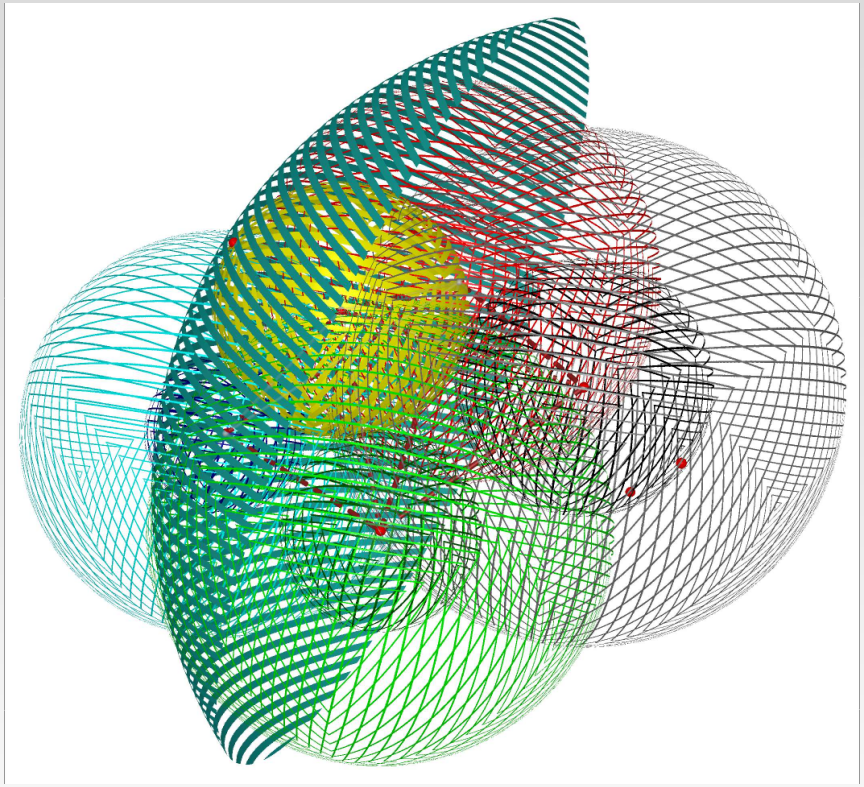
$$F = \begin{vmatrix} a' & c' & a'^2 + b'^2 + c'^2 - s'^2 \\ d' & f' & d'^2 + e'^2 + f'^2 - t'^2 \\ g' & i' & g'^2 + h'^2 + i'^2 - u'^2 \end{vmatrix}$$

$$H = 2\begin{vmatrix} b' & c' & -s' \\ e' & f' & -t' \\ h' & i' & -u' \end{vmatrix}$$

$$I = \begin{vmatrix} b' & c' & a'^2 + b'^2 + c'^2 - s'^2 \\ e' & f' & d'^2 + e'^2 + f'^2 - t'^2 \\ h' & i' & g'^2 + h'^2 + i'^2 - u'^2 \end{vmatrix}$$

Delaunay empty circumsphere predicate for spheres





Relative position of a sphere with respect to the empty sphere

The polynomial stating the difference of squared distances between the Voronoi vertex and S_5 and between the Voronoi vertex and S_1 is:

- $2\left(-m+\frac{jH-kE+lB}{A}\right)\cdot v+$
  $\left(\frac{K}{J}+2\frac{BC+EF+HI}{A^2}-\frac{\left(B^2+E^2+H^2\right)K}{A^2J}\right)\cdot v+$
  $\left(j^2+k^2+l^2-m^2\right)+2\frac{jl-kF+lC}{A}+\frac{L}{J}+$
  $\frac{C^2+F^2+I^2}{A^2}-\frac{\left(B^2+E^2+H^2\right)L}{A^2J}$ for the Voronoi vertex exterior to S_5;
- $2\left(m+\frac{jH-kE+lB}{A}\right)\cdot v+$
  $\left(\frac{K}{J}+2\frac{BC+EF+HI}{A^2}-\frac{\left(B^2+E^2+H^2\right)K}{A^2J}\right)\cdot v+$
  $\left(j^2+k^2+l^2-m^2\right)+2\frac{jl-kF+lC}{A}+\frac{L}{J}+$
  $\frac{C^2+F^2+I^2}{A^2}-\frac{\left(B^2+E^2+H^2\right)L}{A^2J}$ for the Voronoi vertex interior to S_5.

Relative position of a sphere with respect to the empty sphere

The polynomial stating the difference of squared distances between the Voronoi vertex and S_5 and between the Voronoi vertex and S_1 is:

- $2\left(-m+\frac{jH-kE+lB}{A}\right)\cdot v+$
  $\left(\frac{K}{J}+2\frac{BC+EF+HI}{A^2}-\frac{\left(B^2+E^2+H^2\right)K}{A^2J}\right)\cdot v+$
  $\left(j^2+k^2+l^2-m^2\right)+2\frac{jl-kF+lC}{A}+\frac{L}{J}+$
  $\frac{C^2+F^2+I^2}{A^2}-\frac{\left(B^2+E^2+H^2\right)L}{A^2J}$ for the Voronoi vertex exterior to S_5;
- $2\left(m+\frac{jH-kE+lB}{A}\right)\cdot v+$
  $\left(\frac{K}{J}+2\frac{BC+EF+HI}{A^2}-\frac{\left(B^2+E^2+H^2\right)K}{A^2J}\right)\cdot v+$
  $\left(j^2+k^2+l^2-m^2\right)+2\frac{jl-kF+lC}{A}+\frac{L}{J}+$
  $\frac{C^2+F^2+I^2}{A^2}-\frac{\left(B^2+E^2+H^2\right)L}{A^2J}$ for the Voronoi vertex interior to S_5.

Relative position of a sphere with respect to the empty sphere

Proposition

The algebraic degree of the incircle predicate for spheres in the invariants and the variables defining the fifth sphere is 6. *We need* 6 *times longer bits for the exact computation of the incircle predicate than the bits used for the invariants and the variables defining the fifth sphere.*

Proof.

The incircle predicate is given by the sign of G. Since the denominator of v is $2J$, the greatest common divider of all the terms in the expansion of G is $A^2J^2 \geq 0$. In the generic case ($A^2J^2 \neq 0$), we can rewrite G as a rational function, whose numerator degree in the invariants and the variables defining S_5 is the degree of the monomials mKA^2J^2 or $\left(j^2 + k^2 + l^2 - m^2\right) A^2J^2$, which is 6. Bounding all the invariants and the variables defining S_5 as in [9], we need 6 times longer bits. □

Conclusions

This research work provides a significative simplification using invariants for the exact computation of vertices of the Voronoi diagram of spheres and the empty circumsphere criterion as well as their geometric invariants. This work has a direct application in Geodesy: the optimal placement of the system of GPS satellites and the determination of the geometric uncertainty of the determination of coordinates by GPS as a function of the placement of the GPS satellites. Further work will address these applications as well as the application of the automatic derivation and simplification of invariants to the Delaunay graph and Voronoi diagram of quadrics.


François Anton, Darka Mioc, and Christopher Gold, *The voronoi diagram of circles and its application to the visualization of the growth of particles*, Transactions on Computational Science III (Marina L. Gavrilova and C. J. Tan, eds.), Springer-Verlag, Berlin, Heidelberg, 2009, pp. 20–54.


M. L. Gavrilova and J. Rokne, *Updating the topology of the dynamic voronoi diagram for spheres in euclidean d-dimensional space*, Comput. Aided Geom. Des. **20** (2003), 231–242.


Marina Gavrilova, *Proximity and applications in general metrics*, Ph.D. thesis, University of Calgary, Calgary, Alberta, Canada, 1998.


Iddo Hanniel and Gershon Elber, *Computing the voronoi cells of planes, spheres and cylinders in* $\mathbb{R}^3$, Comput. Aided Geom. Des. **26** (2009), 695–710.


Deok-Soo Kim, Youngsong Cho, and Donguk Kim, *Euclidean voronoi diagram of 3d balls and its computation via tracing edges.*, Computer-Aided Design (2005), 1412–1424.


Deok-soo Kim, Youngsong Cho, and Donguk Kim, *Calculating three-dimensional (3d) voronoi diagrams*, Patent no. 7825927, November 2010.


Deok-Soo Kim, Youngsong Cho, Donguk Kim, Sangsoo Kim, Jonghwa Bhak, and Sung-Hoon Lee, *Euclidean voronoi diagrams of 3d spheres and applications to protein structure analysis*, Japan Journal of Industrial and Applied Mathematics **22** (2005), 251–265, 10.1007/BF03167441.


Donguk Kim and Deok-Soo Kim, *Region-expansion for the voronoi diagram of 3d spheres*, Comput. Aided Des. **38** (2006), 417–430.

Tetsushi Nishida and Kokichi Sugihara, *Precision necessary for d−dimensional Sphere Voronoi Diagrams*, Proceedings of the Fifth International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2008), 2008, pp. 157–167.

Tetsushi Nishida, Yoshiaki Tanaka, and Kokichi Sugihara, *Evaluation of the Precision for Exact Computation of a Circle Voronoi Diagram*, Tech. Report UW-CS-TR-1481, Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo, Bunkyo-Ku, Tokyo 113-8656, Japan, October 2007.

Joonghyun Ryu, Donguk Kim, Youngsong Cho, Rhohun Park, and Deok soo Kim, *D-s, Òcomputation of molecular surface using euclidean voronoi diagram*, Computer-Aided Design and Applications, Vol.2, Nos. 1-4, 2005.

Hans-Martin Will, *Fast and efficient computation of additively weighted voronoi cells for applications in molecular biology*,

Proceedings of the 6th Scandinavian Workshop on Algorithm Theory (London, UK), SWAT '98, Springer-Verlag, 1998, pp. 310–321.

Wen-Tsun Wu and Xiao-Shan Gao, *Automated reasoning and equation solving with the characteristic set method*, J. Comput. Sci. & Technol. **21** (2006), no. 5, 756–764.

Third part: Applications to Image Analysis through the dual graph of the Delaunay graph, which is isomorphic to the medial axis

On the Isomorphism Between the Medial Axis and a Dual of the Delaunay Graph

Ojaswa Sharma[†], François Anton[†], and Darka Mioc[‡]

[†]**DTU Informatics,**
The Technical University of Denmark, Denmark

[‡]**Department of Geodesy and Geomatics Engineering,**
University of New Brunswick, Canada

ISVD 2009, Copenhagen

23 June, 2009

Outline

- Research objectives
- Graph isomorphism
- Our automated approach to skeletonization
- Results
- Conclusions

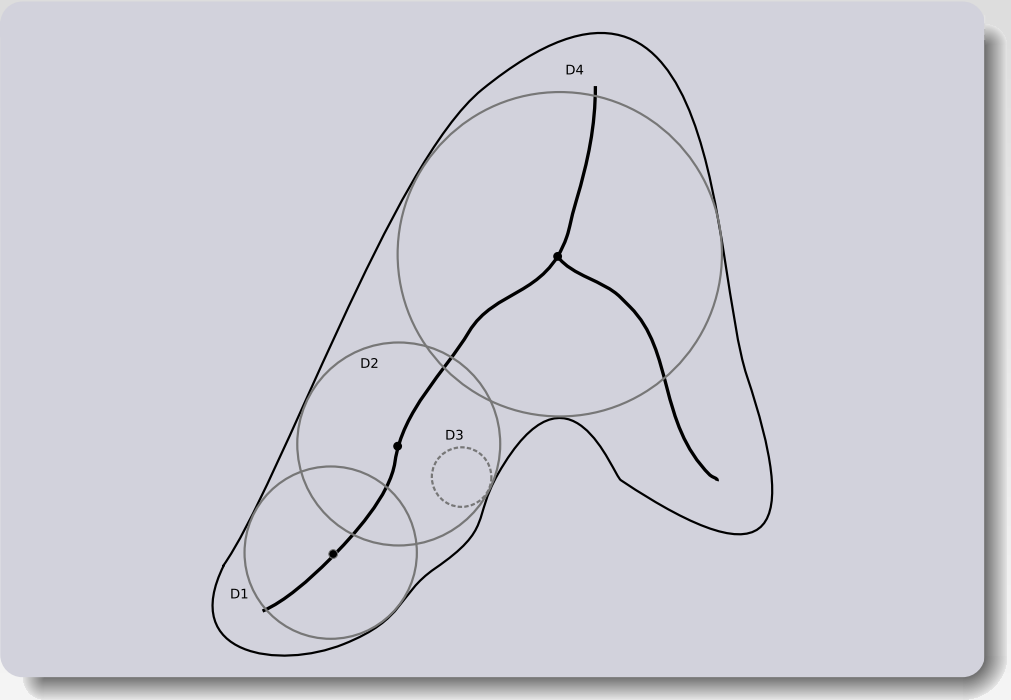
Research Objectives

1. To show graph isomorphism between the medial axis of an object and a dual of its Delaunay graph.
2. To design a methodology to automate delineation of boundary and centreline of objects from digital images.

