

ACCV 2016 Tutorial on Digital Geometry Processing: Extracting High Quality Geometric Features - Part II -

Bertrand Kerautret¹

(work in collaboration with Jacques-Olivier Lachaud², Mouhammad Said²,
and Miguel Colom³)

¹Université de Lorraine, LORIA CNRS, ADAGlo team , UMR 7503, Vandoeuvre-lès-Nancy, F-54506, France

²LAMA (UMR CNRS 5127), Université Savoie Mont Blanc, F-73376, France

³CMLA ENS Cachan, France

13th Asian Conference on Computer Vision
Nov 20-24, 2016, Taipei, Taiwan, ROC



Overview of the presentation - Part II -

- 1 1. Introduction to the Meaningful Scale Detection
 - 1.1 Main idea of the meaningful scale
 - 1.2 Meaningful Scale Profile and Noise Detection
 - 1.3 Experiments and new applications
- 2 2. Extension to the Meaningful Thickness
 - 2.2 Meaningful Thickness profile
 - 2.3 Experiments, comparisons and applications
 - 2.4 Computing the meaningful thickness in the DGtal Library
- 3 3. Highlighting Reproducible Research
 - 3.1 Image Processing On Line Presentation: principle and current form
 - 3.2 Structure of an IPOL demonstration

Overview of the presentation

- 1 1. Introduction to the Meaningful Scale Detection
 - 1.1 Main idea of the meaningful scale
 - 1.2 Meaningful Scale Profile and Noise Detection
 - 1.3 Experiments and new applications
- 2 2. Extension to the Meaningful Thickness
 - 2.2 Meaningful Thickness profile
 - 2.3 Experiments, comparisons and applications
 - 2.4 Computing the meaningful thickness in the DGtal Library
- 3 3. Highlighting Reproducible Research
 - 3.1 Image Processing On Line Presentation: principle and current form
 - 3.2 Structure of an IPOL demonstration

Part I. Introduction to the Meaningful Scale Detection

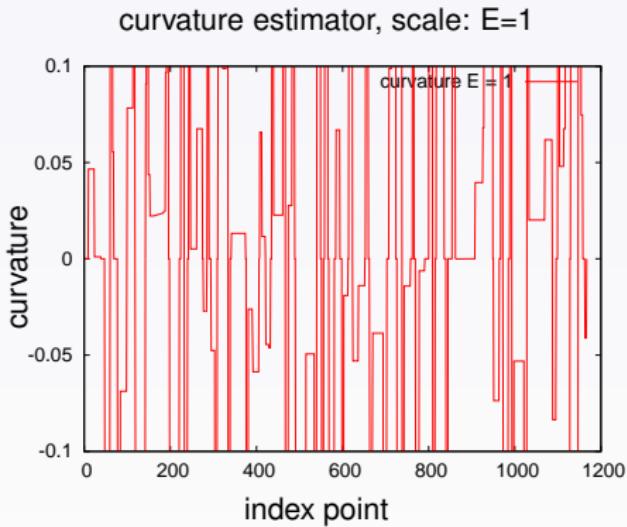
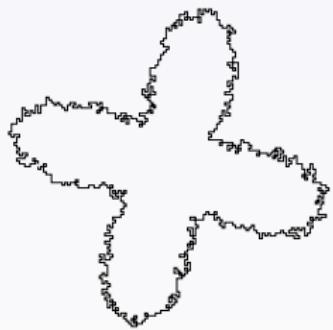
Noise induces generally the tuning of parameters

- Output quality/accuracy depends on the choice of the parameter.

Part I. Introduction to the Meaningful Scale Detection

Noise induces generally the tuning of parameters

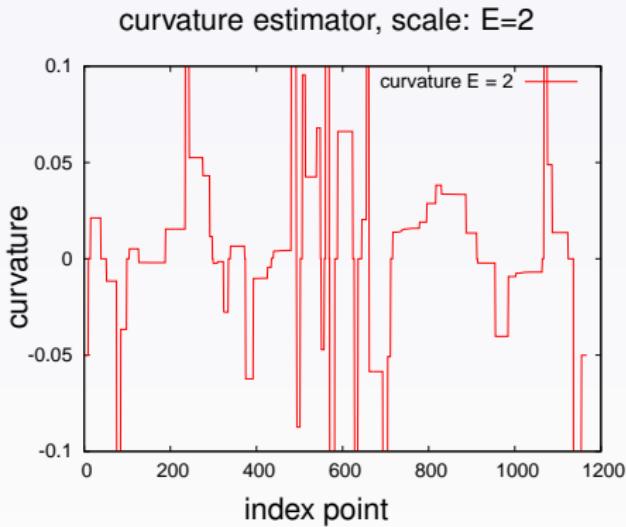
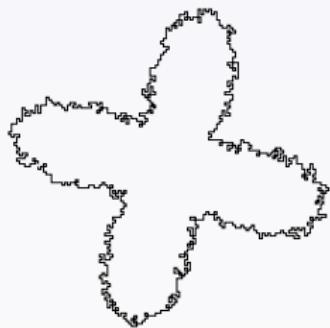
- Output quality/accuracy depends on the choice of the parameter.
- Example: found the best scale to analyze discrete shape.



Part I. Introduction to the Meaningful Scale Detection

Noise induces generally the tuning of parameters

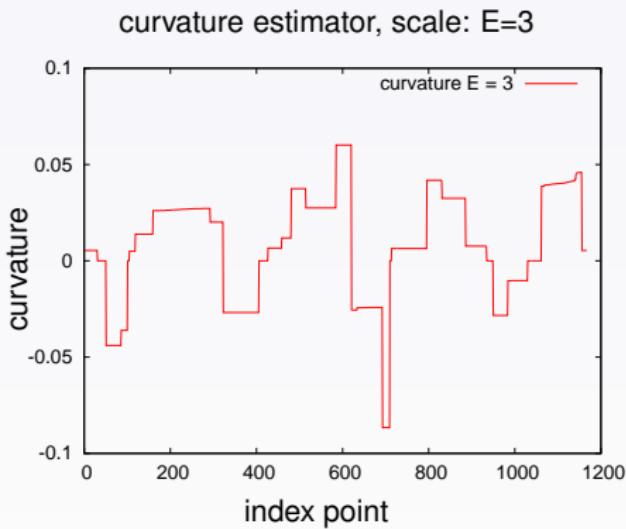
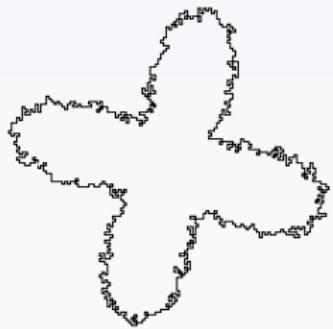
- Output quality/accuracy depends on the choice of the parameter.
- Example: found the best scale to analyze discrete shape.



Part I. Introduction to the Meaningful Scale Detection

Noise induces generally the tuning of parameters

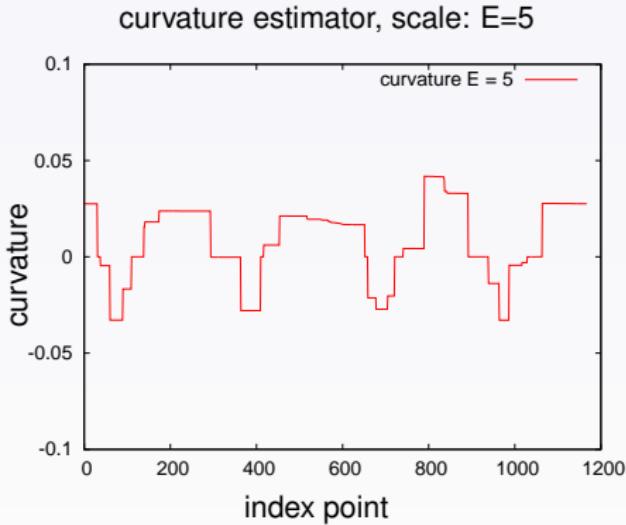
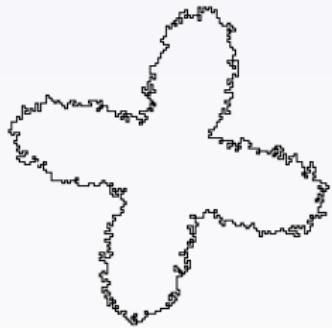
- Output quality/accuracy depends on the choice of the parameter.
- Example: found the best scale to analyze discrete shape.



Part I. Introduction to the Meaningful Scale Detection

Noise induces generally the tuning of parameters

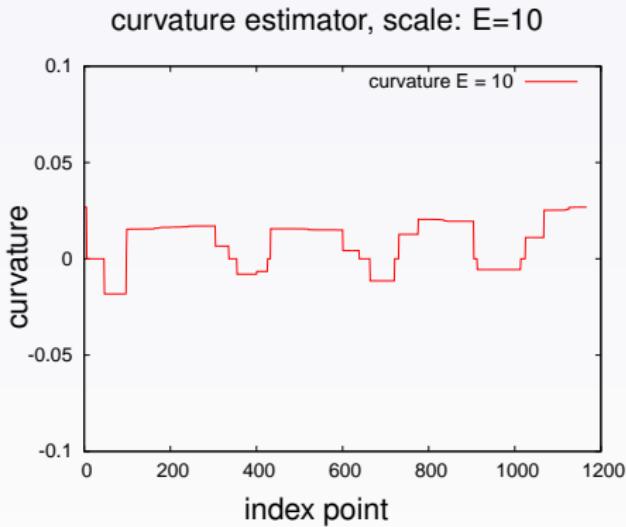
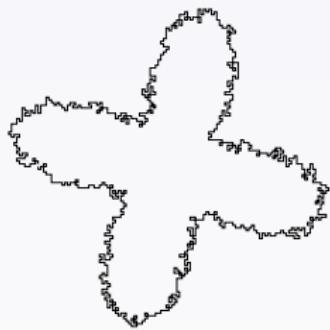
- Output quality/accuracy depends on the choice of the parameter.
- Example: found the best scale to analyze discrete shape.



Part I. Introduction to the Meaningful Scale Detection

Noise induces generally the tuning of parameters

- Output quality/accuracy depends on the choice of the parameter.
- Example: found the best scale to analyze discrete shape.

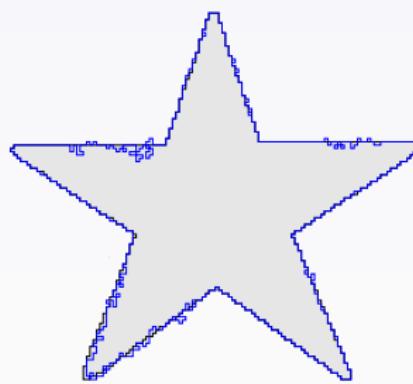


Part I. Introduction to the Meaningful Scale Detection

Noise induces generally the tuning of parameters

- Output quality/accuracy depends on the choice of the parameter.
- Example: found the best scale to analyze discrete shape.
- Algorithm parameter tuning.

Denoising approach [Hoang et al.,2011]
(fidelity parameter $\epsilon = 10$)

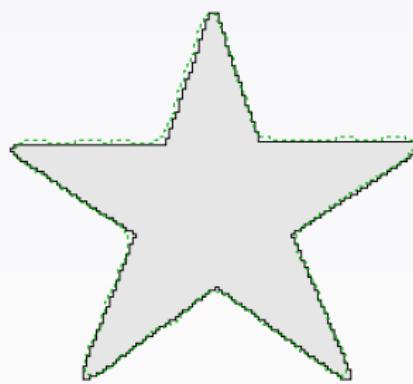


Part I. Introduction to the Meaningful Scale Detection

Noise induces generally the tuning of parameters

- Output quality/accuracy depends on the choice of the parameter.
- Example: found the best scale to analyze discrete shape.
- Algorithm parameter tuning.

Denoising approach [Hoang et al.,2011]
(fidelity parameter $\epsilon = 20$)

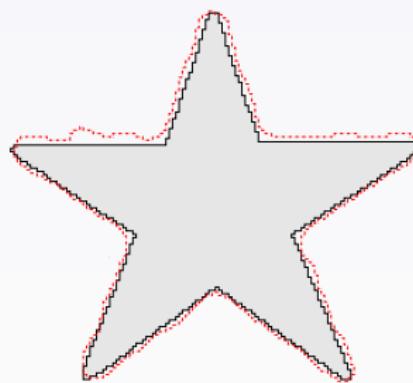


Part I. Introduction to the Meaningful Scale Detection

Noise induces generally the tuning of parameters

- Output quality/accuracy depends on the choice of the parameter.
- Example: found the best scale to analyze discrete shape.
- Algorithm parameter tuning.

Denoising approach [Hoang et al.,2011]
(fidelity parameter $\epsilon = 30$)

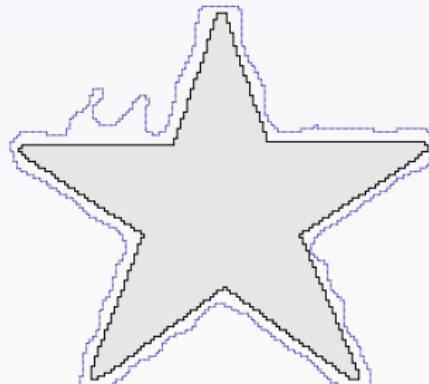


Part I. Introduction to the Meaningful Scale Detection

Noise induces generally the tuning of parameters

- Output quality/accuracy depends on the choice of the parameter.
- Example: found the best scale to analyze discrete shape.
- Algorithm parameter tuning.

Denoising approach [Hoang et al.,2011]
(fidelity parameter $\epsilon = 50$)



Existing works in local noise estimation (1)

- Multi-scale image analysis [Koenderink 84, Perona-Malik 90] ...

Existing works in local noise estimation (1)

- Multi-scale image analysis [Koenderink 84, Perona-Malik 90] ...
- Global optimization (filter scale for edge detection) [Jeong Kim 92]

Existing works in local noise estimation (1)

- Multi-scale image analysis [Koenderink 84, Perona-Malik 90] ...
- Global optimization (filter scale for edge detection) [Jeong Kim 92]
- Local optimization:
 - minimum reliable scale by fitting a blurred step-model of a contour [Elder Zucker 98],
 - non-linear Gaussian filters [Barash 02] or statistical method [Kervrann 04].
 - constrained anisotropic diffusion with discontinuity measure [Chen 05]

Existing works in local noise estimation (1)

- Multi-scale image analysis [Koenderink 84, Perona-Malik 90] ...
- Global optimization (filter scale for edge detection) [Jeong Kim 92]
- Local optimization:
 - minimum reliable scale by fitting a blurred step-model of a contour [Elder Zucker 98],
 - non-linear Gaussian filters [Barash 02] or statistical method [Kervrann 04].
 - constrained anisotropic diffusion with discontinuity measure [Chen 05]

Main drawbacks:

- Objective (image restoration, edge detection) is different.
- Adapted to grey-level or color images, not B & W (or discrete contours).

Existing works in local noise estimation (1)

- Multi-scale image analysis [Koenderink 84, Perona-Malik 90] ...
- Global optimization (filter scale for edge detection) [Jeong Kim 92]
- Local optimization:
 - minimum reliable scale by fitting a blurred step-model of a contour [Elder Zucker 98],
 - non-linear Gaussian filters [Barash 02] or statistical method [Kervrann 04].
 - constrained anisotropic diffusion with discontinuity measure [Chen 05]

Main drawbacks:

- Objective (image restoration, edge detection) is different.
- Adapted to grey-level or color images, not B & W (or discrete contours).
- Related to local SNR estimation and optimization.

Existing works in local noise estimation (1)

- Multi-scale image analysis [Koenderink 84, Perona-Malik 90] ...
- Global optimization (filter scale for edge detection) [Jeong Kim 92]
- Local optimization:
 - minimum reliable scale by fitting a blurred step-model of a contour [Elder Zucker 98],
 - non-linear Gaussian filters [Barash 02] or statistical method [Kervrann 04].
 - constrained anisotropic diffusion with discontinuity measure [Chen 05]

Main drawbacks:

- Objective (image restoration, edge detection) is different.
- Adapted to grey-level or color images, not B & W (or discrete contours).
- Related to local SNR estimation and optimization.
- Require at least one user-given parameter.

Existing works in local noise estimation (2)

Previous work on discrete contours

- Notion of *good continuations* [Cao 03]
 - based on perception principle from the Gestalt theory.
 - False alarm probability defined from curvature approximation.

Existing works in local noise estimation (2)

Previous work on discrete contours

- Notion of *good continuations* [Cao 03]

- based on perception principle from the Gestalt theory.
- False alarm probability defined from curvature approximation.



Existing works in local noise estimation (2)

Previous work on discrete contours

- Notion of *good continuations* [Cao 03]
 - based on perception principle from the Gestalt theory.
 - False alarm probability defined from curvature approximation.
- Meaningful edges detection: [Desolneux et al., 2001]



Proposed concept of Meaningful Scale [Kerautret & Lachaud, 2012]

Main idea: Local shape analysis of digital contours

- 1 Exploit asymptotic properties of **perfect shape** digitizations.

Proposed concept of Meaningful Scale [Kerautret & Lachaud, 2012]

Main idea: Local shape analysis of digital contours

- ① Exploit asymptotic properties of **perfect shape** digitizations.
- ② Estimate them from a multiresolution decomposition of the **input shape**.

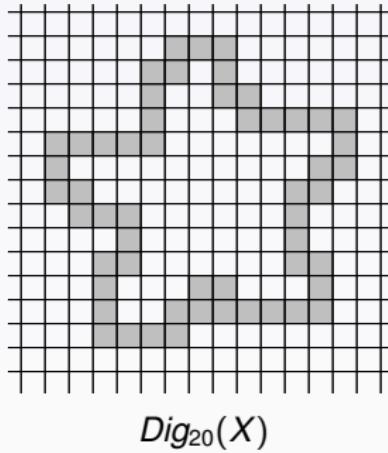


X

Proposed concept of Meaningful Scale [Kerautret & Lachaud, 2012]

Main idea: Local shape analysis of digital contours

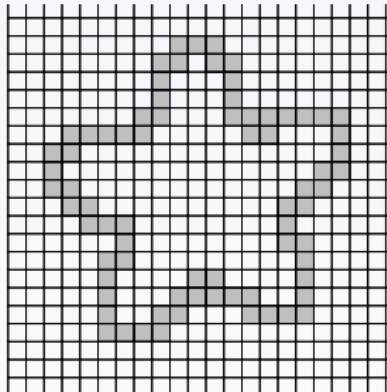
- ① Exploit asymptotic properties of **perfect shape** digitizations.
- ② Estimate them from a multiresolution decomposition of the **input shape**.



Proposed concept of Meaningful Scale [Kerautret & Lachaud, 2012]

Main idea: Local shape analysis of digital contours

- ① Exploit asymptotic properties of **perfect shape** digitizations.
- ② Estimate them from a multiresolution decomposition of the **input shape**.

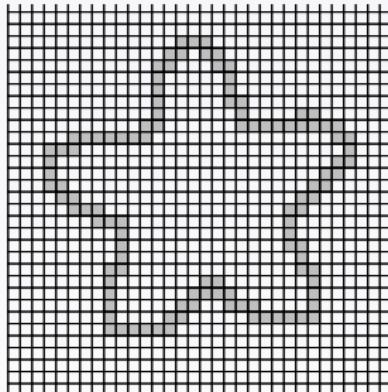


$Dig_{15}(X)$

Proposed concept of Meaningful Scale [Kerautret & Lachaud, 2012]

Main idea: Local shape analysis of digital contours

- ① Exploit asymptotic properties of **perfect shape** digitizations.
- ② Estimate them from a multiresolution decomposition of the **input shape**.

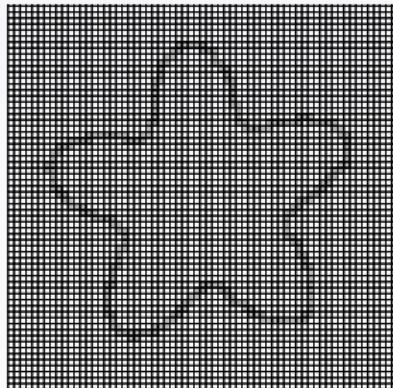


$Dig_{10}(X)$

Proposed concept of Meaningful Scale [Kerautret & Lachaud, 2012]

Main idea: Local shape analysis of digital contours

- ① Exploit asymptotic properties of **perfect shape** digitizations.
- ② Estimate them from a multiresolution decomposition of the **input shape**.

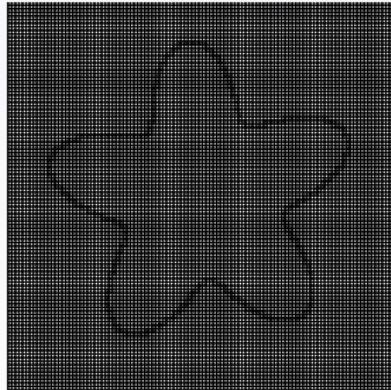


$Dig_5(X)$

Proposed concept of Meaningful Scale [Kerautret & Lachaud, 2012]

Main idea: Local shape analysis of digital contours

- ① Exploit asymptotic properties of **perfect shape** digitizations.
- ② Estimate them from a multiresolution decomposition of the **input shape**.

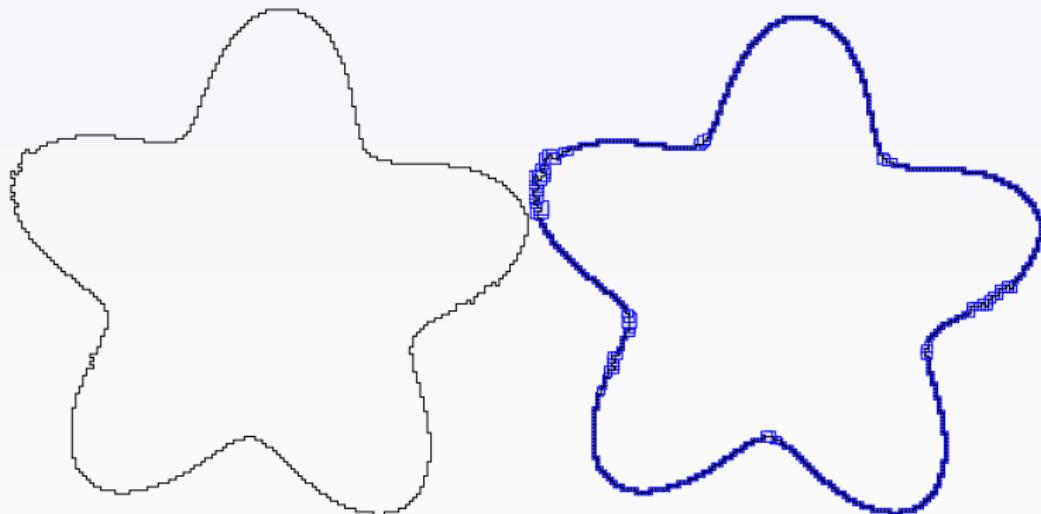


$Dig_3(X)$

Proposed concept of Meaningful Scale [Kerautret & Lachaud, 2012]

Main idea: Local shape analysis of digital contours

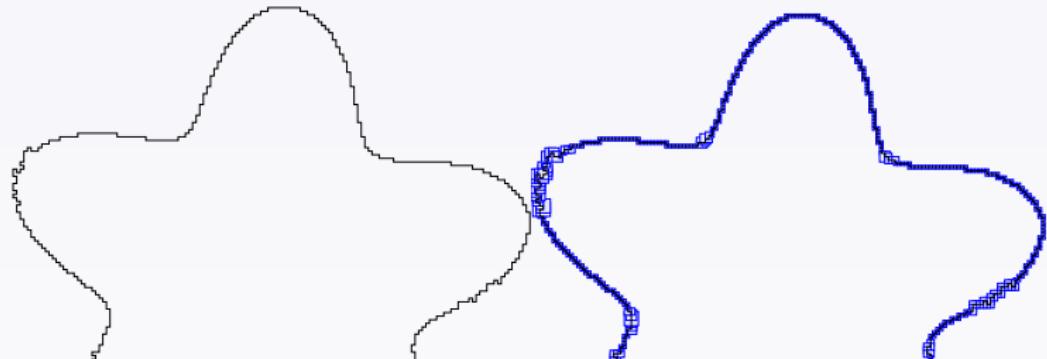
- 1 Exploit asymptotic properties of **perfect shape** digitizations.
- 2 Estimate them from a multiresolution decomposition of the **input shape**.
- 3 Compare them to determine a local meaningful scale.



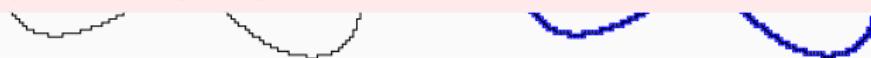
Proposed concept of Meaningful Scale [Kerautret & Lachaud, 2012]

Main idea: Local shape analysis of digital contours

- 1 Exploit asymptotic properties of **perfect shape** digitizations.
- 2 Estimate them from a multiresolution decomposition of the **input shape**.
- 3 Compare them to determine a local meaningful scale.



A digital contour is locally damaged by some noise iff it does not follow the asymptotic properties of perfect shape digitizations.



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.



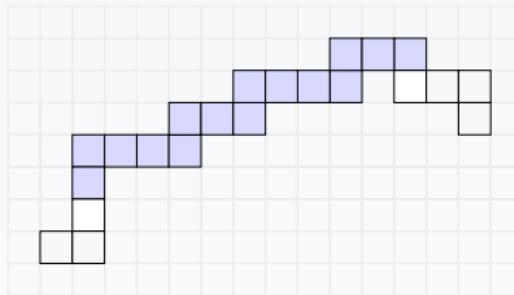
1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

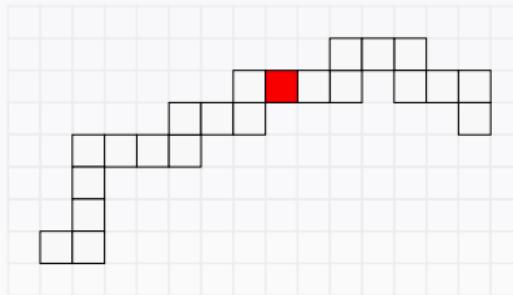
- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

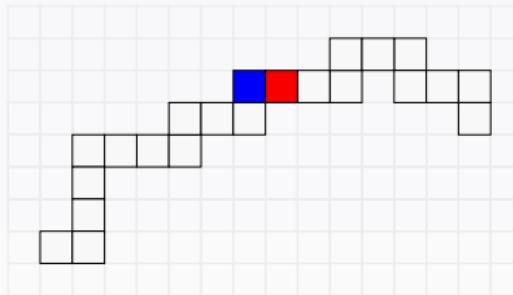
- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

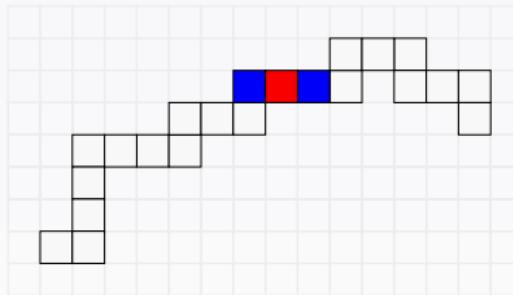
- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

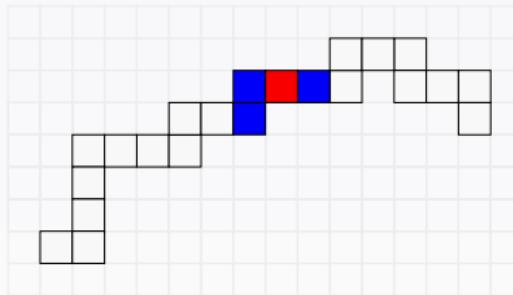
- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

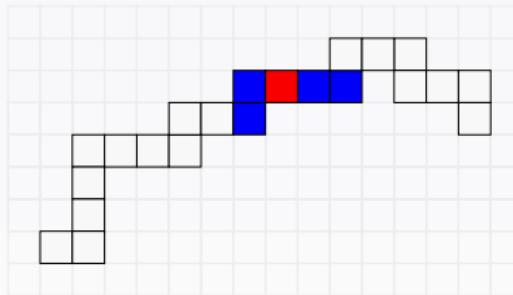
- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

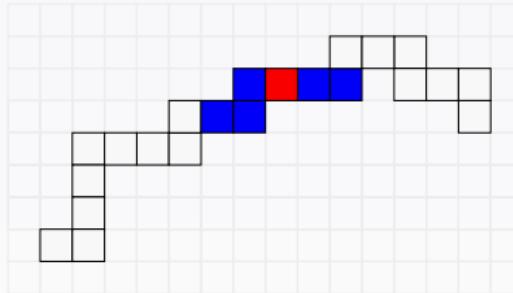
- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

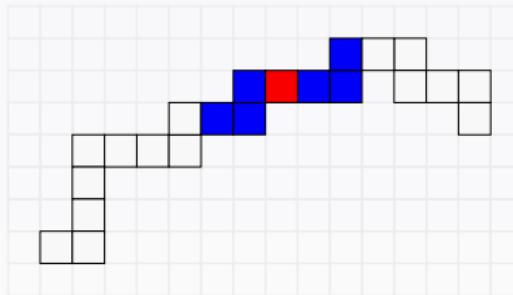
- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

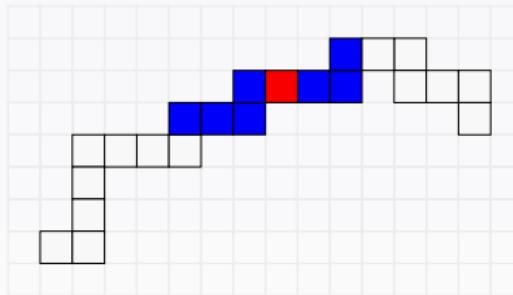
- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

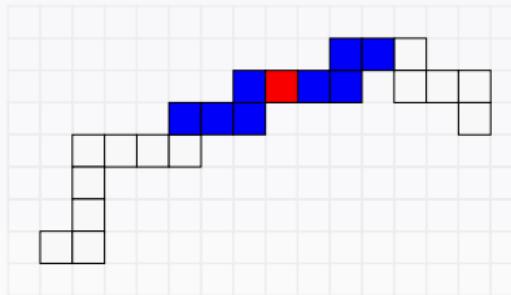
- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

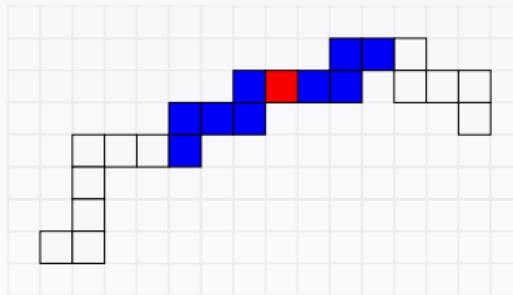
- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

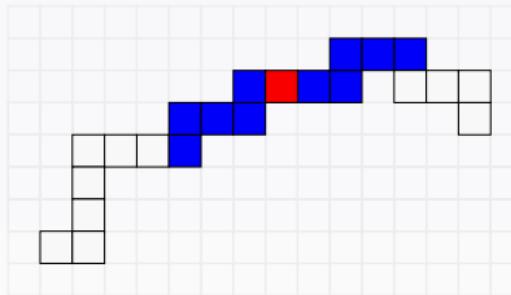
- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

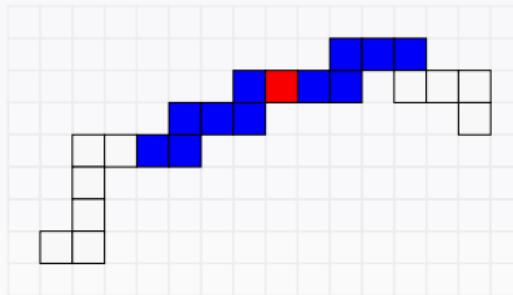
- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (1)

Concept based on the definition of discrete primitives

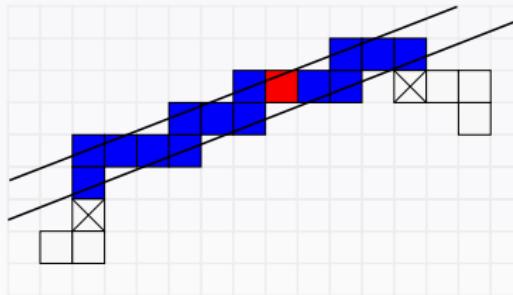
- ① A standard Digital Straight Line (DSL):

$$\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

where (a, b, μ) are integers and $\gcd(a, b) = 1$.

- ② Maximal straight segment:

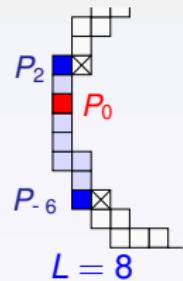
- 4-connected piece (denoted M) of DSL.
- No more a DSL by adding other contour points $C \setminus M$



1.1 Main idea of the meaningful scale (2)

Main idea [Kerautret & Lachaud, 2012]

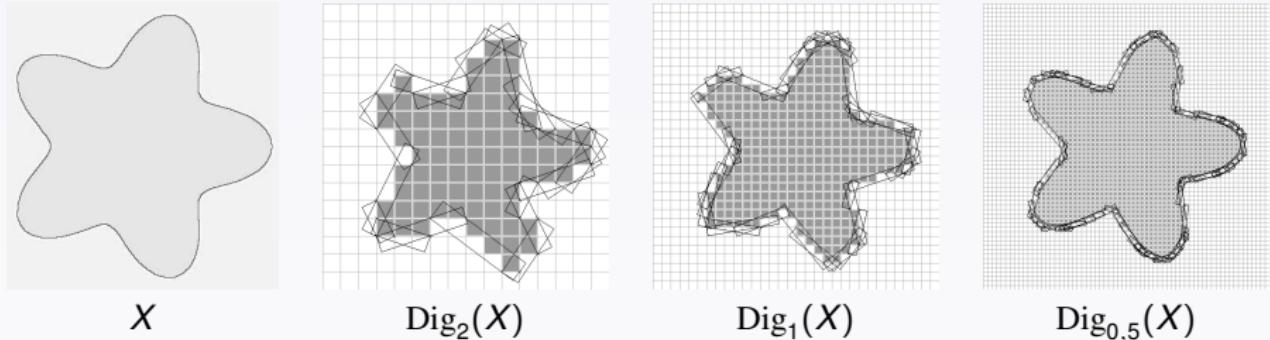
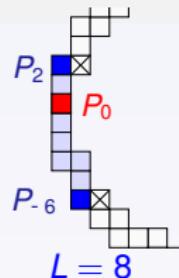
- ① Exploit asymptotic properties of the Length (L) of maximal straight segments (valid on perfect shape digitizations).



1.1 Main idea of the meaningful scale (2)

Main idea [Kerautret & Lachaud, 2012]

- ① Exploit asymptotic properties of the Length (L) of maximal straight segments (valid on perfect shape digitizations).

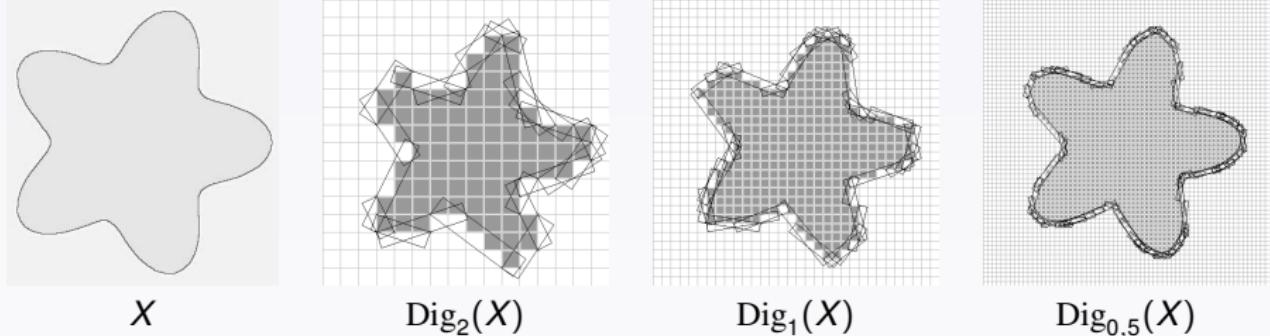
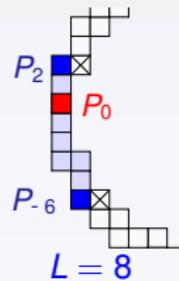


- X some simply connected compact shape of \mathbb{R}^2 .
- $\text{Dig}_h(X)$ = Gauss digitization of X with step h .

1.1 Main idea of the meaningful scale (2)

Main idea [Kerautret & Lachaud, 2012]

- ➊ Exploit asymptotic properties of the Length (L) of maximal straight segments (valid on perfect shape digitizations).
- ➋ They grow longer as h gets finer.

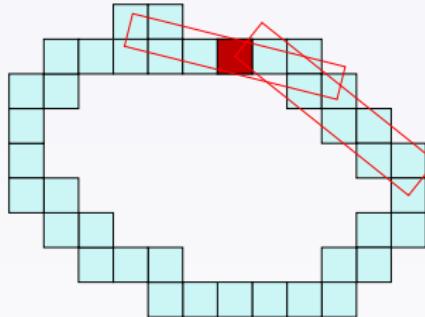
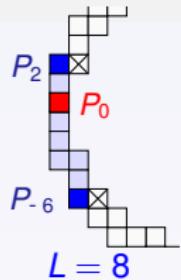


- X some simply connected compact shape of \mathbb{R}^2 .
- $\text{Dig}_h(X)$ = Gauss digitization of X with step h .

1.1 Main idea of the meaningful scale (2)

Main idea [Kerautret & Lachaud, 2012]

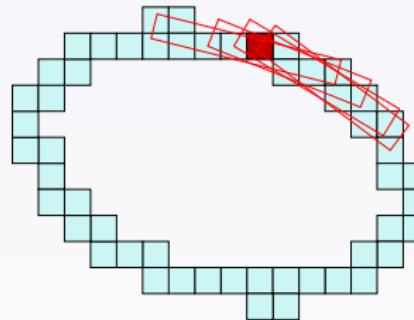
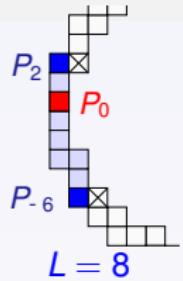
- ① Exploit asymptotic properties of the Length (L) of maximal straight segments (valid on perfect shape digitizations).
- ② They grow longer as h gets finer.



1.1 Main idea of the meaningful scale (2)

Main idea [Kerautret & Lachaud, 2012]

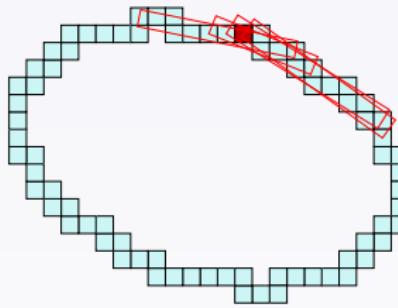
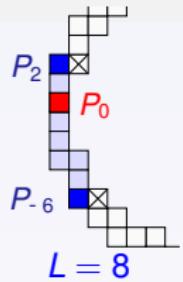
- ① Exploit asymptotic properties of the Length (L) of maximal straight segments (valid on perfect shape digitizations).
- ② They grow longer as h gets finer.



1.1 Main idea of the meaningful scale (2)

Main idea [Kerautret & Lachaud, 2012]

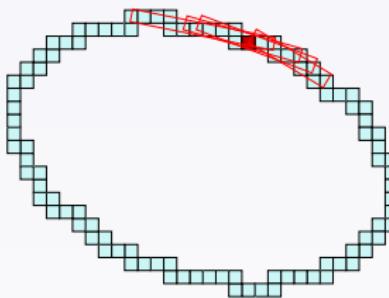
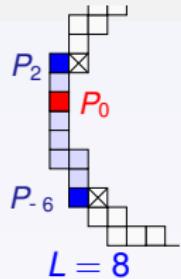
- ① Exploit asymptotic properties of the Length (L) of maximal straight segments (valid on perfect shape digitizations).
- ② They grow longer as h gets finer.



1.1 Main idea of the meaningful scale (2)

Main idea [Kerautret & Lachaud, 2012]

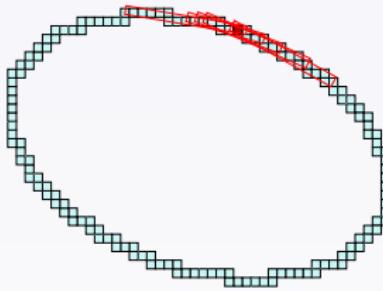
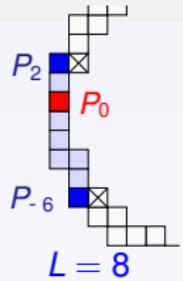
- ① Exploit asymptotic properties of the Length (L) of maximal straight segments (valid on perfect shape digitizations).
- ② They grow longer as h gets finer.



1.1 Main idea of the meaningful scale (2)

Main idea [Kerautret & Lachaud, 2012]

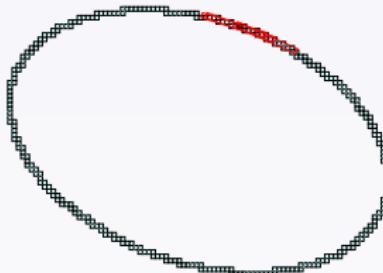
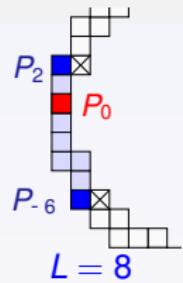
- ① Exploit asymptotic properties of the **Length (L)** of maximal straight segments (valid on **perfect shape** digitizations).
- ② They grow longer as h gets finer.



1.1 Main idea of the meaningful scale (2)

Main idea [Kerautret & Lachaud, 2012]

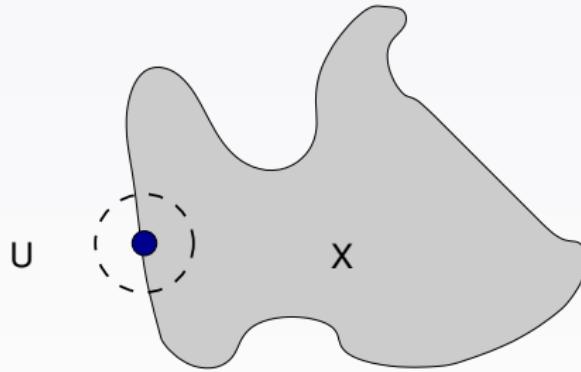
- ① Exploit asymptotic properties of the Length (L) of maximal straight segments (valid on perfect shape digitizations).
- ② They grow longer as h gets finer.



1.1 Main idea of the meaningful scale (3)

Theorem [Lachaud 06]: asymptotic behavior of the length of maximal segments

- X simply connected shape in R^2 with piecewise C^3 boundary ∂X ,
- U an open connected neighborhood of $p \in \partial X$,



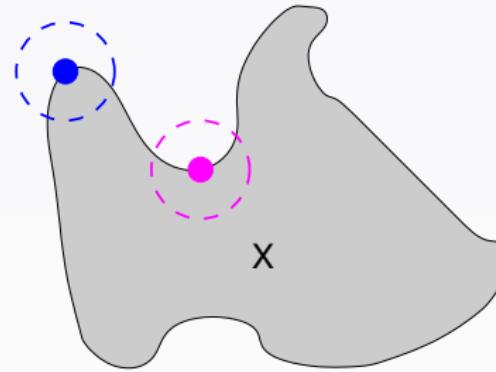
1.1 Main idea of the meaningful scale (3)

Theorem [Lachaud 06]: asymptotic behavior of the length of maximal segments

- X simply connected shape in R^2 with piecewise C^3 boundary ∂X ,
- U an open connected neighborhood of $p \in \partial X$,
- (L_j^h) the digital lengths of the maximal segments of $\text{Dig}_h(X)$ which cover p ,

$$\partial X \cap U \text{ convex or concave}, \text{ then } \Omega(1/h^{1/3}) \leq L_j^h \leq O(1/h^{1/2}) \quad (1)$$

$$\partial X \cap U \text{ has null curvature, then } \Omega(1/h^1) \leq L_j^h \leq O(1/h^1) \quad (2)$$



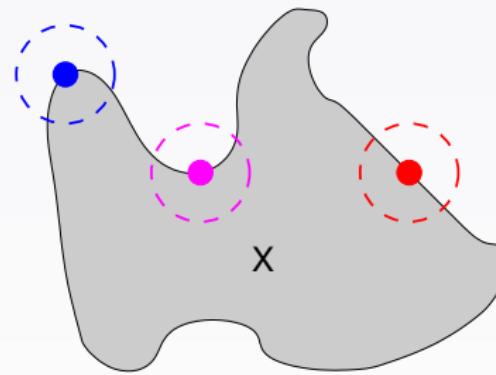
1.1 Main idea of the meaningful scale (3)

Theorem [Lachaud 06]: asymptotic behavior of the length of maximal segments

- X simply connected shape in R^2 with piecewise C^3 boundary ∂X ,
- U an open connected neighborhood of $p \in \partial X$,
- (L_j^h) the digital lengths of the maximal segments of $\text{Dig}_h(X)$ which cover p ,

$$\partial X \cap U \text{ convex or concave}, \text{ then } \Omega(1/h^{1/3}) \leq L_j^h \leq O(1/h^{1/2}) \quad (1)$$

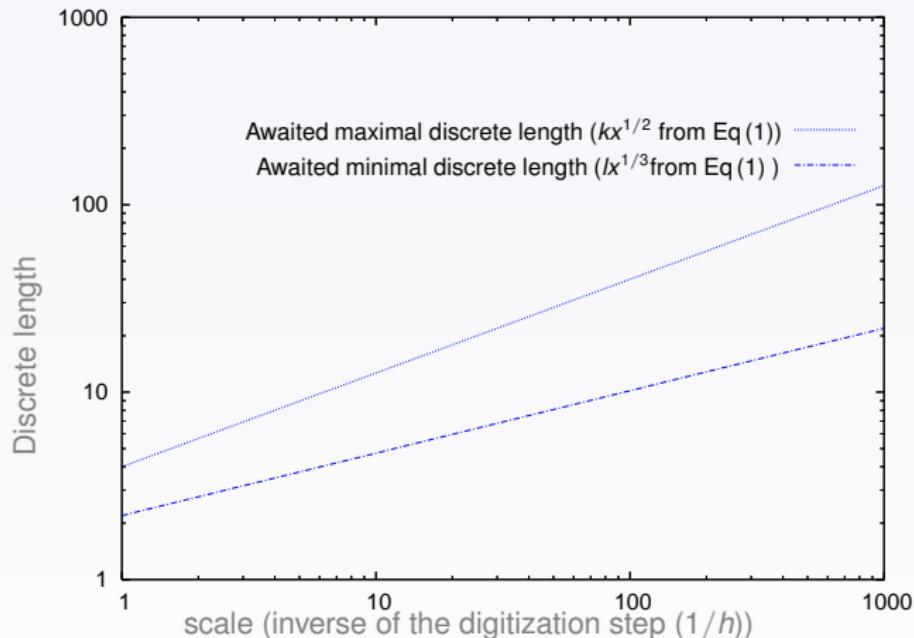
$$\partial X \cap U \text{ has null curvature, then } \Omega(1/h^1) \leq L_j^h \leq O(1/h^1) \quad (2)$$



1.1 Main idea of the meaningful scale (4)

Previous theorem on asymptotic behaviour

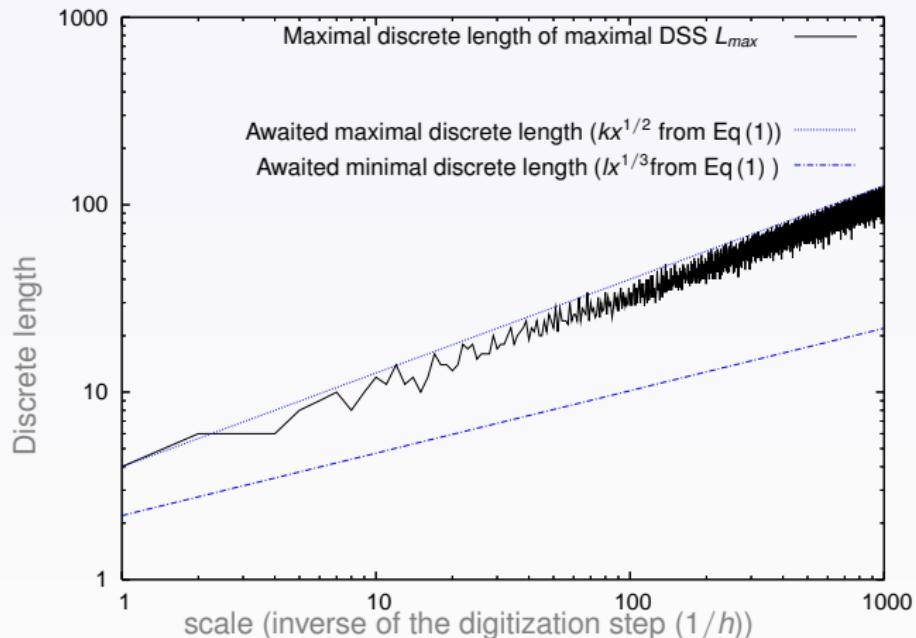
- Proved in [Lachaud 06].
- Can be checked experimentally:



1.1 Main idea of the meaningful scale (4)

Previous theorem on asymptotic behaviour

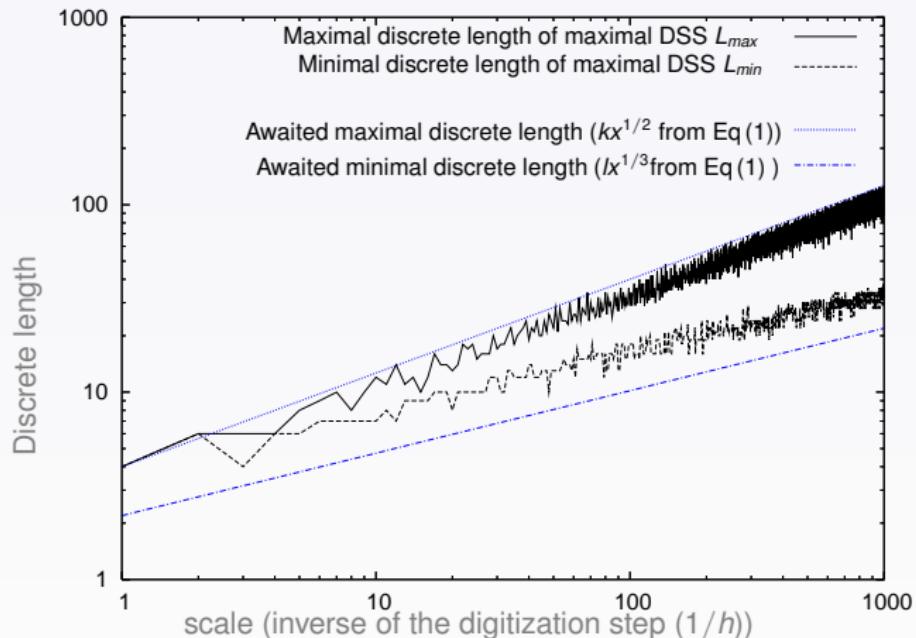
- Proved in [Lachaud 06].
- Can be checked experimentally:



1.1 Main idea of the meaningful scale (4)

Previous theorem on asymptotic behaviour

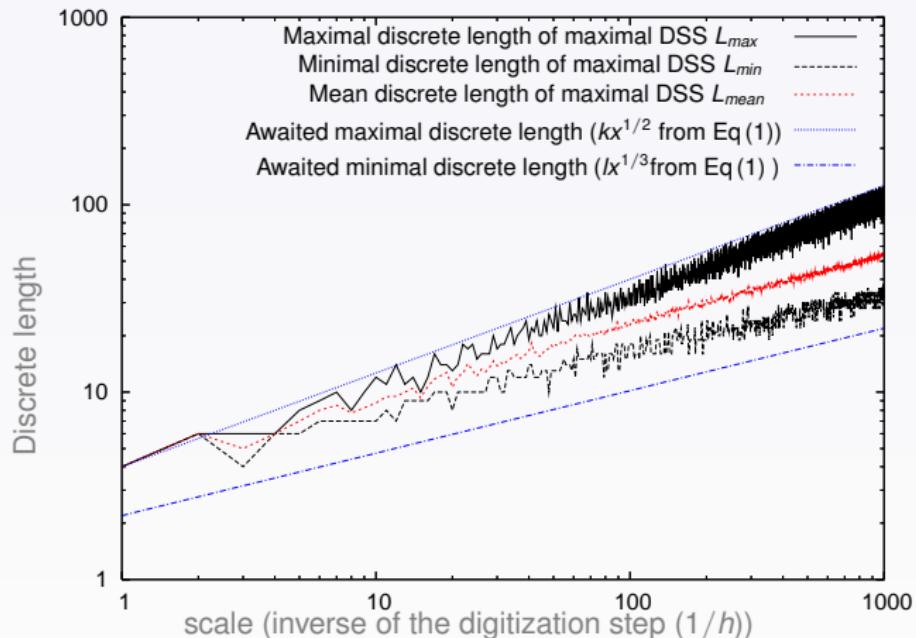
- Proved in [Lachaud 06].
- Can be checked experimentally:



1.1 Main idea of the meaningful scale (4)

Previous theorem on asymptotic behaviour

- Proved in [Lachaud 06].
- Can be checked experimentally:



Asymptotic estimation with multiresolution

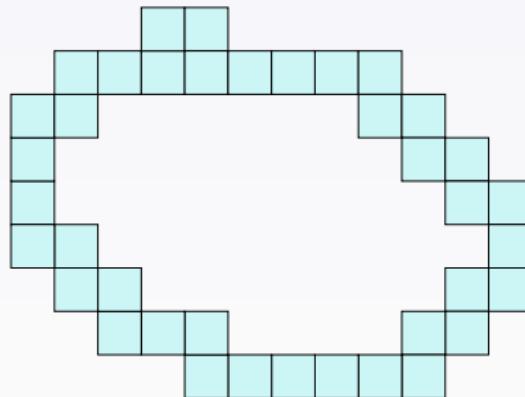
Some criticisms:

- Preceding theorem = digital properties of **perfect** shape.
- A technique for noise detection ?

Asymptotic estimation with multiresolution

Some criticisms:

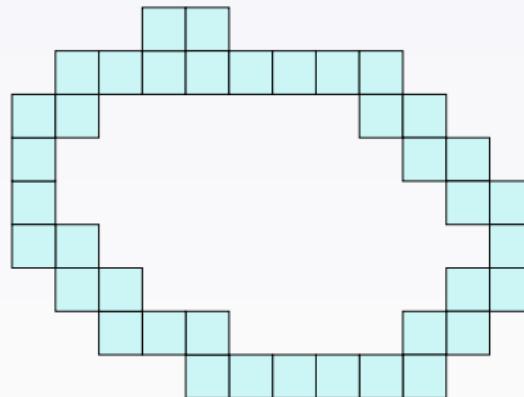
- Preceding theorem = digital properties of **perfect** shape.
- A technique for noise detection ?
- **But:** input is only a digital contour.



Asymptotic estimation with multiresolution

Some criticisms:

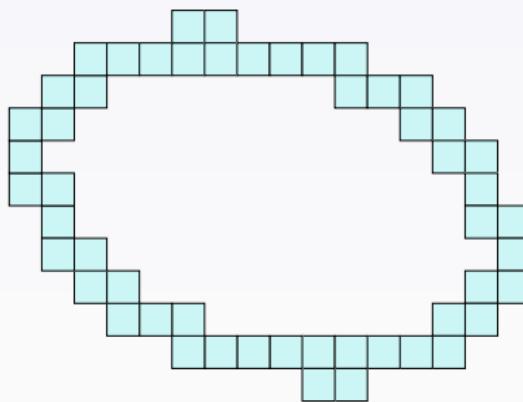
- Preceding theorem = digital properties of **perfect** shape.
- A technique for noise detection ?
- **But:** input is only a digital contour.
- finer and finer digitizations are not accessible.



Asymptotic estimation with multiresolution

Some criticisms:

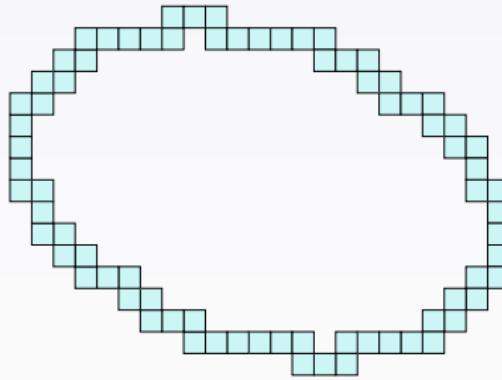
- Preceding theorem = digital properties of **perfect** shape.
- A technique for noise detection ?
- **But:** input is only a digital contour.
- finer and finer digitizations are not accessible.



Asymptotic estimation with multiresolution

Some criticisms:

- Preceding theorem = digital properties of **perfect** shape.
- A technique for noise detection ?
- **But:** input is only a digital contour.
- finer and finer digitizations are not accessible.



Asymptotic estimation with multiresolution

Some criticisms:

- Preceding theorem = digital properties of **perfect** shape.
- A technique for noise detection ?
- **But:** input is only a digital contour.
- finer and finer digitizations are not accessible.

Idea: reverse asymptote with **multiresolution**

- coarser and coarser sampling of the contour
- measure digital lengths of maximal segments
- should also follow approximately asymptotic behavior

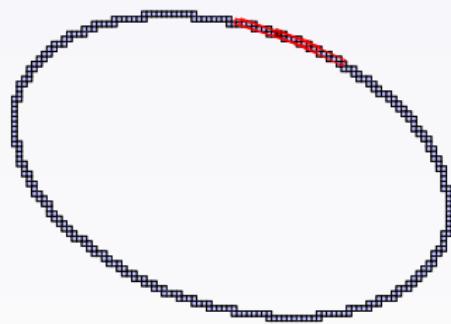
Asymptotic estimation with multiresolution

Some criticisms:

- Preceding theorem = digital properties of **perfect** shape.
- A technique for noise detection ?
- **But:** input is only a digital contour.
- finer and finer digitizations are not accessible.

Idea: reverse asymptote with **multiresolution**

- coarser and coarser sampling of the contour
- measure digital lengths of maximal segments
- should also follow approximately asymptotic behavior



grid size 1

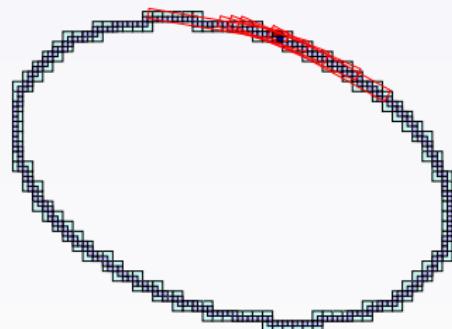
Asymptotic estimation with multiresolution

Some criticisms:

- Preceding theorem = digital properties of **perfect** shape.
- A technique for noise detection ?
- **But:** input is only a digital contour.
- finer and finer digitizations are not accessible.

Idea: reverse asymptote with **multiresolution**

- coarser and coarser sampling of the contour
- measure digital lengths of maximal segments
- should also follow approximately asymptotic behavior



grid size 2

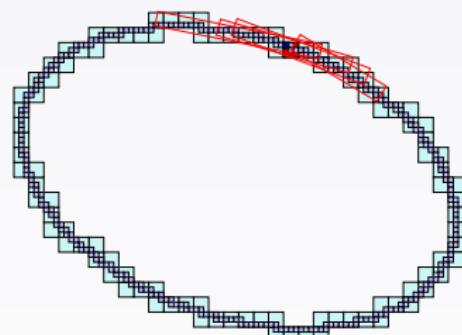
Asymptotic estimation with multiresolution

Some criticisms:

- Preceding theorem = digital properties of **perfect** shape.
- A technique for noise detection ?
- **But:** input is only a digital contour.
- finer and finer digitizations are not accessible.

Idea: reverse asymptote with multiresolution

- coarser and coarser sampling of the contour
- measure digital lengths of maximal segments
- should also follow approximately asymptotic behavior



grid size 3

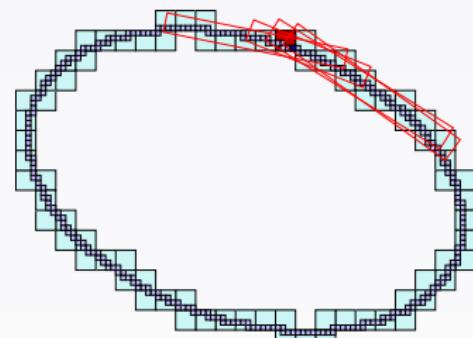
Asymptotic estimation with multiresolution

Some criticisms:

- Preceding theorem = digital properties of **perfect** shape.
- A technique for noise detection ?
- **But:** input is only a digital contour.
- finer and finer digitizations are not accessible.

Idea: reverse asymptote with multiresolution

- coarser and coarser sampling of the contour
- measure digital lengths of maximal segments
- should also follow approximately asymptotic behavior



grid size 4

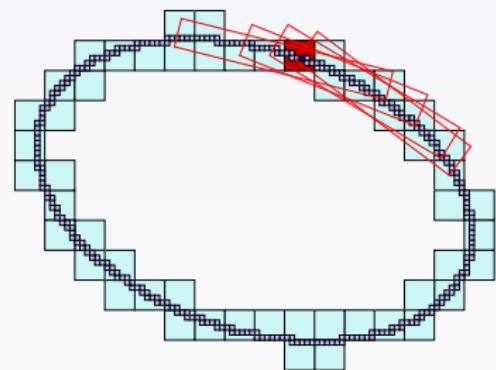
Asymptotic estimation with multiresolution

Some criticisms:

- Preceding theorem = digital properties of **perfect** shape.
- A technique for noise detection ?
- **But:** input is only a digital contour.
- finer and finer digitizations are not accessible.

Idea: reverse asymptote with multiresolution

- coarser and coarser sampling of the contour
- measure digital lengths of maximal segments
- should also follow approximately asymptotic behavior



grid size 6

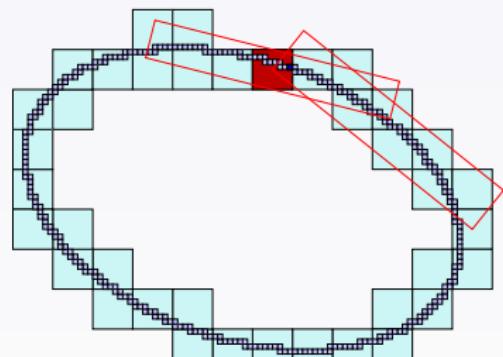
Asymptotic estimation with multiresolution

Some criticisms:

- Preceding theorem = digital properties of **perfect** shape.
- A technique for noise detection ?
- **But:** input is only a digital contour.
- finer and finer digitizations are not accessible.

Idea: reverse asymptote with multiresolution

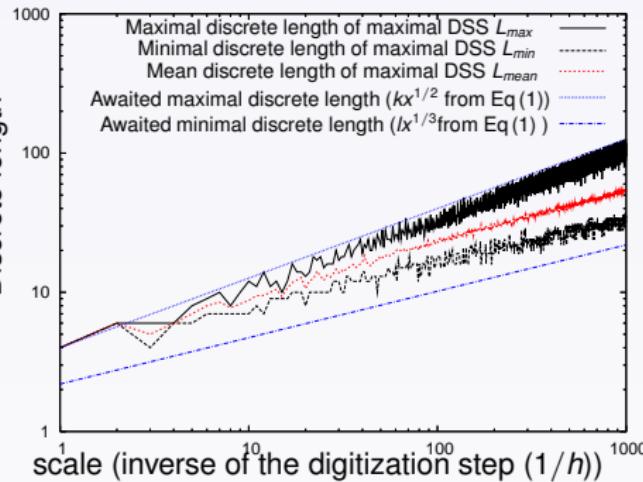
- coarser and coarser sampling of the contour
- measure digital lengths of maximal segments
- should also follow approximately asymptotic behavior



grid size 8

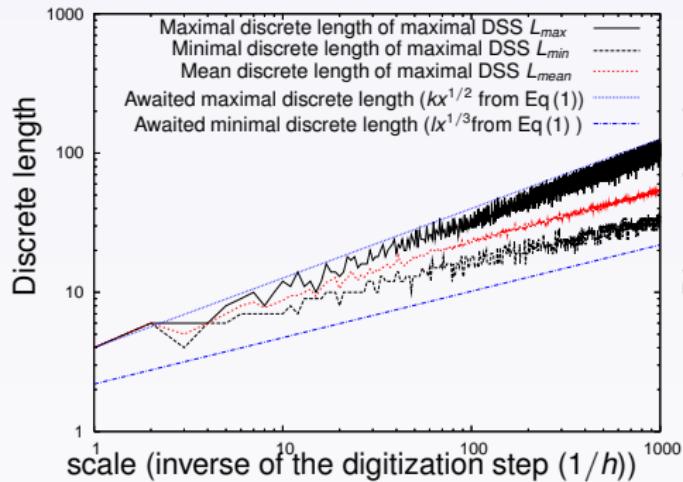
Preliminary experiments about reverse asymptotic behavior

Experiments of asymptotic behaviour

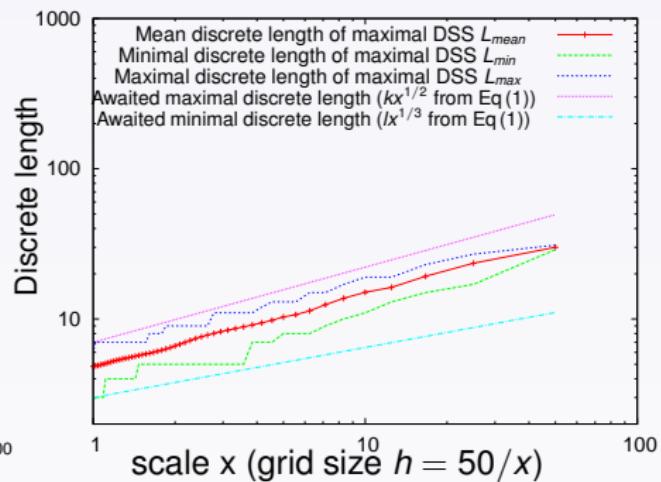


Preliminary experiments about reverse asymptotic behavior

Experiments of asymptotic behaviour



Experiments from subsampling



Subsampling a digital contour

Subsampling process:

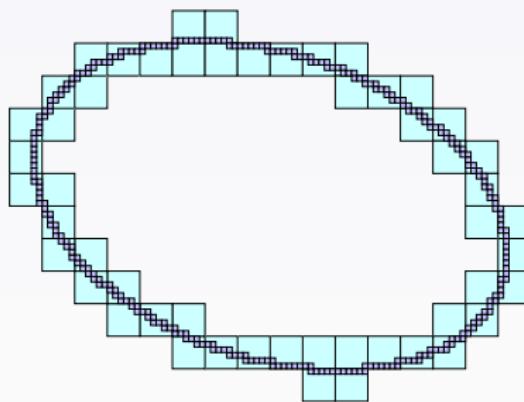
- Not spatial but applied to the sequence of points (x_j, y_j) of the contour C .
- All subsampled contours $\phi_i^{(x_0, y_0)}$ are computed for all shifts (x_0, y_0) :
$$(X_j, Y_j) = ((x_j - x_0) \div i, (y_j - y_0) \div i)$$

Subsampling a digital contour

Subsampling process:

- Not spatial but applied to the sequence of points (x_j, y_j) of the contour C .
- All subsampled contours $\phi_i^{(x_0, y_0)}$ are computed for all shifts (x_0, y_0) :
$$(X_j, Y_j) = ((x_j - x_0) \div i, (y_j - y_0) \div i)$$

Subsampling contours $\phi_6^{0,0}$

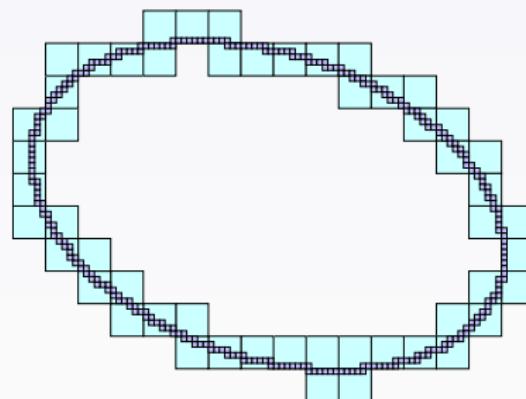


Subsampling a digital contour

Subsampling process:

- Not spatial but applied to the sequence of points (x_j, y_j) of the contour C .
- All subsampled contours $\phi_i^{(x_0, y_0)}$ are computed for all shifts (x_0, y_0) :
$$(X_j, Y_j) = ((x_j - x_0) \div i, (y_j - y_0) \div i)$$

Subsampling contours $\phi_6^{1,0}$

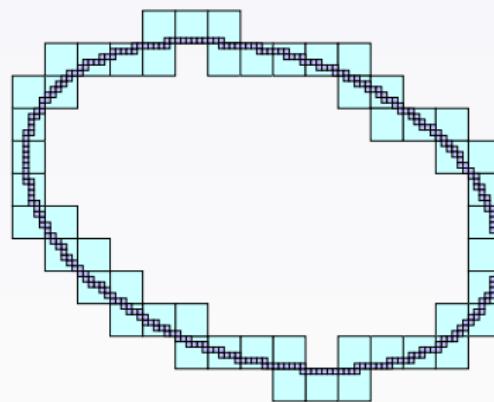


Subsampling a digital contour

Subsampling process:

- Not spatial but applied to the sequence of points (x_j, y_j) of the contour C .
- All subsampled contours $\phi_i^{(x_0, y_0)}$ are computed for all shifts (x_0, y_0) :
$$(X_j, Y_j) = ((x_j - x_0) \div i, (y_j - y_0) \div i)$$

Subsampling contours $\phi_6^{2,0}$

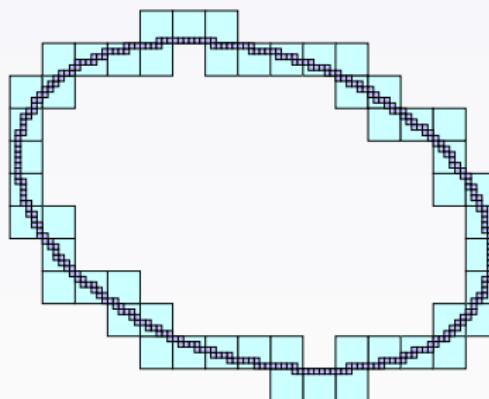


Subsampling a digital contour

Subsampling process:

- Not spatial but applied to the sequence of points (x_j, y_j) of the contour C .
- All subsampled contours $\phi_i^{(x_0, y_0)}$ are computed for all shifts (x_0, y_0) :
$$(X_j, Y_j) = ((x_j - x_0) \div i, (y_j - y_0) \div i)$$

Subsampling contours $\phi_6^{3,0}$

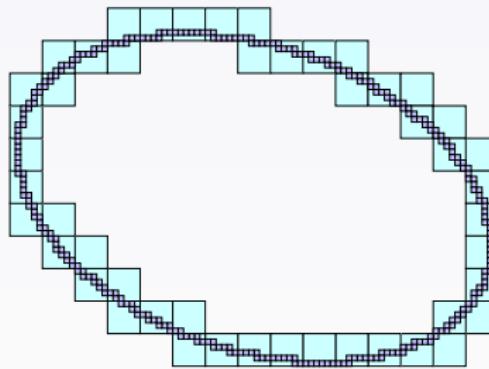


Subsampling a digital contour

Subsampling process:

- Not spatial but applied to the sequence of points (x_j, y_j) of the contour C .
- All subsampled contours $\phi_i^{(x_0, y_0)}$ are computed for all shifts (x_0, y_0) :
$$(X_j, Y_j) = ((x_j - x_0) \div i, (y_j - y_0) \div i)$$

Subsampling contours $\phi_6^{3,1}$



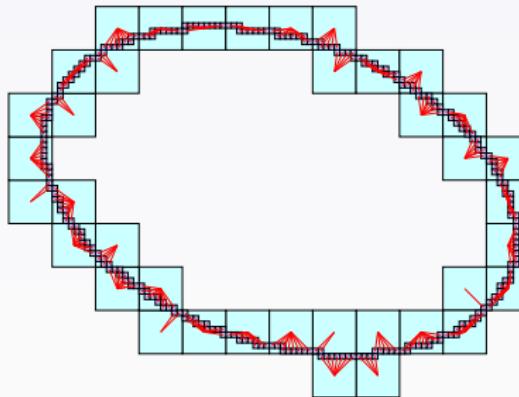
Subsampling a digital contour

Subsampling process:

- Not spatial but applied to the sequence of points (x_j, y_j) of the contour C .
- All subsampled contours $\phi_i^{(x_0, y_0)}$ are computed for all shifts (x_0, y_0) :

$$(X_j, Y_j) = ((x_j - x_0) \div i, (y_j - y_0) \div i)$$
- Compute the surjective map $f_i^{x_0, y_0}(P)$

Subsampling contours $f_8^{0,0}$



1.2 Meaningful Scale Profile and Noise Detection

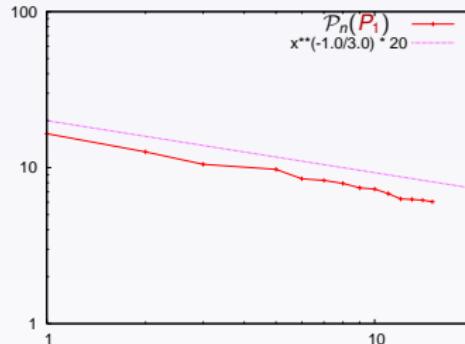
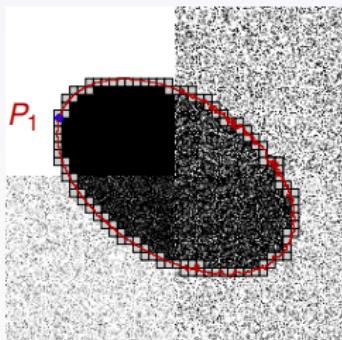
Multiscale profile \mathcal{P}_n of a point P on a digital contour:

- $\mathcal{P}_n(P) = \text{sequence } (\log i, \log(E(L^{h_i})))_{i=1..n}$, where:
 - E is the average operator.
 - L^{h_i} are the digital lengths of the maximal segments covering P for all subsampling $i \times i$.

1.2 Meaningful Scale Profile and Noise Detection

Multiscale profile \mathcal{P}_n of a point P on a digital contour:

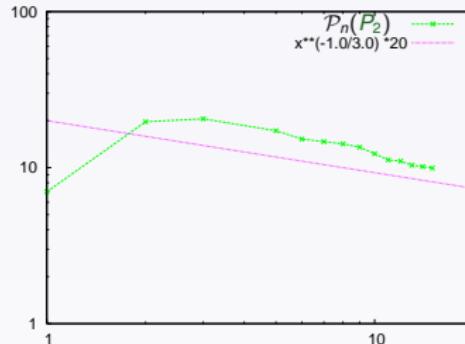
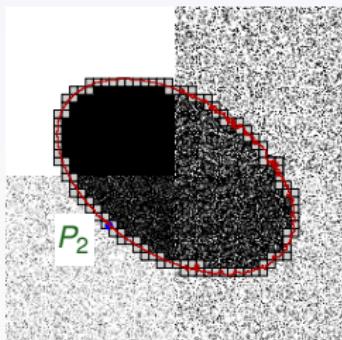
- $\mathcal{P}_n(P) = \text{sequence } (\log i, \log(E(L^{h_i})))_{i=1..n}$, where:
 - E is the average operator.
 - L^{h_i} are the digital lengths of the maximal segments covering P for all subsampling $i \times i$.



1.2 Meaningful Scale Profile and Noise Detection

Multiscale profile \mathcal{P}_n of a point P on a digital contour:

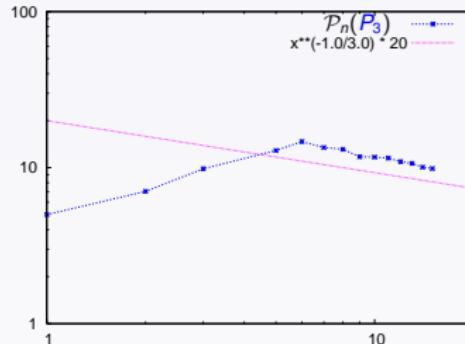
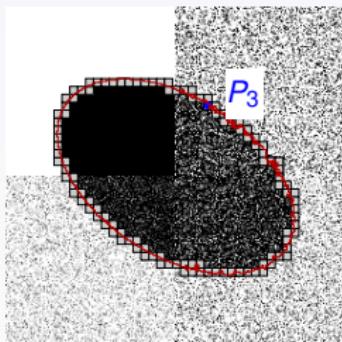
- $\mathcal{P}_n(P) = \text{sequence } (\log i, \log(E(L^{h_i})))_{i=1..n}$, where:
 - E is the average operator.
 - L^{h_i} are the digital lengths of the maximal segments covering P for all subsampling $i \times i$.



1.2 Meaningful Scale Profile and Noise Detection

Multiscale profile \mathcal{P}_n of a point P on a digital contour:

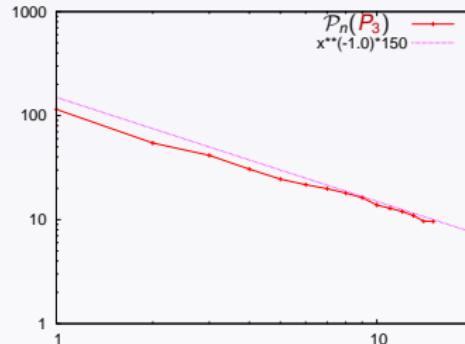
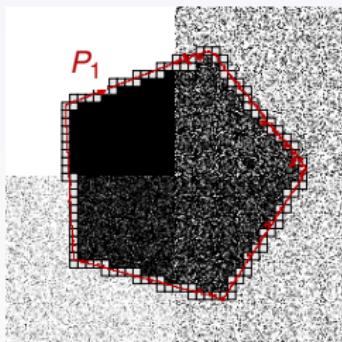
- $\mathcal{P}_n(P) = \text{sequence } (\log i, \log(E(L^{h_i})))_{i=1..n}$, where:
 - E is the average operator.
 - L^{h_i} are the digital lengths of the maximal segments covering P for all subsampling $i \times i$.



1.2 Meaningful Scale Profile and Noise Detection

Multiscale profile \mathcal{P}_n of a point P on a digital contour:

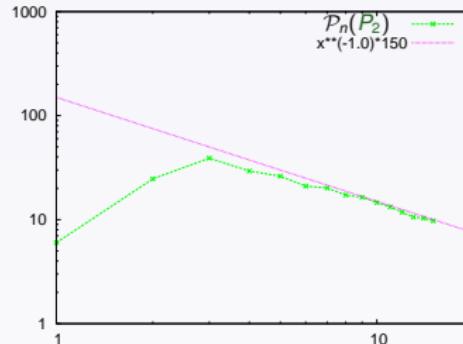
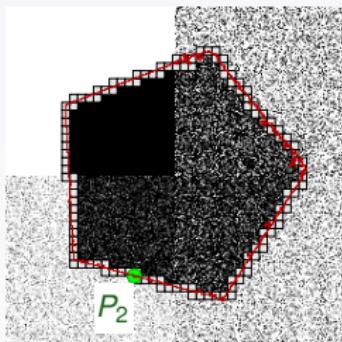
- $\mathcal{P}_n(P) = \text{sequence } (\log i, \log(E(L^{h_i})))_{i=1..n}$, where:
 - E is the average operator.
 - L^{h_i} are the digital lengths of the maximal segments covering P for all subsampling $i \times i$.