What is a Skeleton?

A skeleton of an object can be described as its centreline.

More formally, a skeleton is the locus of the centre of maximal inscribed discs.



Existing Methods

- Raster based processing to extract centerline.
- Mainly process binary images.
- Do not preserve topology and mediality simultaneously.

Example

- Morphological thinning
- Distance transform

Existing Methods

- Raster based processing to extract centerline.
- Mainly process binary images.
- Do not preserve topology and mediality simultaneously.

Example

Grayscale and colour images are processed by thresholding.

Existing Methods

- Raster based processing to extract centerline.
- Mainly process binary images.
- Do not preserve topology and mediality simultaneously.

Example

Thinning does not guarantee mediality while Distance transform does not ensure connectivity.

Graph Isomorphism between Medial Axis and a Dual of Delaunay Graph

Our approach to medial axis is based on the graph isomorphism between the medial axis and a dual of the Delaunay graph. In order to show the isomorphism, let's define the following graphs

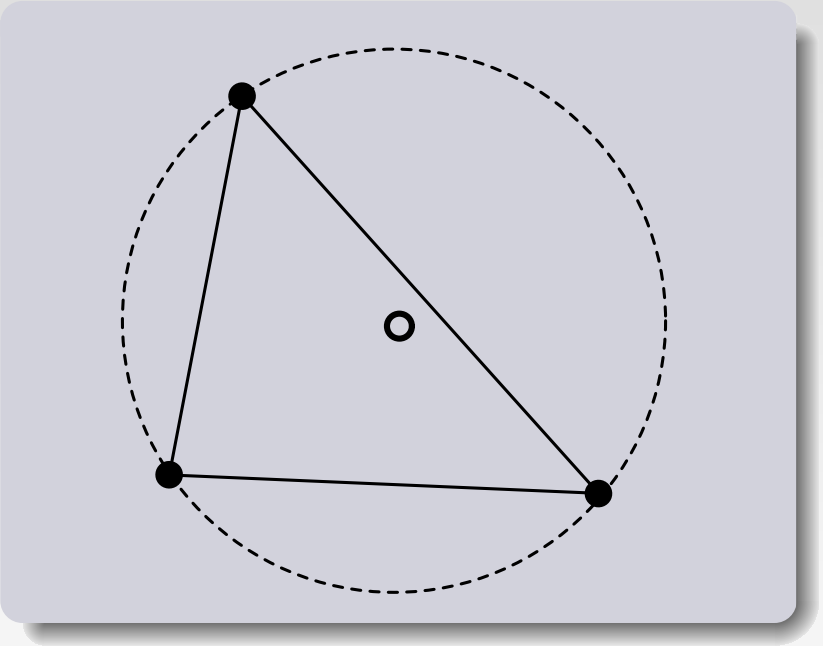
- *IDG*: Interior Delaunay Graph (a subset of Delaunay graph) which represents the object boundary.
- *DIDG*: Dual of *IDG* constructed by application of our rules.

Rules for Construction of the Dual of Internal Delaunay Graph

(a) The vertices of *DIDG* are the isobarycenters of the vertices of each triangle of *IDG* that does not belong to a complete subgraph of at least four vertices.

Rules for Construction of the Dual of Internal Delaunay Graph

(a) The vertices of *DIDG* are the isobarycenters of the vertices of each triangle of *IDG* that does not belong to a complete subgraph of at least four vertices.

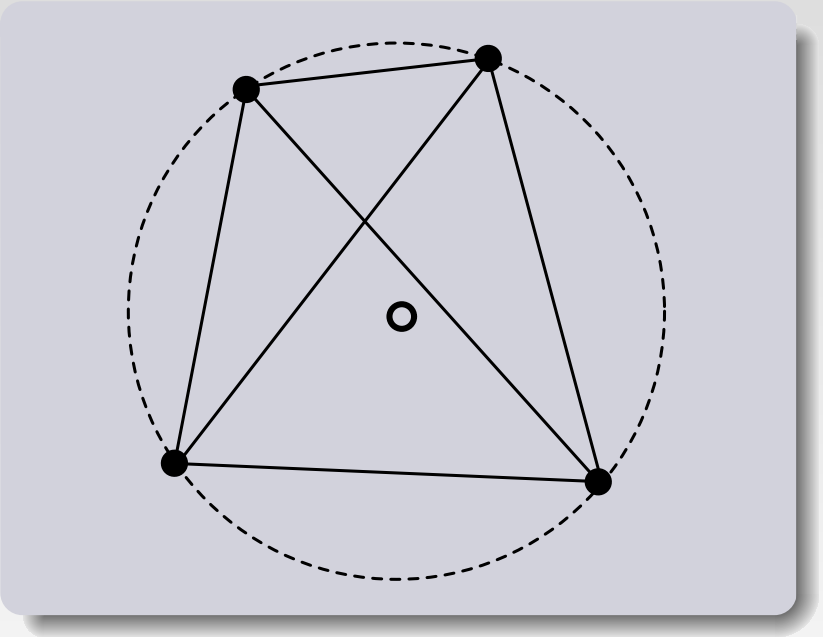


Rules for Construction of the Dual of Internal Delaunay Graph

(b) For each complete subgraph of at least 4 vertices of *IDG* (that are cocircular), the corresponding subgraph of *DIDG* is reduced to a point.

Rules for Construction of the Dual of Internal Delaunay Graph

(b) For each complete subgraph of at least 4 vertices of *IDG* (that are cocircular), the corresponding subgraph of *DIDG* is reduced to a point.

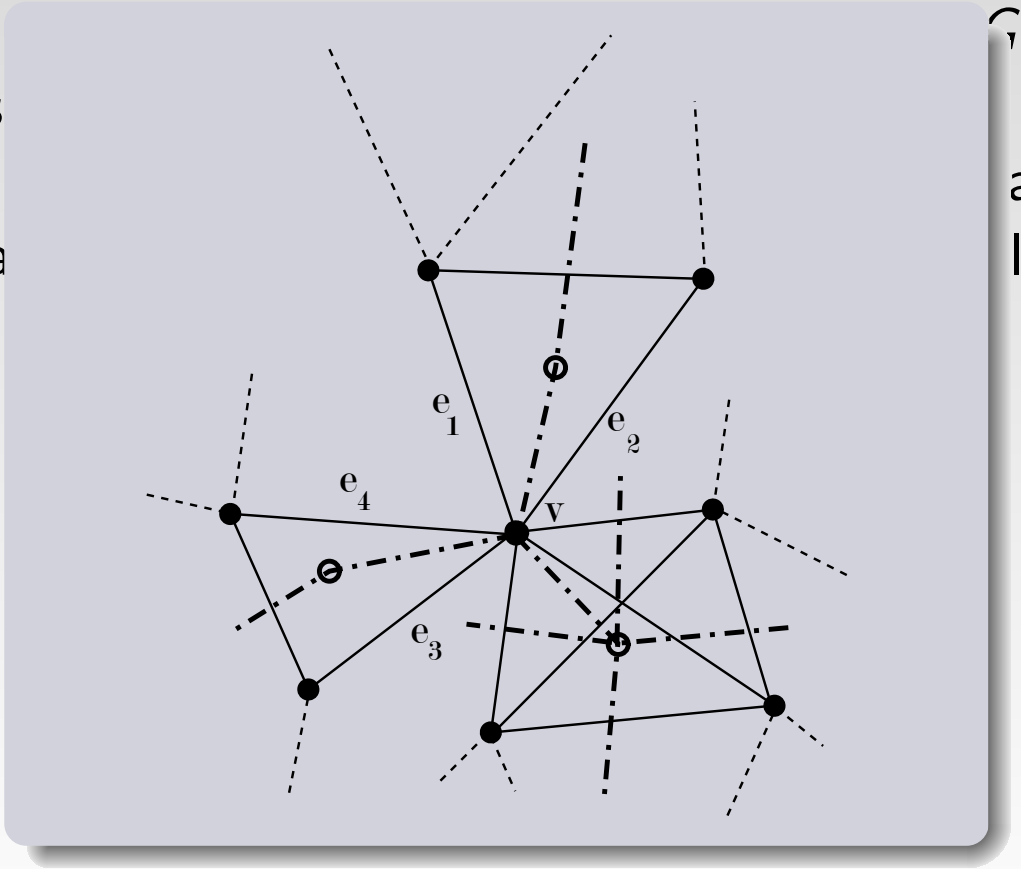


Rules for Construction of the Dual of Internal Delaunay Graph

(c) For each set E of edges $e_1 ... e_k, k > 2$ of the boundary of IDG that share one common vertex v that is a vertex of a complete subgraph K of at least 4 vertices of IDG, there is a set of edges of $DIDG$ that link v to each one of the isobarycenters of the triangles $t_1 ... t_j$ such that $t_i \in IDG$ and t_i has two of its edges in E that are not edges of K, and there is one edge of $DIDG$ that links v to the center of the circumcircle of the vertices of K.

Rules for Construction of the Dual of Internal Delaunay Graph

(c) For each set E of edges $e_1 ... e_k, k > 2$ of the boundary of IDG that share one common vertex v that is a vertex of a complete subgraph K of at least 4 vertices [illegible] that link v to each one of the is [illegible] that $t_i \in IDG$ and t_i has two of [illegible] and there is one edge of $DIDG$ tha [illegible] le of the vertices of K.

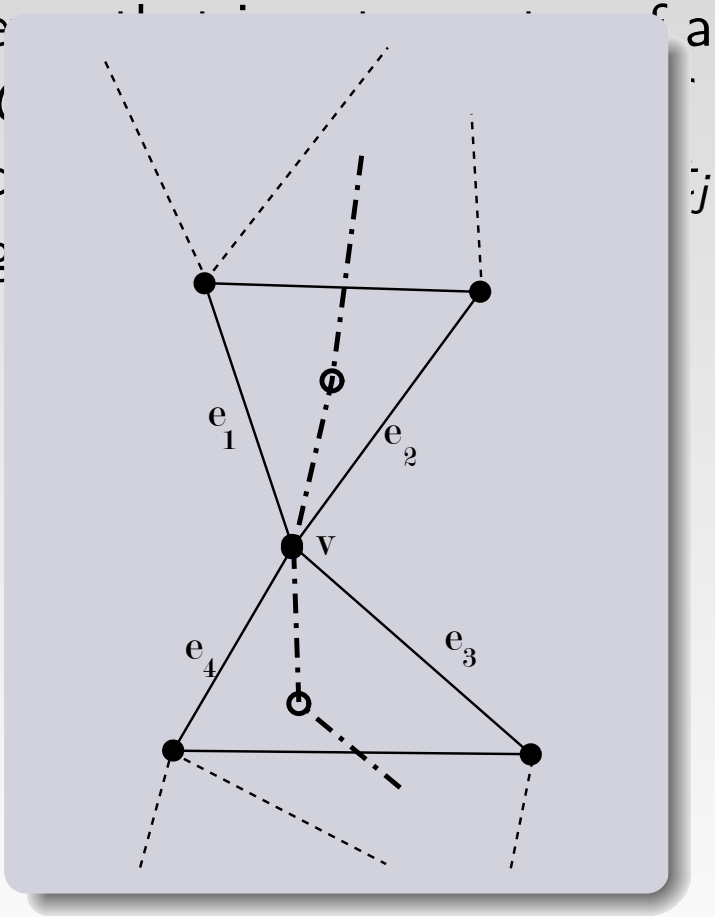


Rules for Construction of the Dual of Internal Delaunay Graph

(d) For each set E of edges $e_1 ... e_k, k > 2$ of the boundary of IDG that share one common vertex v that is not a vertex of a complete subgraph of at least 4 vertices of IDG, there is a set of edges of $DIDG$ that link v to each one of the isobarycenters of the triangles $t_1 ... t_j$ such that $t_i \in IDG$ and t_i has two of its edges in E.

Rules for Construction of the Dual of Internal Delaunay Graph

(d) For each set E of edges $e_1 ... e_k, k > 2$ of the boundary of IDG that share one common verte… a complete subgraph of at least 4 vertices of IDG … $DIDG$ that link v to each one of the isobaryc… $_j$ such that $t_i \in IDG$ and t_i has two of its edg…

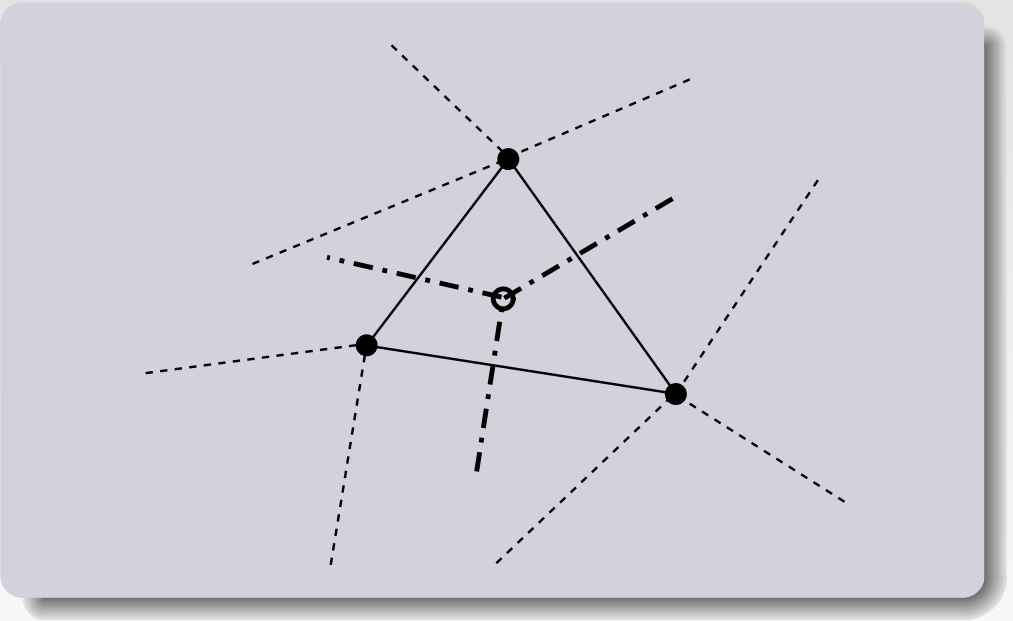


Rules for Construction of the Dual of Internal Delaunay Graph

(e) For each edge e that is not on the boundary of IDG and that does not link two vertices of a complete subgraph of at least 4 vertices of IDG, there exists an edge of $DIDG$ that links the isobarycenters of the vertices of each one of the triangles that share e.

Rules for Construction of the Dual of Internal Delaunay Graph

(e) For each edge e that is not on the boundary of *IDG* and that does not link two vertices of a complete subgraph of at least 4 vertices of *IDG*, there exists an edge of *DIDG* that links the isobarycenters of the vertices of each one of the triangles that share e.



Proof of Isomorphism

Let's call

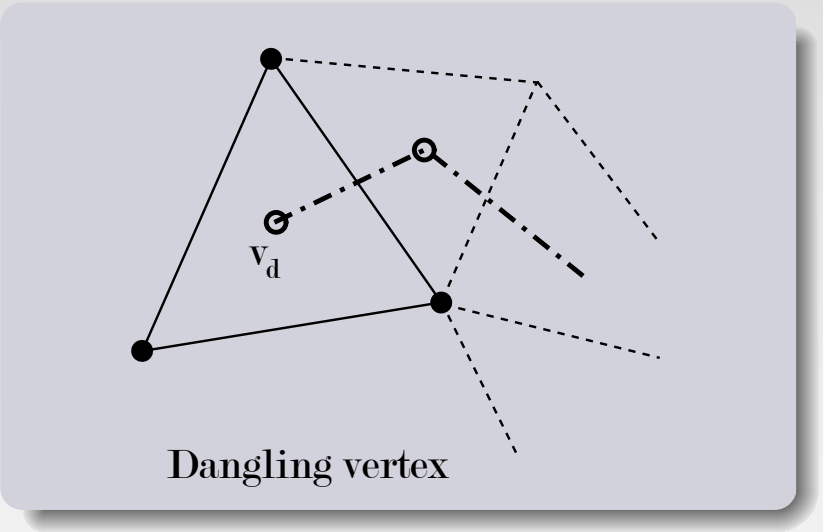
- **Ramification vertices**: the vertices *DIDG* that have a degree greater than two.
- **Dangling vertices**: the vertices of *DIDG* that have degree one.

Proof of Isomorphism

(a) The dangling vertices of *DIDG* correspond to triangles of *IDG* that have two of their edges on the boundary of *DIDG* (which are therefore adjacent). The corresponding vertices of the medial axis are the centers of maximal circles that touch two adjacent edges of the boundary of *IDG*.

Proof of Isomorphism

(a) The dangling vertices of *DIDG* correspond to triangles of *IDG* that have two of their edges on the boundary of *DIDG* (which are therefore adjacent). The corresponding vertices of the medial axis are the centers of maximal circles that touch two adjacent edges of the boundary of *IDG*.



Proof of Isomorphism

(b) The ramification vertices of *DIDG* correspond either to the common vertex of a set E of edges $e_1 ... e_k, k > 2$ of the boundary of *IDG*, or to the isobarycenters of the triangles of *DIDG* that have no edge in the boundary of *IDG*.

The later kind of ramification vertices (that we will call type I ramification vertices) correspond to Voronoi vertices that are at the same distance with respect to 3 distinct vertices on 3 distinct portions of the boundary of *IDG*.

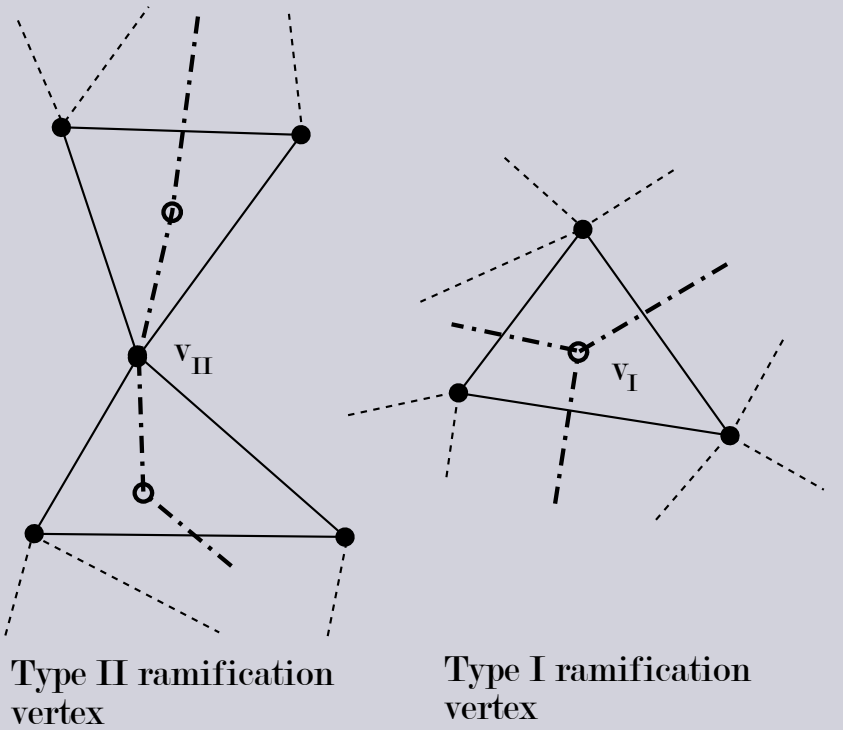
The earlier kind of ramification vertices (that we will call type II ramification vertices) correspond to singular points of the boundary of *IDG*.

Proof of Isomorphism

(b) The ramification vertices of *DIDG* correspond either to the common vertex of a set E of edges $e_1 ... e_k, k > 2$ of the boundary of *IDG*, or to the isobarycenters of the triangles of *DIDG* that have no edge in the boundary of *IDG*.

The later kind … l ramification vertices) corre… distance with respect to 3 di… ndary of *IDG*.

The earlier kin… e II ramification ve… ndary of *IDG*.



Proof of Isomorphism

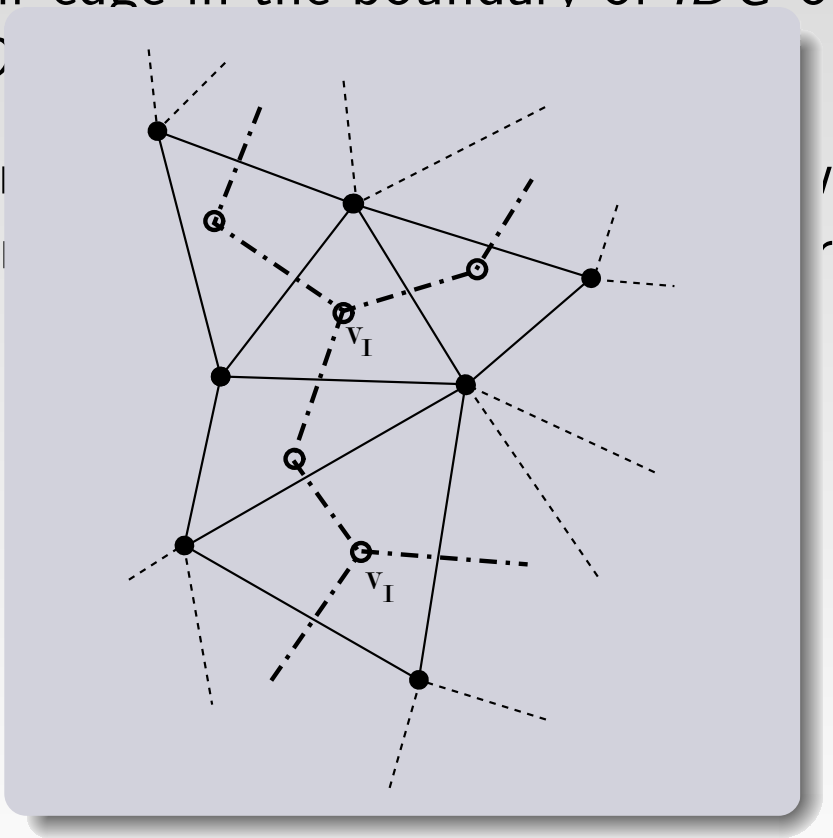
(c) The internal vertices on the paths between two type I ramification vertices of *DIDG* correspond to triangles of *IDG* that have one edge in the boundary of *IDG*. The edges in such paths link isobarycenters of triangles of *IDG* that have their edge in the boundary of *IDG* on different portions of the boundary of *IDG*.

These edges correspond to edges of the medial axis, whose points are the centers of maximal circles that touch two different portions of the boundary of *IDG*.

Proof of Isomorphism

(c) The internal vertices on the paths between two type I ramification vertices of *DIDG* correspond to triangles of *IDG* that have one edge in the boundary of *IDG*. The edges in such paths link isobarycenters of triangles of *IDG* that have their edge in the boundary of *IDG* on different portions of the boundary of *ID*

These edges corresp hose points are the centers of maximal ci rtions of the boundary of *IDG*.



Proof of Isomorphism

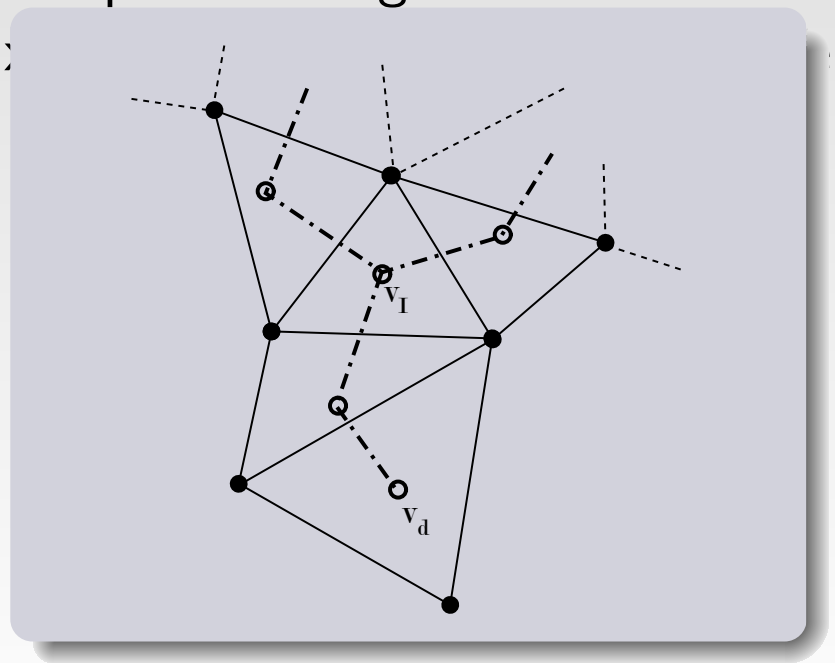
(d) The internal vertices on paths between one type I ramification vertices of *DIDG* and a dangling vertex of *DIDG* correspond to triangles of *IDG* that have one edge in the boundary of *IDG*. Again, the edges in such paths link isobarycenters of triangles of *IDG* that have their edge in the boundary of *IDG* on different portions of the boundary of *IDG*.