1.2 Meaningful Scale Profile and Noise Detection

Multiscale profile \mathcal{P}_n of a point P on a digital contour:

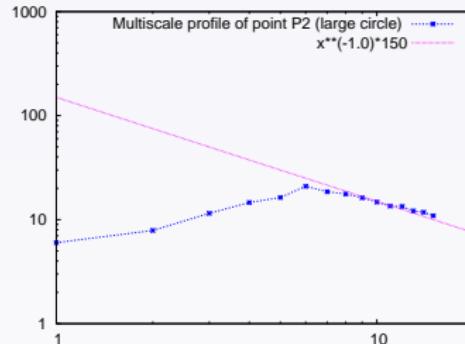
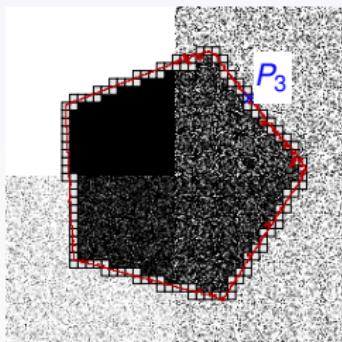
- $\mathcal{P}_n(P) = \text{sequence } (\log i, \log(E(L^{h_i})))_{i=1..n}$, where:
 - E is the average operator.
 - L^{h_i} are the digital lengths of the maximal segments covering P for all subsampling $i \times i$.



1.2 Meaningful Scale Profile and Noise Detection

Multiscale profile \mathcal{P}_n of a point P on a digital contour:

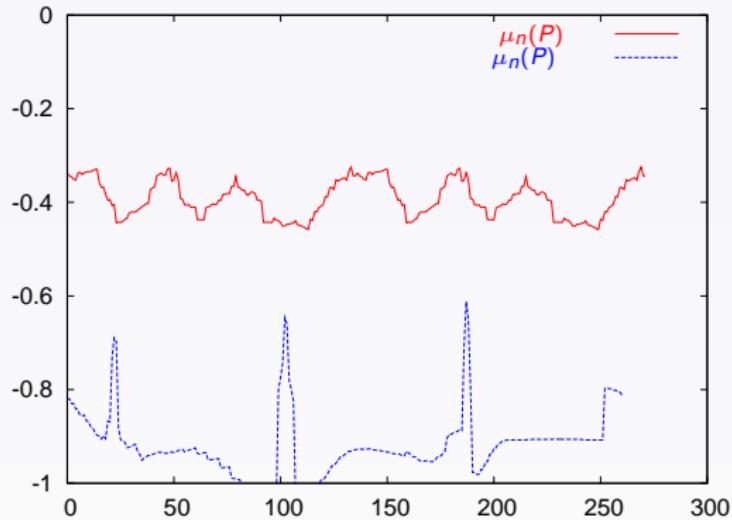
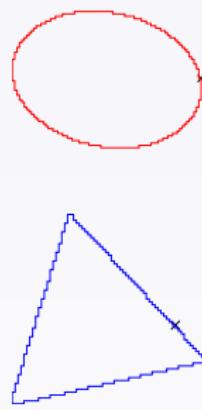
- $\mathcal{P}_n(P) = \text{sequence } (\log i, \log(E(L^{h_i})))_{i=1..n}$, where:
 - E is the average operator.
 - L^{h_i} are the digital lengths of the maximal segments covering P for all subsampling $i \times i$.



Local geometric evaluation with multi-scale profile

Ideal multiscale criterion: $\mu_n(P)$

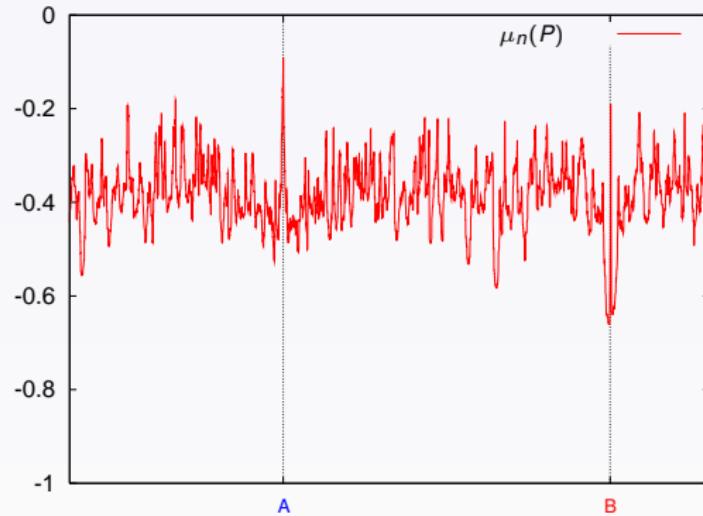
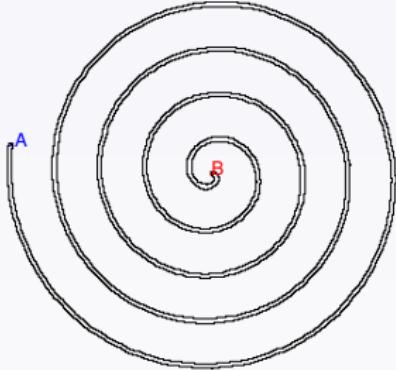
Defined as the slope of the linear regression of $\mathcal{P}_n(P)$



Local geometric evaluation with multi-scale profile

Ideal multiscale criterion: $\mu_n(P)$

Defined as the slope of the linear regression of $\mathcal{P}_n(P)$



Local geometric evaluation with multi-scale profile

Ideal multiscale criterion: $\mu_n(P)$

Defined as the slope of the linear regression of $\mathcal{P}_n(P)$

Evaluation on several shapes with different grid sizes:

		Distribution of slopes $\mu_n(.)$				
shape X	value	h=1	h=1/2	h=1/4	h=1/8	h=1/16
circle r=20	E	-0.391	-0.359	-0.356	-0.343	-0.342
	σ	0.028	0.046	0.041	0.035	0.055
ellipse $a = 20$ $b = 14, \theta = 0.2$	E	-0.412	-0.392	-0.377	-0.353	-0.346
	σ	0.047	0.039	0.054	0.056	0.056

Local geometric evaluation with multi-scale profile

Ideal multiscale criterion: $\mu_n(P)$

Defined as the slope of the linear regression of $\mathcal{P}_n(P)$

Evaluation on several shapes with different grid sizes:

		Distribution of slopes $\mu_n(.)$				
shape X	value	h=1	h=1/2	h=1/4	h=1/8	h=1/16
triangle $r = 20, \theta = 0.3$	E	-0.860	-0.931	-0.956	-0.923	-0.920
	σ	0.1	0.060	0.052	0.068	0.047
pentagon $r = 20, \theta = 0.2$	E	-0.695	-0.815	-0.890	-0.914	-0.924
	σ	0.081	0.070	0.064	0.066	0.059

Local meaningful scale and noise detection

Meaningful scale:

A **meaningful scale** of a multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$ is the pair (i_1, i_2) $1 \leq i_1 \leq i_2 \leq n$ such that for all i , $i_1 \leq i < i_2$,

$$\frac{Y_{i+1} - Y_i}{X_{i+1} - X_i} \leq t_m,$$

while not true for $i_1 - 1$ and i_2 .

Parameter t_m = noise threshold to discriminate curved from noisy areas.

Local meaningful scale and noise detection

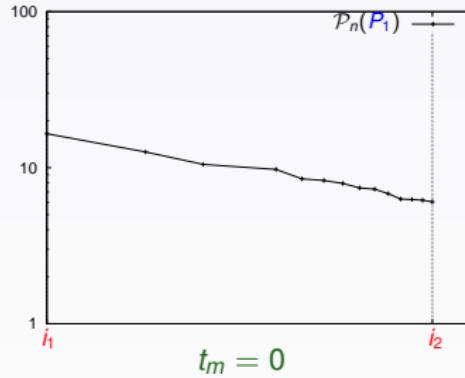
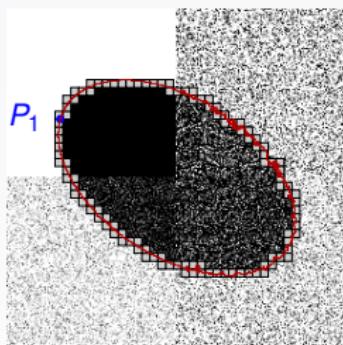
Meaningful scale:

A **meaningful scale** of a multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$ is the pair (i_1, i_2) $1 \leq i_1 \leq i_2 \leq n$ such that for all i , $i_1 \leq i < i_2$,

$$\frac{Y_{i+1} - Y_i}{X_{i+1} - X_i} \leq t_m,$$

while not true for $i_1 - 1$ and i_2 .

Parameter t_m = noise threshold to discriminate curved from noisy areas.



Local meaningful scale and noise detection

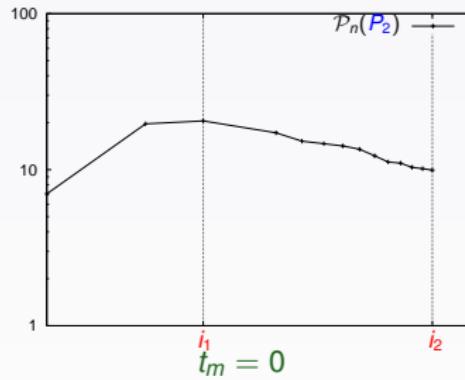
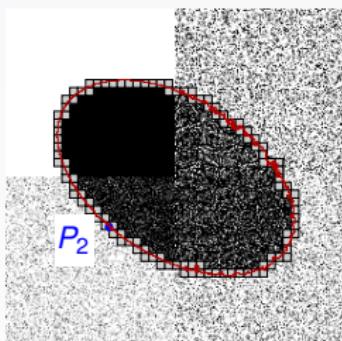
Meaningful scale:

A **meaningful scale** of a multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$ is the pair (i_1, i_2) $1 \leq i_1 \leq i_2 \leq n$ such that for all i , $i_1 \leq i < i_2$,

$$\frac{Y_{i+1} - Y_i}{X_{i+1} - X_i} \leq t_m,$$

while not true for $i_1 - 1$ and i_2 .

Parameter t_m = noise threshold to discriminate curved from noisy areas.



Local meaningful scale and noise detection

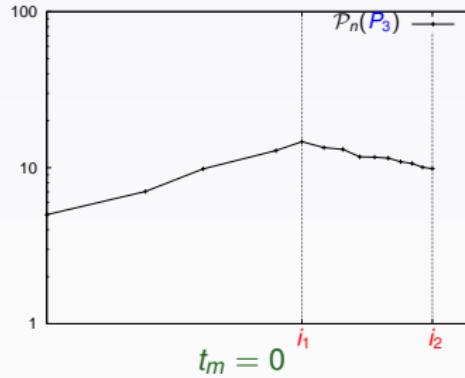
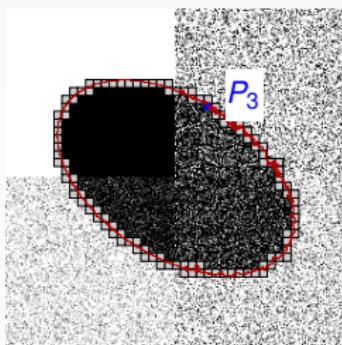
Meaningful scale:

A **meaningful scale** of a multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$ is the pair (i_1, i_2) $1 \leq i_1 \leq i_2 \leq n$ such that for all i , $i_1 \leq i < i_2$,

$$\frac{Y_{i+1} - Y_i}{X_{i+1} - X_i} \leq t_m,$$

while not true for $i_1 - 1$ and i_2 .

Parameter t_m = noise threshold to discriminate curved from noisy areas.



Local meaningful scale and noise detection

Meaningful scale:

A **meaningful scale** of a multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$ is the pair (i_1, i_2) $1 \leq i_1 \leq i_2 \leq n$ such that for all i , $i_1 \leq i < i_2$,

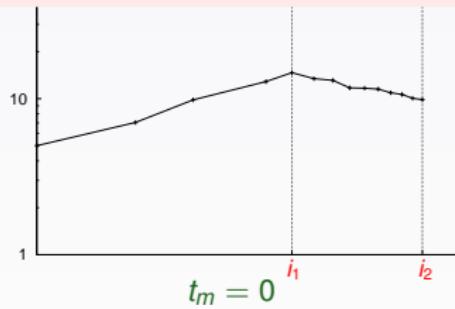
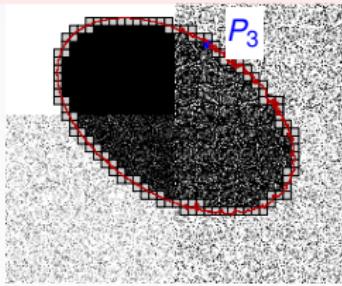
$$\frac{Y_{i+1} - Y_i}{X_{i+1} - X_i} \leq t_m,$$

while not true for $i_1 - 1$ and i_2 .

Parameter t_m = noise threshold to discriminate curved from noisy areas.

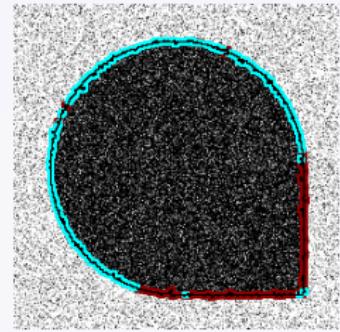
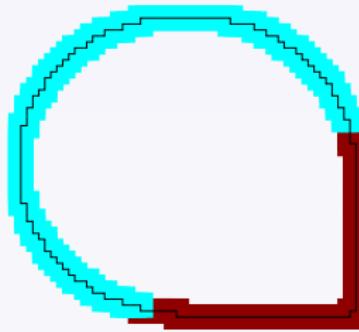
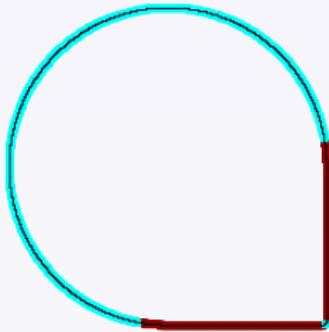
Noise level at point P

If (i_1, i_2) is the first meaningful scale at point P , the **noise level** at point P is $i_1 - 1$.



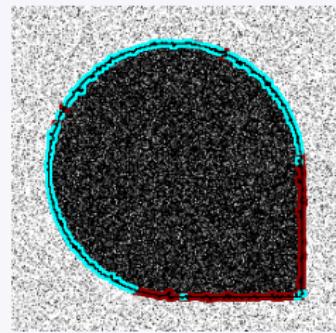
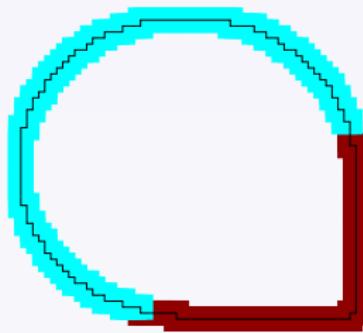
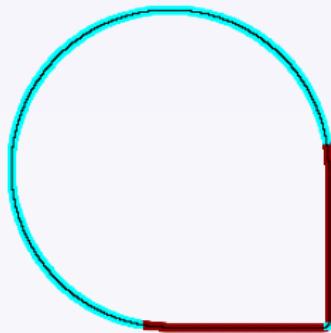
3. Experiments: flat and curved areas detection

Discrimination between curved and flat parts ($t_{f/c} = -0.52$):

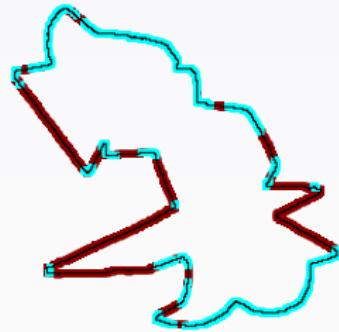
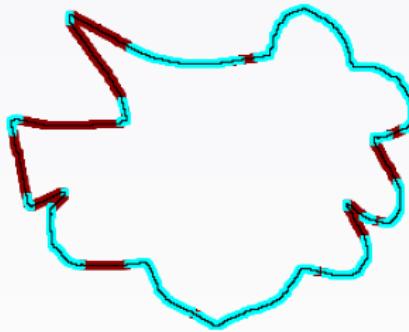


3. Experiments: flat and curved areas detection

Discrimination between curved and flat parts ($t_{f/c} = -0.52$):

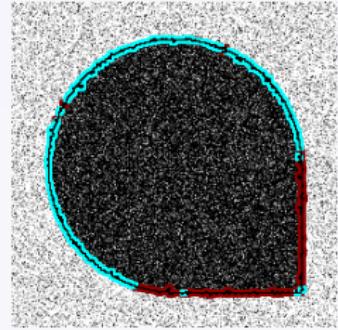
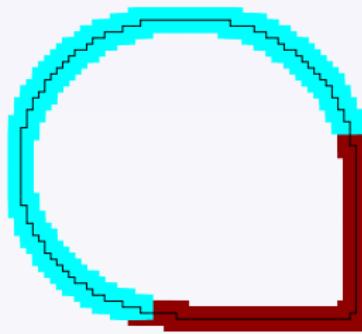
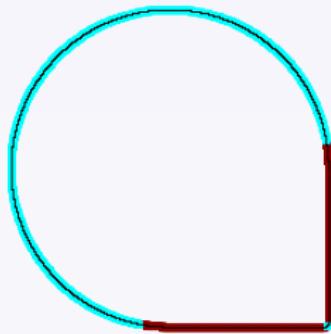


Test on real shapes:

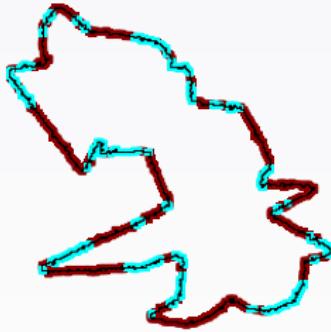
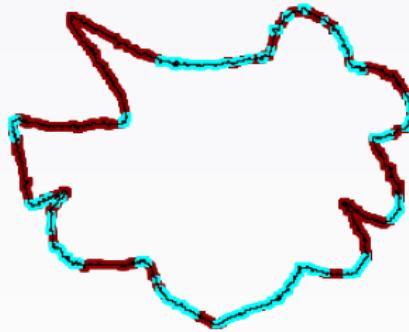


3. Experiments: flat and curved areas detection

Discrimination between curved and flat parts ($t_{f/c} = -0.52$):

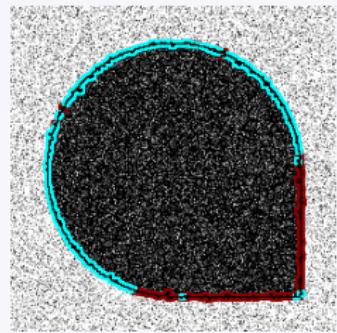
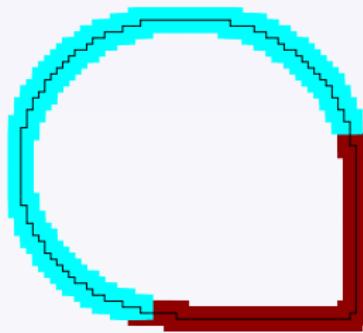
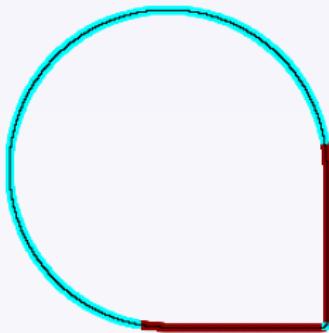


Test on damaged versions of these shapes:



3. Experiments: flat and curved areas detection

Discrimination between curved and flat parts ($t_{f/c} = -0.52$):



Test on damaged versions of these shapes:

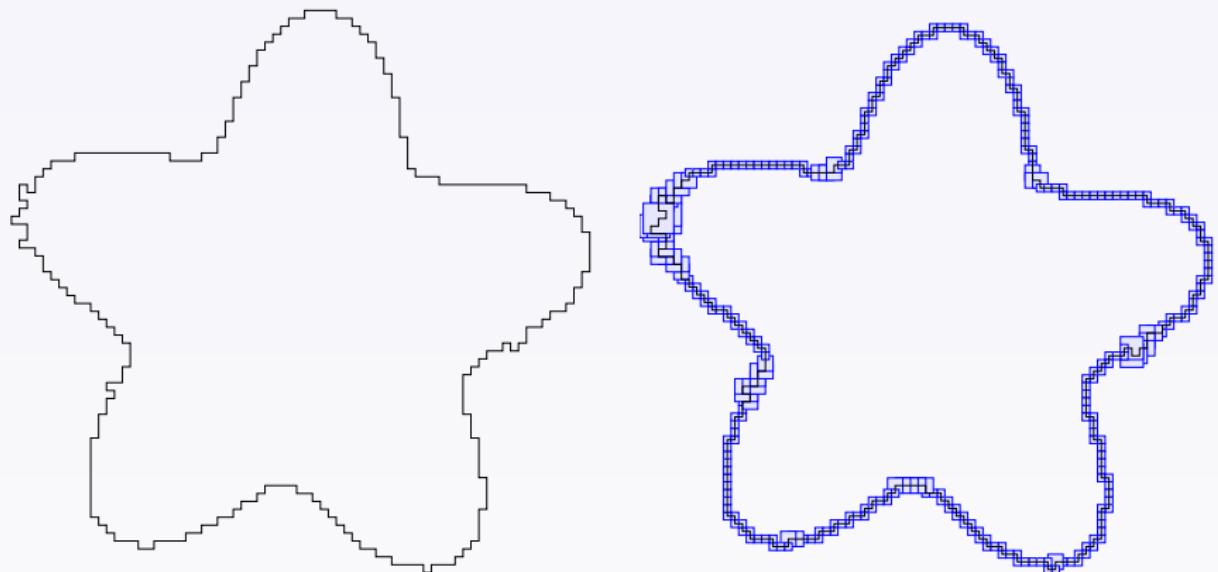


All computations are pointwise. No postprocessing.



Experiments: local noise level detection

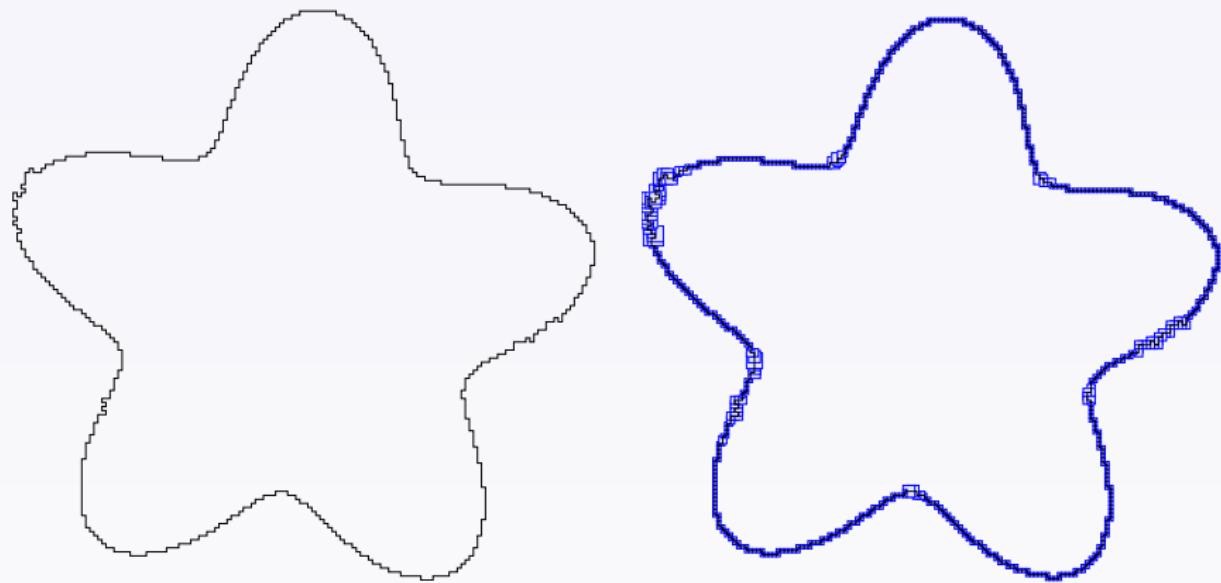
Flower with local noise



Local noise on resolution $R0$

Experiments: local noise level detection

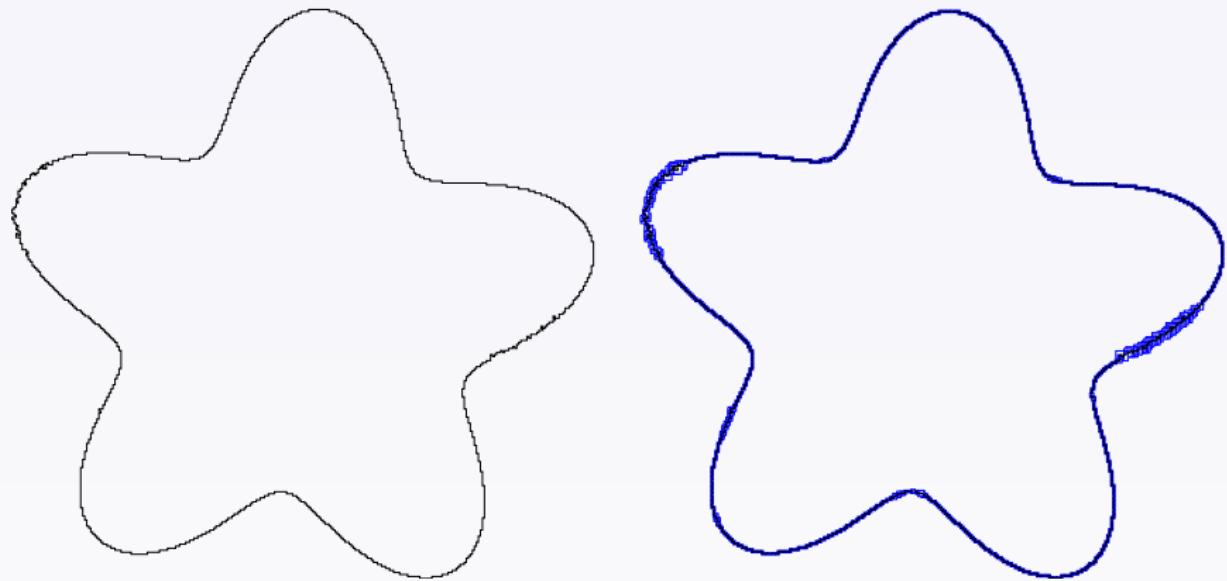
Flower with local noise



Local noise on resolution $R1$

Experiments: local noise level detection

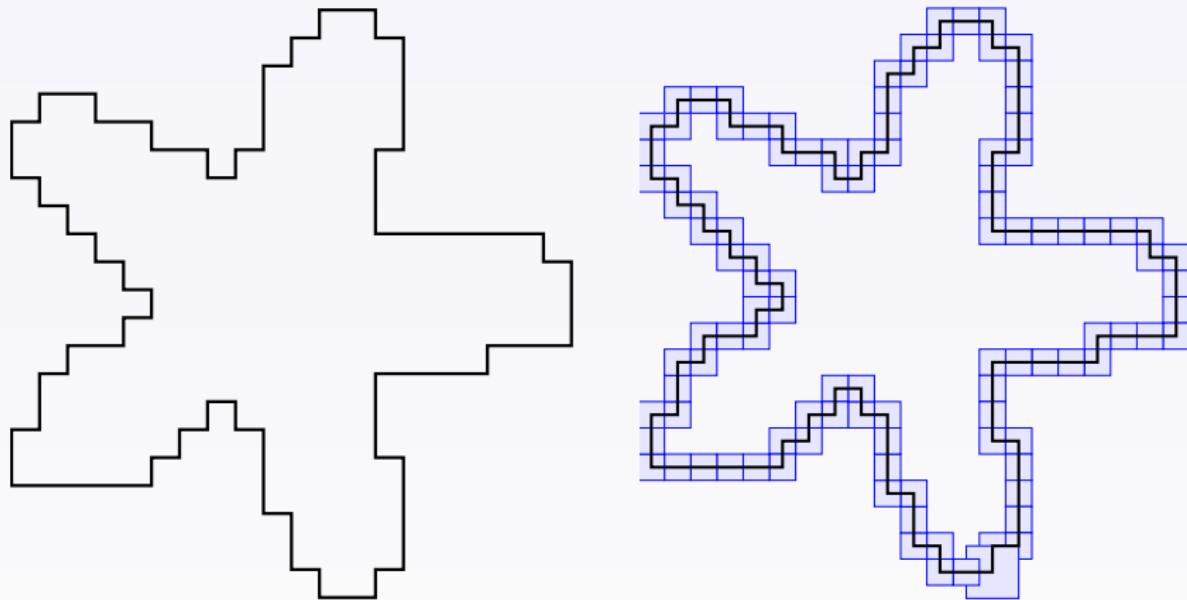
Flower with local noise



Local noise on resolution $R2$

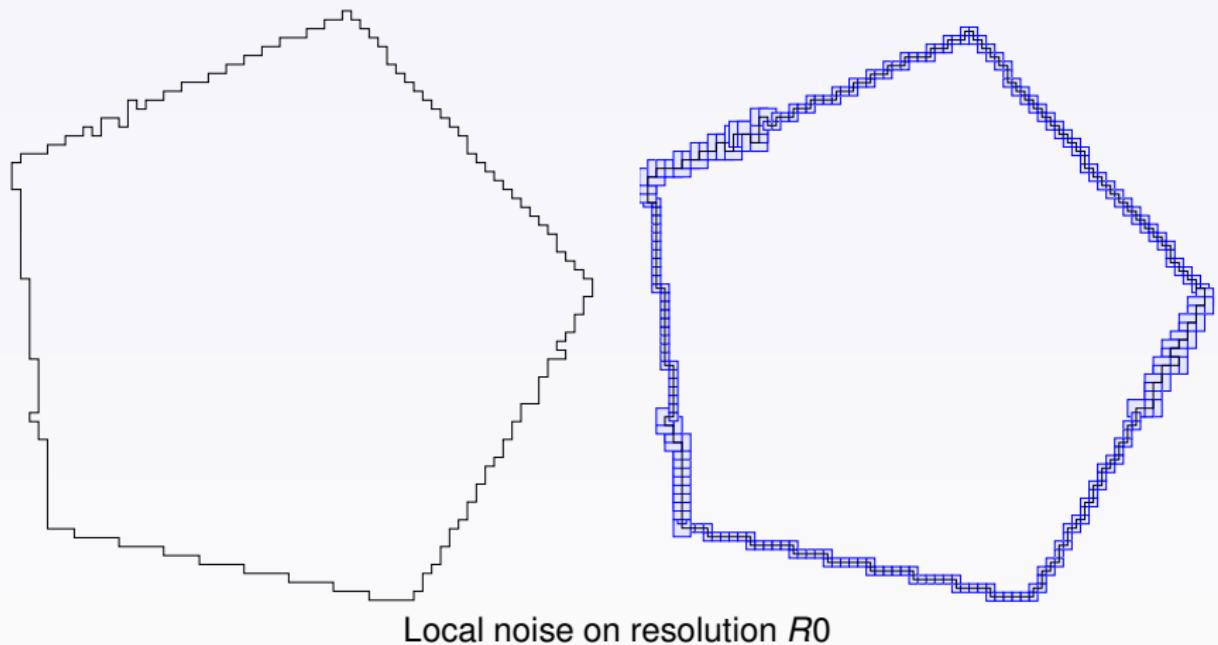
Experiments: local noise level detection

Tiny flower without noise



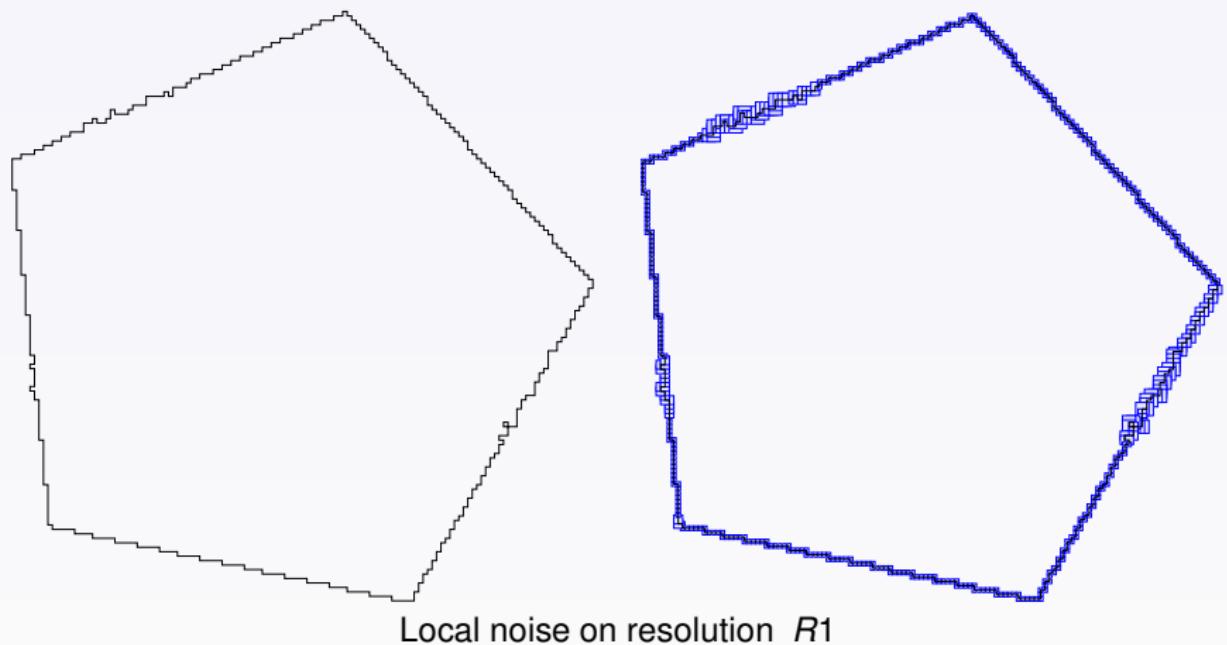
Local noise detection

Polygon with local noise



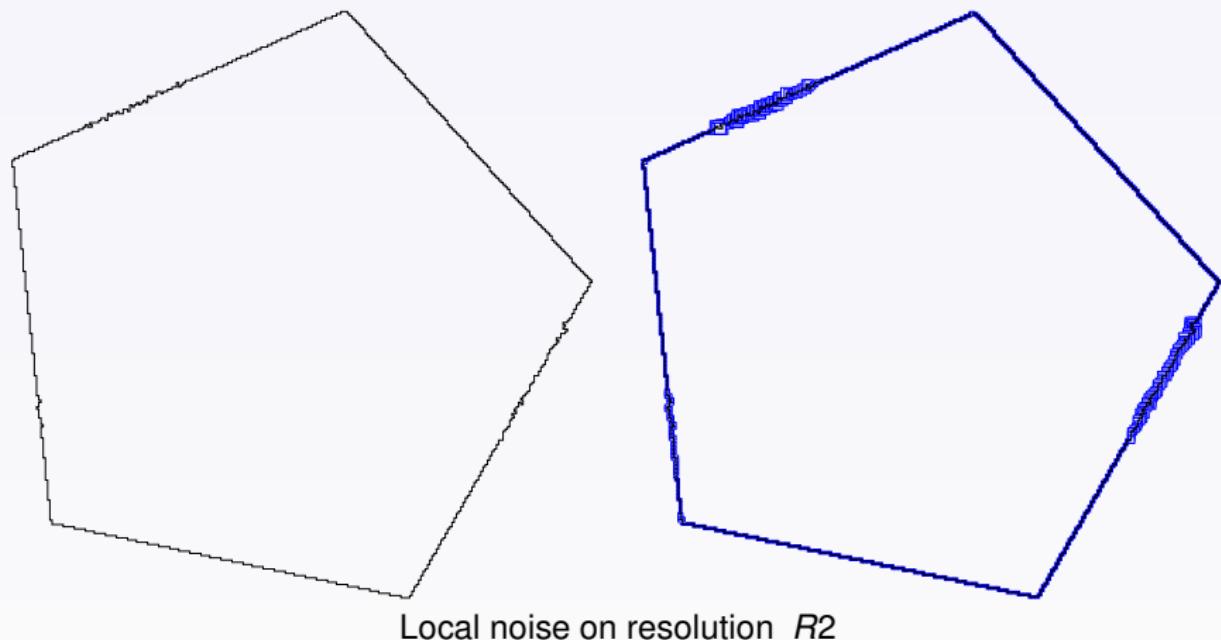
Local noise detection

Polygon with local noise



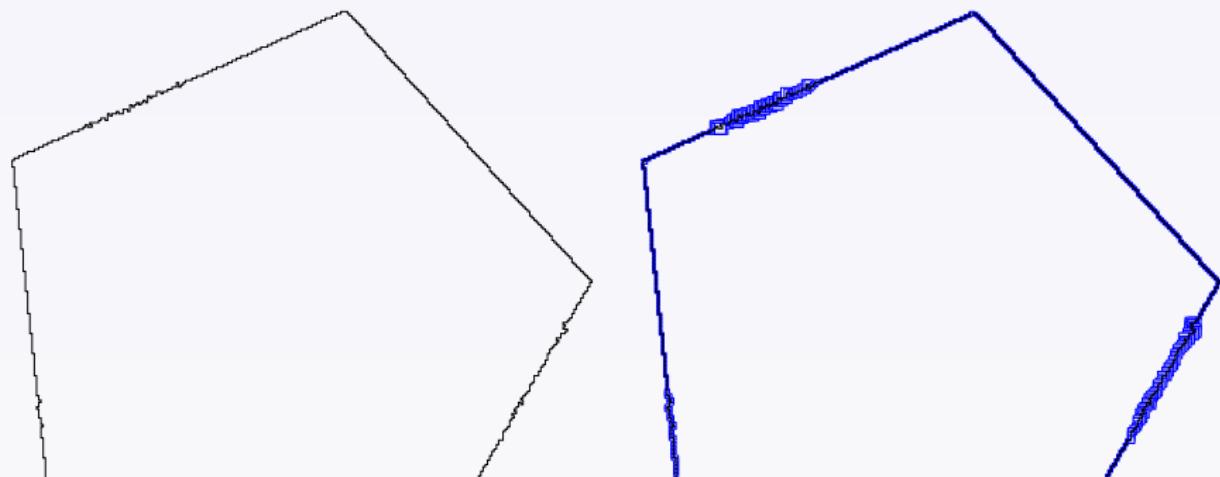
Local noise detection

Polygon with local noise



Local noise detection

Polygon with local noise

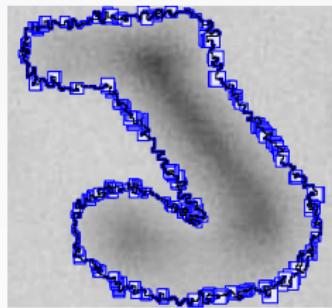
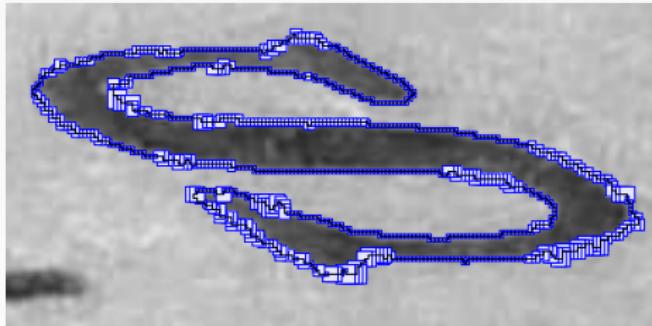
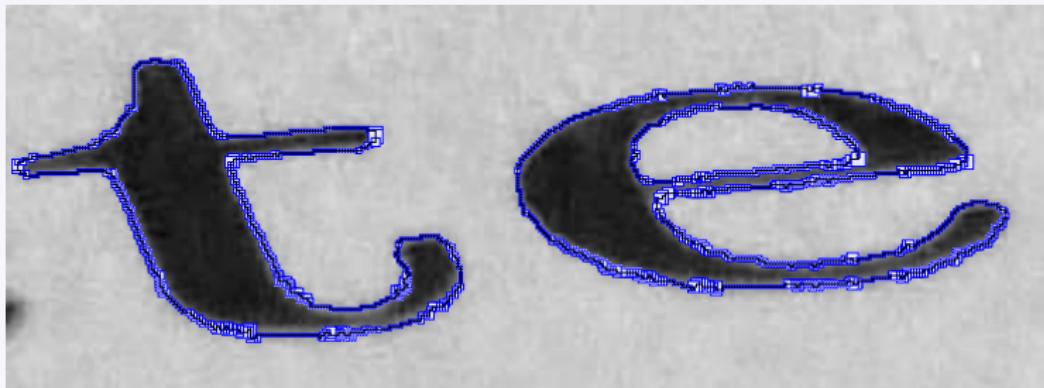


- Accuracy of noise detection independent of shape geometry, independent of shape resolution.
- Only one parameter : maximum level of subsampling (always 10 here).

Noise detection on real images

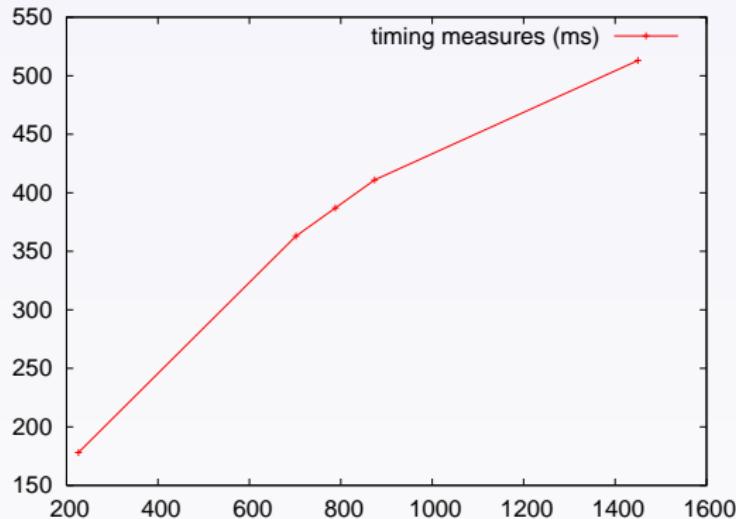


Noise detection on real images



Timing measures

Timing measures (including computation of all subsampled contours $\phi_i^{(x_0, y_0)}(C)$ and their maximal segments).

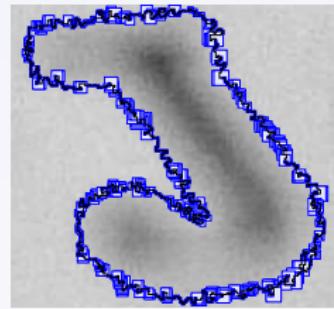


Obtained on a 2.4Ghz *Intel Core Duo*.

1.3. New applications: standard scale

Limits of the meaningful scale:

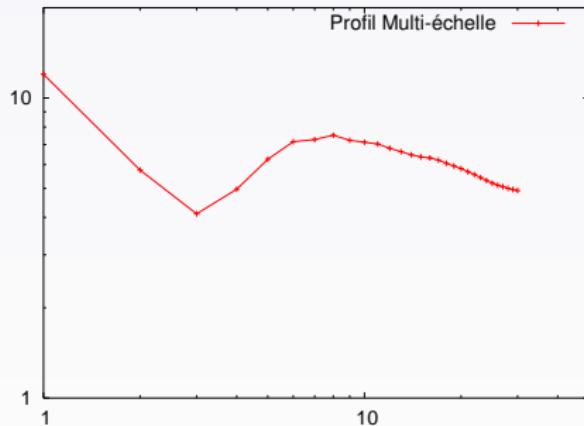
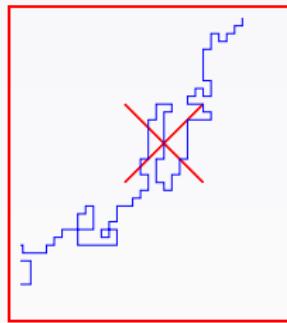
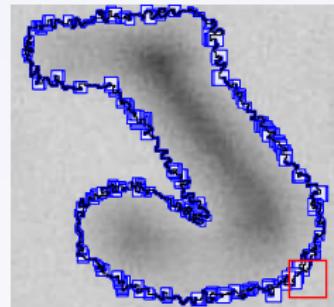
- The **first** significant scale.
- Not necessarily the awaited scale.



1.3. New applications: standard scale

Limits of the meaningful scale:

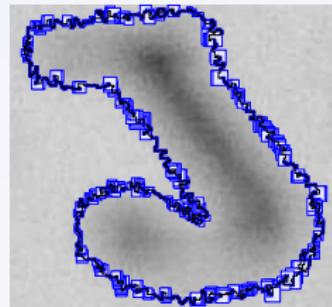
- The **first** significant scale.
- Not necessarily the awaited scale.



1.3. New applications: standard scale

Limits of the meaningful scale:

- The **first** significant scale.
- Not necessarily the awaited scale.



Multicale profile decomposition:

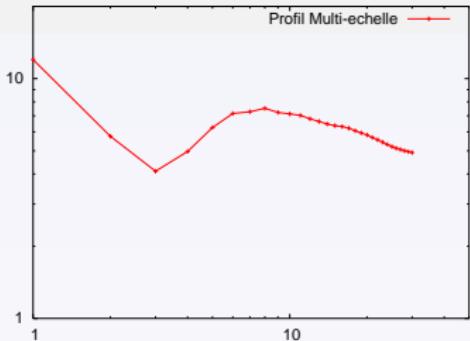
Decomposition of the multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$:

- In a sequence S_k of k pairs $(i_1, i_2), (i_2, i_3), \dots, (i_k, i_{k+1})$
- Each one is associated to a linear regression starting from the scale i_{k+1} and which is not true for $i_k - 1$.

1.3. New applications: standard scale

Limits of the meaningful scale:

- The **first** significant scale.
- Not necessarily the awaited scale.



Multicale profile decomposition:

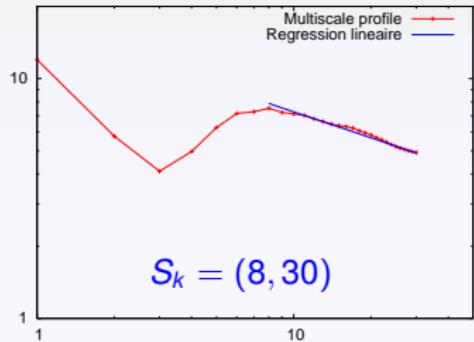
Decomposition of the multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$:

- In a sequence S_k of k pairs $(i_1, i_2), (i_2, i_3), \dots, (i_k, i_{k+1})$
- Each one is associated to a linear regression starting from the scale i_{k+1} and which is not true for $i_k - 1$.

1.3. New applications: standard scale

Limits of the meaningful scale:

- The **first** significant scale.
- Not necessarily the awaited scale.



Multicale profile decomposition:

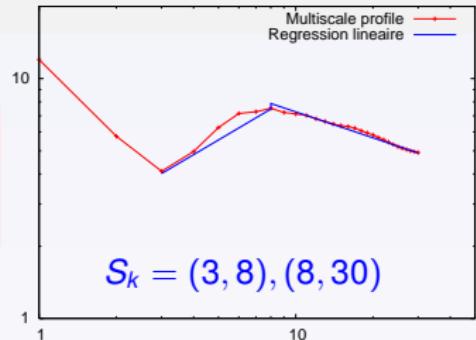
Decomposition of the multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$:

- In a sequence S_k of k pairs $(i_1, i_2), (i_2, i_3), \dots, (i_k, i_{k+1})$
- Each of one is associated to a linear regression starting from the scale i_{k+1} and which is not true for $i_k - 1$.

1.3. New applications: standard scale

Limits of the meaningful scale:

- The **first** significant scale.
- Not necessarily the awaited scale.



Multicale profile decomposition:

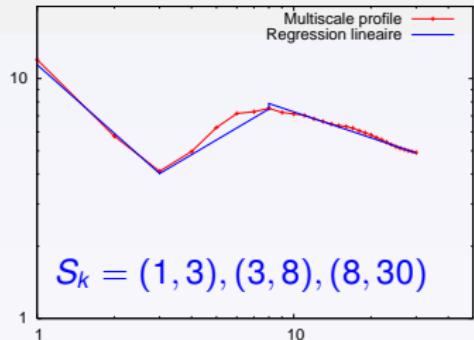
Decomposition of the multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$:

- In a sequence S_k of k pairs $(i_1, i_2), (i_2, i_3), \dots, (i_k, i_{k+1})$
- Each one is associated to a linear regression starting from the scale i_{k+1} and which is not true for $i_k - 1$.

1.3. New applications: standard scale

Limits of the meaningful scale:

- The **first** significant scale.
- Not necessarily the awaited scale.



Multicale profile decomposition:

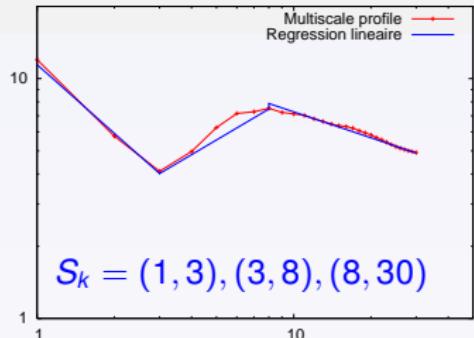
Decomposition of the multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$:

- In a sequence S_k of k pairs $(i_1, i_2), (i_2, i_3), \dots, (i_k, i_{k+1})$
- Each one is associated to a linear regression starting from the scale i_{k+1} and which is not true for $i_k - 1$.

1.3. New applications: standard scale

Limits of the meaningful scale:

- The **first** significant scale.
- Not necessarily the awaited scale.



Multicale profile decomposition:

Decomposition of the multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$:

- In a sequence S_k of k pairs $(i_1, i_2), (i_2, i_3), \dots, (i_k, i_{k+1})$
- Each one is associated to a linear regression starting from the scale i_{k+1} and which is not true for $i_k - 1$.

Standard scale

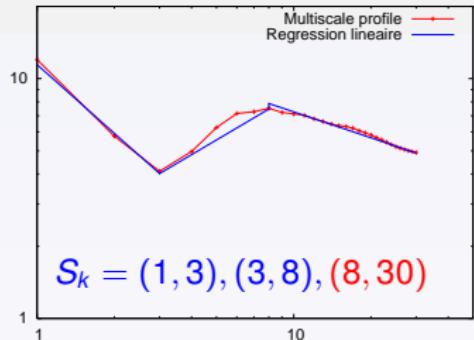
If we denote θ_{ij} the slope of the linear regression (i_j, i_{j+1}) , the **standard scale** is defined by the first interval (i_l, i_m) , $1 \leq l < m \leq k + 1$ such that:

$$\forall p, l \leq p < m, \theta_{ip} < 0 \text{ and } \theta_{i_{l-1}} \geq 0$$

1.3. New applications: standard scale

Limits of the meaningful scale:

- The **first** significant scale.
- Not necessarily the awaited scale.



Multicale profile decomposition:

Decomposition of the multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$:

- In a sequence S_k of k pairs $(i_1, i_2), (i_2, i_3), \dots, (i_k, i_{k+1})$
- Each one is associated to a linear regression starting from the scale i_{k+1} and which is not true for $i_k - 1$.

Standard scale

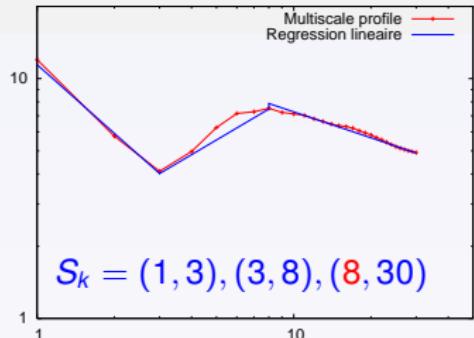
If we denote θ_{ij} the slope of the linear regression (i_j, i_{j+1}) , the **standard scale** is defined by the first interval (i_l, i_m) , $1 \leq l < m \leq k + 1$ such that:

$$\forall p, l \leq p < m, \theta_{ip} < 0 \text{ and } \theta_{i_{l-1}} \geq 0$$

1.3. New applications: standard scale

Limits of the meaningful scale:

- The **first** significant scale.
- Not necessarily the awaited scale.



Multicale profile decomposition:

Decomposition of the multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$:

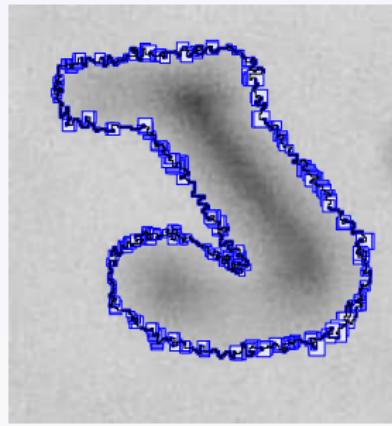
- In a sequence S_k of k pairs $(i_1, i_2), (i_2, i_3), \dots, (i_k, i_{k+1})$
- Each one is associated to a linear regression starting from the scale i_{k+1} and which is not true for $i_k - 1$.

Standard scale

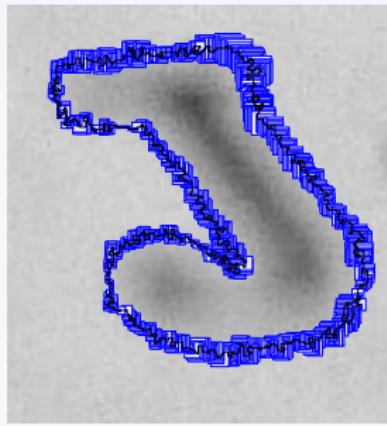
If we denote θ_{ij} the slope of the linear regression (i_j, i_{j+1}) , the **standard scale** is defined by the first interval (i_l, i_m) , $1 \leq l < m \leq k + 1$ such that:

$$\forall p, l \leq p < m, \theta_{ip} < 0 \text{ and } \theta_{i_{l-1}} \geq 0$$

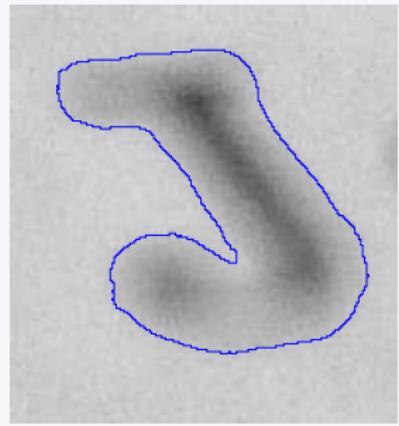
1.3. New applications: standard scale (results)



(a) *Meaningful Scale*

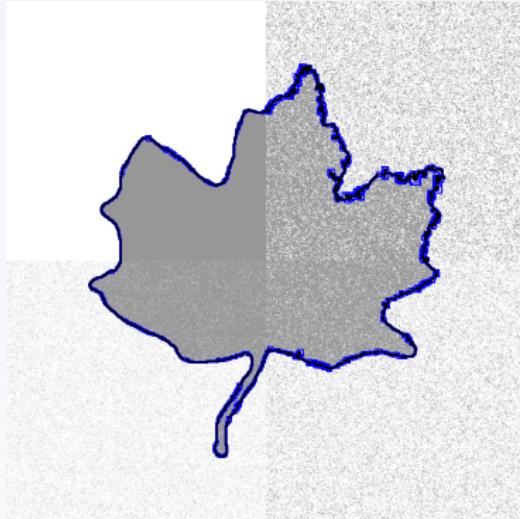


(b) *Standard Scale*

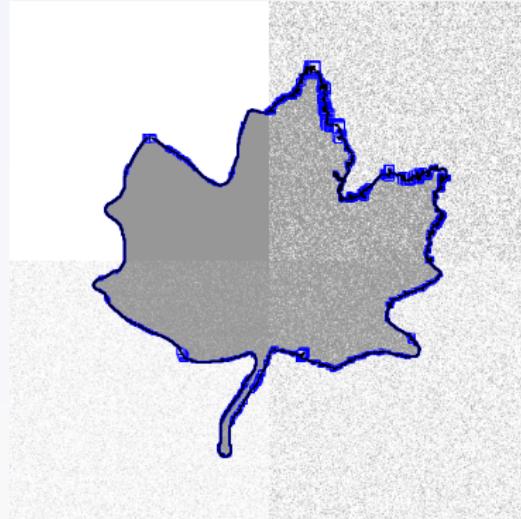


(c) median smoothing (E Std)

1.3. New applications: standard scale (results)

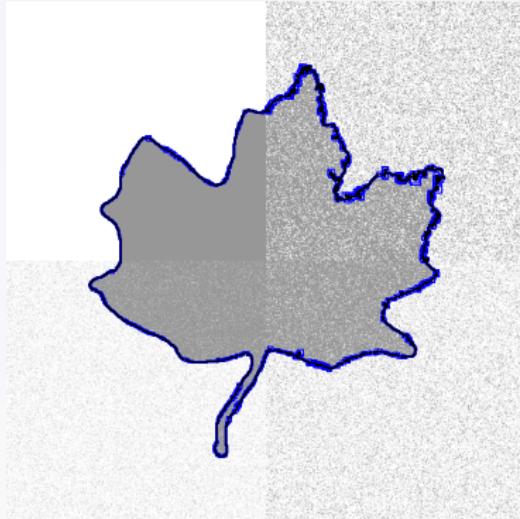


(a) *Meaningful Scale*
(max scale =30)

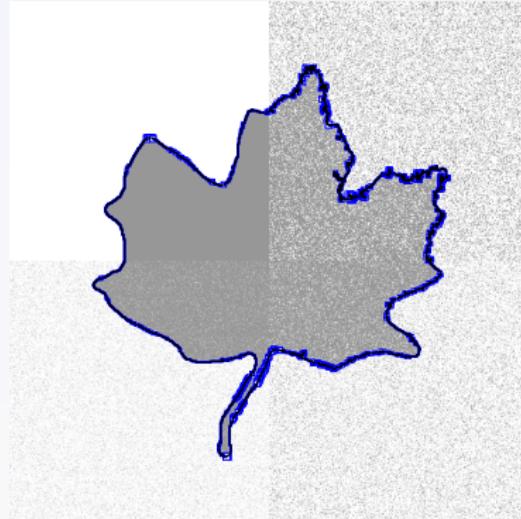


(b) *Standard Scale*
(max scale =30)

1.3. New applications: standard scale (results)



(a) *Meaningful Scale*
(max scale = 20)



(b) *Standard Scale*
(max scale = 20)

1.3. New applications: noise removal

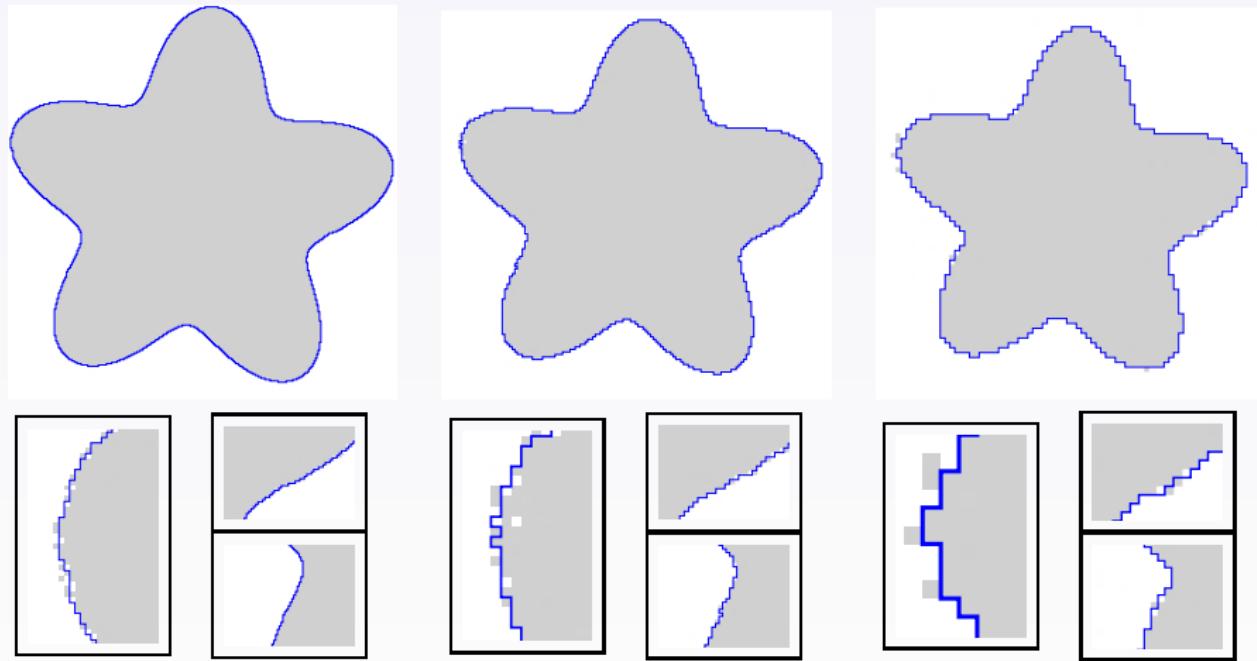
Noise removal on digital contours:

- Application of simple median filter.
- Filter size defined according to the noise level detected: $2\nu(P) + 1$

1.3. New applications: noise removal

Noise removal on digital contours:

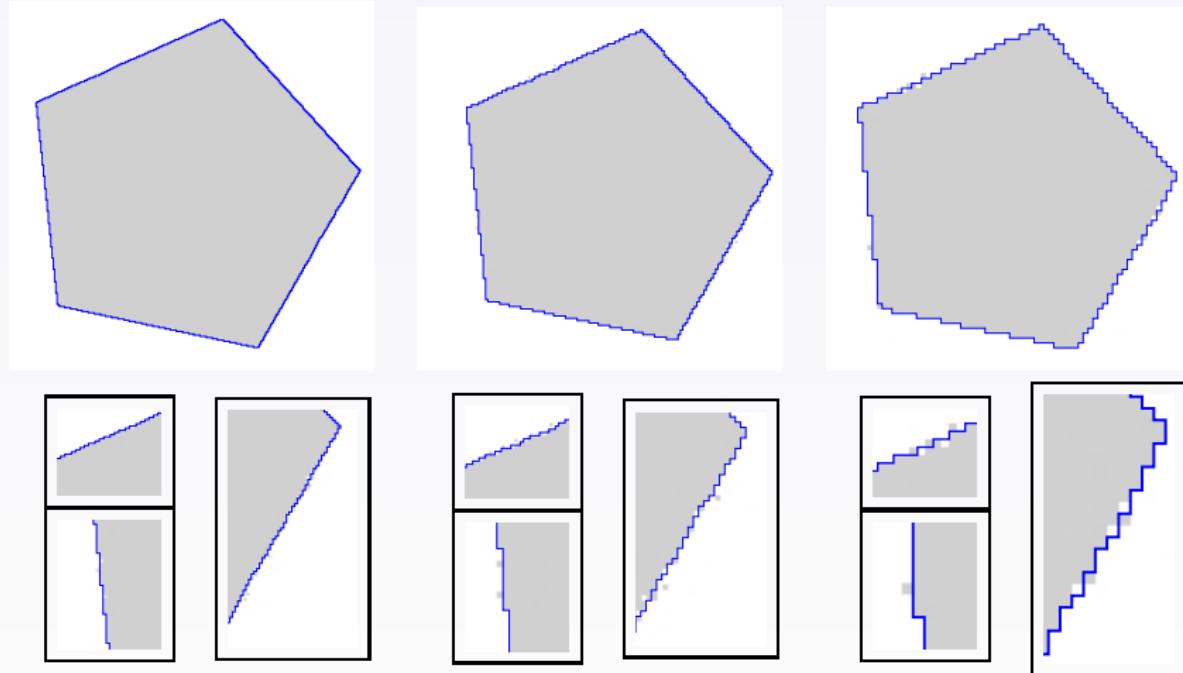
- Application of simple median filter.
- Filter size defined according to the noise level detected: $2\nu(P) + 1$



1.3. New applications: noise removal

Noise removal on digital contours:

- Application of simple median filter.
- Filter size defined according to the noise level detected: $2\nu(P) + 1$



1.3. New applications: noise removal

Noise removal on digital contours:

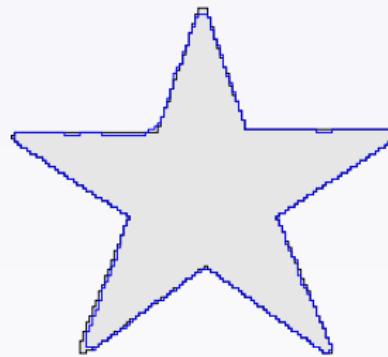
- Application of simple median filter.
- Filter size defined according to the noise level detected: $2\nu(P) + 1$



1.3. New applications: noise removal

Noise removal on digital contours:

- Application of simple median filter.
- Filter size defined according to the noise level detected: $2\nu(P) + 1$

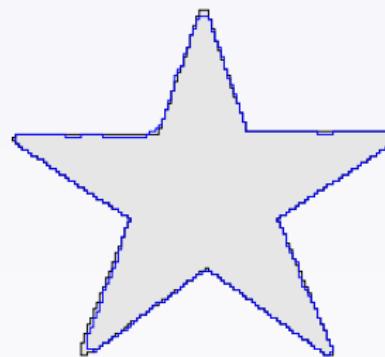


median filter
(meaningful scales)

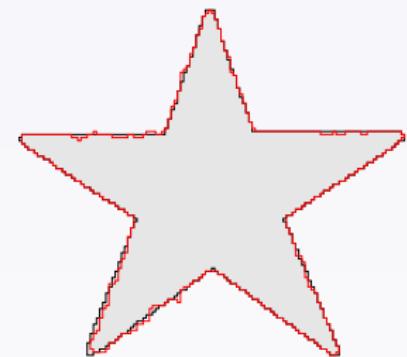
1.3. New applications: noise removal

Noise removal on digital contours:

- Application of simple median filter.
- Filter size defined according to the noise level detected: $2\nu(P) + 1$



median filter
(meaningful scales)

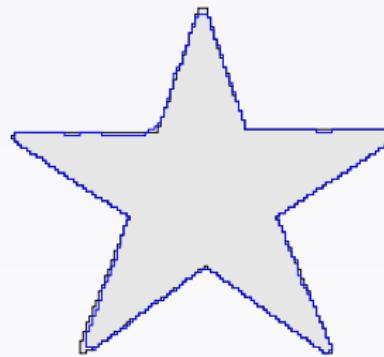


median filter size=2

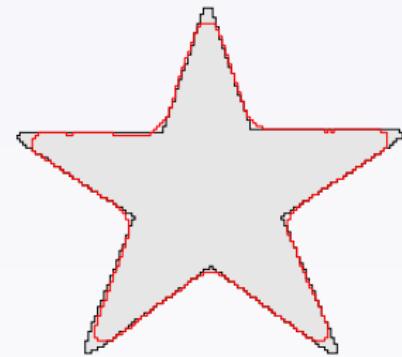
1.3. New applications: noise removal

Noise removal on digital contours:

- Application of simple median filter.
- Filter size defined according to the noise level detected: $2\nu(P) + 1$



median filter
(meaningful scales)

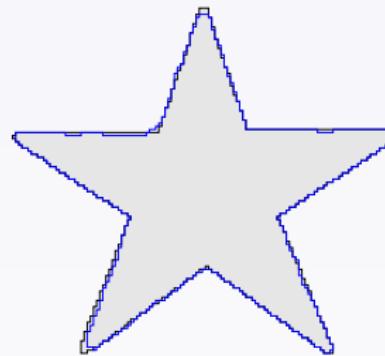


median filter size=6

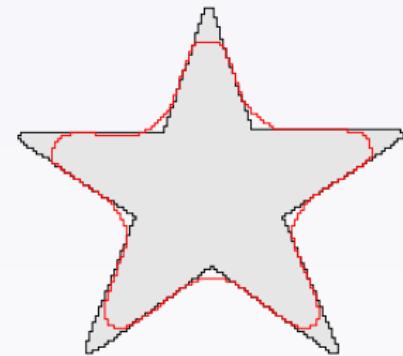
1.3. New applications: noise removal

Noise removal on digital contours:

- Application of simple median filter.
- Filter size defined according to the noise level detected: $2\nu(P) + 1$



median filter
(meaningful scales)



median filter size=10

1.3. New applications: noise removal

Noise removal on digital contours:

- Application of simple median filter.
- Filter size defined according to the noise level detected: $2\nu(P) + 1$



1.3. New applications: noise removal

Noise removal on digital contours:

- Application of simple median filter.
- Filter size defined according to the noise level detected: $2\nu(P) + 1$



1.3. New applications: noise removal

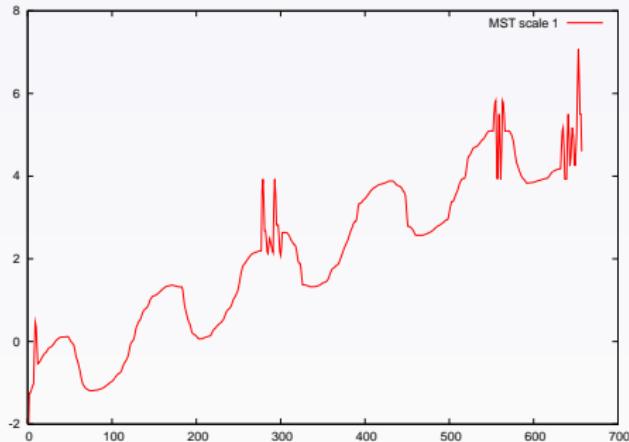
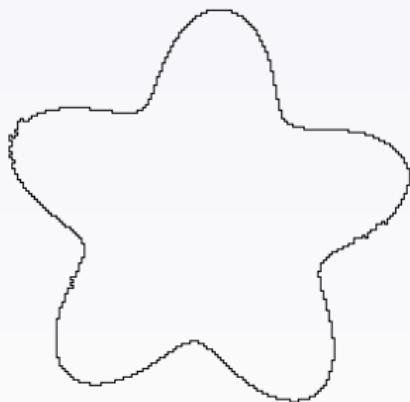
Application to geometric estimator:

- Normal estimator $\lambda - MST$ [Lachaud07 et al].
- Principle: for each contour point C_k , we associate the normal computed from the meaningful scale level: $\phi_{\nu(C_k)+1}^{0,0}(C_k)$.

1.3. New applications: noise removal

Application to geometric estimator:

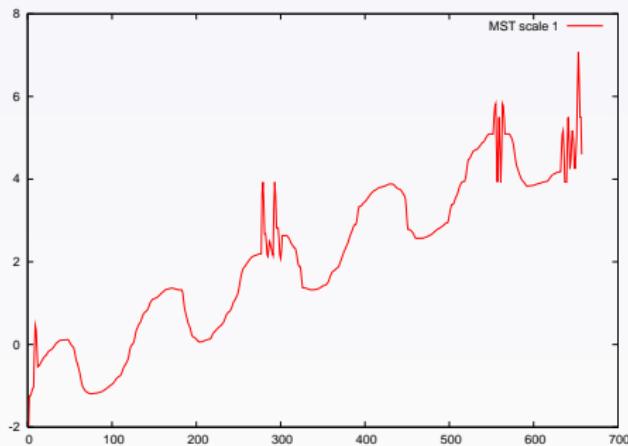
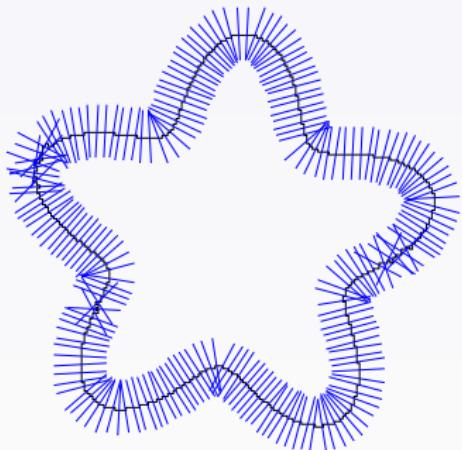
- Normal estimator λ – MST [Lachaud07 et al].
- Principle: for each contour point C_k , we associate the normal computed from the meaningful scale level: $\phi_{\nu(C_k)+1}^{0,0}(C_k)$.



1.3. New applications: noise removal

Application to geometric estimator:

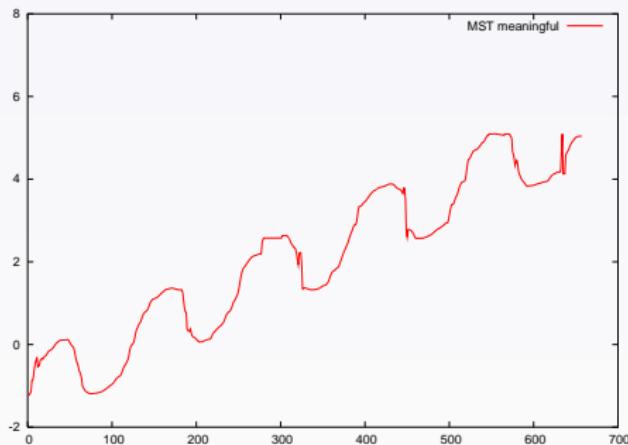
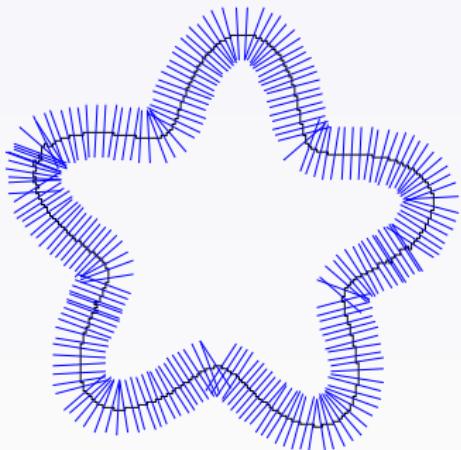
- Normal estimator λ – *MST* [Lachaud07 et al].
- Principle: for each contour point C_k , we associate the normal computed from the meaningful scale level: $\phi_{\nu(C_k)+1}^{0,0}(C_k)$.



1.3. New applications: noise removal

Application to geometric estimator:

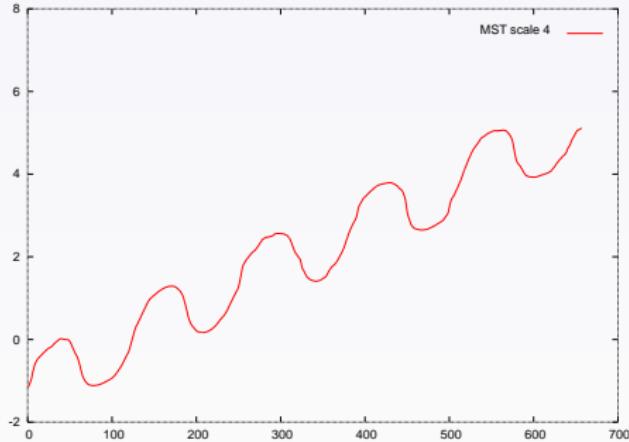
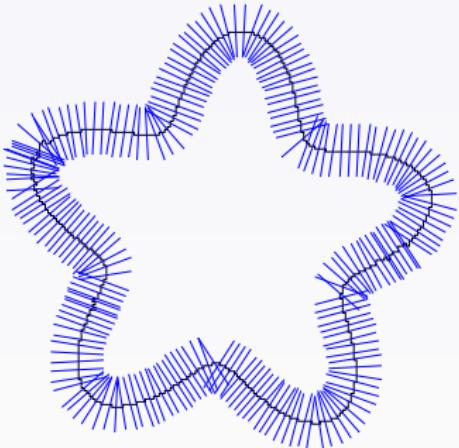
- Normal estimator λ – MST [Lachaud07 et al].
- Principle: for each contour point C_k , we associate the normal computed from the meaningful scale level: $\phi_{\nu(C_k)+1}^{0,0}(C_k)$.



1.3. New applications: noise removal

Application to geometric estimator:

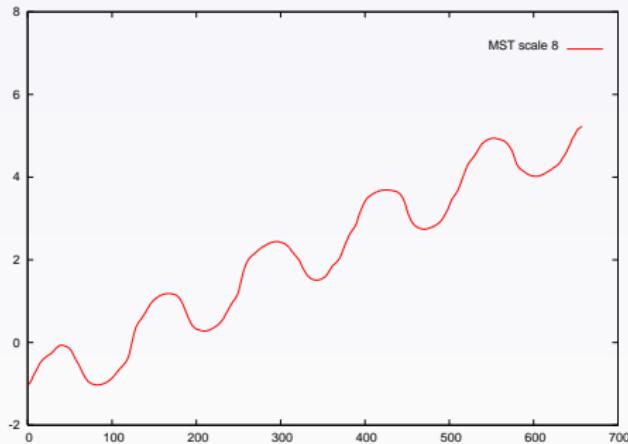
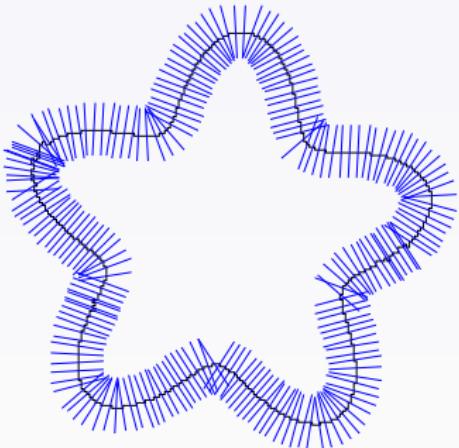
- Normal estimator $\lambda - MST$ [Lachaud07 et al].
- Principle: for each contour point C_k , we associate the normal computed from the meaningful scale level: $\phi_{\nu(C_k)+1}^{0,0}(C_k)$.



1.3. New applications: noise removal

Application to geometric estimator:

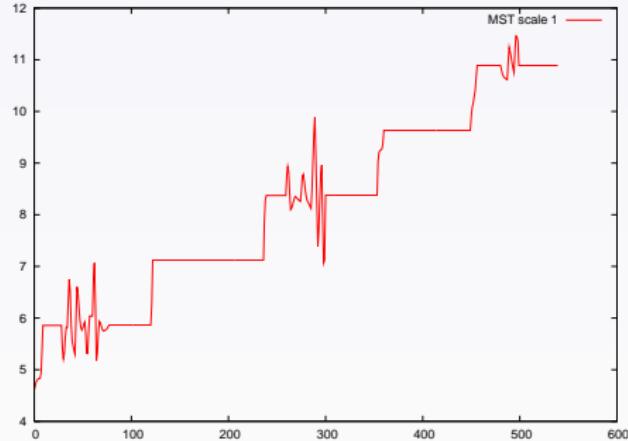
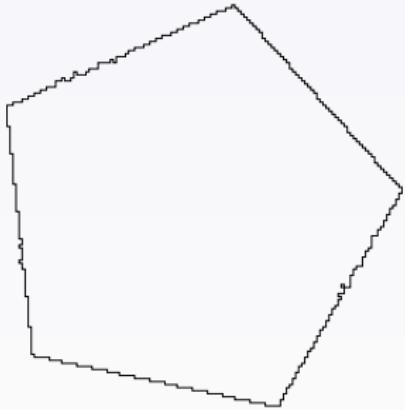
- Normal estimator λ – MST [Lachaud07 et al].
- Principle: for each contour point C_k , we associate the normal computed from the meaningful scale level: $\phi_{\nu(C_k)+1}^{0,0}(C_k)$.



1.3. New applications: noise removal

Application to geometric estimator:

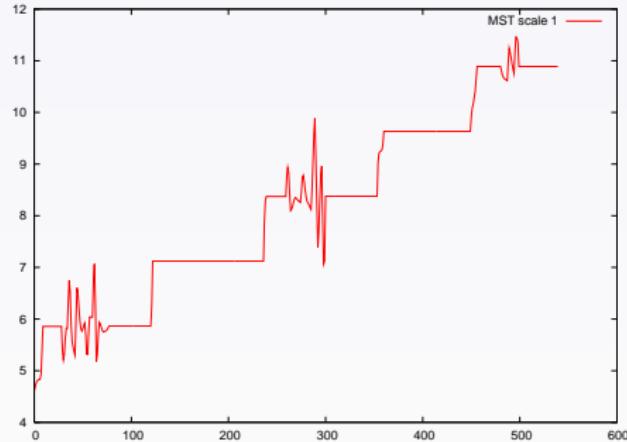
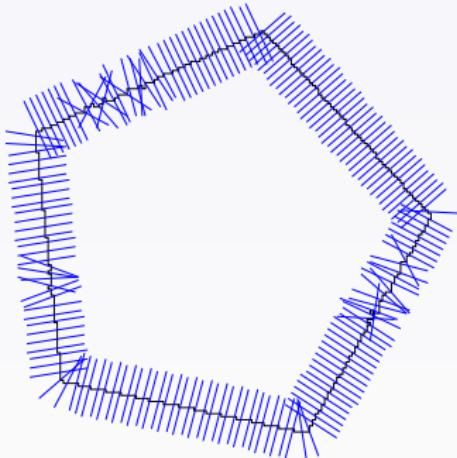
- Normal estimator λ – *MST* [Lachaud07 et al].
- Principle: for each contour point C_k , we associate the normal computed from the meaningful scale level: $\phi_{\nu(C_k)+1}^{0,0}(C_k)$.