Again, these edges correspond to edges of the medial axis, whose points are the centers of maximal circles that touch two different portions of the boundary of *IDG*.

Proof of Isomorphism

(d) The internal vertices on paths between one type I ramification vertices of *DIDG* and a dangling vertex of *DIDG* correspond to triangles of *IDG* that have one edge in the boundary of *IDG*. Again, the edges in such paths link isobarycenters of triangles of *IDG* that have their edge in the boundary of *IDG* on different portions of the boundary of *IDG*.

Again, these edges correspond to edges of the medial axis, whose points are the centers of ma rent portions of the boundary of *IDG*.



Proof of Isomorphism

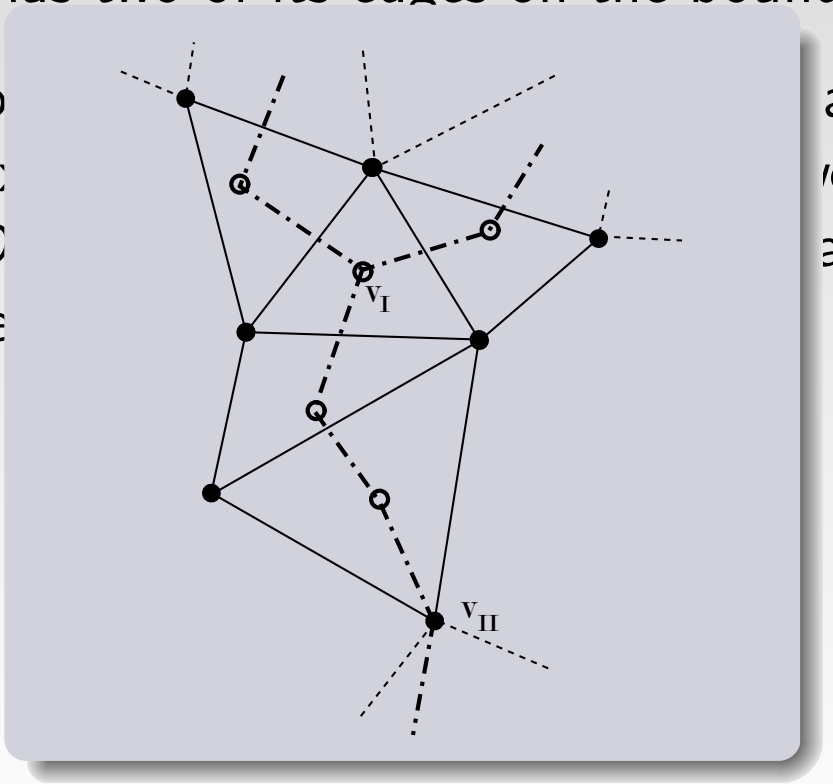
(e)The internal vertices on paths between a type II ramification vertex and a type I ramification vertex correspond to triangles of *IDG* that have one edge in the boundary of *IDG*, except for the vertex that is connected to the type II ramification vertex by a single edge, which corresponds to a triangle of *IDG* that has two of its edges on the boundary of *DIDG*.

Again, these edges correspond to edges of the medial axis, whose points are the centers of maximal circles that touch either two different portions of the boundary of *IDG*, or two portions of the boundary of *IDG* that have a common singular vertex.

Proof of Isomorphism

(e)The internal vertices on paths between a type II ramification vertex and a type I ramification vertex correspond to triangles of *IDG* that have one edge in the boundary of *IDG*, except for the vertex that is connected to the type II ramification vertex by a single edge, which corresponds to a triangle of *IDG* that has two of its edges on the boundary of *DIDG*.

Again, these edges co axis, whose points are the centers of ma o different portions of the boundary of *ID* ary of *IDG* that have a common singular ve



Proof of Isomorphism

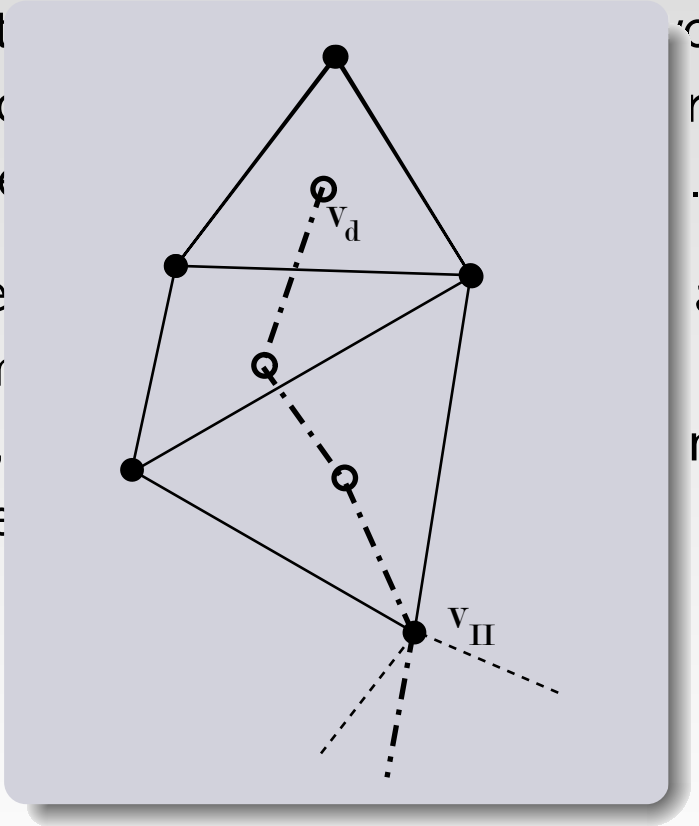
(f) The internal vertices on paths between a type II ramification vertex of *DIDG* and a dangling vertex of *DIDG* correspond to triangles of *IDG* that have one edge in the boundary of *IDG*, except for the vertex that is connected to the type II ramification vertex by a single edge, which corresponds either to a triangle of *IDG* that has two of its edges on the boundary of *DIDG*, or to the circumcircle of the vertices of a complete subgraph K of 4 or more cocircular vertices of *IDG*.

Again, these edges correspond to edges of the medial axis, whose points are the centers of maximal circles that touch either two different portions of the boundary of *IDG*, or two portions of the boundary of *IDG* that have a common singular vertex.

Proof of Isomorphism

(f) The internal vertices on paths between a type II ramification vertex of *DIDG* and a dangling vertex of *DIDG* correspond to triangles of *IDG* that have one edge in the boundary of *IDG*, except for the vertex that is connected to the type II ramification vertex by a single edge, which corresponds either to a t ...o of its edges on the boundary of *DIDG*, or t... rtices of a complete subgraph *K* of 4 or mor... .

Again, these edges corre... al axis, whose points are the centers of maxim... two different portions of the boundary of *IDG*, ...ndary of *IDG* that have a common singular verte...



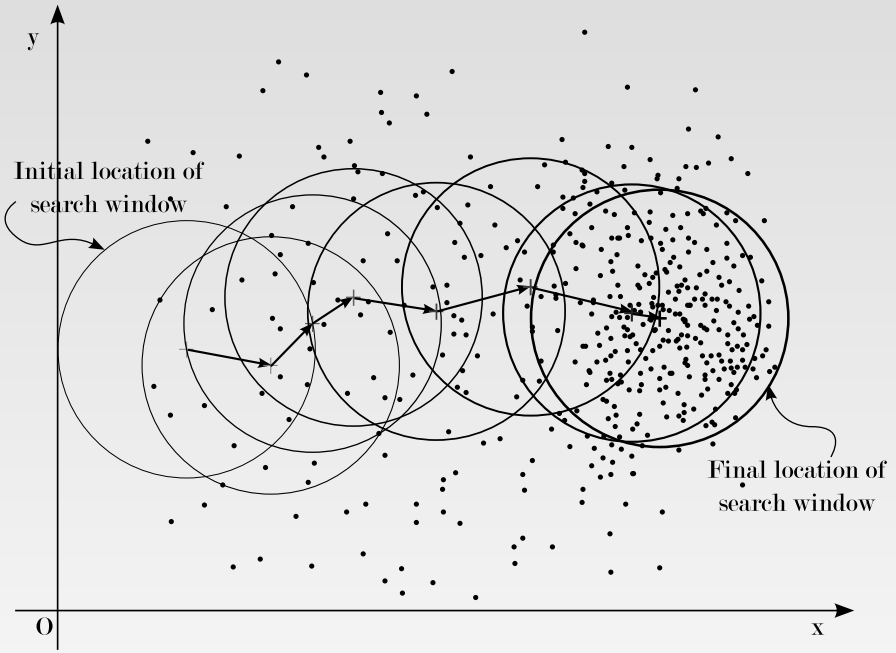
Automated Approach to Skeletonization

Following is our general approach to skeletonization:

1. Segment a color image into prominent objects.
2. Ask the user if he or she wants to process all the objects independently (automatic process) or select an object (semi-automatic process).
3. Collect sample points for each object to be processed.
4. Construct the Delaunay triangulation and its dual from the sample points.
5. Extract the medial axis using rules described before.

Image Segmentation using Mean Shift Algorithm

Mode seeking using mean shift algorithm



The final window location gives the local maxima of the distribution

Image segmentation

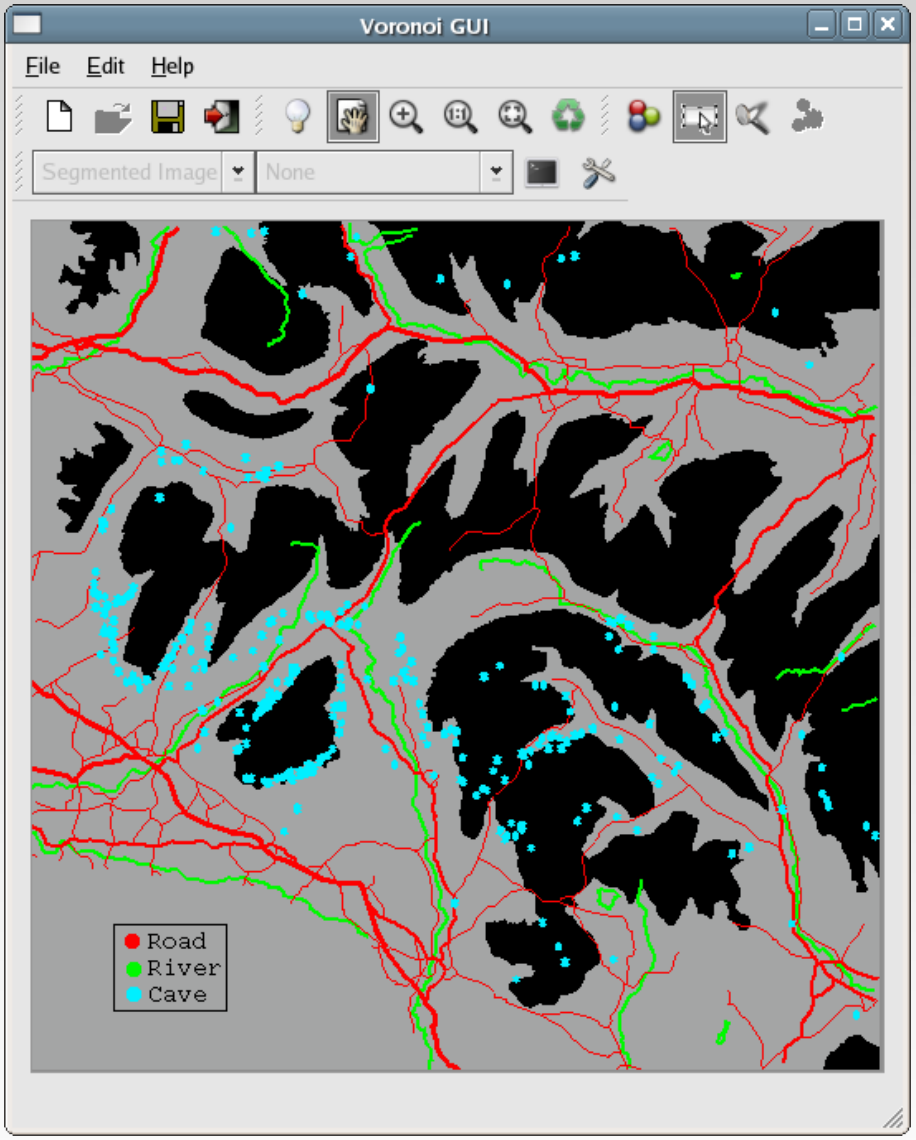
Mean shift algorithm can be applied to segment images [4]. Following general procedure is used:

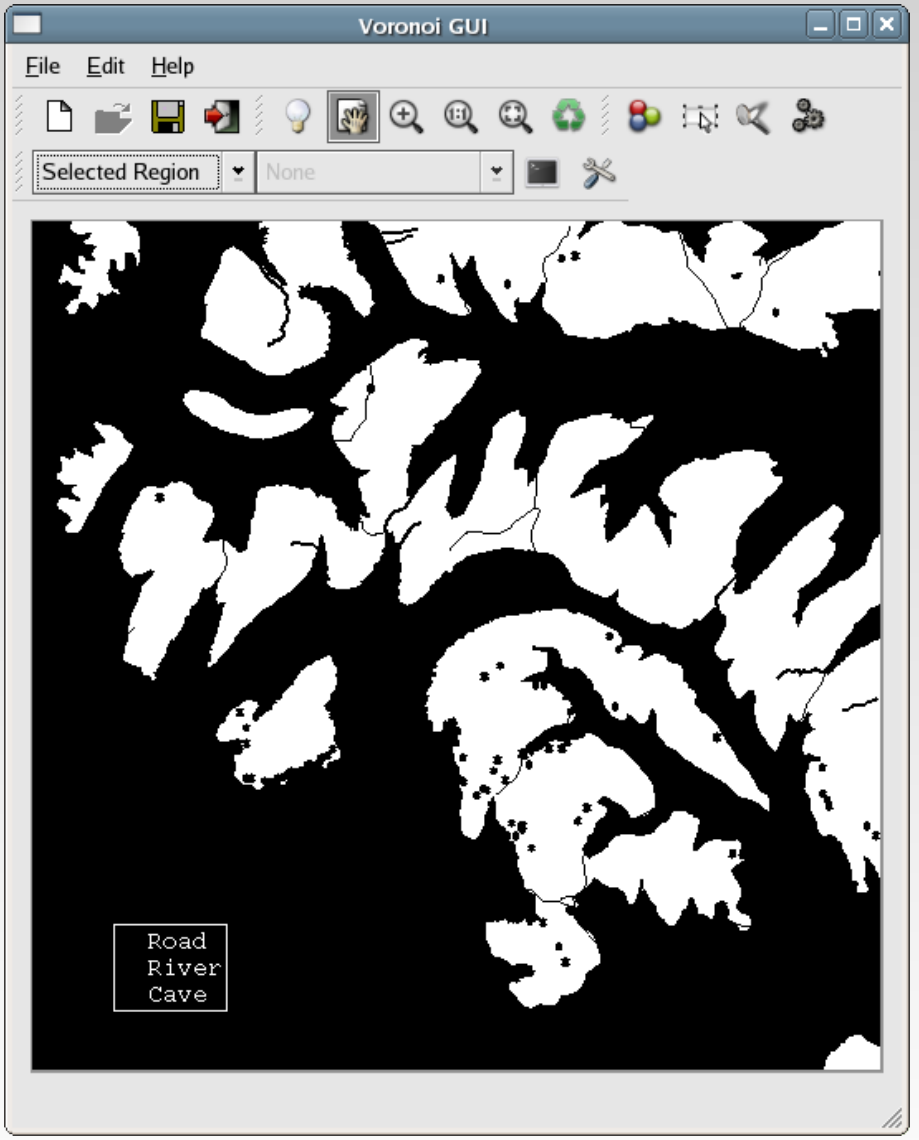
1. Definition of segmentation parameters
2. Definition of search window
3. Mean shift algorithm
4. Removal of detected feature
5. Iterations
6. Determining initial feature palette
7. Determining final feature palette
8. Postprocessing