1.3. New applications: noise removal

Application to geometric estimator:

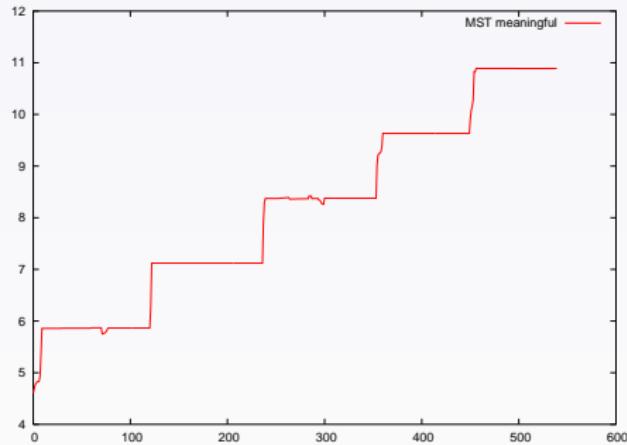
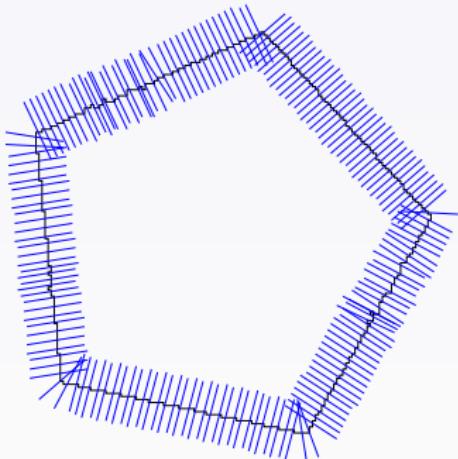
- Normal estimator λ – MST [Lachaud07 et al].
- Principle: for each contour point C_k , we associate the normal computed from the meaningful scale level: $\phi_{\nu(C_k)+1}^{0,0}$.



1.3. New applications: noise removal

Application to geometric estimator:

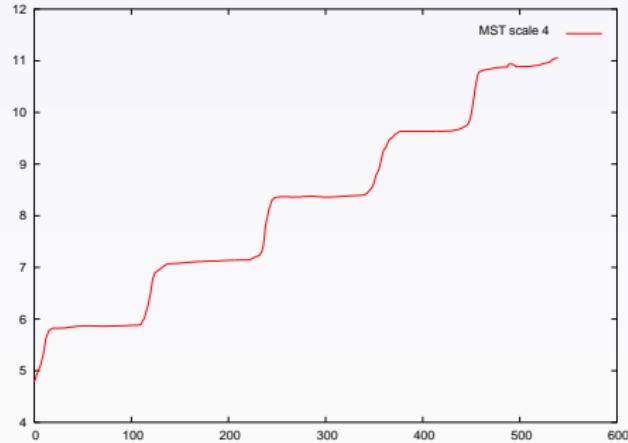
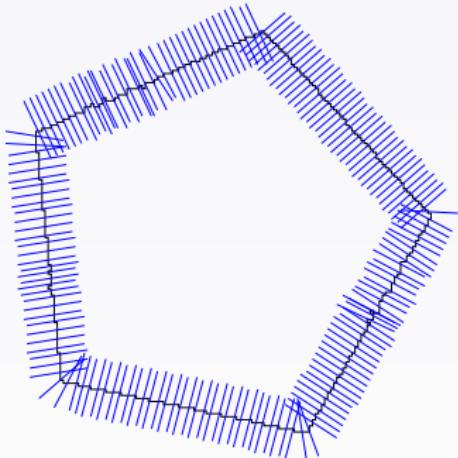
- Normal estimator λ – MST [Lachaud07 et al].
- Principle: for each contour point C_k , we associate the normal computed from the meaningful scale level: $\phi_{\nu(C_k)+1}^{0,0}$.



1.3. New applications: noise removal

Application to geometric estimator:

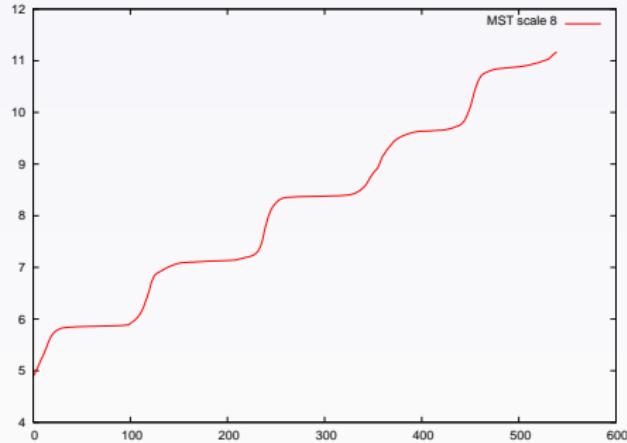
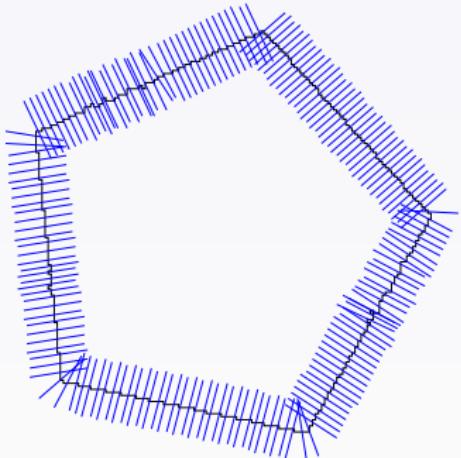
- Normal estimator λ – MST [Lachaud07 et al].
- Principle: for each contour point C_k , we associate the normal computed from the meaningful scale level: $\phi_{\nu(C_k)+1}^{0,0}(C_k)$.



1.3. New applications: noise removal

Application to geometric estimator:

- Normal estimator λ – MST [Lachaud07 et al].
- Principle: for each contour point C_k , we associate the normal computed from the meaningful scale level: $\phi_{\nu(C_k)+1}^{0,0}(C_k)$.



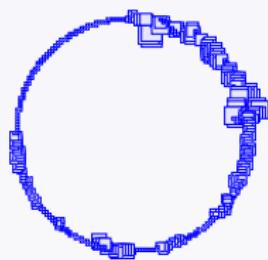
1.3. Other application examples

Polygonalisation from irregular isothetic grid [Vacavant 13 et al.,]

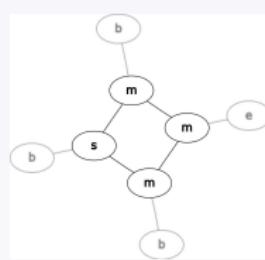
- Use the meaningful boxes as input to irregular isothetic grid.
- Reeb graph.



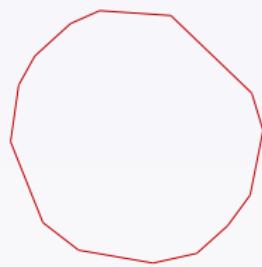
source



meaningful scale



reeb graph



polygonal representation

1.3. Other application examples

Polygonalisation from irregular isothetic grid [Vacavant 13 et al.,]

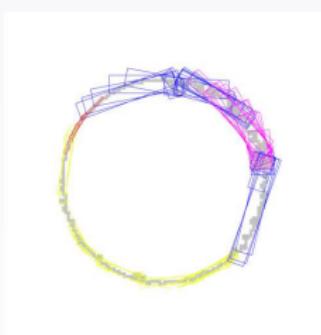
- Use the meaningful boxes as input to irregular isothetic grid.
- Reeb graph.

Adaptive tangential cover [Ngo et al., 15]

- Adapt the thickness according the meaningful scales.
- Local adaption.
- Useful to geometric estimators or polygonalisation.



source

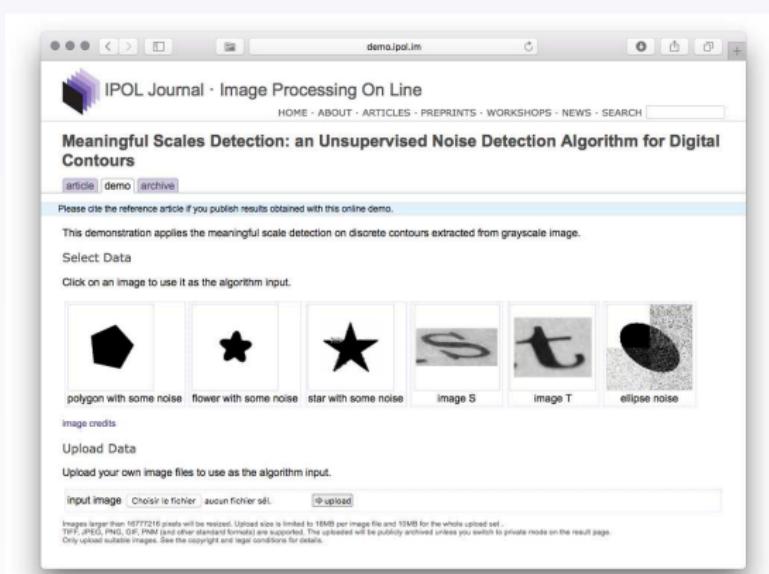


adaptive tangential cover

1.3 Online demonstration and limitations

Online demonstration available in an IPOL publication: [Kerautret & Lachaud, 2014]

- The algorithm can be tested here: <http://www.ipol.im/pub/art/2014/75/>
- IPOL article with source code (based on the ImaGene Library).



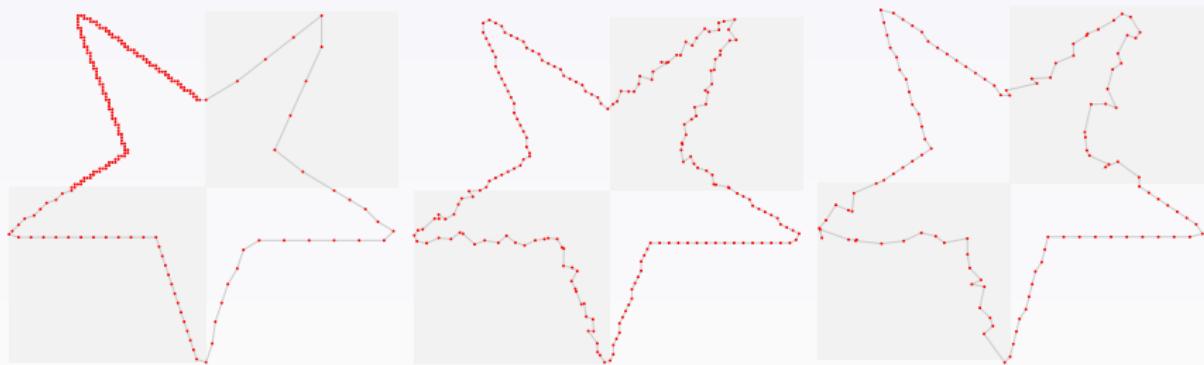
1.3 Online demonstration and limitations

Online demonstration available in an IPOL publication: [Kerautret & Lachaud, 2014]

- The algorithm can be tested here: <http://www.ipol.im/pub/art/2014/75/>
- IPOL article with source code (based on the ImaGene Library).

Limitation

- Can only be applied on discrete contours.



Overview of the presentation

1. Introduction to the Meaningful Scale Detection

- 1.1 Main idea of the meaningful scale
- 1.2 Meaningful Scale Profile and Noise Detection
- 1.3 Experiments and new applications

2. Extension to the Meaningful Thickness

- 2.2 Meaningful Thickness profile
- 2.3 Experiments, comparisons and applications
- 2.4 Computing the meaningful thickness in the DGtal Library

3. Highlighting Reproducible Research

- 3.1 Image Processing On Line Presentation: principle and current form
- 3.2 Structure of an IPOL demonstration

2.1 Extension to the Meaningful Thickness

Main idea

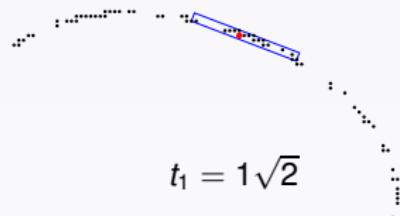
- Use another primitive to process set of points of the Euclidean plane.



2.1 Extension to the Meaningful Thickness

Main idea

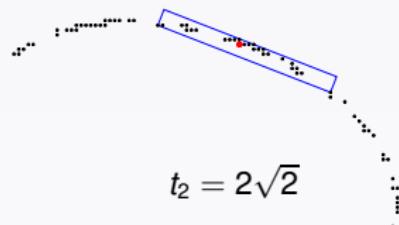
- Use another primitive to process set of points of the Euclidean plane.
- **alpha-thick segments** [Debled-Rennesson et al., 2006, Faure et al., 2009]:
 - Defined with a thick parameter: t



2.1 Extension to the Meaningful Thickness

Main idea

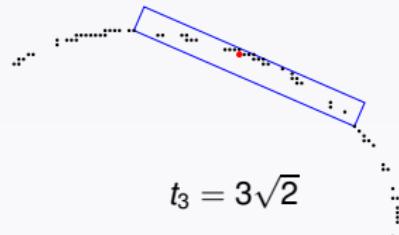
- Use another primitive to process set of points of the Euclidean plane.
- **alpha-thick segments** [Debled-Rennesson et al., 2006, Faure et al., 2009]:
 - Defined with a thick parameter: t



2.1 Extension to the Meaningful Thickness

Main idea

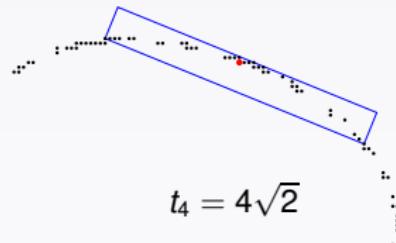
- Use another primitive to process set of points of the Euclidean plane.
- **alpha-thick segments** [Debled-Rennesson et al., 2006, Faure et al., 2009]:
 - Defined with a thick parameter: t



2.1 Extension to the Meaningful Thickness

Main idea

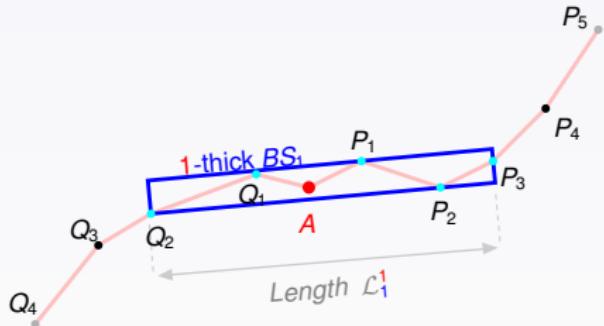
- Use another primitive to process set of points of the Euclidean plane.
- **alpha-thick segments** [Debled-Rennesson et al., 2006, Faure et al., 2009]:
 - Defined with a thick parameter: t



2.1 Extension to the Meaningful Thickness

Main idea

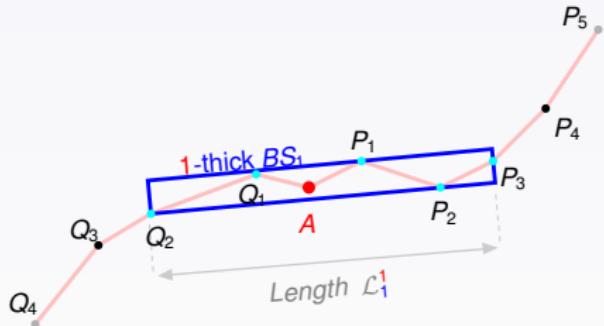
- Use another primitive to process set of points of the Euclidean plane.
- alpha-thick segments [Debled-Rennesson et al., 2006, Faure et al., 2009]:
 - Defined with a thick parameter: t
 - maximal isothetic thickness of the convex hull.
 $\Rightarrow (P_1, Q_1, Q_2, P_2, P_3)$



2.1 Extension to the Meaningful Thickness

Main idea

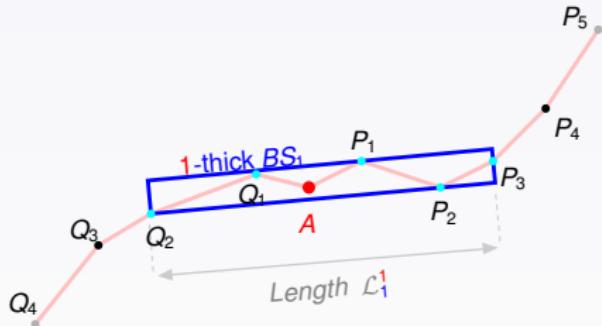
- Use another primitive to process set of points of the Euclidean plane.
- **alpha-thick segments** [Debled-Rennesson et al., 2006, Faure et al., 2009]:
 - Defined with a thick parameter: t
 - maximal isothetic thickness of the convex hull.
 $\Rightarrow (P_1, Q_1, Q_2, P_2, P_3)$
- **Maximal alpha-thick segments.**



2.1 Extension to the Meaningful Thickness

Main idea

- Use another primitive to process set of points of the Euclidean plane.
- **alpha-thick segments** [Deblieck-Rennesson et al., 2006, Faure et al., 2009]:
 - Defined with a thick parameter: t
 - maximal isothetic thickness of the convex hull.
 $\Rightarrow (P_1, Q_1, Q_2, P_2, P_3)$
- **Maximal alpha-thick segments.**
- The multi scale behaviour is obtained from the t parameter.



Thickness asymptotic behaviour

Multi-thickness property (from experiments)

The plots of the lengths $\mathcal{L}_j^{t_i} / t_i$ in log-scale are approximately affine with negative slopes as specified besides:

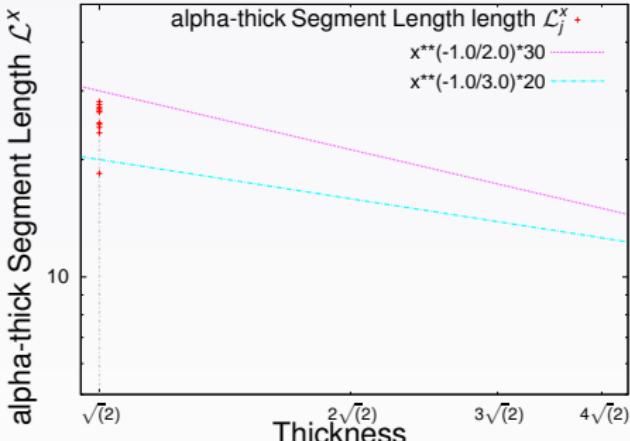
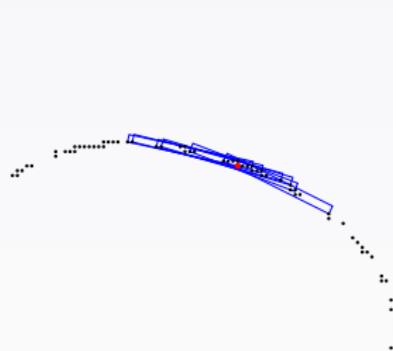
plot	expected slope	
	curved part	flat part
$(\log(t_i), \log(\max_j \mathcal{L}_j^{t_i} / t_i))$	$\approx -\frac{1}{2}$	≈ -1
$(\log(t_i), \log(\min_j \mathcal{L}_j^{t_i} / t_i))$	$\approx -\frac{1}{3}$	≈ -1

Thickness asymptotic behaviour

Multi-thickness property (from experiments)

The plots of the lengths $\mathcal{L}_j^{t_i} / t_i$ in log-scale are approximately affine with negative slopes as specified besides:

plot	expected slope	
	curved part	flat part
$(\log(t_i), \log(\max_j \mathcal{L}_j^{t_i} / t_i))$	$\approx -\frac{1}{2}$	≈ -1
$(\log(t_i), \log(\min_j \mathcal{L}_j^{t_i} / t_i))$	$\approx -\frac{1}{3}$	≈ -1

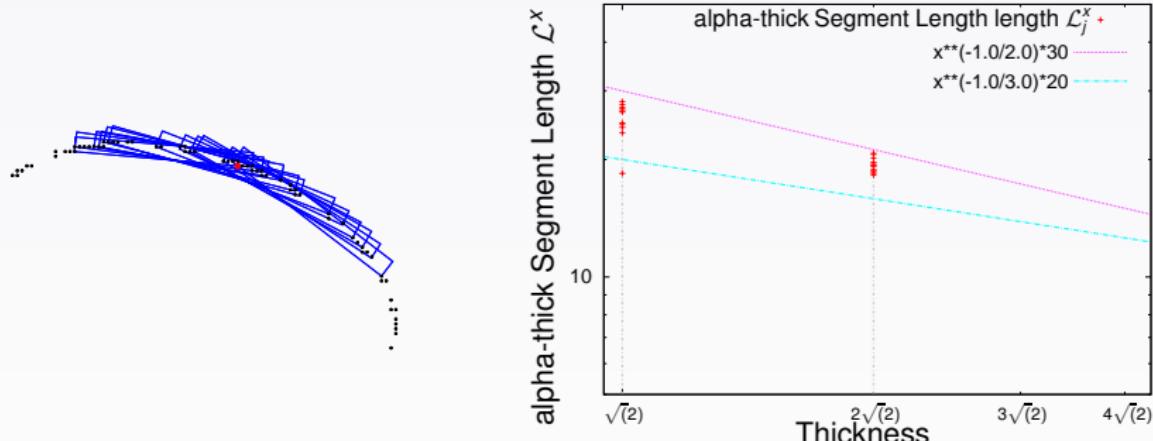


Thickness asymptotic behaviour

Multi-thickness property (from experiments)

The plots of the lengths $\mathcal{L}_j^{t_i} / t_i$ in log-scale are approximately affine with negative slopes as specified besides:

plot	expected slope	
	curved part	flat part
$(\log(t_i), \log(\max_j \mathcal{L}_j^{t_i} / t_i))$	$\approx -\frac{1}{2}$	≈ -1
$(\log(t_i), \log(\min_j \mathcal{L}_j^{t_i} / t_i))$	$\approx -\frac{1}{3}$	≈ -1

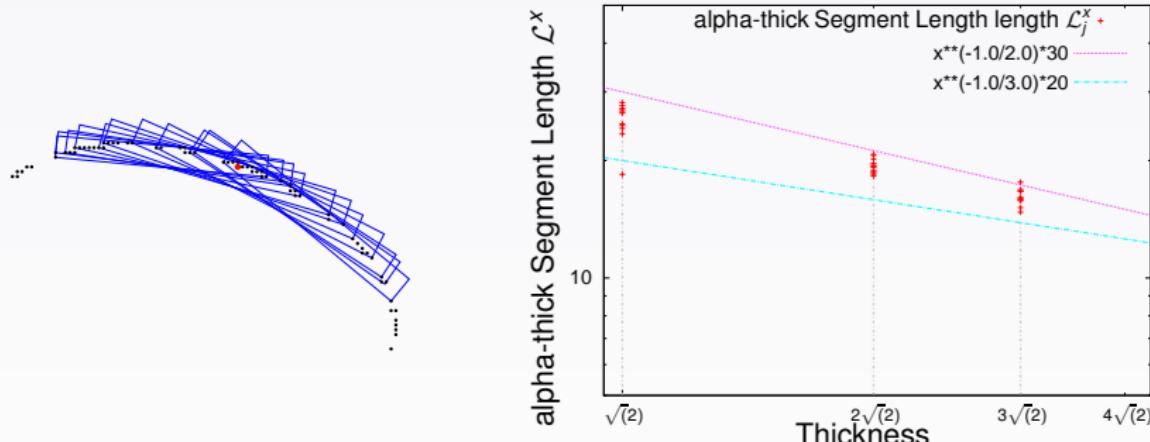


Thickness asymptotic behaviour

Multi-thickness property (from experiments)

The plots of the lengths $\mathcal{L}_j^{t_i} / t_i$ in log-scale are approximately affine with negative slopes as specified besides:

plot	expected slope	
	curved part	flat part
$(\log(t_i), \log(\max_j \mathcal{L}_j^{t_i} / t_i))$	$\approx -\frac{1}{2}$	≈ -1
$(\log(t_i), \log(\min_j \mathcal{L}_j^{t_i} / t_i))$	$\approx -\frac{1}{3}$	≈ -1

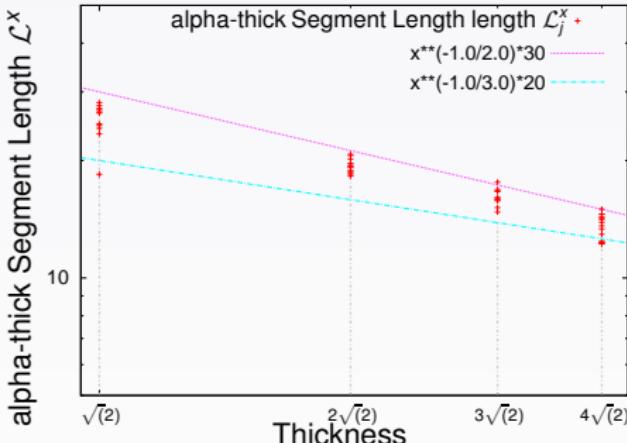
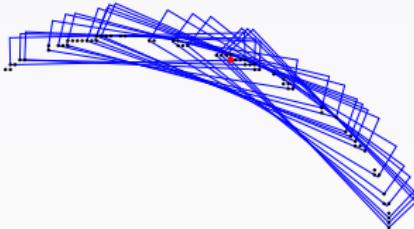


Thickness asymptotic behaviour

Multi-thickness property (from experiments)

The plots of the lengths $\mathcal{L}_j^{t_i} / t_i$ in log-scale are approximately affine with negative slopes as specified besides:

plot	expected slope	
	curved part	flat part
$(\log(t_i), \log(\max_j \mathcal{L}_j^{t_i} / t_i))$	$\approx -\frac{1}{2}$	≈ -1
$(\log(t_i), \log(\min_j \mathcal{L}_j^{t_i} / t_i))$	$\approx -\frac{1}{3}$	≈ -1

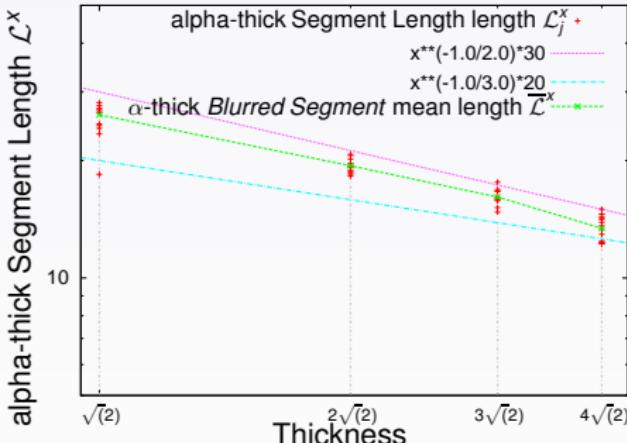
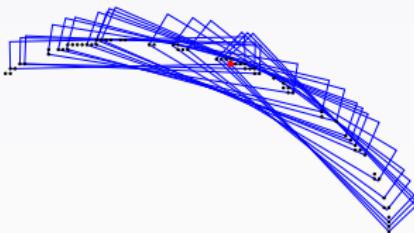


Thickness asymptotic behaviour

Multi-thickness property (from experiments)

The plots of the lengths $\mathcal{L}_j^{t_i} / t_i$ in log-scale are approximately affine with negative slopes as specified besides:

plot	expected slope	
	curved part	flat part
$(\log(t_i), \log(\max_j \mathcal{L}_j^{t_i} / t_i))$	$\approx -\frac{1}{2}$	≈ -1
$(\log(t_i), \log(\min_j \mathcal{L}_j^{t_i} / t_i))$	$\approx -\frac{1}{3}$	≈ -1

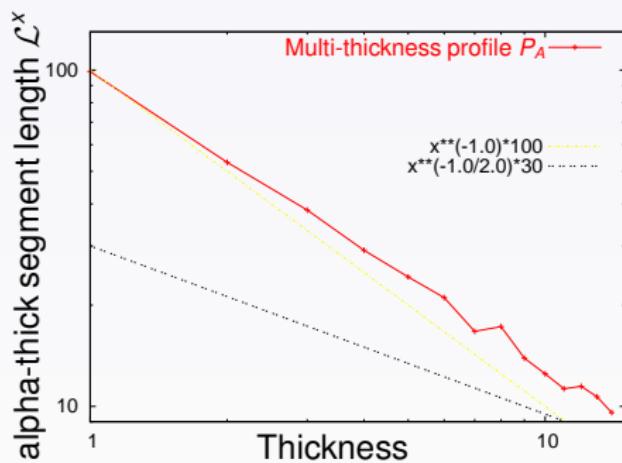
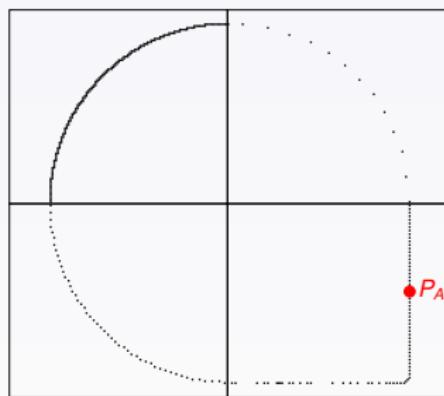


2.2 Meaningful Thickness profile

Multi-thickness Profile

The ***multi-thickness profile*** $\mathcal{P}_n(P)$ of a point P is defined as the graph $(\log(t_i), \log(\overline{\mathcal{L}}^{t_i} / t_i))_{i=1,\dots,n}$.

Example obtained from a shape with different samplings:

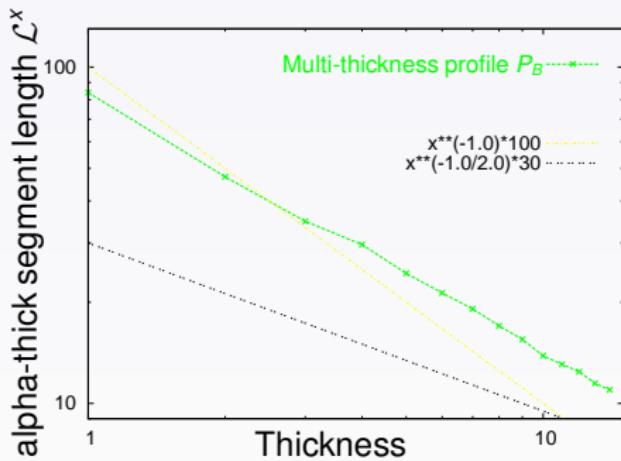
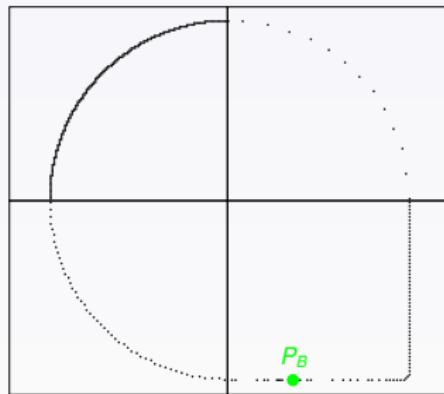


2.2 Meaningful Thickness profile

Multi-thickness Profile

The **multi-thickness profile** $\mathcal{P}_n(P)$ of a point P is defined as the graph $(\log(t_i), \log(\overline{\mathcal{L}}^{t_i} / t_i))_{i=1,\dots,n}$.

Example obtained from a shape with different samplings:

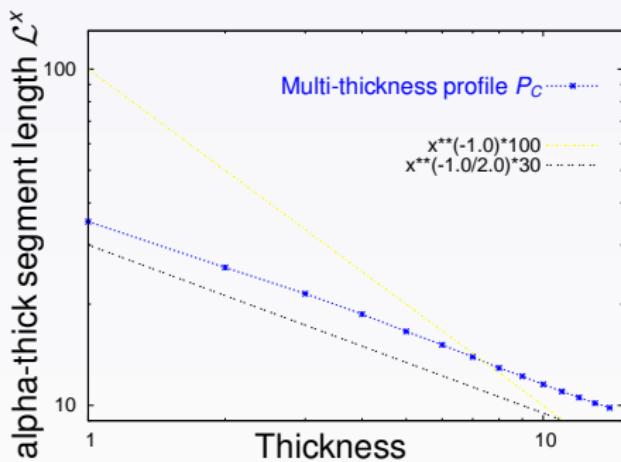
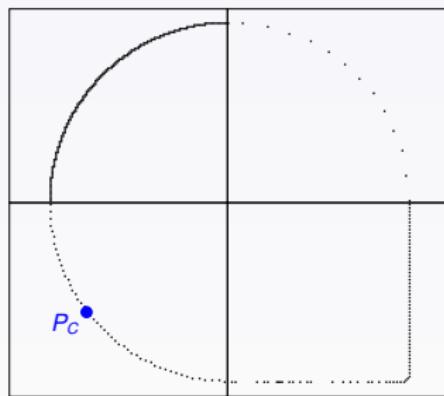


2.2 Meaningful Thickness profile

Multi-thickness Profile

The **multi-thickness profile** $\mathcal{P}_n(P)$ of a point P is defined as the graph $(\log(t_i), \log(\overline{\mathcal{L}}^{t_i} / t_i))_{i=1,\dots,n}$.

Example obtained from a shape with different samplings:

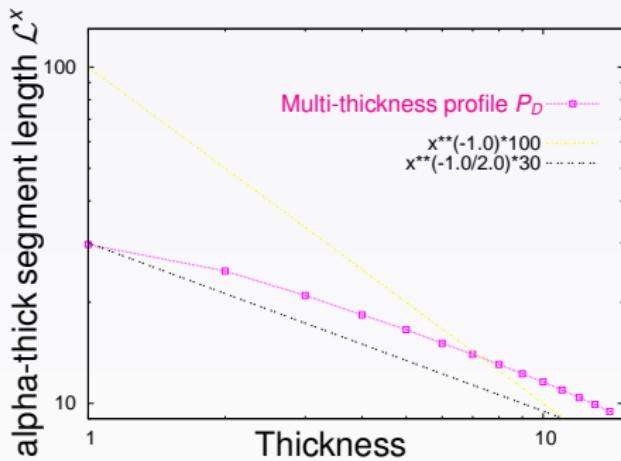
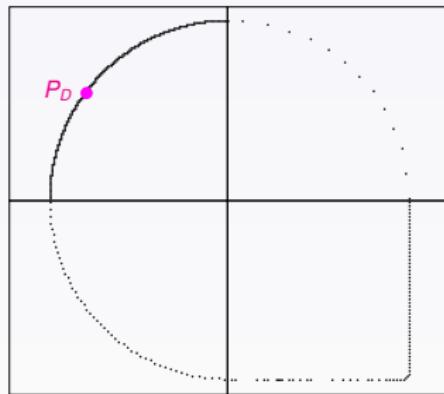


2.2 Meaningful Thickness profile

Multi-thickness Profile

The ***multi-thickness profile*** $\mathcal{P}_n(P)$ of a point P is defined as the graph $(\log(t_i), \log(\overline{\mathcal{L}}^{t_i} / t_i))_{i=1,\dots,n}$.

Example obtained from a shape with different samplings:

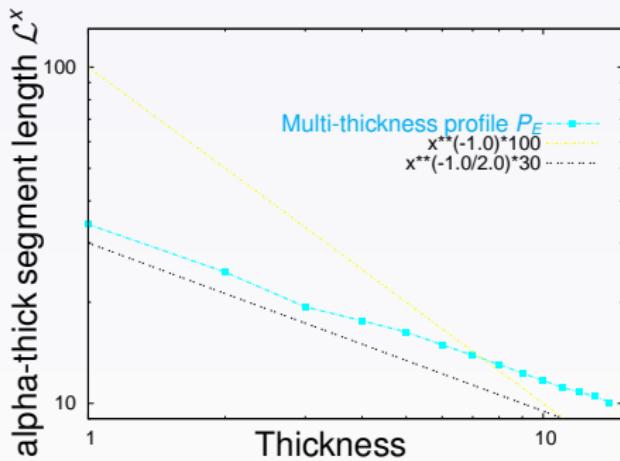
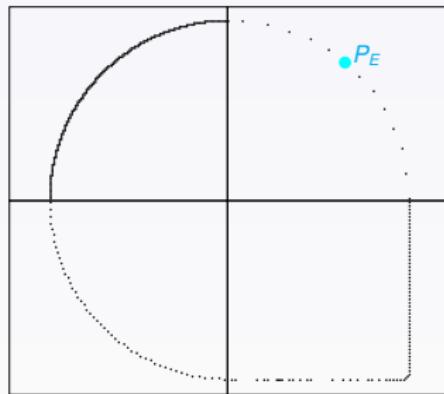


2.2 Meaningful Thickness profile

Multi-thickness Profile

The ***multi-thickness profile*** $\mathcal{P}_n(P)$ of a point P is defined as the graph $(\log(t_i), \log(\overline{\mathcal{L}}^{t_i} / t_i))_{i=1,\dots,n}$.

Example obtained from a shape with different samplings:



2.2 Meaningful Thickness profile

Multi-thickness Profile

The ***multi-thickness profile*** $\mathcal{P}_n(P)$ of a point P is defined as the graph $(\log(t_i), \log(\overline{\mathcal{L}}^{t_i} / t_i))_{i=1,\dots,n}$.