Image Segmentation using Mean Shift Algorithm

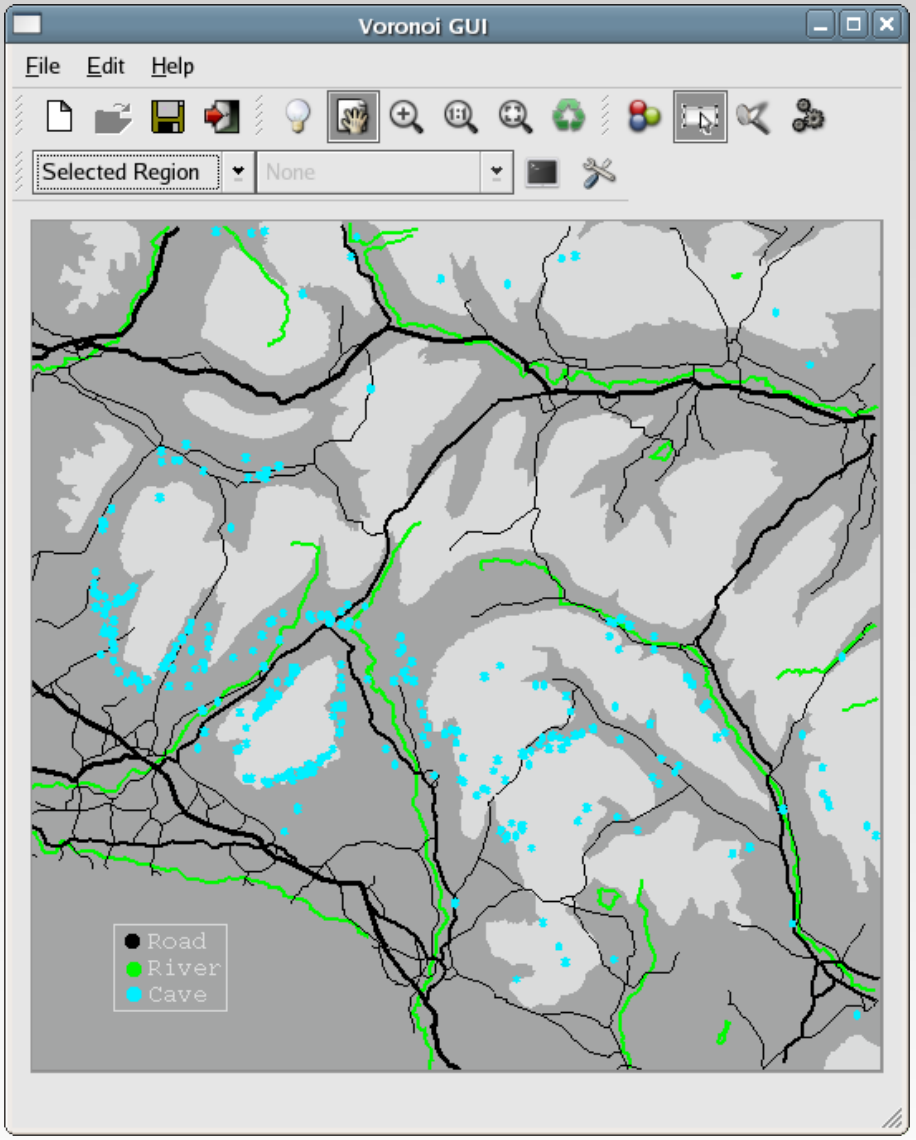
Image segmentation results

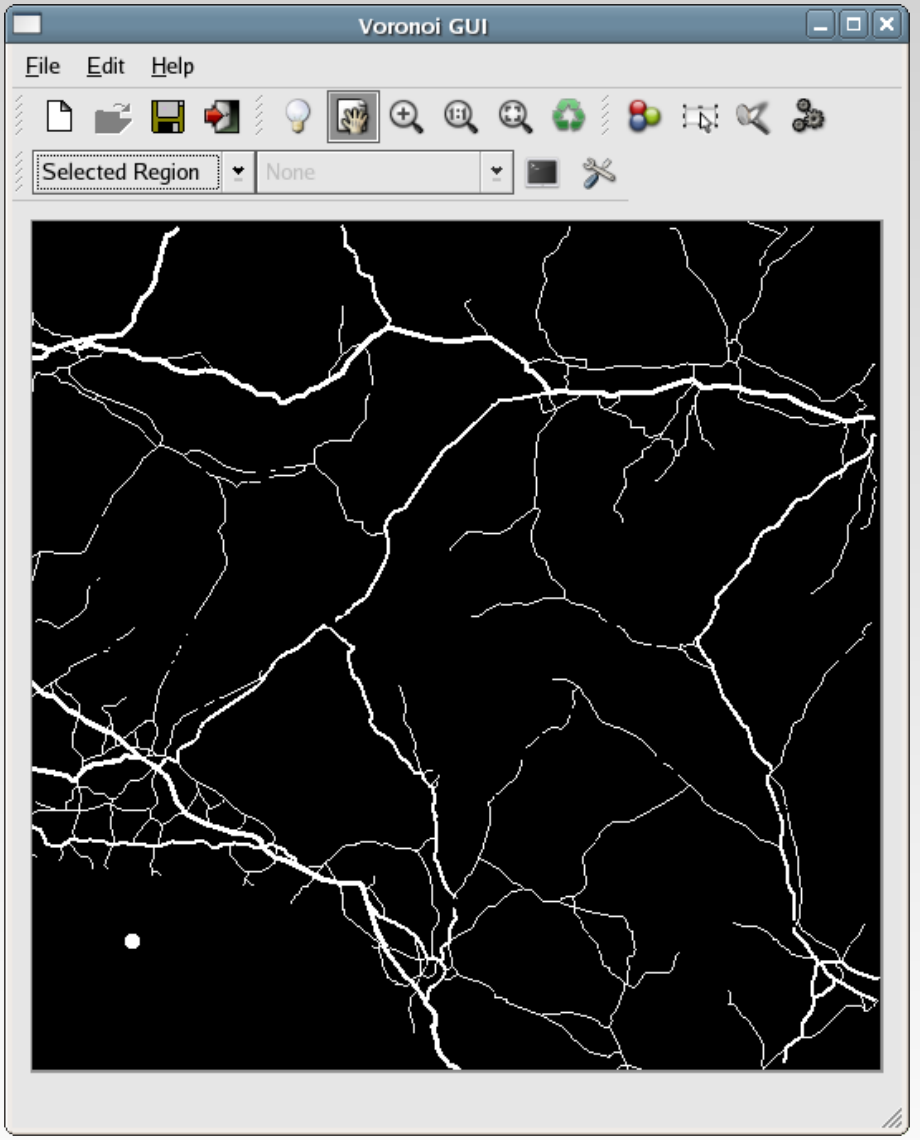
Object Selection from Segmented Image





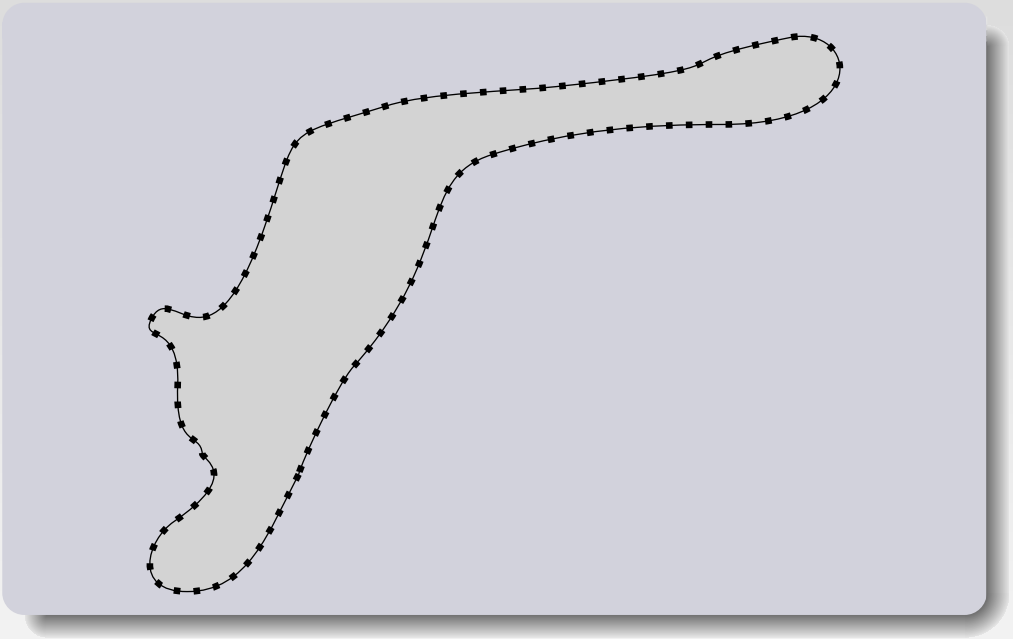
Object Selection from Segmented Image





Sampling the Object Boundary

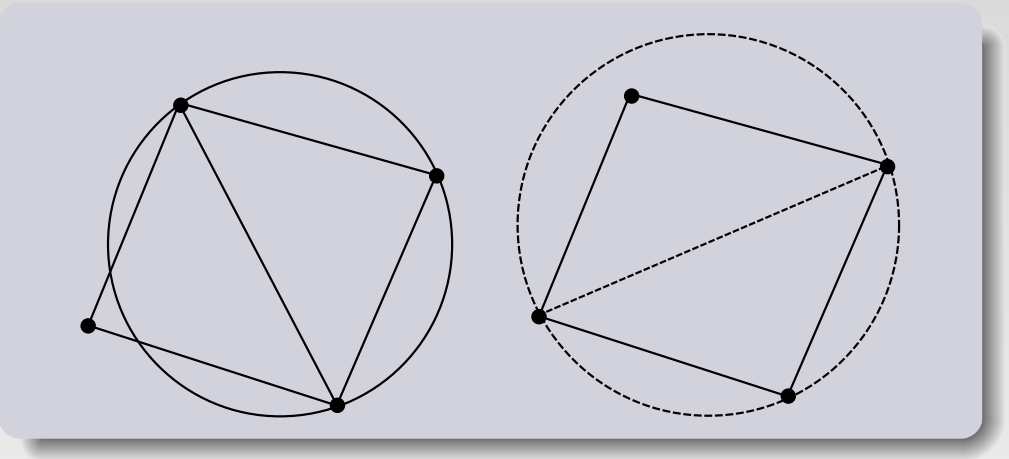
A selected object must be sampled before it any processing.
The sampled points are then used to compute the Delaunay triangulation and the Voronoi diagram.



This can be achieved by edge detection. Edge pixels are calculated using morphological edge detection from the binary image of the selected object.

Computation of the Delaunay Triangulation

The **Delaunay triangulation** of a set of points is a triangulation such that no point in P is inside the circumcircle of any triangle of the triangulation.

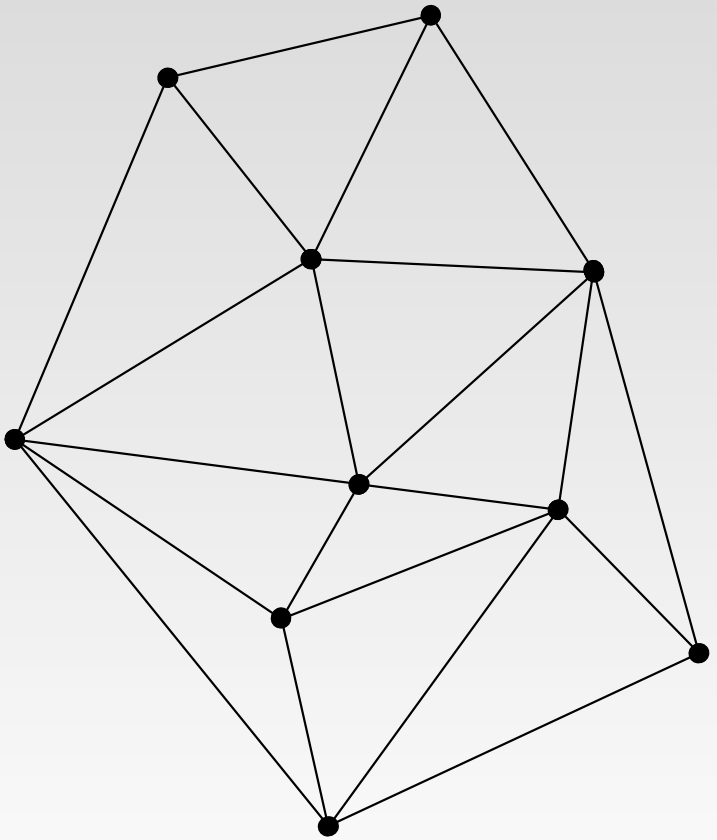


Incircle criterion:

$$H(p_a, p_b, p_c, p_d) = \begin{vmatrix} 1 & x_a & y_a & x_a^2 + y_a^2 \\ 1 & x_b & y_b & x_b^2 + y_b^2 \\ 1 & x_c & y_c & x_c^2 + y_c^2 \\ 1 & x_d & y_d & x_d^2 + y_d^2 \end{vmatrix}$$

Computation of the Delaunay Triangulation

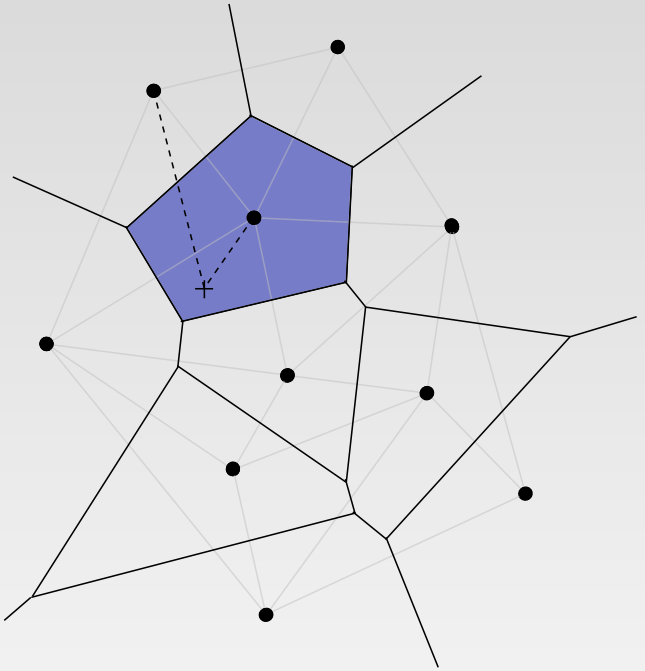
The **Delaunay triangulation** of a set of points is a triangulation such that no point in P is inside the circumcircle of any triangle of the triangulation.



Computed using the **incremental algorithm**.

Computation of the Voronoi Diagram

Voronoi diagram is an irregular (polygonal) tessellation of space that adapts to spatial objects. They are synthesis of raster and vector model.

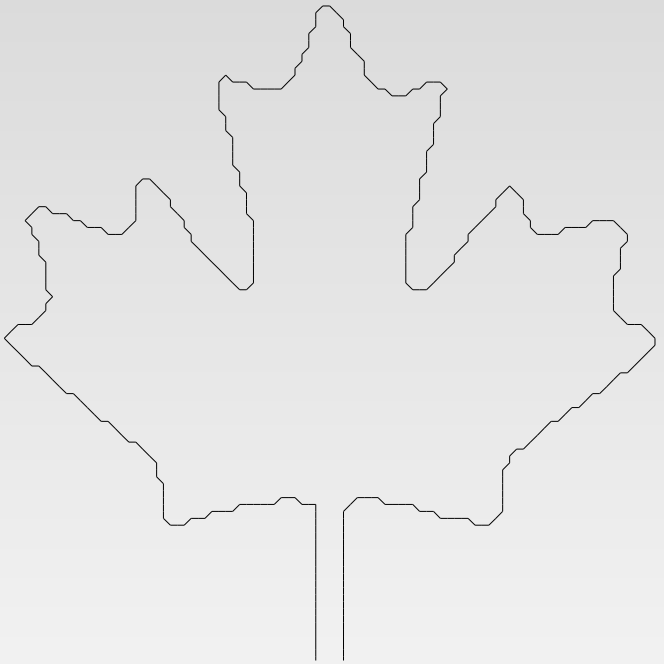


The Voronoi region of point p_i is given by:

$$V(p_i) = \{x | \, \|x - x_i\| \leq \|x - x_j\| \text{ for } j \neq i, j \in I_n\}$$

Extraction of Crust (Boundary)

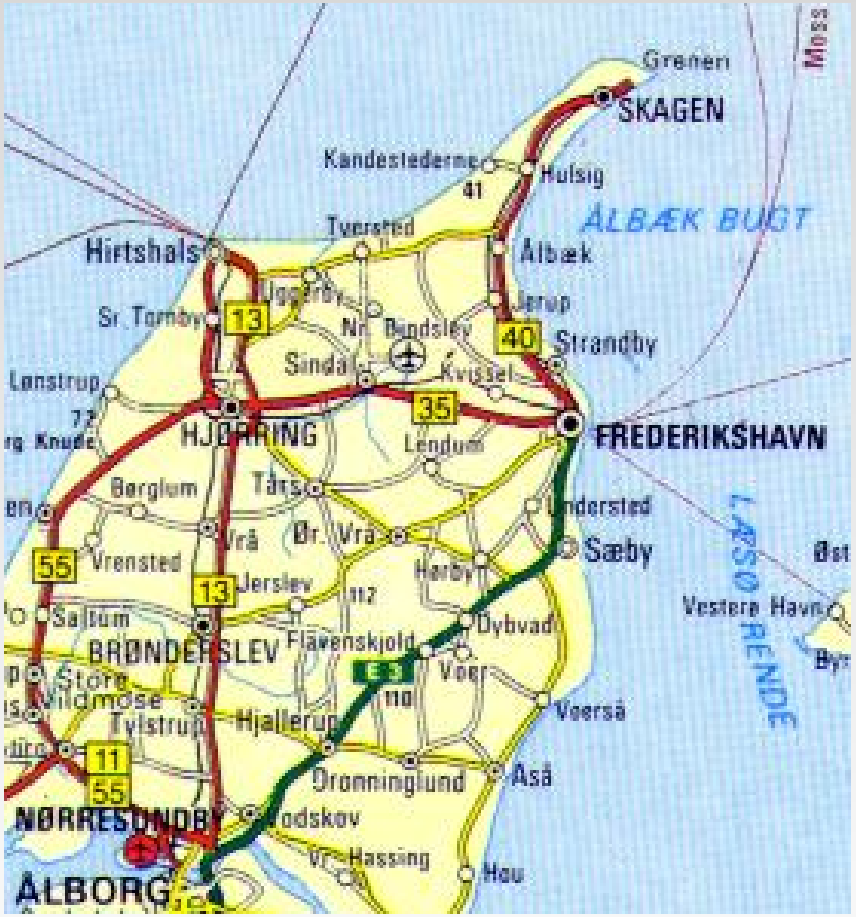
Any Delaunay edge that has a circle that does not contain the vertices of its dual Voronoi edge belongs to the **Crust**[2, 3].

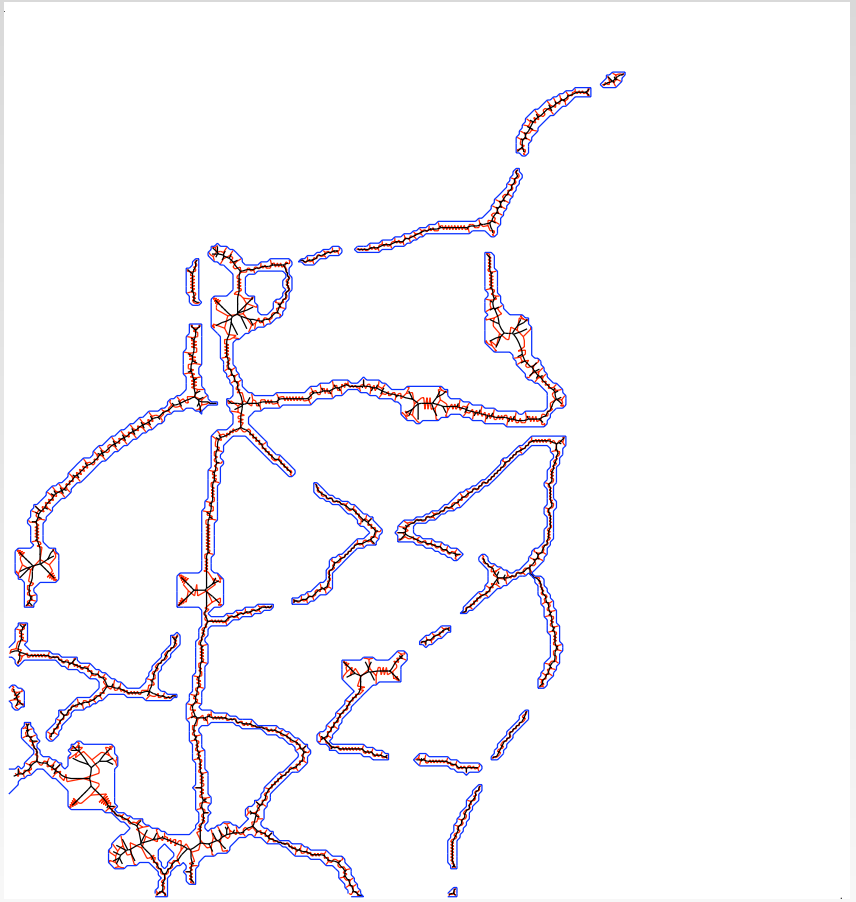


The crust forms the *IDG*. We obtain the *DIDG* by application of our set of rules to *IDG*.

Results

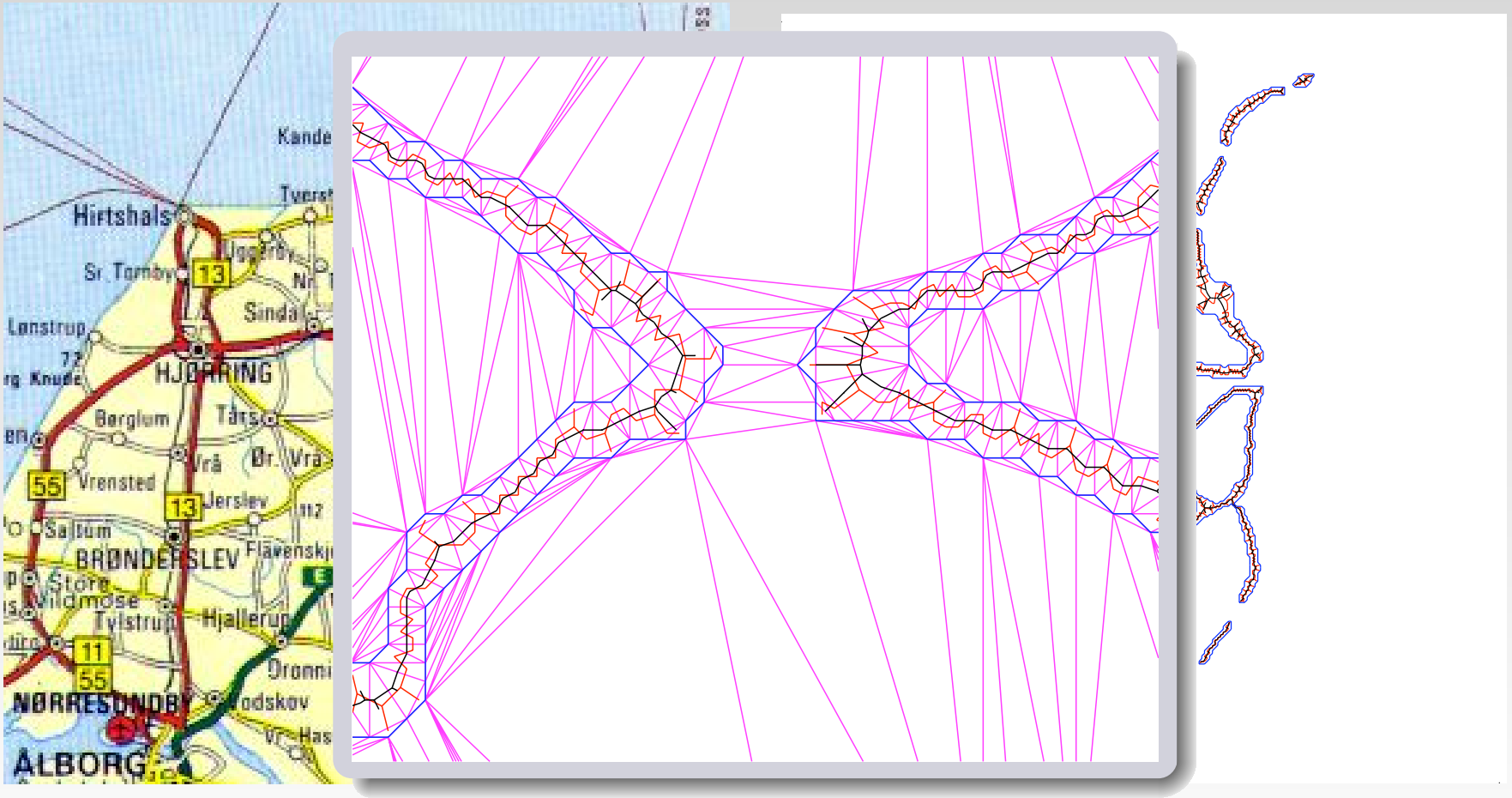
Denmark Road Network





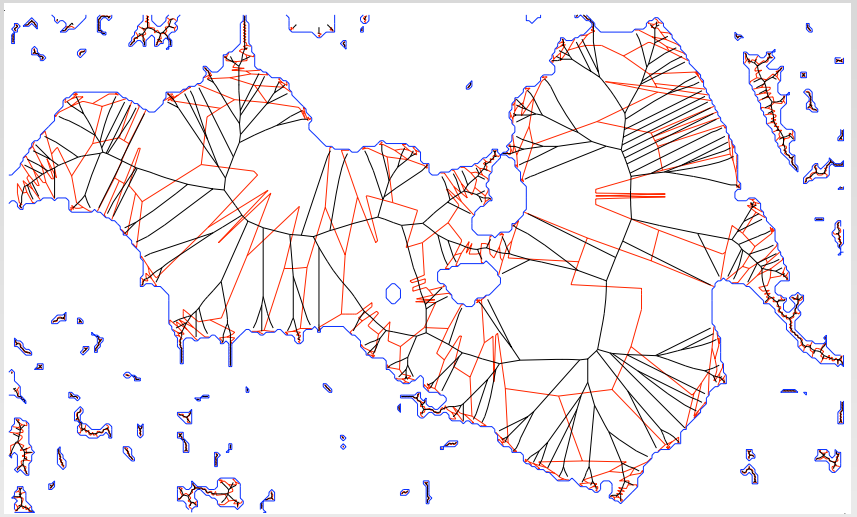
Results

Denmark Road Network



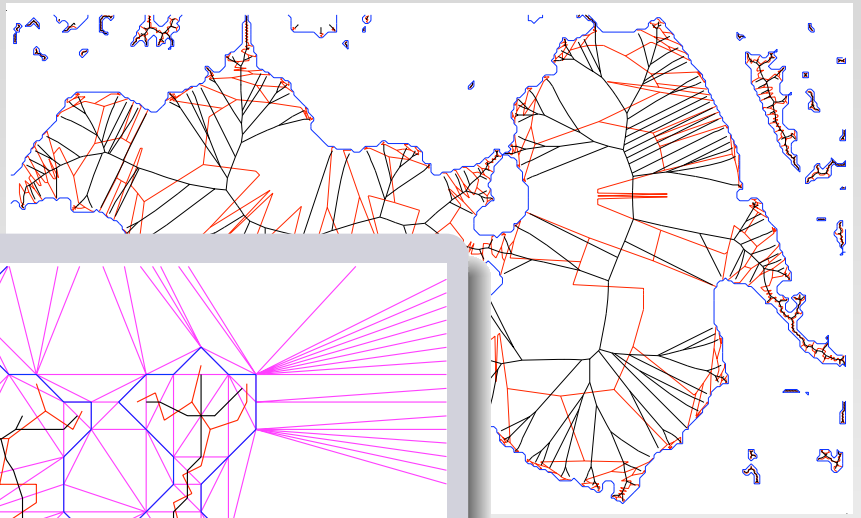
Results

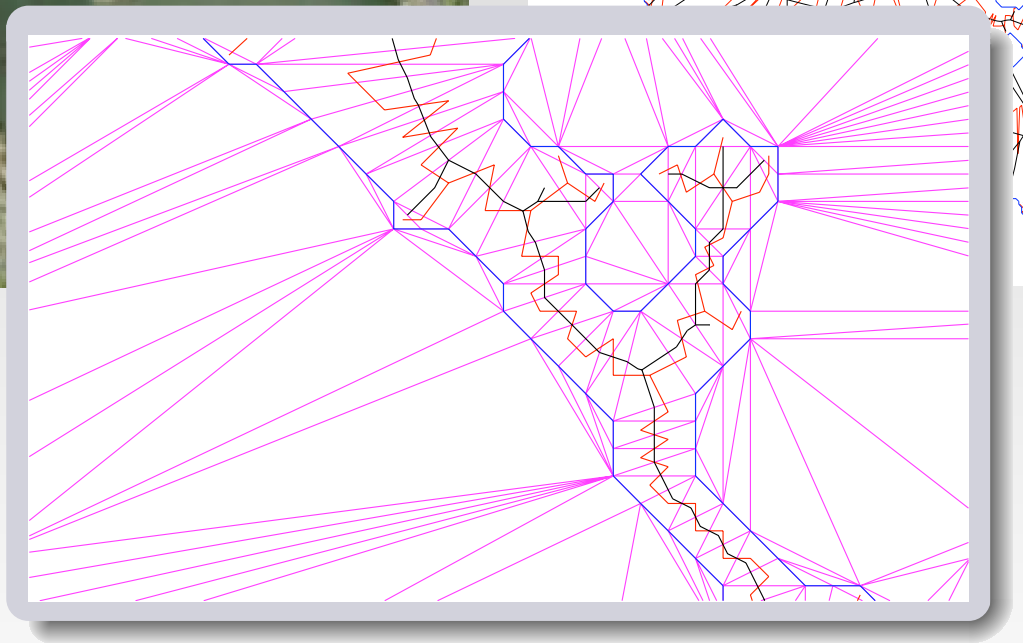
Lake Lyngby, Denmark



Results

Lake Lyngby, Denmark





Conclusions

- We have shown medial axis computation by utilising the graph isomorphism between the medial axis and a dual of the Delaunay graph.
- We outlined a methodology to automate various parts of digitization from scanned maps has been suggested.
- We developed an interactive software application to computed medial axis of objects from digital images and digitize features.
- We show Applicability of the designed method to two types of digital images, scanned maps and satellite images.

Thank You

Bibliography

[1] N. Amenta, M. Bern, and D. Eppstein, "The crust and the β -skeleton: Combinatorial curve reconstruction," Graphical models and image processing: GMIP, vol. 60, no. 2, pp. 125–135, 1998.

[2] C. M. Gold, "Crust and anti-crust: A one-step boundary and skeleton extraction algorithm," in Symposium on Computational Geometry. New York, NY, USA: ACM Press, 1999, pp. 189–196.

[3] C. M. Gold and D. Thibault, "Map generalization by skeleton retraction," in Proceedings of the 20th International Cartographic Conference (ICC), Beijing, China, August 2001, pp. 2072–2081.

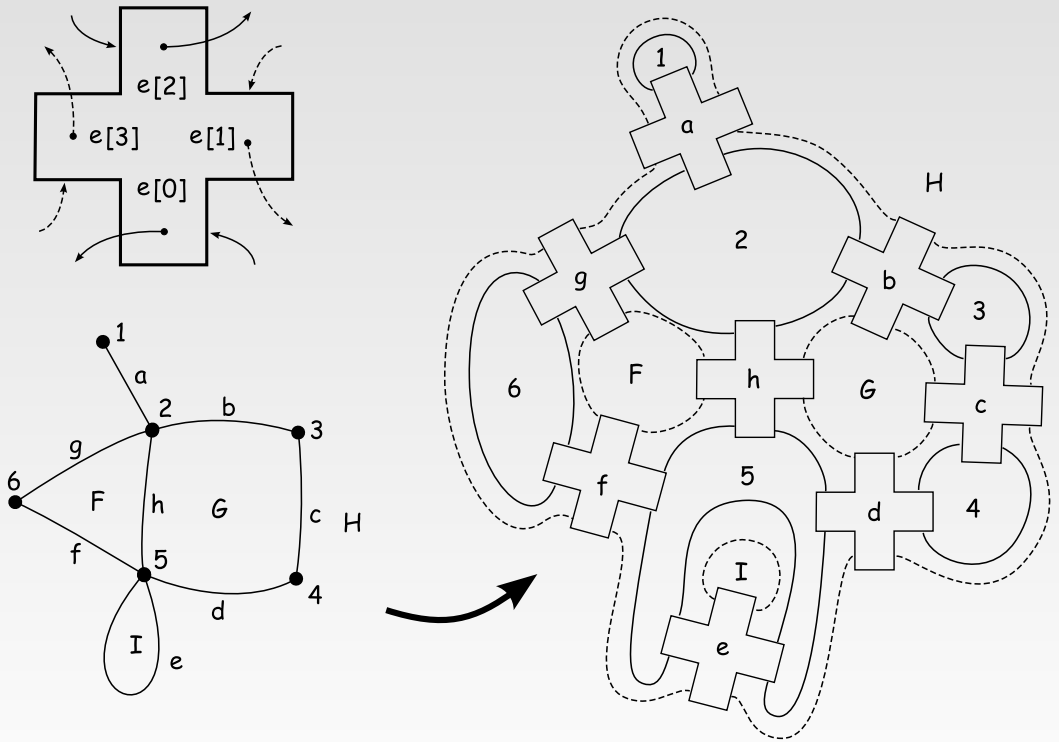
[4] D. Comaniciu and P. Meer, "Robust analysis of feature spaces: color image segmentation," in Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97). Washington, DC, USA: IEEE Computer Society, 1997, pp. 750–755.