Example obtained from a shape with different samplings:

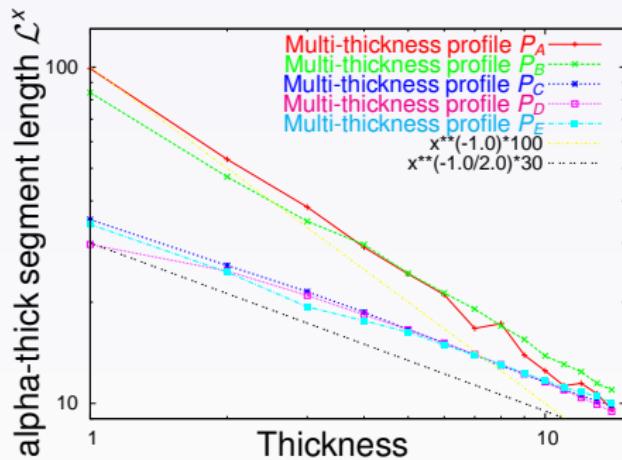
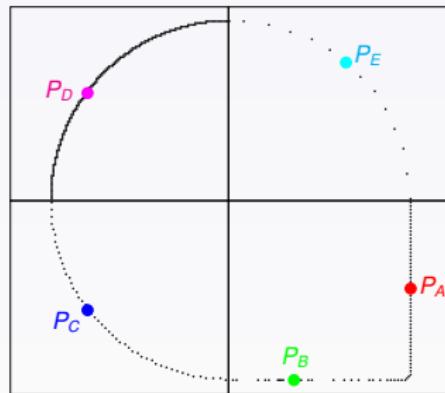


Illustration of multi-thickness profile (2)

Example obtained by adding noise:

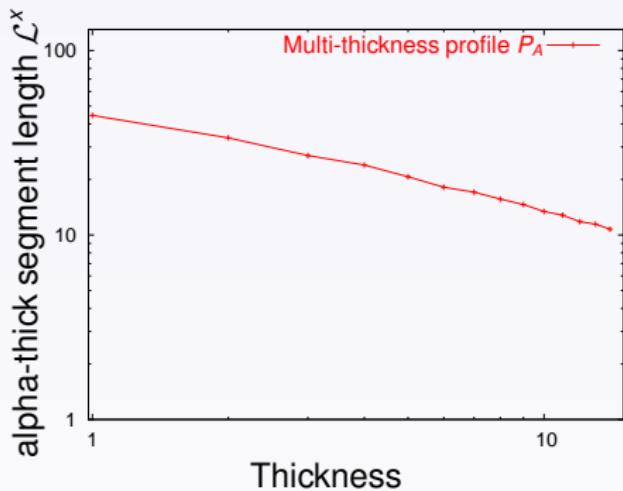
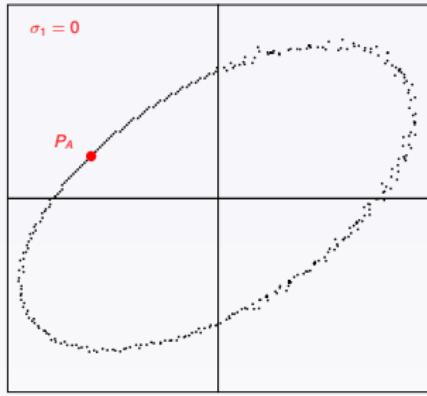


Illustration of multi-thickness profile (2)

Example obtained by adding noise:

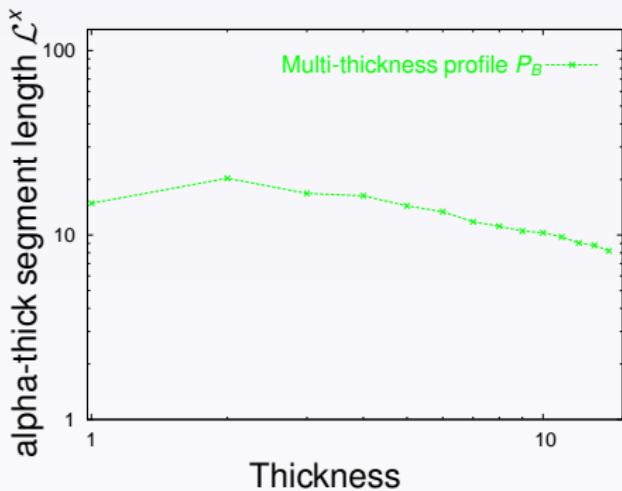
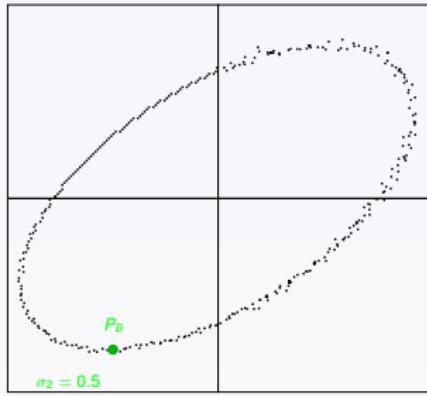


Illustration of multi-thickness profile (2)

Example obtained by adding noise:

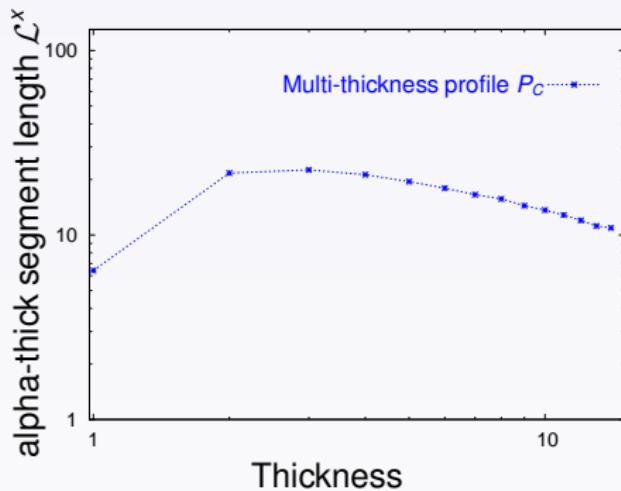
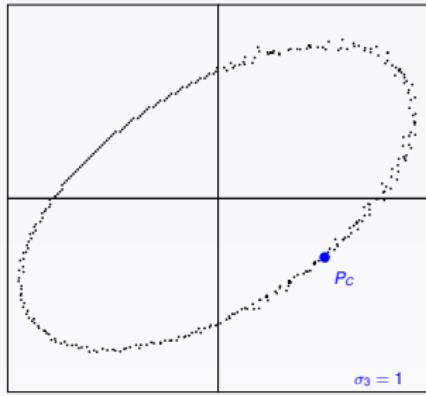


Illustration of multi-thickness profile (2)

Example obtained by adding noise:

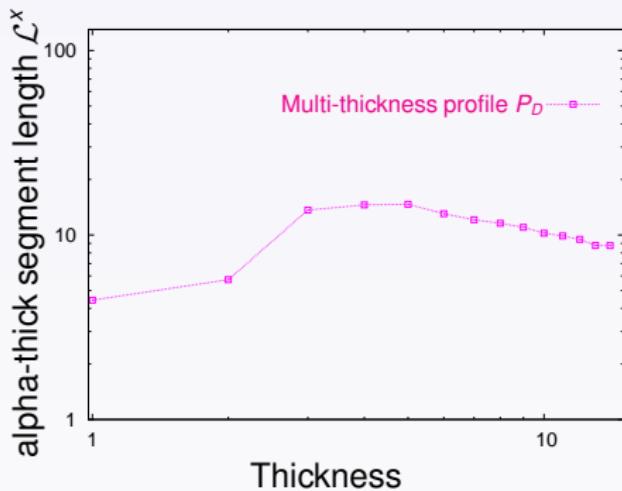
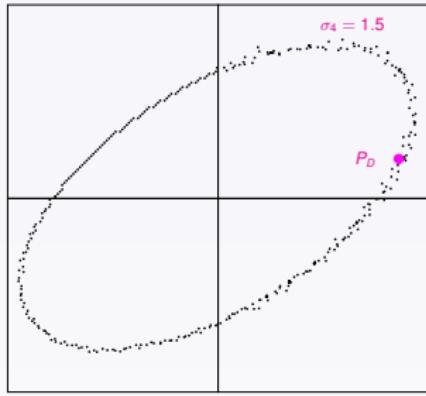


Illustration of multi-thickness profile (2)

Example obtained by adding noise:

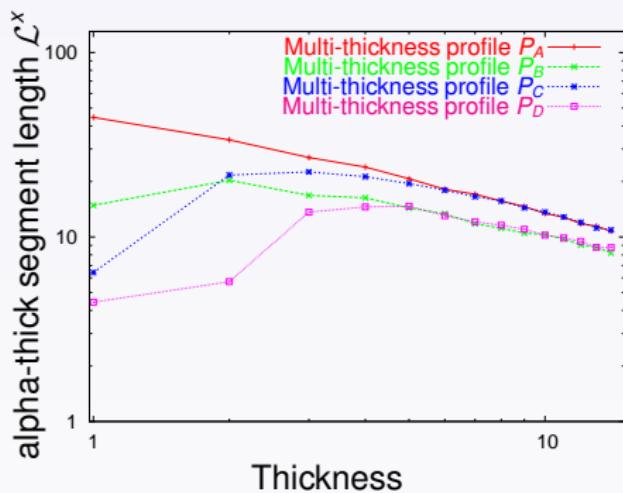
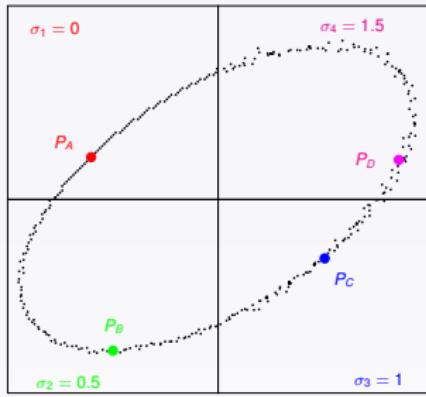
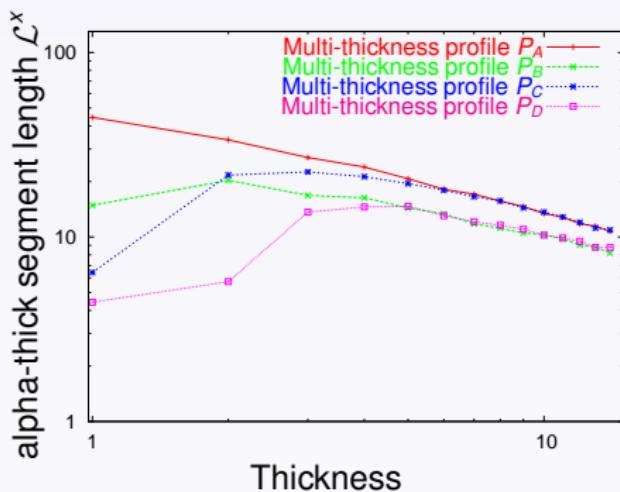
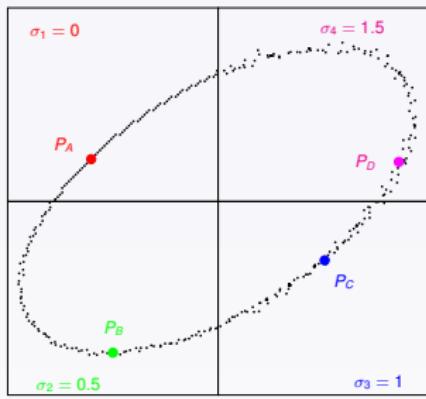


Illustration of multi-thickness profile (2)

Example obtained by adding noise:



⇒ Define a noise threshold T_m to discriminate the curved and noisy zone.

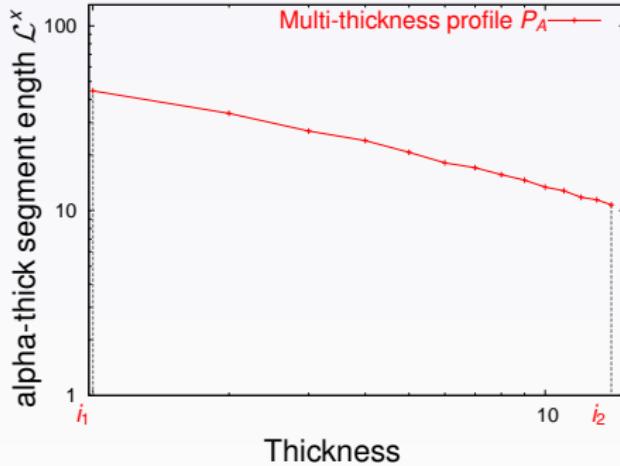
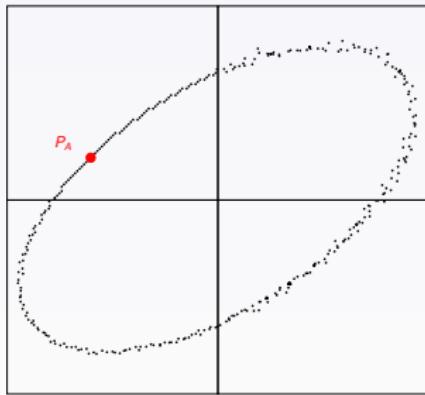
Noise detection and Meaningful Thickness

Meaningful thickness

A **Meaningful thickness** of a multi-thickness profile $(X_i, Y_i)_{1 \leq i \leq n}$ is then a pair (i_1, i_2) , $1 \leq i_1 < i_2 \leq n$, such that for all i , $i_1 \leq i < i_2$,

$$\frac{Y_{i+1} - Y_i}{X_{i+1} - X_i} \leq T_m,$$

and the property is not true for $i_1 - 1$ and i_2 .



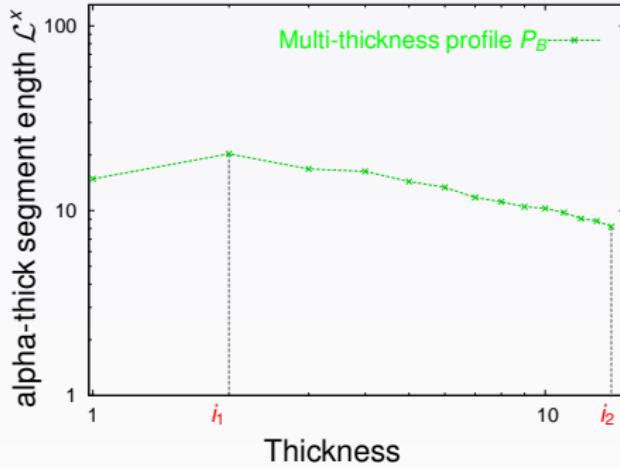
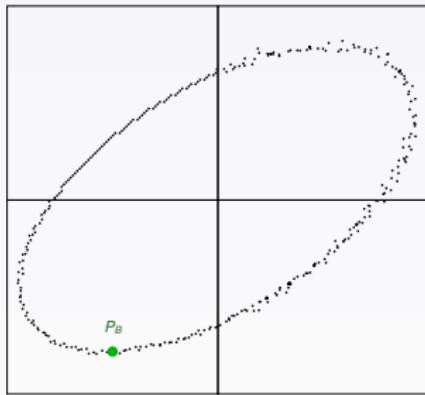
Noise detection and Meaningful Thickness

Meaningful thickness

A **Meaningful thickness** of a multi-thickness profile $(X_i, Y_i)_{1 \leq i \leq n}$ is then a pair (i_1, i_2) , $1 \leq i_1 < i_2 \leq n$, such that for all i , $i_1 \leq i < i_2$,

$$\frac{Y_{i+1} - Y_i}{X_{i+1} - X_i} \leq T_m,$$

and the property is not true for $i_1 - 1$ and i_2 .



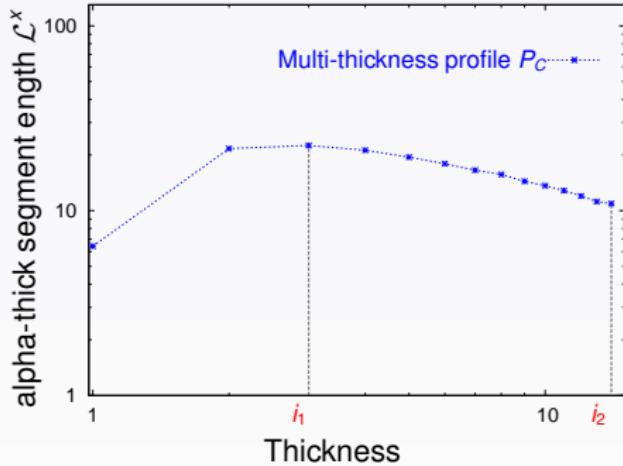
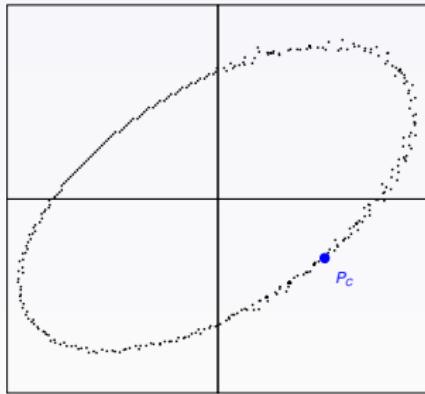
Noise detection and Meaningful Thickness

Meaningful thickness

A **Meaningful thickness** of a multi-thickness profile $(X_i, Y_i)_{1 \leq i \leq n}$ is then a pair (i_1, i_2) , $1 \leq i_1 < i_2 \leq n$, such that for all i , $i_1 \leq i < i_2$,

$$\frac{Y_{i+1} - Y_i}{X_{i+1} - X_i} \leq T_m,$$

and the property is not true for $i_1 - 1$ and i_2 .



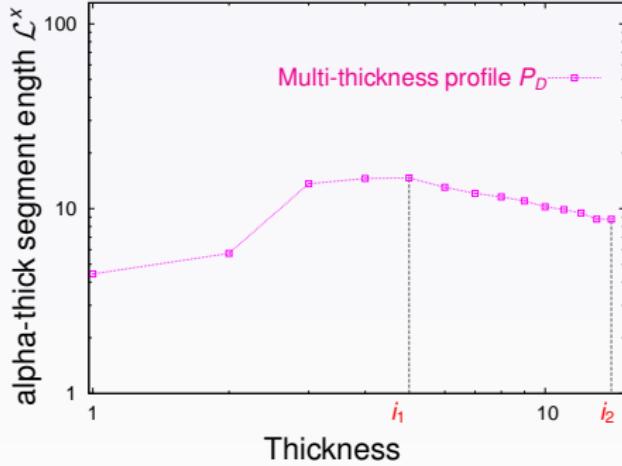
Noise detection and Meaningful Thickness

Meaningful thickness

A **Meaningful thickness** of a multi-thickness profile $(X_i, Y_i)_{1 \leq i \leq n}$ is then a pair (i_1, i_2) , $1 \leq i_1 < i_2 \leq n$, such that for all i , $i_1 \leq i < i_2$,

$$\frac{Y_{i+1} - Y_i}{X_{i+1} - X_i} \leq T_m,$$

and the property is not true for $i_1 - 1$ and i_2 .



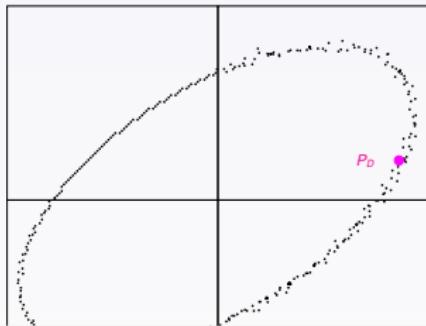
Noise detection and Meaningful Thickness

Meaningful thickness

A **Meaningful thickness** of a multi-thickness profile $(X_i, Y_i)_{1 \leq i \leq n}$ is then a pair (i_1, i_2) , $1 \leq i_1 < i_2 \leq n$, such that for all i , $i_1 \leq i < i_2$,

$$\frac{Y_{i+1} - Y_i}{X_{i+1} - X_i} \leq T_m,$$

and the property is not true for $i_1 - 1$ and i_2 .

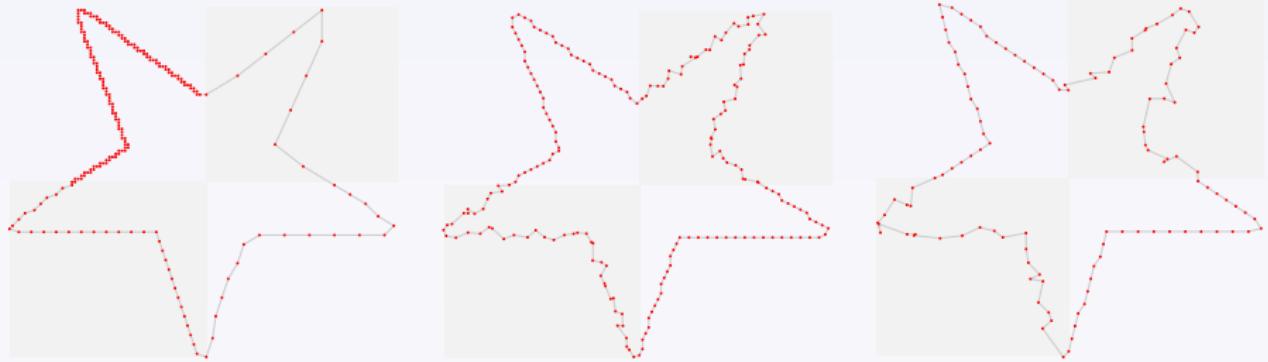


Noise level

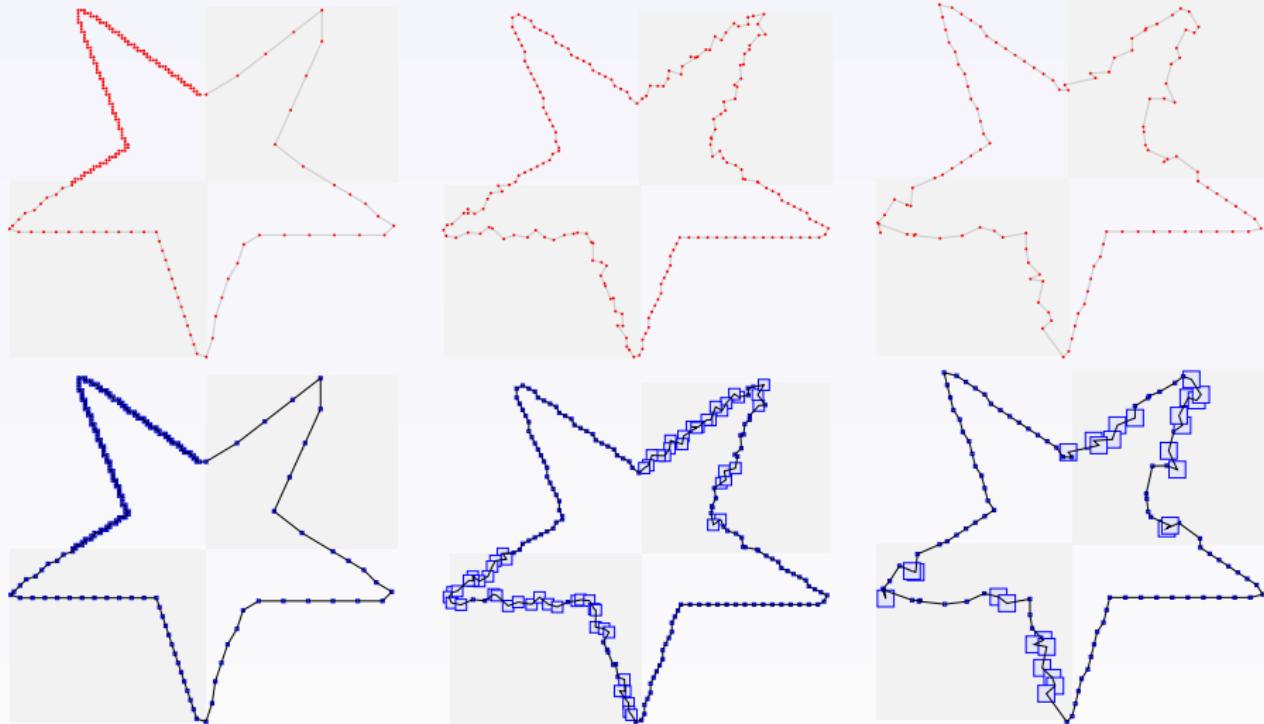
if (i_1, i_2) is the first meaningful scale at point P the noise level is $i_1 - 1$.

Thickness

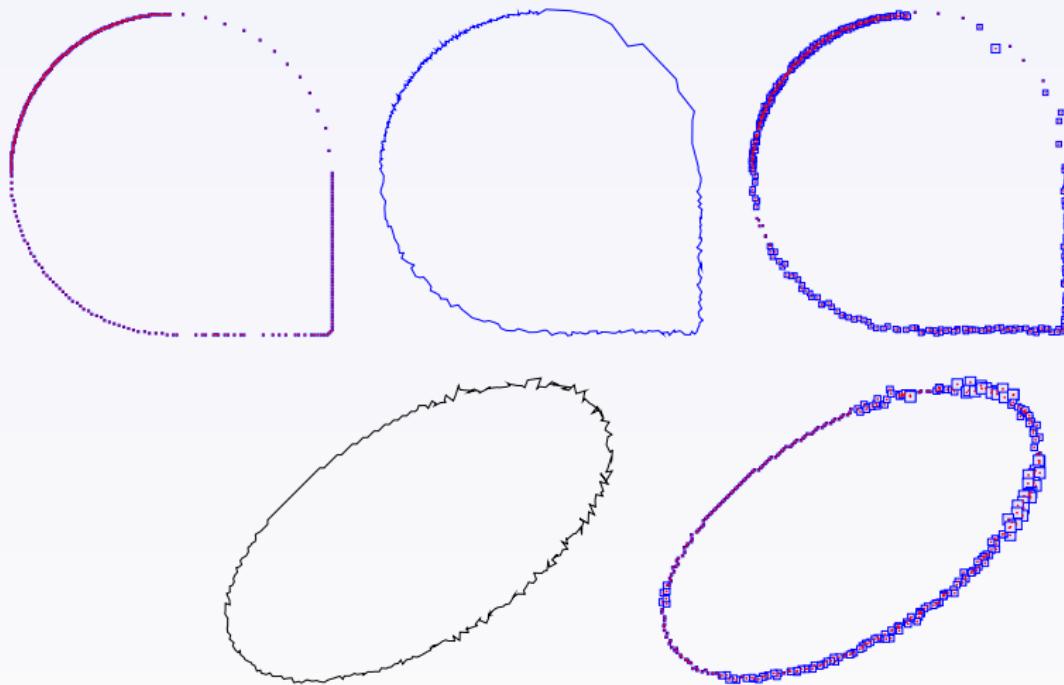
Experiments on polygonal shapes (1)



Experiments on polygonal shapes (1)

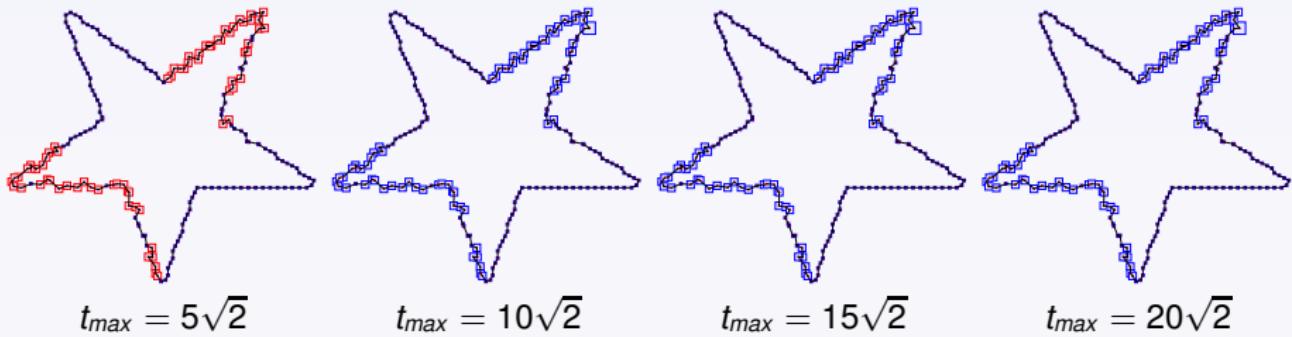


Experiments on polygonal shapes (2)



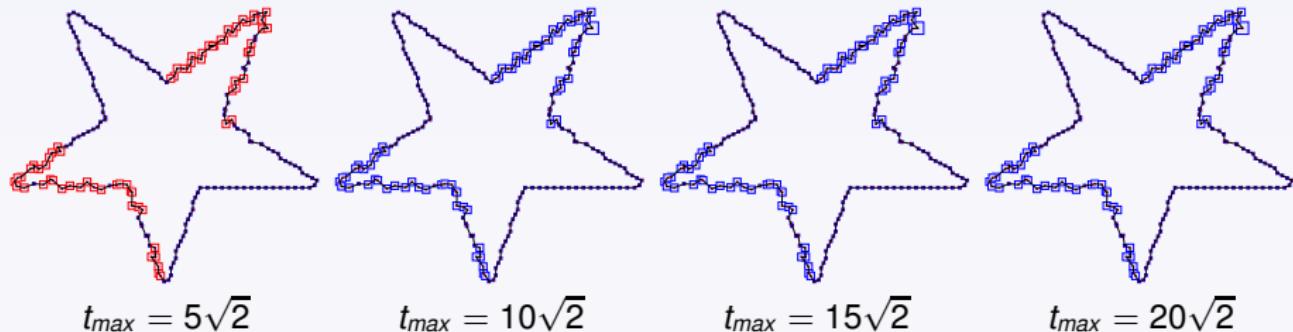
Stability from intern parameters

- Maximal thickness t_{max}

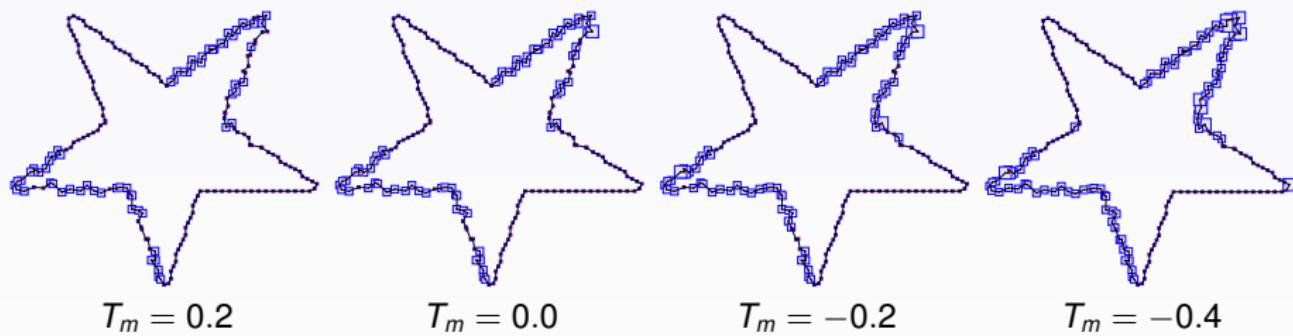


Stability from intern parameters

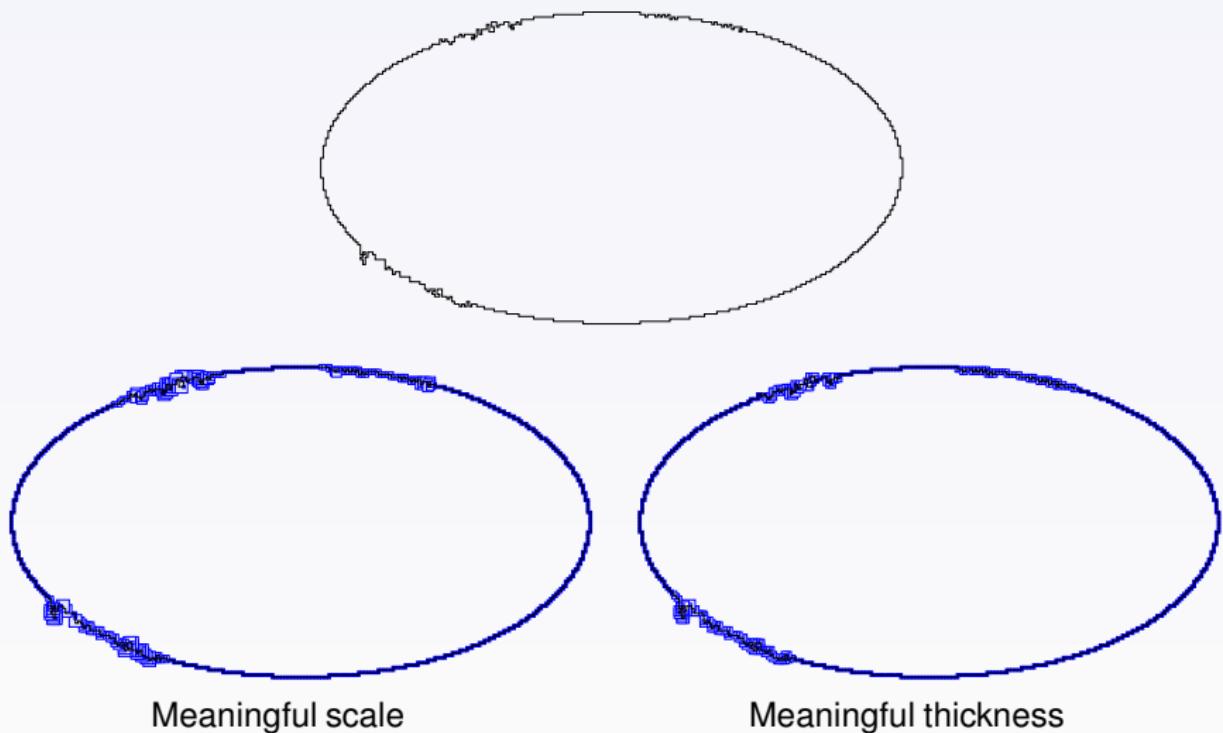
- Maximal thickness t_{max}



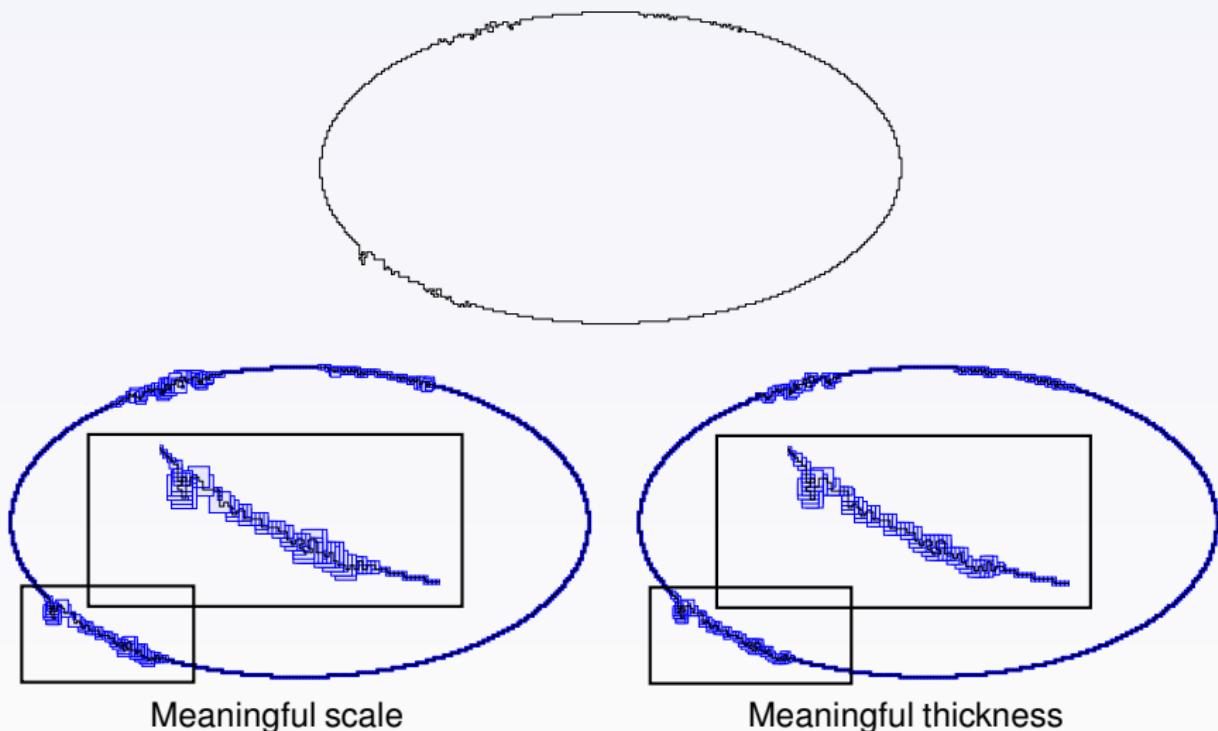
- Noise threshold T_m



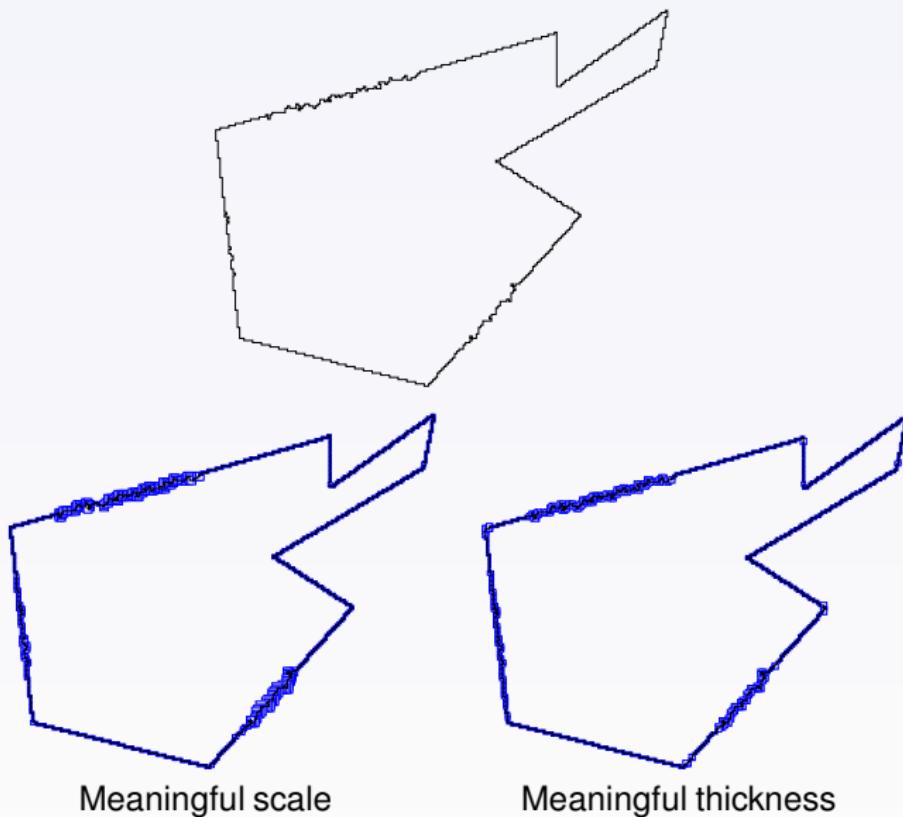
Comparison with the Meaningful Scales [Kerautret & Lachaud, 2012]



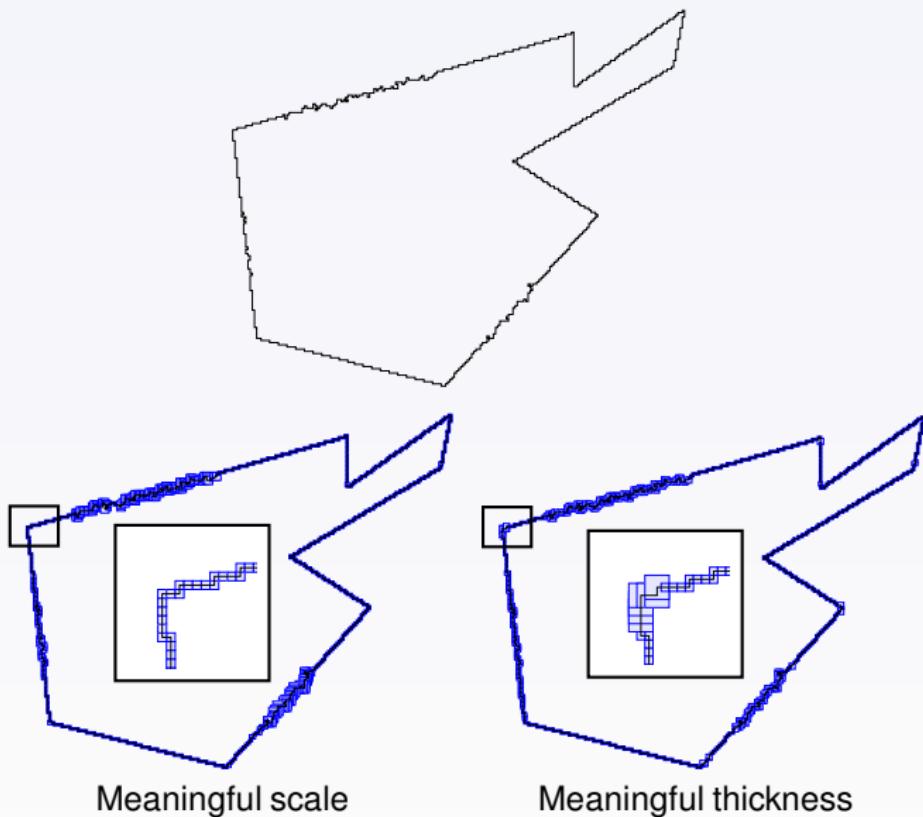
Comparison with the Meaningful Scales [Kerautret & Lachaud, 2012]