[5] P. Green and R. Sibson, "Computing dirichlet tessellations in the plane," The Computer Journal, vol. 21, no. 2, pp. 168–173, 1977.

(Please refer to the full paper for complete bibliography.)

Quad-Edge Data Structure

- A Quad-edge simultaneously represents a graph and its dual.
- Dual of a mesh exchanges faces and vertices.
- The edges of the graph are directed and there are four such edges representing two symmetric edges from both the graph and its dual.



Quad-Edge Data Structure

Two atomic operators for navigation:

- **Rot()** points to a 90^o counterclockwise rotated edge.
- **Onext()** points to the next counterclockwise edge sharing the same origin with the current edge.

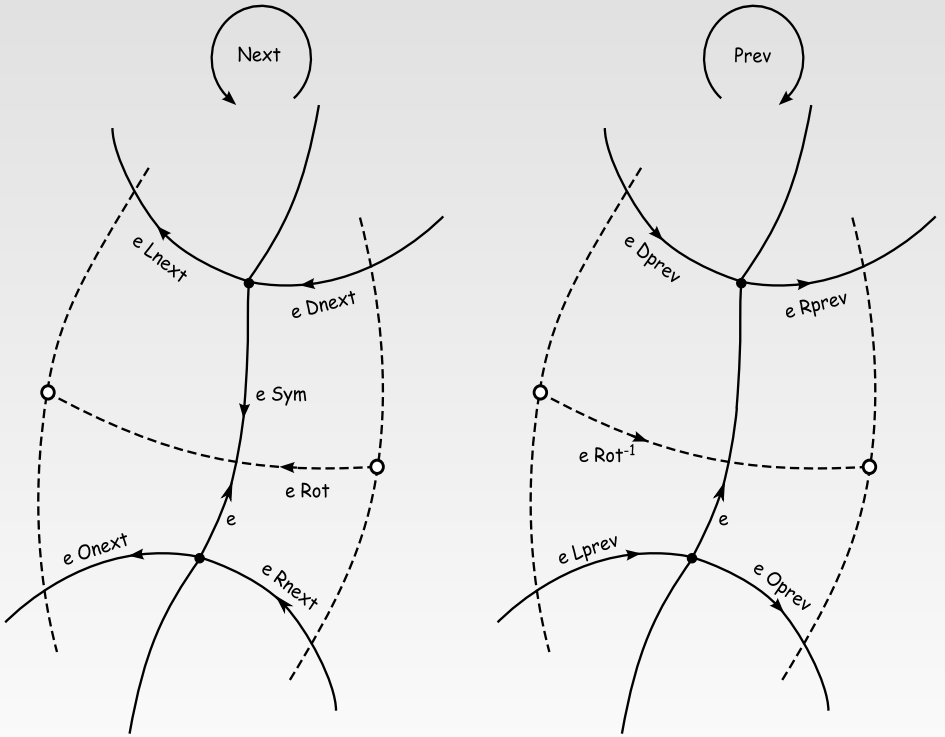
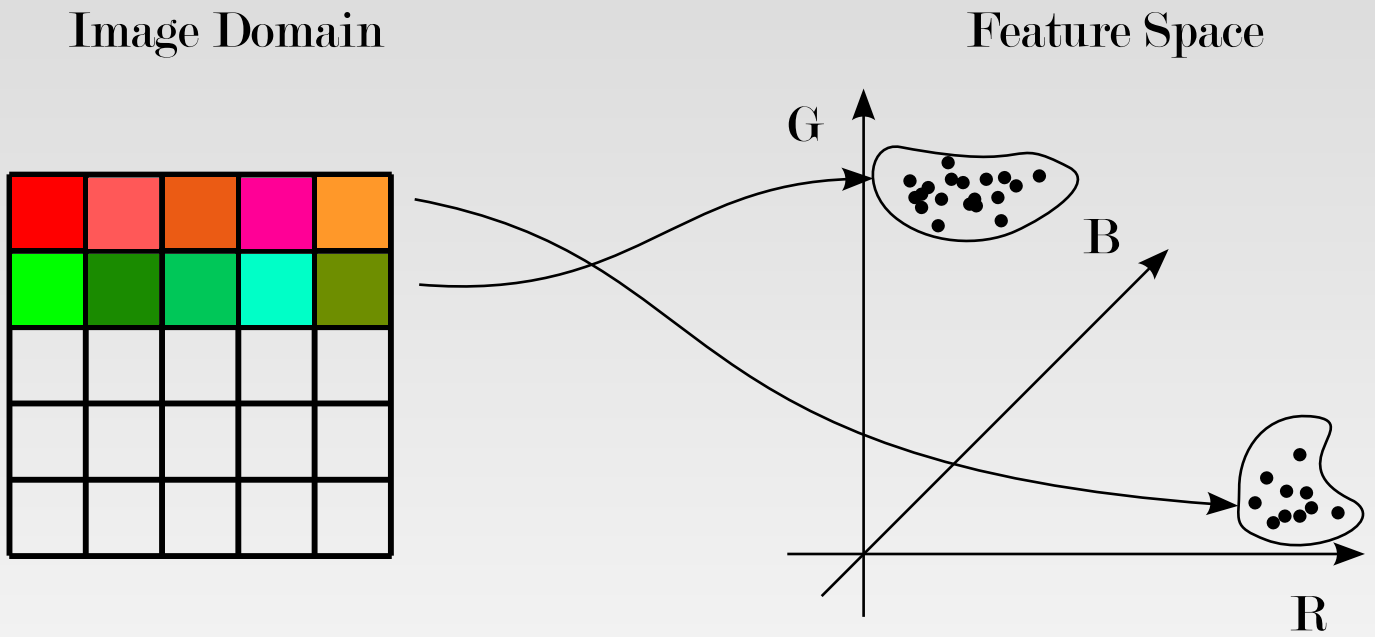


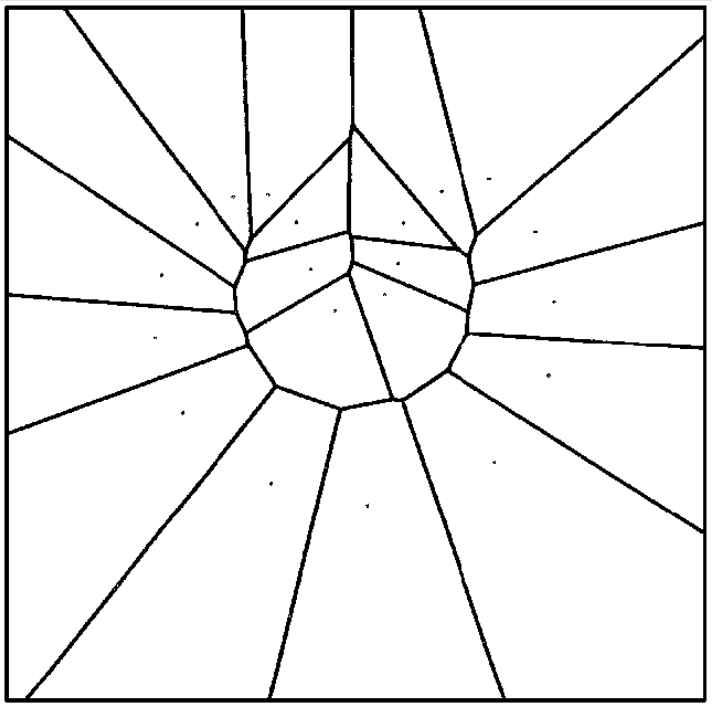
Image Space to Feature Space

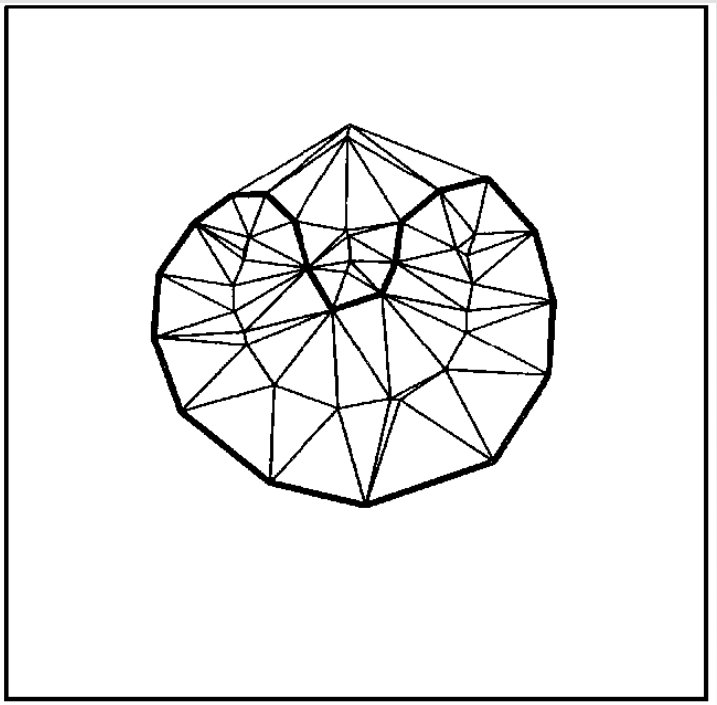
Clustering is done in **Feature Space** using **Mean Shift algorithm**. Conversion from Image Space to Feature Space is performed before clustering.



Crust Condition

[1]:
Let S be a finite set of points in a plane, and let V be the vertices of the Voronoi diagram of S. Let S' be the union $S \cup V$, and consider the Delaunay triangulation of S'. An edge of the Delaunay triangulation of S' belongs to the *crust* of S if both of its endpoints belong to S.





Crust Condition

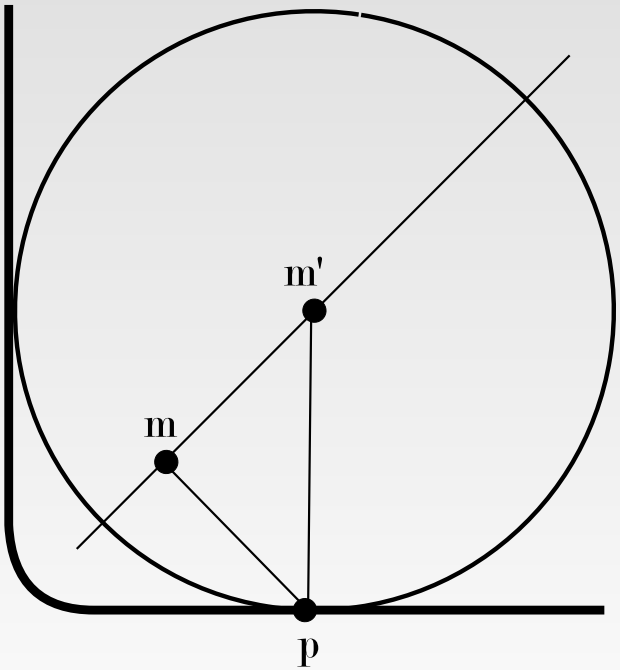
One step formulation by Christopher Gold:

$$(s-q)\cdot(s-r) * (p-q)\cdot(p-r) \geq -(s-r)\cdot v * (p-q)\cdot v$$

Where, edge(q, r) is the Delaunay edge under the test and edge(p, s) froms the dual Voronoi edge. v is a vector 90^o clockwise from $(r-q)$.

Sampling Condition

Let F be a smooth curve and $S \subset F$ be a finite set of sample points on F. The sampling should be done such that the distance from any point p on F to the nearest sample $s \in S$ is at most a constant factor r times the *local feature size* at p, which is defined as the distance from p to the *medial axis* of F.



Distance $d(p, m)$ defines the local feature size.