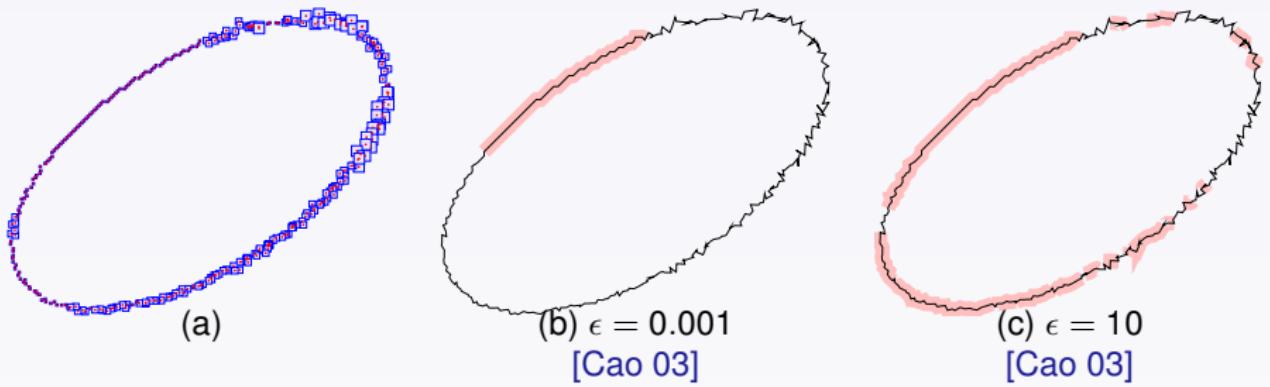
Comparison with the Meaningful Scales (2) [Kerautret & Lachaud, 2012]



Comparison with the Meaningful Scales (2) [Kerautret & Lachaud, 2012]



Comparison with the *Good Continuation* approach [Cao 03]



2.3 Simple applications (1)

Meaningful contour extraction from image level sets:

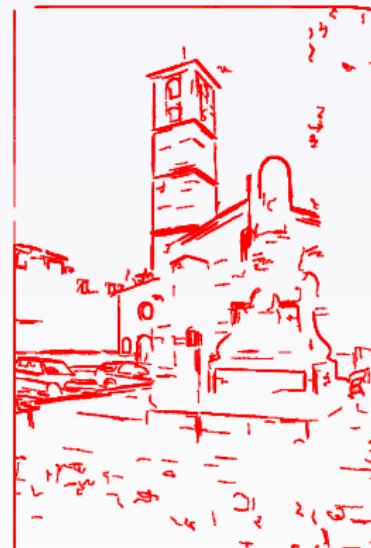
- Extraction of all contours.
- Apply meaningful thickness detection.
- Detection of straight parts.



source



contour from level set



meaningful parts

2.3 Simple applications (1)

Meaningful contour extraction from image level sets:

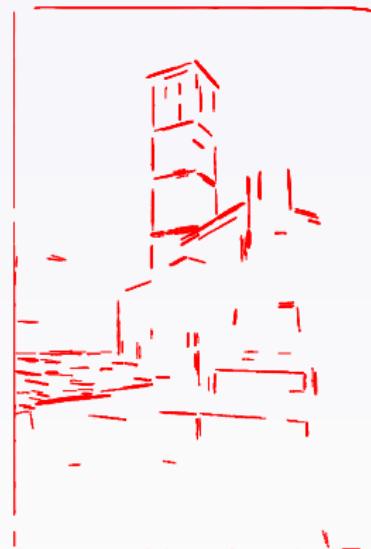
- Extraction of all contours.
- Apply meaningful thickness detection.
- Detection of straight parts.



source



contour from level set

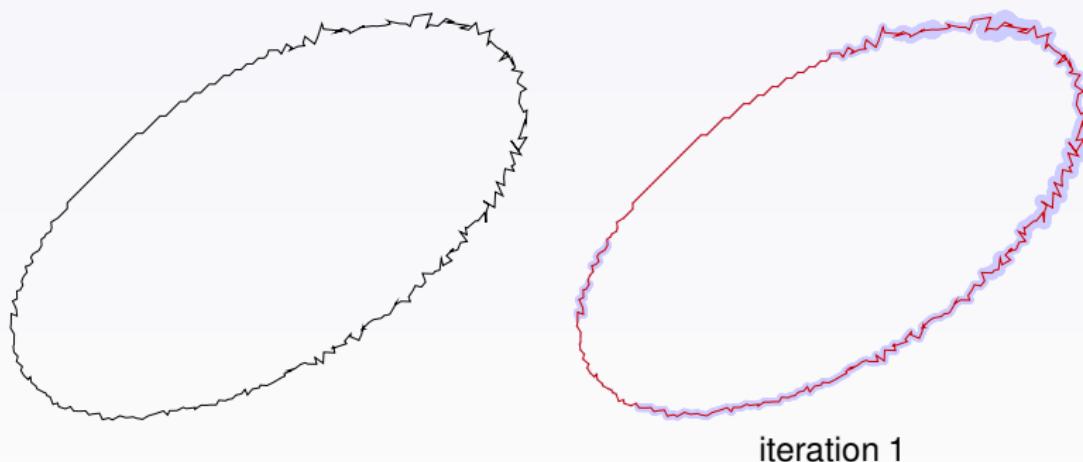


meaningful straight parts

2.3 Simple applications (2)

Polygon denoising

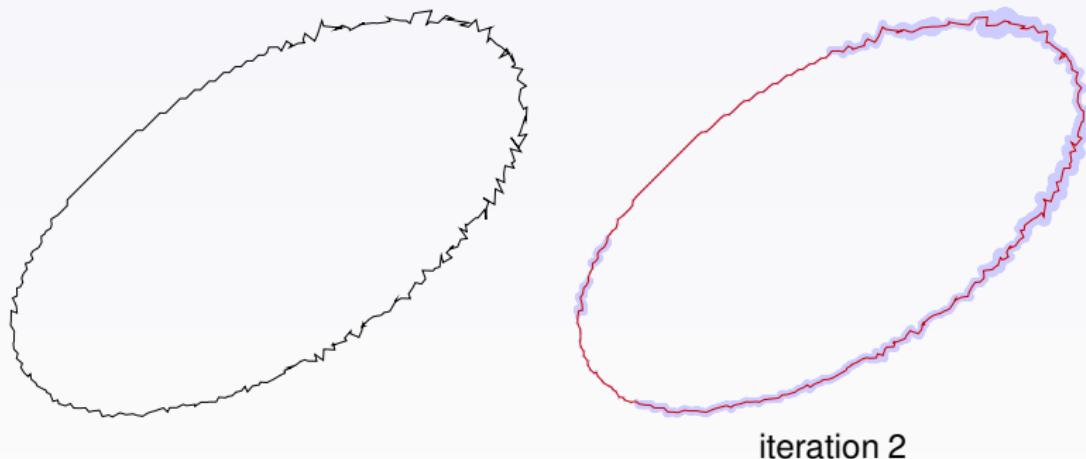
- Applying an iterative process on contour points P_i .
- Each points are moved:
 - by a weighted average of its two neighbors.
 - from constraint defined from meaningful thickness.
- Constraints are also defined between polygon vertex by linear interpolation.



2.3 Simple applications (2)

Polygon denoising

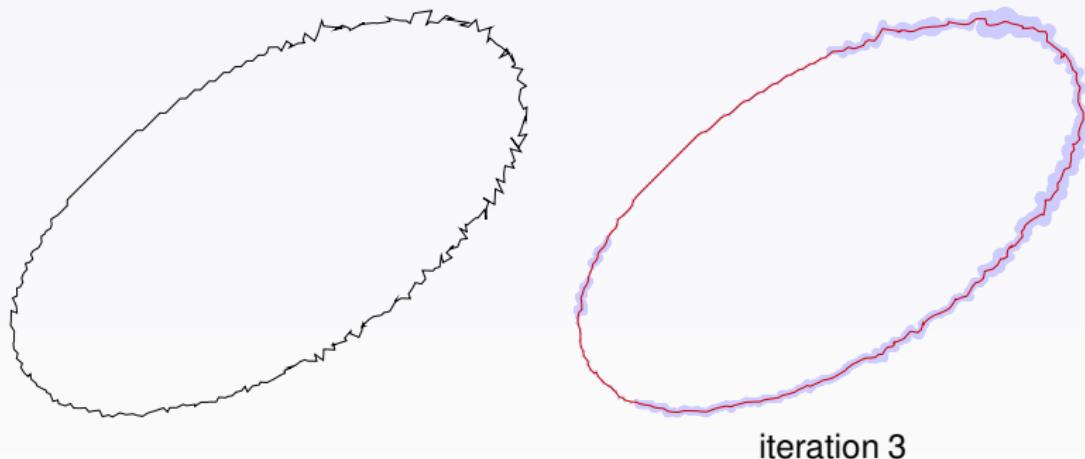
- Applying an iterative process on contour points P_i .
- Each points are moved:
 - by a weighted average of its two neighbors.
 - from constraint defined from meaningful thickness.
- Constraints are also defined between polygon vertex by linear interpolation.



2.3 Simple applications (2)

Polygon denoising

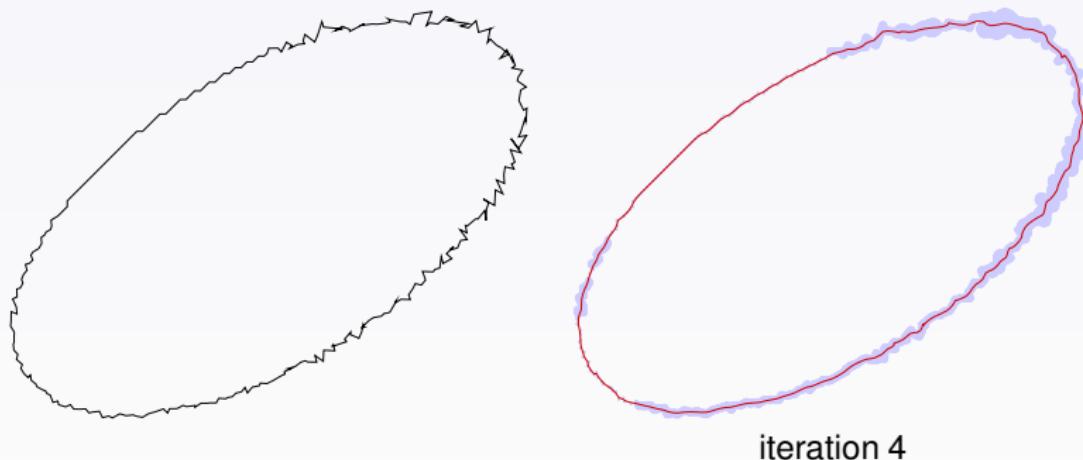
- Applying an iterative process on contour points P_i .
- Each points are moved:
 - by a weighted average of its two neighbors.
 - from constraint defined from meaningful thickness.
- Constraints are also defined between polygon vertex by linear interpolation.



2.3 Simple applications (2)

Polygon denoising

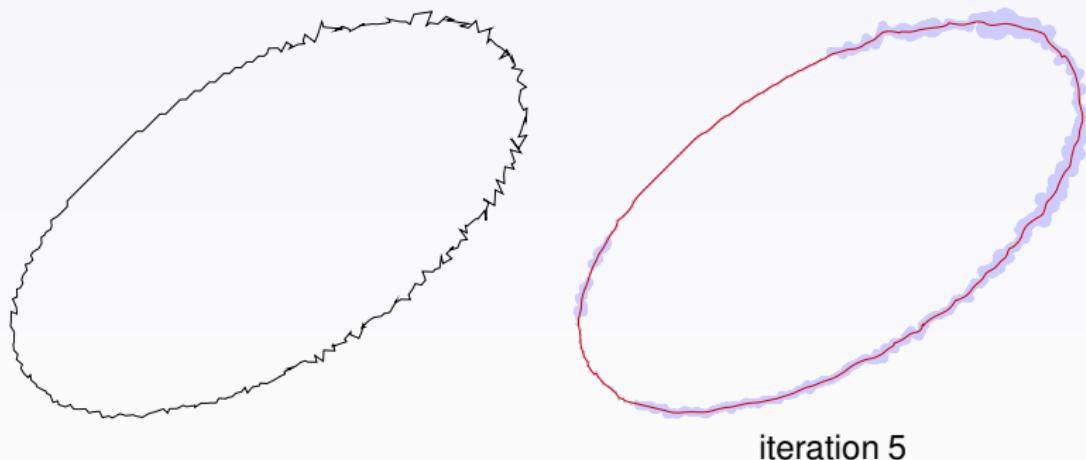
- Applying an iterative process on contour points P_i .
- Each points are moved:
 - by a weighted average of its two neighbors.
 - from constraint defined from meaningful thickness.
- Constraints are also defined between polygon vertex by linear interpolation.



2.3 Simple applications (2)

Polygon denoising

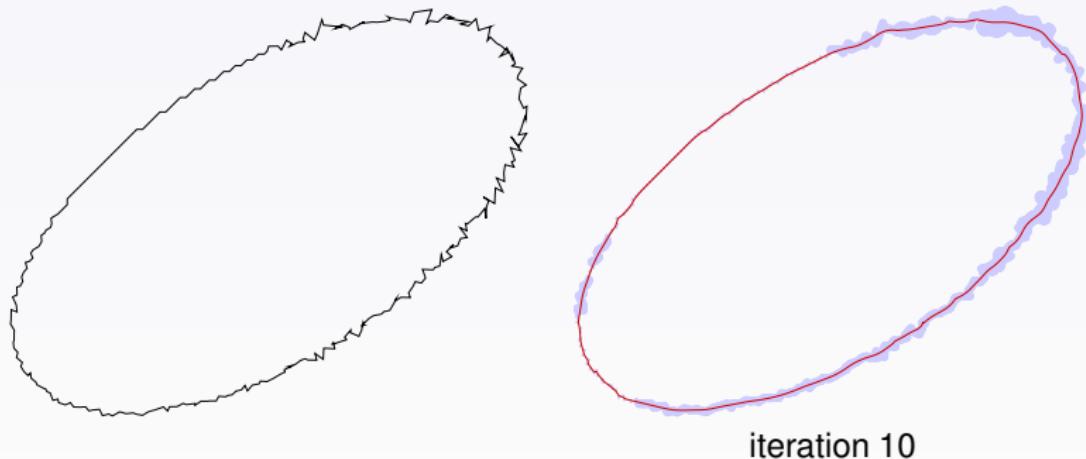
- Applying an iterative process on contour points P_i .
- Each points are moved:
 - by a weighted average of its two neighbors.
 - from constraint defined from meaningful thickness.
- Constraints are also defined between polygon vertex by linear interpolation.



2.3 Simple applications (2)

Polygon denoising

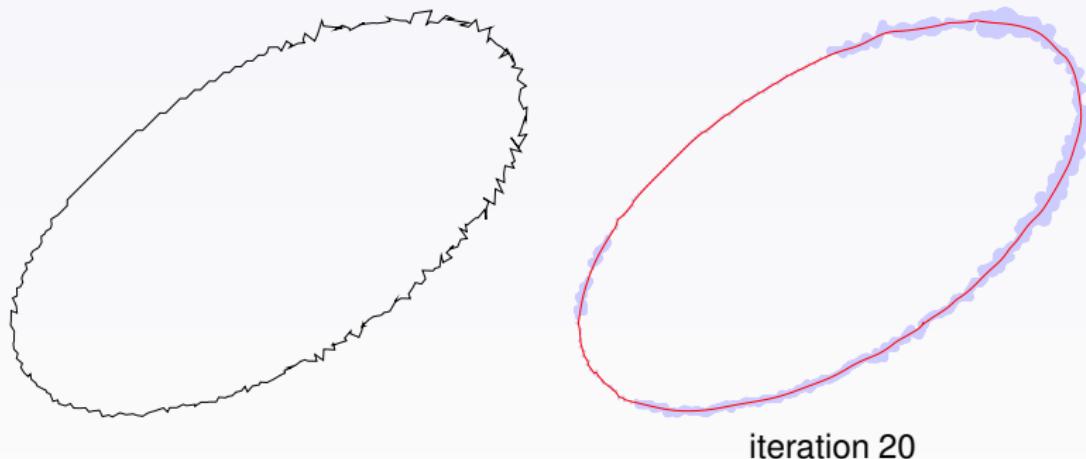
- Applying an iterative process on contour points P_i .
- Each points are moved:
 - by a weighted average of its two neighbors.
 - from constraint defined from meaningful thickness.
- Constraints are also defined between polygon vertex by linear interpolation.



2.3 Simple applications (2)

Polygon denoising

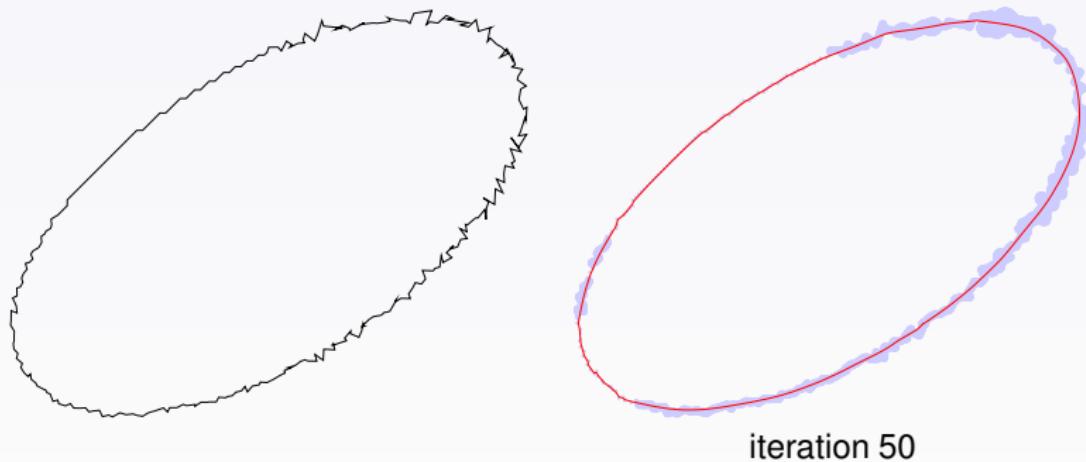
- Applying an iterative process on contour points P_i .
- Each points are moved:
 - by a weighted average of its two neighbors.
 - from constraint defined from meaningful thickness.
- Constraints are also defined between polygon vertex by linear interpolation.



2.3 Simple applications (2)

Polygon denoising

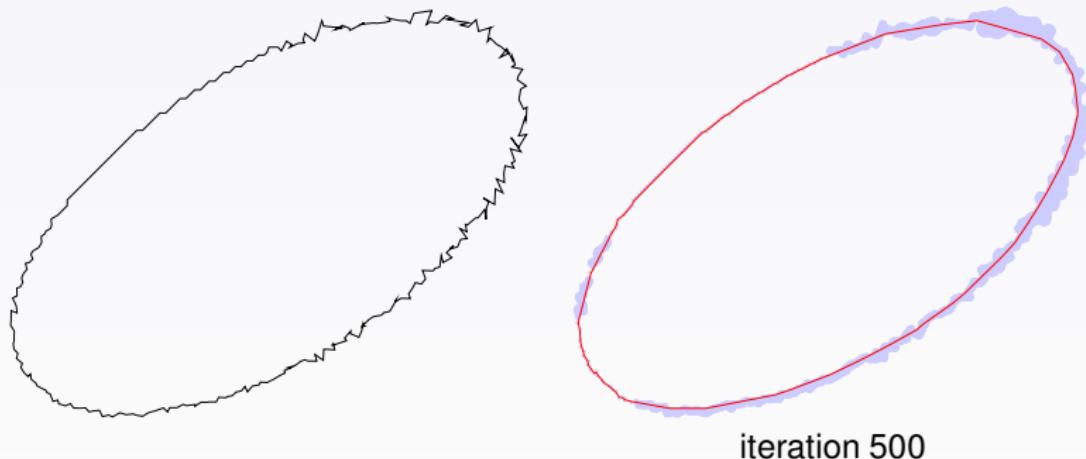
- Applying an iterative process on contour points P_i .
- Each points are moved:
 - by a weighted average of its two neighbors.
 - from constraint defined from meaningful thickness.
- Constraints are also defined between polygon vertex by linear interpolation.



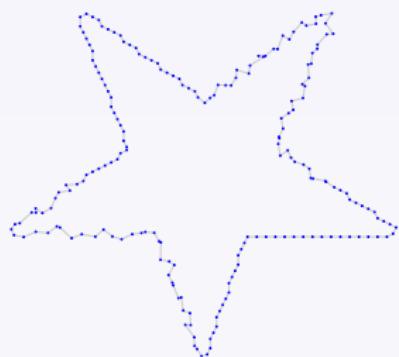
2.3 Simple applications (2)

Polygon denoising

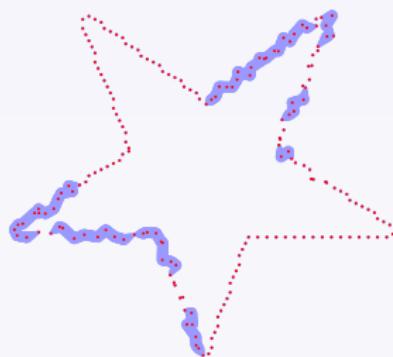
- Applying an iterative process on contour points P_i .
- Each points are moved:
 - by a weighted average of its two neighbors.
 - from constraint defined from meaningful thickness.
- Constraints are also defined between polygon vertex by linear interpolation.



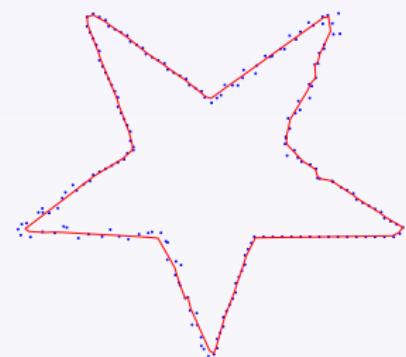
2.3 Simple applications (2)



(d) source contour



(e) meaningful thickness constraint



(f) resulting reconstruction



(g) source contour



(h) meaningful thickness constraint



(i) resulting reconstruction

First step: Alpha-Thick Tangential Cover computation (1)

Alpha Thick Segment

- Since DGtal v. 0.9.1 the [AlphaThickSegmentComputer](#) is implemented.
- You can easily exploit it to define geometric estimators.

First step: Alpha-Thick Tangential Cover computation (1)

Alpha Thick Segment

- Since DGtal v. 0.9.1 the [AlphaThickSegmentComputer](#) is implemented.
- You can easily exploit it to define geometric estimators.
- Example of basic use:

(see file: `tuto4_compATS.cpp`)

```
1 // types definition
2 typedef Z2i::Point Point
3 typedef AlphaThickSegmentComputer<Point> AlphaThickSegmentComputer2D;
4 // Construction of the AlphaThickSegmentComputer from a maximal thickness = 5
5 AlphaThickSegmentComputer2D aComputer(5);
6
7 // Initialisation from the point of index 30:
8 aComputer.init(contour.begin() + 30);
9
10 while (aComputer.extendFront()) {
11 }
```

First step: Alpha-Thick Tangential Cover computation (1)

Alpha Thick Segment

- Since DGtal v. 0.9.1 the [AlphaThickSegmentComputer](#) is implemented.
- You can easily exploit it to define geometric estimators.
- Example of basic use:

(see file: `tuto4_compATS.cpp`)

```
1 // types definition
2 typedef Z2i::Point Point
3 typedef AlphaThickSegmentComputer<Point> AlphaThickSegmentComputer2D;
4 // Construction of the AlphaThickSegmentComputer from a maximal thickness = 5
5 AlphaThickSegmentComputer2D aComputer(5);
6
7 // Initialisation from the point of index 30:
8 aComputer.init(contour.begin() + 30);
9
10 while (aComputer.extendFront()) {
11 }
```

with [Board2D](#) display:

```
1 aBoard << CustomStyle( aComputer.className(), new CustomColors( DGtal::Color::Blue,
2                                         DGtal::Color::None));
3 aBoard << aComputer;
4 aBoard.saveEPS("resultTuto2.eps");
```

First step: Alpha-Thick Tangential Cover computation (1)

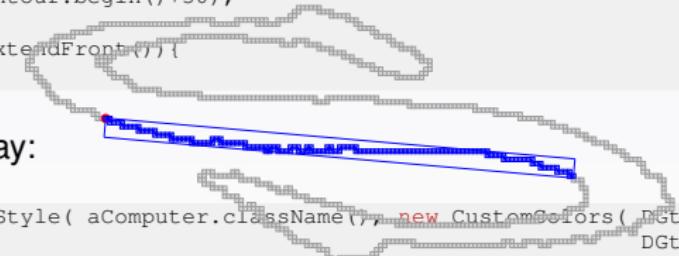
Alpha Thick Segment

- Since DGtal v. 0.9.1 the `AlphaThickSegmentComputer` is implemented.
- You can easily exploit it to define geometric estimators.
- Exemple of basic use:

(see file: `tuto4_compATS.cpp`)

```

1 // types definition
2 typedef Z2i::Point Point
3 typedef AlphaThickSegmentComputer<Point> AlphaThickSegmentComputer2D;
4 // Construction of the AlphaThickSegmentComputer from a maximal thickness = 5
5 AlphaThickSegmentComputer2D aComputer(5);
6
7 // Initialisation from the point of index 30:
8 aComputer.init(contour.begin() + 30);
9
10 while(aComputer.extendFront()) {
11 }
```



with `Board2D` display:

```

1 aBoard << CustomStyle( aComputer.className(), new CustomColors( DGtal::Color::Blue,
2 DGtal::Color::None));
3 aBoard << aComputer;
4 aBoard.saveEPS("resultTuto2.eps");
```

First step: Alpha-Thick Tangential Cover computation (2)

Tangential Cover

- Commonly used to define geometric estimator: length, tangent, curvature.
⇒ ex: λ – MST tangent estimator (see [\[Lachaud 2010\]](#) for a survey)

First step: Alpha-Thick Tangential Cover computation (2)

Tangential Cover

- Commonly used to define geometric estimator: length, tangent, curvature.
⇒ ex: λ – MST tangent estimator (see [\[Lachaud 2010\]](#) for a survey)
- Thanks to the alpha-thick segments, they can now handle noise.

First step: Alpha-Thick Tangential Cover computation (2)

Tangential Cover

- Commonly used to define geometric estimator: length, tangent, curvature.
⇒ ex: λ – MST tangent estimator (see [\[Lachaud 2010\]](#) for a survey)
- Thanks to the alpha-thick segments, they can now handle noise.
- Example of use: compute all maximal [AlphaThickSegmentComputer](#) covering a point.

First step: Alpha-Thick Tangential Cover computation (2)

Tangential Cover

- Commonly used to define geometric estimator: length, tangent, curvature.
⇒ ex: λ – MST tangent estimator (see [\[Lachaud 2010\]](#) for a survey)
- Thanks to the alpha-thick segments, they can now handle noise.
- Example of use: compute all maximal `AlphaThickSegmentComputer` covering a point.

(see file: `tuto5_compATSTC.cpp`)

```
1 // to get the functions firstMaximalSegment(), lastMaximalSegment() and
2 // nextMaximalSegment()
3 #include "DGtal/geometry/curves/SegmentComputerUtils.h"
4
5 // Get the first and last AlphaThickSegment
6 AlphaThickSegmentComputer2D aComputer(5);
7 firstMaximalSegment(aComputer, contour.begin()+30, contour.begin(), contour.end());
8 AlphaThickSegmentComputer2D first (aComputer);
9 lastMaximalSegment(aComputer, contour.begin()+30, contour.begin(), contour.end());
10 AlphaThickSegmentComputer2D last (aComputer);
11
12 // And iterate to display all the set of maximal segments
13 while(first.end() != last.end()){
14     aBoard << first;
15     nextMaximalSegment(first, contour.end());
16 }
17 aBoard << first;
```

First step: Alpha-Thick Tangential Cover computation (2)

Tangential Cover

- Commonly used to define geometric estimator: length, tangent, curvature.
⇒ ex: λ – MST tangent estimator (see [Lachaud 2010] for a survey)
- Thanks to the alpha-thick segments, they can now handle noise.
- Example of use: compute all maximal `AlphaThickSegmentComputer` covering a point.

(see file: `tuto5_compATSTC.cpp`)

```

1 // to get the functions firstMaximalSegment(), lastMaximalSegment() and
2 // nextMaximalSegment()
3 #include "DGtal/geometry/curves/SegmentComputerUtils.h"
4
5 // Get the first and last AlphaThickSegment
6 AlphaThickSegmentComputer2D aComputer(5);
7 firstMaximalSegment(aComputer, contour.begin()+30, contour.begin(), contour.end());
8 AlphaThickSegmentComputer2D first(aComputer);
9 lastMaximalSegment(aComputer, contour.begin()+30, contour.begin(), contour.end());
10 AlphaThickSegmentComputer2D last(aComputer);
11
12 // And iterate to display the sequence of maximal segments
13 while(first.end() != last.end()){
14     aBoard << first;
15     nextMaximalSegment(first, contour.end());
16 }
17 aBoard << first;

```

First step: Alpha-Thick Tangential Cover computation (3)

Tangential cover from saturated segmentation

- The class `AlphaThickSegmentComputer` is a model of the concept `CForwardComputer`.

First step: Alpha-Thick Tangential Cover computation (3)

Tangential cover from saturated segmentation

- The class `AlphaThickSegmentComputer` is a model of the concept `CForwardComputer`.
- So the `SaturatedSegmentation` can be applied on this computer.

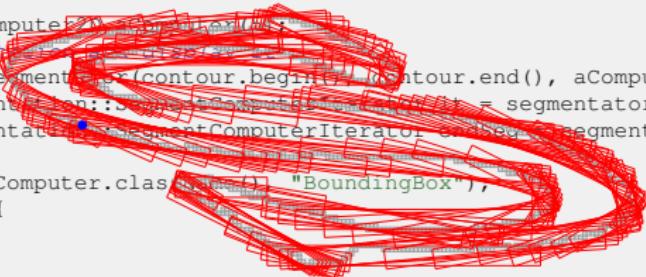
```
1 #include "DGtal/geometry/curves/SaturatedSegmentation.h"
2 ...
3 typedef AlphaThickSegmentComputer<Z2i::Point> AlphaThickSegmentComputer2D;
4 typedef SaturatedSegmentation<AlphaThickSegmentComputer2D> AlphaSegmentation;
5 ...
6 AlphaThickSegmentComputer2D aComputer(5);
7 // Apply the segmentation and display it
8 AlphaSegmentation segmentator(contour.begin(), contour.end(), aComputer);
9 typename AlphaSegmentation::SegmentComputerIterator it = segmentator.begin();
10 typename AlphaSegmentation::SegmentComputerIterator endSeg = segmentator.end();
11
12 aBoard << SetMode(aComputer.className(), "BoundingBox");
13 while(it != endSeg) {
14     aBoard << *it;
15     ++it;
16 }
```

First step: Alpha-Thick Tangential Cover computation (3)

Tangential cover from saturated segmentation

- The class `AlphaThickSegmentComputer` is a model of the concept `CForwardComputer`.
- So the `SaturatedSegmentation` can be applied on this computer.
- Thanks to this generic programming we just need to change the primitive to obtain the segmentation.

```
1 #include "DGtal/geometry/curves/SaturatedSegmentation.h"
2 ...
3 typedef AlphaThickSegmentComputer<Z2i::Point> AlphaThickSegmentComputer2D;
4 typedef SaturatedSegmentation<AlphaThickSegmentComputer2D> AlphaSegmentation;
5 ...
6 AlphaThickSegmentComputer2D aComputer(...);
7 // Apply the segmentation
8 AlphaSegmentation segmentation(contour.begin(), contour.end(), aComputer);
9 typename AlphaSegmentation::iterator segmator = segmentation.begin();
10 typename AlphaSegmentation::iterator endSeg = segmentation.end();
11 ...
12 aBoard << SetMode(aComputer.className(), "BoundingBox");
13 while(it != endSeg) {
14     aBoard << *it;
15     ++it;
16 }
```



Second step: Extracting the Meaningful Thickness (1)

Main steps of the Meaningful Thickness algorithm

- Tangential cover computation: for each thickness 1..N.

(see file: `tuto6_compMT1pt.cpp`)

```
1 // To represent the multiscale profile
2 #include "DGtal/math/Profile.h"
3 ...
4 Profile<> sp;
5 sp.init(5);
6 ...
7 // compute all the segments covering a point (for each thickness):
8 for(unsigned int thickness = 1; thickness<=5; thickness++) {
9     AlphaThickSegmentComputer2D aComputer(thickness);
10    firstMaximalSegment(aComputer, contour.begin()+index, contour.begin(), contour.end());
11    AlphaThickSegmentComputer2D first (aComputer);
12    lastMaximalSegment(aComputer, contour.begin()+index, contour.begin(), contour.end());
13    AlphaThickSegmentComputer2D last (aComputer);
14    ...
15 }
```

Second step: Extracting the Meaningful Thickness (1)

Main steps of the Meaningful Thickness algorithm

- Tangential cover computation: for each thickness $1..N$.
- \forall alpha-thick segment S_i : update a multiscale profile on all points covered by S_i .

(see file: `tuto6-compMT1pt.cpp`)

```

1 // To represent the multiscale profile
2 #include "DGtal/math/Profile.h"
3 ...
4 Profile<> sp;
5 sp.init(5);
6 ...
7 // compute all the segments covering a point (for each thickness):
8 for(unsigned int thickness = 1; thickness<=5; thickness++) {
9     AlphaThickSegmentComputer2D aComputer(thickness);
10    firstMaximalSegment(aComputer, contour.begin()+index, contour.begin(), contour.end());
11    AlphaThickSegmentComputer2D first (aComputer);
12    lastMaximalSegment(aComputer, contour.begin()+index, contour.begin(), contour.end());
13    AlphaThickSegmentComputer2D last (aComputer);
14    ...
15 }
```

updating the profile: (at line 14 of previous code)

```

1 while(first.end() != last.end()){
2     sp.addValue(thickness - 1, first.getSegmentLength()/thickness);
3     nextMaximalSegment(first, contour.end());
4 }
```

Second step: Extracting the Meaningful Thickness

Main steps of the Meaningful Thickness algorithm

- Tangential cover computation: **for each** thickness 1..N.
- \forall alpha-thick segment S_i : update a **multiscale profile** on all points covered by S_i .
- Find a first meaningful scale.

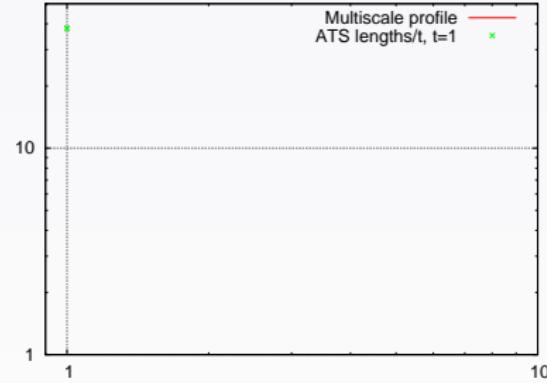
```
1 std::vector<double> x,y;
2 s.getProfile(x, y);
3 trace.info() << "Noise_level:" << sp.noiseLevel() << std::endl;
```

Second step: Extracting the Meaningful Thickness

Main steps of the Meaningful Thickness algorithm

- Tangential cover computation: **for each** thickness $1..N$.
- \forall alpha-thick segment S_i : update a **multiscale profile** on all points covered by S_i .
- Find a first first meaningful scale.

```
1 std::vector<double> x, y;  
2 s.getProfile(x, y);  
3 trace.info() << "Noise_level: " << sp.noiseLevel() << std::endl;
```

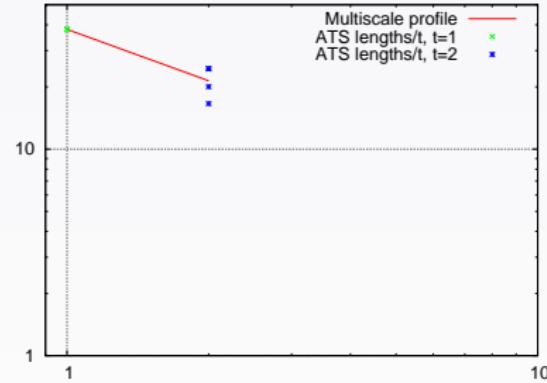


Second step: Extracting the Meaningful Thickness

Main steps of the Meaningful Thickness algorithm

- Tangential cover computation: **for each** thickness $1..N$.
- \forall alpha-thick segment S_i : update a **multiscale profile** on all points covered by S_i .
- Find a first first meaningful scale.

```
1 std::vector<double> x, y;
2 s.getProfile(x, y);
3 trace.info() << "Noise_level: " << sp.noiseLevel() << std::endl;
```

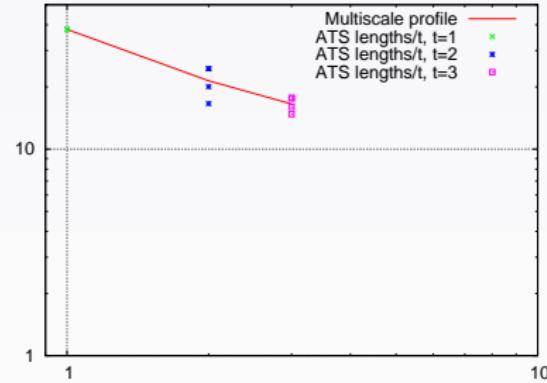
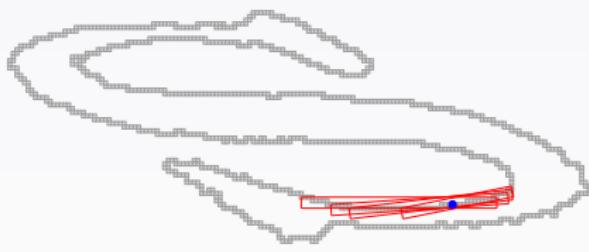


Second step: Extracting the Meaningful Thickness

Main steps of the Meaningful Thickness algorithm

- Tangential cover computation: **for each** thickness $1..N$.
- \forall alpha-thick segment S_i : update a **multiscale profile** on all points covered by S_i .
- Find a first first meaningful scale.

```
1 std::vector<double> x, y;
2 s.getProfile(x, y);
3 trace.info() << "Noise_level: " << sp.noiseLevel() << std::endl;
```



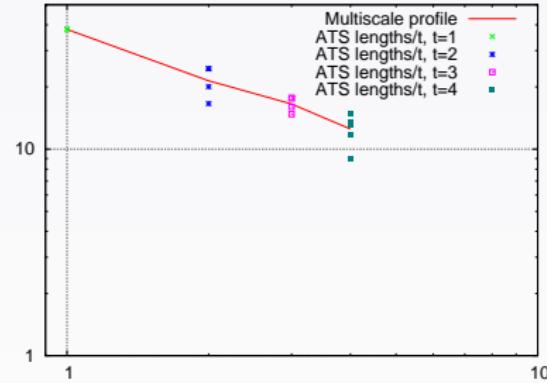
Second step: Extracting the Meaningful Thickness

Main steps of the Meaningful Thickness algorithm

- Tangential cover computation: **for each** thickness $1..N$.
- \forall alpha-thick segment S_i : update a **multiscale profile** on all points covered by S_i .
- Find a first first meaningful scale.

```

1 std::vector<double> x, y;
2 s.getProfile(x, y);
3 trace.info() << "Noise_level: " << sp.noiseLevel() << std::endl;
```

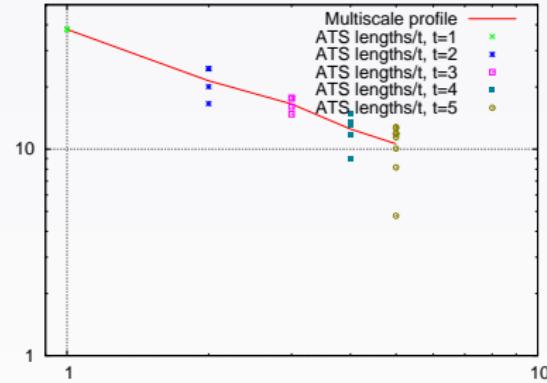
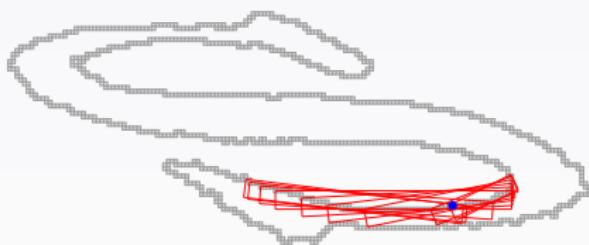


Second step: Extracting the Meaningful Thickness

Main steps of the Meaningful Thickness algorithm

- Tangential cover computation: **for each** thickness $1..N$.
- \forall alpha-thick segment S_i : update a **multiscale profile** on all points covered by S_i .
- Find a first first meaningful scale.

```
1 std::vector<double> x, y;
2 s.getProfile(x, y);
3 trace.info() << "Noise_level: " << sp.noiseLevel() << std::endl;
```

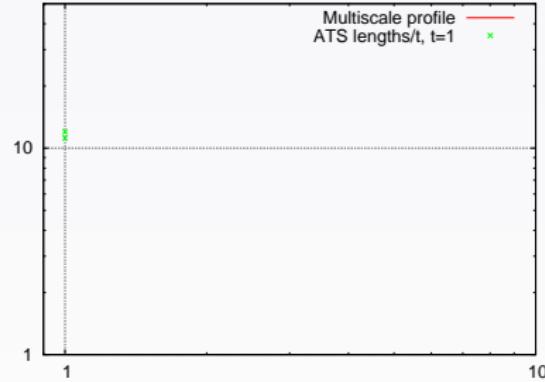
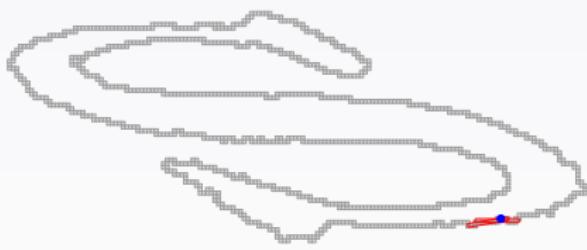


Second step: Extracting the Meaningful Thickness

Main steps of the Meaningful Thickness algorithm

- Tangential cover computation: **for each** thickness $1..N$.
- \forall alpha-thick segment S_i : update a **multiscale profile** on all points covered by S_i .
- Find a first first meaningful scale.

```
1 std::vector<double> x, y;  
2 s.getProfile(x, y);  
3 trace.info() << "Noise_level: " << sp.noiseLevel() << std::endl;
```



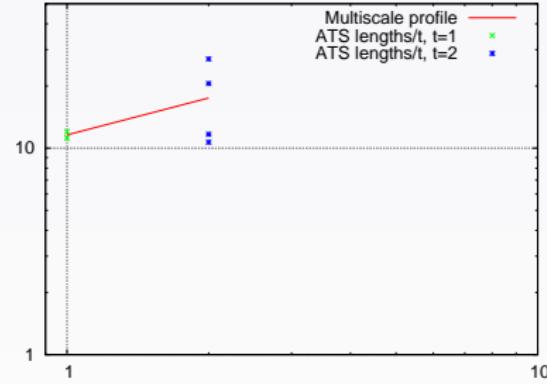
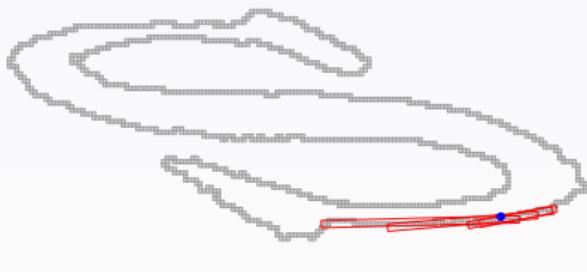
Second step: Extracting the Meaningful Thickness

Main steps of the Meaningful Thickness algorithm

- Tangential cover computation: **for each** thickness $1..N$.
- \forall alpha-thick segment S_i : update a **multiscale profile** on all points covered by S_i .
- Find a first first meaningful scale.

```

1 std::vector<double> x, y;
2 s.getProfile(x, y);
3 trace.info() << "Noise_level: " << sp.noiseLevel() << std::endl;
```



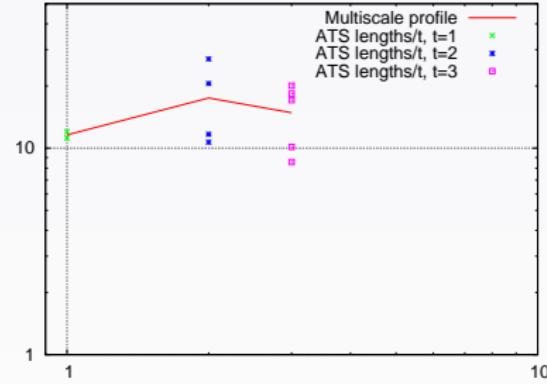
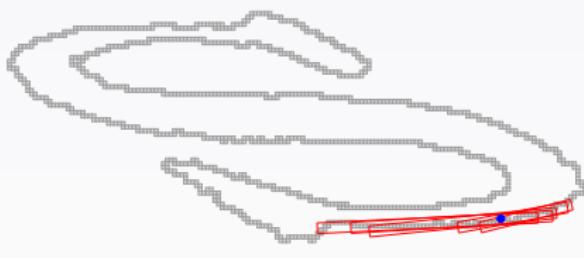
Second step: Extracting the Meaningful Thickness

Main steps of the Meaningful Thickness algorithm

- Tangential cover computation: **for each** thickness $1..N$.
- \forall alpha-thick segment S_i : update a **multiscale profile** on all points covered by S_i .
- Find a first first meaningful scale.

```

1 std::vector<double> x, y;
2 s.getProfile(x, y);
3 trace.info() << "Noise_level: " << sp.noiseLevel() << std::endl;
```

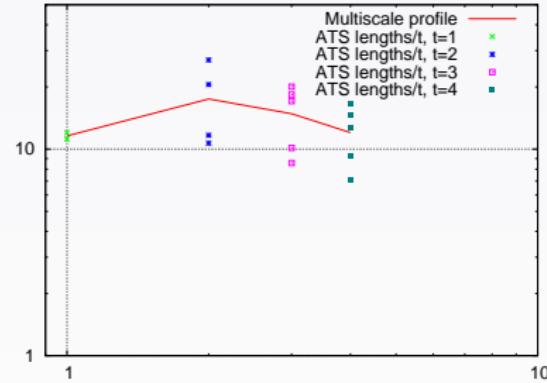
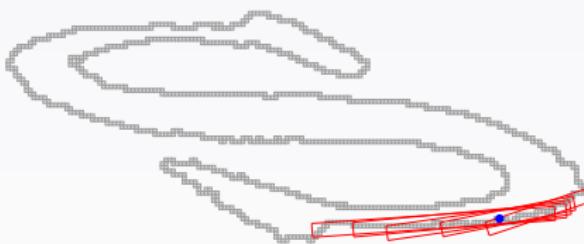


Second step: Extracting the Meaningful Thickness

Main steps of the Meaningful Thickness algorithm

- Tangential cover computation: **for each** thickness $1..N$.
- \forall alpha-thick segment S_i : update a **multiscale profile** on all points covered by S_i .
- Find a first first meaningful scale.

```
1 std::vector<double> x, y;
2 s.getProfile(x, y);
3 trace.info() << "Noise_level: " << sp.noiseLevel() << std::endl;
```



Second step: Extracting the Meaningful Thickness

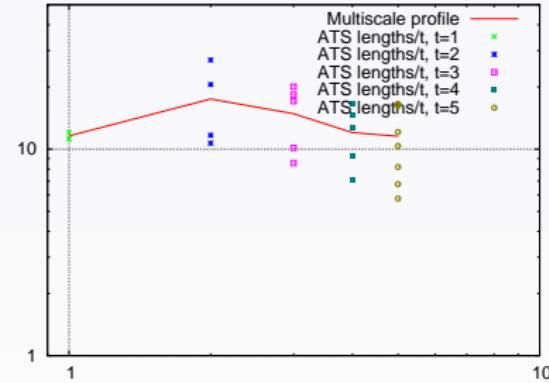
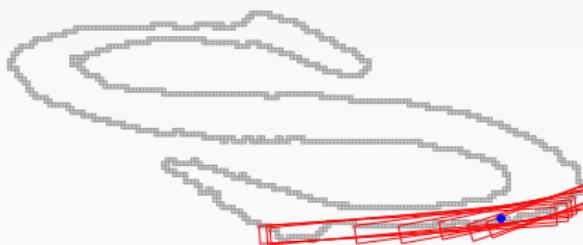
Main steps of the Meaningful Thickness algorithm

- Tangential cover computation: **for each** thickness $1..N$.
- \forall alpha-thick segment S_i : update a **multiscale profile** on all points covered by S_i .
- Find a first first meaningful scale.

```

1 std::vector<double> x, y;
2 s.getProfile(x, y);
3 trace.info() << "Noise_level: " << sp.noiseLevel() << std::endl;

```



Second step: Extracting the Meaningful Thickness (2)

Meaningful thickness computation on all points

- Compute the full tangential cover from [SaturatedSegmentation](#).
⇒ similarly to example:tuto4_SaturatedSegmentation.cpp

Second step: Extracting the Meaningful Thickness (2)

Meaningful thickness computation on all points

- Compute the full tangential cover from [SaturatedSegmentation](#).
⇒ similary to example:tuto4_SaturatedSegmentation.cpp
- As previously we need the tangential cover for each thickness.
- Maintian the [ScaleProfile](#) object.

(see file [tuto6_MeaningfulThicknessAllPt.cpp](#))

```
1 std::vector<Point>::const_iterator itInit = contour.begin();
2 for(unsigned int thickness = 1; thickness <= scaleMax; thickness++) {
3     AlphaThickSegmentComputer2D aComputer(thickness);
4     AlphaSegmentation segmentator(contour.begin(), contour.end(), aComputer);
5     typename AlphaSegmentation::SegmentComputerIterator it = segmentator.begin();
6     typename AlphaSegmentation::SegmentComputerIterator endSeg = segmentator.end();
7     for( ; it != endSeg; ++it){
8         trace.info() << ".";
9         AlphaThickSegmentComputer2D seg(*it);
10        double lengthSegment = seg.getSegmentLength();
11        std::vector<Point>::const_iterator its = seg.begin();
12        for ( ; its != seg.end(); ++its) {
13            unsigned int i = std::distance(itInit, its);
14            vectProfiles[i].addValue(thickness-1, lengthSegment/thickness);
15        }
16    }
17 }
```

Second step: Extracting the Meaningful Thickness (2)

Meaningful thickness computation on all points

- Compute the full tangential cover from [SaturatedSegmentation](#).
⇒ similary to example:tuto4_SaturatedSegmentation.cpp
- As previously we need the tangential cover for each thickness.
- Maintian the [ScaleProfile](#) object.

(see file [tuto6_MeaningfulThicknessAllPt.cpp](#))

```
1 std::vector<Point>::const_iterator itInit = contour.begin();
2 for(unsigned int thickness = 1; thickness <= scaleMax; thickness++){
3     AlphaThickSegmentComputer2D aComputer(thickness);
4     AlphaSegmentation segmentator(contour.begin(), contour.end(), aComputer);
5     typename AlphaSegmentation::SegmentComputerIterator it = segmentator.begin();
6     typename AlphaSegmentation::SegmentComputerIterator endSeg = segmentator.end();
7     for( ; it != endSeg; ++it){
8         trace.info() << " ";
9         AlphaThickSegmentComputer2D seg(*it);
10        double lengthSegment = seg.getSegmentLength();
11        std::vector<Point>::const_iterator its = seg.begin();
12        for ( ; its != seg.end(); ++its){
13            unsigned int i = std::distance(itInit, its);
14            vectProfiles[i].addValue((double) i - 1, lengthSegment/thickness);
15        }
16    }
17 }
```

Overview of the presentation

- 1 1. Introduction to the Meaningful Scale Detection
 - 1.1 Main idea of the meaningful scale
 - 1.2 Meaningful Scale Profile and Noise Detection
 - 1.3 Experiments and new applications
- 2 2. Extension to the Meaningful Thickness
 - 2.2 Meaningful Thickness profile
 - 2.3 Experiments, comparisons and applications
 - 2.4 Computing the meaningful thickness in the DGtal Library
- 3 3. Highlighting Reproducible Research
 - 3.1 Image Processing On Line Presentation: principle and current form
 - 3.2 Structure of an IPOL demonstration

3. Highlighting Reproducible Research

3. Highlighting Reproducible Research (1)

Reproducible research in sciences:

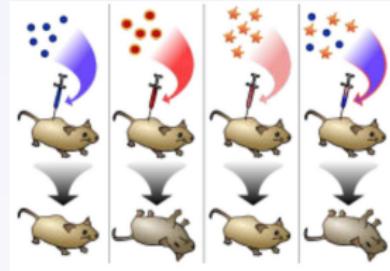
- Theoretical scientists share demonstrations;

$$\begin{aligned}
 CPT^2 &= 2a \cdot 1 + c(Ta^2) \\
 &= 2a \cdot 1 + 2a \cdot T a^2 + c(T^2 a^4) \\
 &= 2a \cdot 1 + 2a \cdot T a^2 + 2a \cdot T^2 a^4 + c(T^3 a^6) \dots \text{etc} \\
 CPT^3 &= 2a \cdot 1 + 2a \cdot T a^2 + 2a \cdot T^2 a^4 + 2a \cdot T^3 a^6 + \dots \\
 \text{The recurrence equation above then leads us to the summation equation:} \\
 2a \cdot (1 + a^2 + a^4 + a^6 + \dots) &= 2a \frac{1}{1-a^2} \\
 \text{Since we know the converging infinite geometric series states:} \\
 \sum_{n=0}^{\infty} ra^n &= \frac{r}{1-r}, \quad \text{if } |r| < 1 \\
 2a \frac{1}{1-a^2} &= 2a \cdot \frac{1}{1-a^2} - \frac{2a}{a^2-1} \\
 \text{This in turn leads us to the closed form formula:} \\
 2a \frac{1}{1-a^2} &= 2a \cdot \frac{1}{1-a^2} - \frac{2a}{a^2-1} \\
 \text{Proof by induction: } 2a \cdot 1 + c(Ta^2) \cdot 1 &= \frac{2a}{1-a^2} \\
 \text{Inductive Hypothesis: } c(TK)^2 &= \frac{2K}{1-K^2} \quad \forall K < 1 \\
 \text{Inductive Step: } & \\
 \frac{2K}{1-K^2} \cdot K + \frac{2K}{1-K^2} \cdot \frac{2K}{1-K^2} &= \frac{2K}{1-K^2} \\
 2K \cdot K + \frac{2K}{1-K^2} \cdot \frac{2K}{1-K^2} &= \frac{2K}{1-K^2} \\
 2K^2 + \frac{4K^2}{1-K^2} &= \frac{2K}{1-K^2} \\
 2K^2 + \frac{4K^2}{1-K^2} - \frac{2K^2}{1-K^2} &= \frac{2K}{1-K^2} \\
 2K^2 + \frac{2K^2}{1-K^2} &= \frac{2K}{1-K^2} \\
 2K^2 \cdot \frac{1+1}{1-K^2} &= \frac{2K}{1-K^2} \\
 2K^2 \cdot \frac{2}{1-K^2} &= \frac{2K}{1-K^2} \\
 2K^2 \cdot \frac{2}{1-K^2} &= \frac{2K}{1-K^2}
 \end{aligned}$$

3. Highlighting Reproducible Research (1)

Reproducible research in sciences:

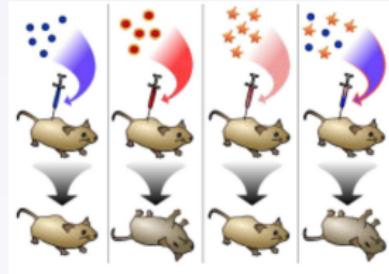
- *Theoretical scientists share demonstrations;*
- *Experimental scientists share procedures;*



3. Highlighting Reproducible Research (1)

Reproducible research in sciences:

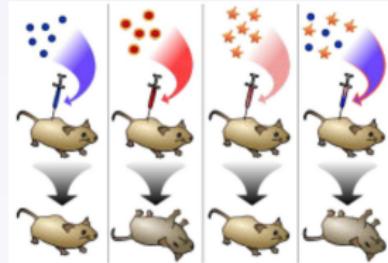
- *Theoretical scientists* share demonstrations;
- *Experimental scientists* share procedures;
- *Computational scientists... ?*



3. Highlighting Reproducible Research (1)

Reproducible research in sciences:

- *Theoretical scientists* share demonstrations;
- *Experimental scientists* share procedures;
- *Computational scientists...* ?



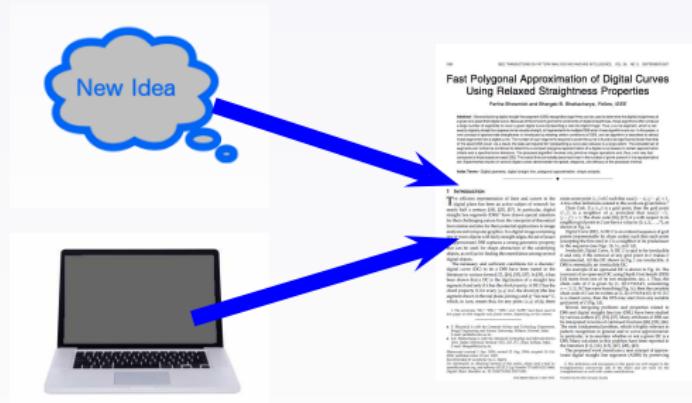
Computer Science:

- Description of methods/algorithms;
- description often limited (constraints on page limits);
- parameters not given or not well described;
- steps of pre/post processing missing.

3. Highlighting Reproducible Research (2)

Research in Computer Science:

- 1 New idea;
- 2 demonstration, implementation;
- 3 article publication.



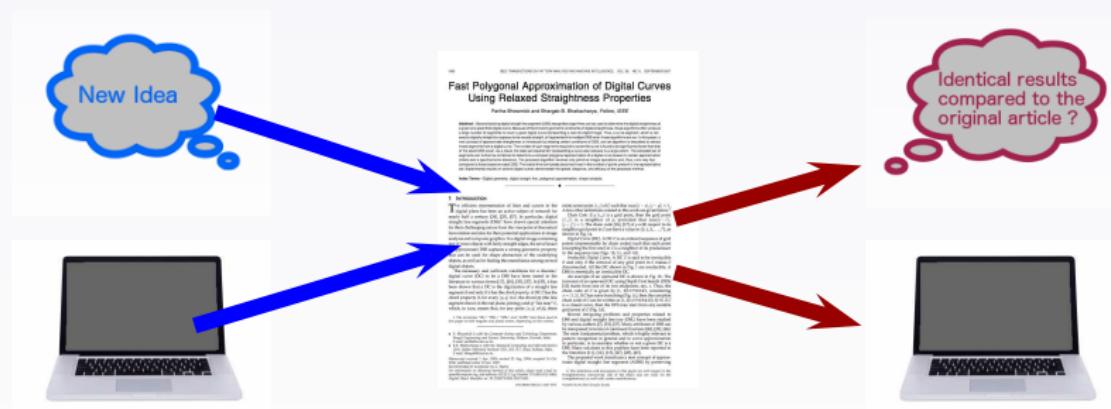
3. Highlighting Reproducible Research (2)

Research in Computer Science:

- 1 New idea;
- 2 demonstration, implementation;
- 3 article publication.

Reusable Research:

- 1 Article which seems interesting;
- 2 re-implement the algorithm;
- 3 conformity of the results with the original.



3. Highlighting Reproducible Research (3)

Providing source code/data

- ⊕ A real added value for the publication;
- ⊕ increases the impact/comparisons;
- ⊖ software is not really acknowledged;
- ⊖ important effort (documentation, tests, user maintenance).

```
#include <limits.h>
#include <Ostal/headers/Common.h>
#include <Ostal/headers/headers/rofHeader.h>
#include <Ostal/headers/rofDisplay3D.h>

#include <Ostal/headers/rofDisplay3DModifier.h>
#include <Ostal/images/ImageSelector.h>
#include <Ostal/headers/headers/rofImageSetFromImage.h>
#include <Ostal/headers/rofImageSet.h>
#include <ConfigSamples.h>

using namespace std;
using namespace OStal;
using namespace Z3d;

int main( int argc, char** argv )
{
    std::string inputFileName = examplesPath + "samples/AI100.vol";
// { [exampleDisplay3DTest]
    Display3DViewer viewer;
    ImageSelectorImage < Z3d::Image > Z3d::Image;
    Image image = VolHeader::Image(); image.VolName(inputFileName);
    Z3d::Image < Z3d::Image > set3d;
    SetFromImageZ3d::bigImageSet(appendImage(&set3d, image, 0, 350);

    viewer << set3d;
    viewer >> "exportMeshToOFF";
// }
    return 0;
}
```

3. Highlighting Reproducible Research (3)

Providing source code/data

- ⊕ A real added value for the publication;
- ⊕ increases the impact/comparisons;
- ⊖ software is not really acknowledged;
- ⊖ important effort (documentation, tests, user maintenance).

Software Diffusion

- Specialized journals in software:

Source Code for Biology and Medicine, Journal of Open Research Software, Computing in Science and Engineering, ...

- Diffusion platform:

RunMyCode / Run&Share, Figshare, DataDryad, Harvard Dataverse, Ubiquity Metajournals, Zenodo (with doi) ...

⇒ no validation, no scientific review (reliability and durability problem).

```
#include <iostream>
#include <Ostal/Volume/Common.h>
#include <Ostal/Volume/render/vtHeader.h>
#include <Ostal/Volume/Display3D.h>

#include <Ostal/Volume/Display3DModifier.h>
#include <Ostal/Images/ImageSelector.h>
#include <Ostal/Images/VolumeImage.h>
#include <Ostal/Helpers/Volume3d.h>
#include <ConfigSamples.h>

using namespace std;
using namespace Ostal;
using namespace Z3D;

int main( int argc, char** argv )
{
    string inputFilename = examplesPath + "samples/AI100.vol";
    // [ExampleDisplay3DFF]
    Display3DModifier viewer;
    Image3D image;
    VolumeImage volumeImage;
    Image3D image3d;
    volumeImage.readVolumeFile(inputFilename);
    image3d.setVolumeImage(volumeImage);
    SetFromImage3D(image3d, image);
    viewer.set3D();
    viewer.set("outputHeight720FF");
    // [/ExampleDisplay3DFF]
    return 0;
}
```

3.1 Image Processing On Line Presentation: origin and motivation

Origin:

- Journal started in October 2009;
- under the initiative of Nicolas Limare, Jean-Michel Morel and the Image Processing team at the CMLA lab (ENS-Cachan);
- first article published in 2010.

3.1 Image Processing On Line Presentation: origin and motivation

Origin:

- Journal started in October 2009;
- under the initiative of Nicolas Limare, Jean-Michel Morel and the Image Processing team at the CMLA lab (ENS-Cachan);
- first article published in 2010.

Motivation [Limare & Morel 2009]:

- Reproducible research;
- new way to publish research results;
- allows everybody to test the algorithms;
 ⇒ with their own images
- independent of the platform (the demos execute on the server side and the results are shown to the user using a web interface).

3.1 Image Processing On Line Presentation: principle and current form

Characteristics:

- Research journal in **image** processing;
- each article contains a description of **one algorithm** and its **source code**;
- association of each **article** with its **online demonstration**, with **archived experiments**;
- the peer-review process includes the **article**, **demo**, and **source code**;
- *Open Science* journal and **Reproducible Research**.



IPOL Journal · Image Processing On Line

HOME · ABOUT · ARTICLES · PREPRINTS · WORKSHOPS · NEWS · SEARCH

IPOL is a research journal of image processing and image analysis which emphasizes the role of mathematics as a source for algorithm design and the reproducibility of the research. Each article contains a text on an algorithm and its source code, with an online demonstration facility and an archive of experiments. Text and source code are peer-reviewed and the demonstration is controlled. IPOL is an Open Science and Reproducible Research journal.

[Editorial Policy](#) [Editorial Board](#) [Submit an Article](#) [Follow IPOL](#)

[Index](#) · [Articles 2011](#) [2012](#) [2013](#) [2014](#) [2015](#) · [Preprints](#) · [News](#) · [Citation score](#)

3.1 Image Processing On Line Presentation: principle and current form

Philosophy of the journal:

- Follows the guideline on reproducible research topics;
- reproducible research standard [\[Stodden 09a\]](#) [\[Stodden 09b\]](#);
- answer to credibility crisis in scientific computation (as pointed out by Donoho [\[Donoho et al. 09\]](#)).

3.1 Image Processing On Line Presentation: principle and current form

Philosophy of the journal:

- Follows the guideline on reproducible research topics;
- reproducible research standard [\[Stodden 09a\]](#) [\[Stodden 09b\]](#);
- answer to credibility crisis in scientific computation (as pointed out by Donoho [\[Donoho et al. 09\]](#)).

What IPOL is not:

- IPOL publishes **algorithms** along with **their implementation**, but not compiled software;
- IPOL is not a software library (each code has minimal dependencies);
- IPOL is not a software or code diffusion platform.

3.1 Image Processing On Line Presentation: principle and current form

- Current form: “classic” (with online PDF).

The screenshot shows a web browser window with the following details:

- Title Bar:** IPOL Journal - LSD: a Line Segment Detector
- Address Bar:** www.ipol.im/pub/art/2012/gjmr-lsd/
- Page Content:**
 - Header:** IPOL Journal - Image Processing On Line
 - Article Title:** LSD: a Line Segment Detector
 - Authors:** Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, Gregory Randall
 - Navigation:** article | demo | archive
 - Published:** 2012-03-24
 - Reference:** Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall, LSD: a Line Segment Detector, Image Processing On Line, vol. 2012. <http://dx.doi.org/10.5201/ipol.2012.gjmr-lsd>
 - Communication:** Communicated by Lionel Moisan
 - Demo:** Demo edited by Rafael Grompone
 - Abstract:** LSD is a linear-time Line Segment Detector giving subpixel accurate results. It is designed to work on any digital image without parameter tuning. It controls its own number of false detections: On average, one false alarm is allowed per image. The method is based on Burns, Hanson, and Riseman's method, and uses an a-contrario validation approach according to Desolneux, Moisan, and Morel's theory. The version described here includes some further improvement over the one described in the original article.
 - Download:**
 - * full text manuscript: PDF (554K) PDF high-res. (1.4M)
 - * source code: ZIP
 - Preview:** Loading takes a few seconds. Images and graphics are degraded here for faster rendering. See the downloadable PDF documents for original high-quality versions.

3.1 Image Processing On Line Presentation: principle and current form

- Current form: “classic” (with online PDF).
- Associated demos.

The screenshot shows a web browser window with the title "IPOL Journal - LSD: a Line Segment Detector". The URL is "demo.ipol.im/demo/gjmr_line_segment_detector/". The page content includes:

- IPOL Journal - Image Processing On Line**
- LSD: a Line Segment Detector**
- Navigation links: HOME, ABOUT, ARTICLES, PREPRINTS, NEWS, SEARCH
- Buttons: article, demo, archive
- A note: "Please cite the reference article if you publish results obtained with this online demo."
- Select Data**: A section with four image thumbnails:
 - Chairs image
 - Le Piree
 - LSD molecule
 - Noise
- Upload Data**: A section with an "input image" field containing "Choisir le fichier aucun fichier sélectionné" and a "upload" button.
- Small text at the bottom: "Images larger than 1000x2000 pixels will be resized. Upload size is limited to 250MB per image file and 10MB for the whole upload set. TIFF, JPEG, PNG, GIF, PNM (and other standard formats) are supported. The uploaded will be publicly archived unless you switch to private mode on the result page. Only upload suitable images. See the copyright and legal conditions for details."
- Page footer: feeds & twitter • sitemap • contact • privacy policy • ISBN: 2105-1232 • DOI: 10.5201/ipol
supported by CMLA, ENS Cachan • DM, Universitat de les Illes Balears • Fing, Universidad de la República • GdR-net
© 2009-2013, IPOL Image Processing On Line & the authors

3.1 Image Processing On Line Presentation: principle and current form

- Current form: “classic” (with online PDF).
 - Associated demos.
 - Archive containing experiments with data uploaded by users.

IPOL Journal - LSD: a Line Segment Detector

demo.ipol.im/demo/gjmr_line_segment_detector/archive/ Lecteur

IPOL Journal - Image Processing On Line

LSD: a Line Segment Detector

article demo archive

Please cite the reference article if you publish results obtained with this online demo.

12/26/2010 public releases of online experiments with original images since 2009/05/13 18:16.

If you have any problem with this online demo, please contact us to request the removal of some images. Some archived content may be deleted by the editorial board for size matters, inadequate content, user requests, or other reasons.

pages: < <> > >> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145
146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184
185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262
263 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301
302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379
380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418
419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457
487 488 489 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535
536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565

key F1259852849590741C1B79D003087663
date 2015/11/12 10:06
LSD Version 1.6.3 of November 11, 2011, compiled
version Jul 11 2013 16:46:31
run time # 579286416397
[s] files output.txt output.png output.svg

images 

key 463FB1B7C5A9D0209BA0CC20CB64F86
date 2015/11/12 10:06
LSD Version 1.6.3 of November 11, 2011, compiled
version Jul 11 2013 16:46:31
run time # 786853047867
[s] files output.txt output.png output.svg

images 

key 41FB8DFCE7C7D7B8EBCDF89DC55AF5
date 2013/11/12 10:13
LSD Version 1.6.3 of November 11, 2011, compiled
version Jul 11 2013 16:46:31
run time # 218708992004
[s] files output.txt output.png output.svg

images 

key 41FB8DFCE7C7D7B8EBCDF89DC55AF5
date 2013/11/12 10:13
LSD Version 1.6.3 of November 11, 2011, compiled
version Jul 11 2013 16:46:31
run time # 218708992004
[s] files output.txt output.png output.svg

images 

key 41FB8DFCE7C7D7B8EBCDF89DC55AF5
date 2013/11/12 10:13
LSD Version 1.6.3 of November 11, 2011, compiled
version Jul 11 2013 16:46:31
run time # 218708992004
[s] files output.txt output.png output.svg

images 

3.1 Image Processing On Line Presentation: editorial structure

Same aspects as a classical journal:

- Editorial project, editorial committee;
- articles, authors, editors;
- reviewing process and validation;
- ISSN, DOI;
- special issues;
- currently indexed by:

Scirus, Google Scholar, DBLP, DOAJ, SHERPA/RoMEO, Héloïse, WorldCat, CrossRef, Ulrich, Index Copernicus, PBN, JGate, VisionBib, CVonline, JournalSeek, and NewJour.

3.1 Image Processing On Line Presentation: editorial structure

Same aspects as a classical journal:

- Editorial project, editorial committee;
- articles, authors, editors;
- reviewing process and validation;
- ISSN, DOI;
- special issues;
- currently indexed by:

Scirus, Google Scholar, DBLP, DOAJ, SHERPA/RoMEO, Héloïse, WorldCat, CrossRef, Ulrich, Index Copernicus, PBN, JGate, VisionBib, CVonline, JournalSeek, and NewJour.

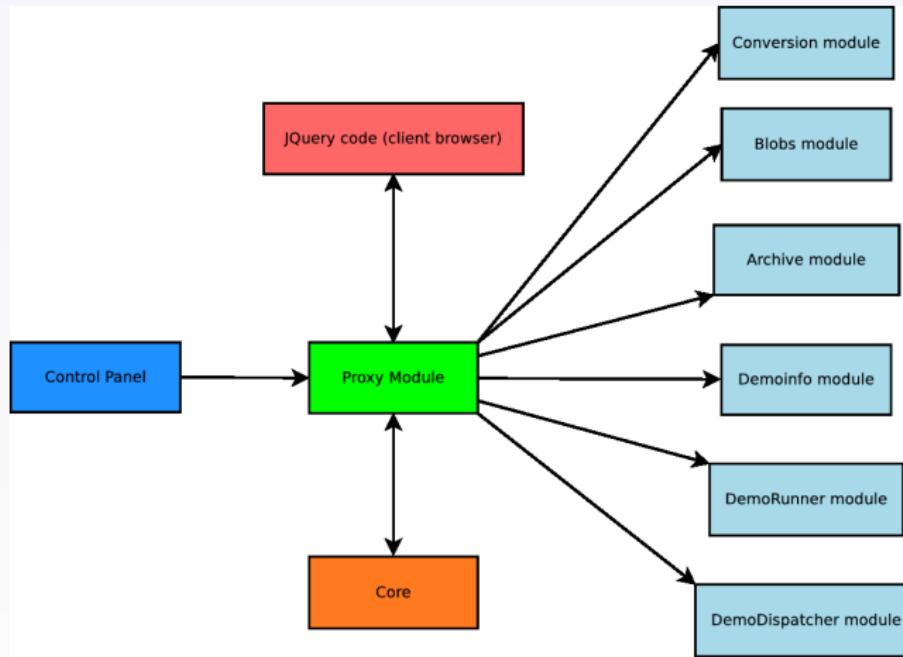
Software point of view:

- Each article should propose an implementation;
- reviewing step, verification, validation, and publication;
- reviewer: **check the correspondence between the algorithm description in the article and code** (+ code readability and code documentation).

3.2 Structure of an IPOL demonstration

New Architecture (upcomming end of year)

- Based on micro-services to allow scalability and fast development.
- Better than previous architecture for distributed algorithm computing.

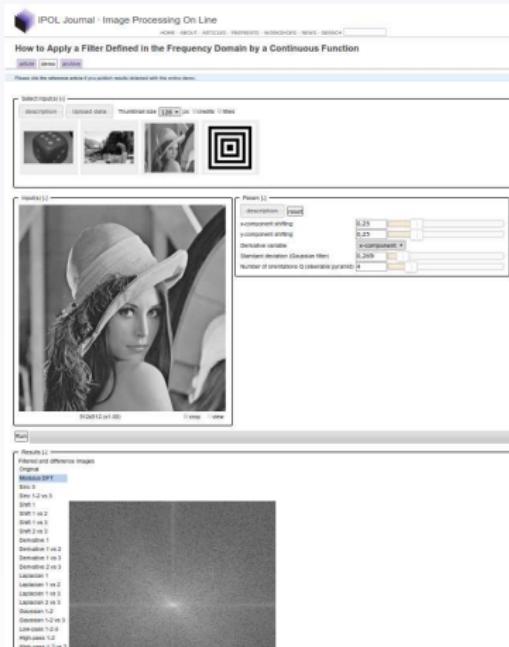


3.2 Structure of an IPOL demonstration (2)

Preview of new interface

- Web interface more ergonomic and simple..
- Example of the demo of a published article:

<http://ipolcore.ipol.im/demo/clientApp/demo.html?id=116>



3.2 Structure of an IPOL demonstration (3)

New Demo Control Panel

- Replace previous demo source code.
- The IPOL Editors create the demo using an internal web tool called:
"Control Panel"

The screenshot shows a web browser window with the URL `integration.ipol.im/cp/demoinfo_demos/`. The page title is "DEMONS LIST". There is a search bar and a "Add New Demo" button. Below is a table listing three demos:

ID	Title	URL	State	Last Modification	Actions
#1234567890	Non-local patch-based image inpainting	http://dev.ipol.im/~parism/189-2011-1-SP.gz	preprint	14 Oct 2016	Delete Edit
#9876543210	arthur TEST - Gaussian Texture Inpainting	http://www.math-info.univ-paris5.fr/~aleclair/ipol/gauss texinpaint.zip	inactive	10 Oct 2016	Delete Edit
#1234567890	pascal test - rectification	http://imagine.enpc.fr/~monasse/	preprint	17 Jun 2016	Delete Edit

Below the table are four buttons: DOL, AUTHORS, EDITORS, and DEMO EXTRAS.

3.2 Structure of an IPOL demonstration (3)

New Demo Control Panel

- Replace previous demo source code.
- The IPOL Editors create the demo using an internal web tool called:
"Control Panel"
- Demo Description Language: an abstract syntax to describe the IPOL demos

The screenshot shows the IPOL Demo Control Panel interface. At the top, there are three tabs: 'Demos' (selected), 'Authors', and 'Editors'. Below the tabs is a search bar and a button labeled 'Add New Demo'. The main area is titled 'DEMONS LIST' and contains a table with three rows of demo entries. Each row has a 'DDL' button, an 'AUTHORS' button, an 'EDITORS' button, and a 'DEMO EXTRAS' button. The first row's DDL button is highlighted in blue. The second row's DDL button is also highlighted in blue. The third row's DDL button is not highlighted. An 'Edit DDL data' modal window is open over the list, centered on the second row. The modal has a title 'Edit DDL data' and a code editor containing JSON-like DDL code. The code includes fields for general information like title and abstract, and specific validation rules like 'The images must have the same size'. At the bottom of the modal are 'Save', 'Close', 'Pretty Print', and 'Ugly Print' buttons. At the very bottom of the page, there is a navigation bar with links: Status, Demos, Archive Module, Blobs Module, DemoInfo Module, DemoRunner Module, DemoDispatcher Module, Proxy Module, User Manual, and Logout. The 'Demos' link in the navigation bar is also highlighted in blue.

Thanks for your attention !



Kerautret, B. and Lachaud, J.-O. (2012).

Meaningful Scales Detection along Digital Contours for Unsupervised Local Noise Estimation detection.
IEEE. Trans. on PAMI, in press (10.1109/TPAMI.2012.38).



Bertrand Kerautret, and Jacques-Olivier Lachaud,

Meaningful Scales Detection: an Unsupervised Noise Detection Algorithm for Digital Contours,
Image Processing On Line, 4 (2014), pp. 98–115. <http://dx.doi.org/10.5201/pol.2014.75>



T. V. Hoang, E. H. Barney Smith, and S. Tabbone (2011)

Edge noise removal in bilevel graphical document images using sparse representation,
IEEE International Conference on Image Processing - ICIP'2011, Brussels, Belgique, Sep 2011.



[Koenderink 84] Koenderink, J.J.:

The structure of images.

Biol. Cyb. 50 (1984) 363–370



[Perona-Malik 90] Perona, P. , Malik, J.

Scale-space and edge detection using anisotropic diffusion

IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990, 12, 629-639



[Jeong Kim 92] Barash, D. :

Adaptive determination of filter scales for edge detection

IEEE Trans. Patt. Anal. Mach. Intell., 24(6): 844-847, 2002.



[Elder Zucker 98] Elder, J.H., Zucker, S.W.:

Local scale control for edge detection and blur estimation.

IEEE Trans. Pattern Anal. Mach. Intell. 20(7) (1998) 669–716

**[Barash 02] Barash, D. :**

A fundamental relationship between bilateral filtering, adaptive smoothing and the nonlinear diffusion equation
IEEE Trans. Patt. Anal. Mach. Intell., 24(6): 844-847, 2002.

**[Kervrann 04] Kervrann, C.:**

An Adaptive Window Approach for Image Smoothing and Structures Preserving.
ECCV 04, Springer, 2004, 132-144

**[Chen 05] Chen, K.:**

Adaptive smoothing via contextual and local discontinuities.
IEEE Trans. Pattern Anal. Mach. Intell. 27(10) (2005) 1552–1566

**Cao, F. (2003).**

Good continuations in digital image level lines.
In *Proc. of ICCV 2003*, volume 1, pages 440–448. IEEE.

**Desolneux, A., Moisan, L., and Morel, J.-M. (2001).**

Edge detection by helmholtz principle.
Journal of Mathematical Imaging and Vision, 14:271–284.

**Lachaud, J.-O (2006).**

Espaces non-euclidiens et analyse d'image : modèles déformables riemanniens et discrets, topologie et géométrie discrète.

Habilitation à diriger des recherches, Université Bordeaux 1, Talence, France (2006) (en français).

**Vacavant, A. and Roussillon, T. and Kerautret, B. and Lachaud, J.-O.**

A combined multi-scale/irregular algorithm for the vectorization of noisy digital contours
Computer Vision and Image Understanding, 117(4) 1077-3142, 2014



[Lachaud07] Lachaud, J.; Vialard, A. ; de Vieilleville, F.

Fast, Accurate and Convergent Tangent Estimation on Digital Contours

Image and Vision Computing, 2007, 25, 1572-1587



P. Ngo1, H. Nasser, I Deblé-Rennesson, B. Keratret

Adaptive Tangential Cover for Noisy Digital Contours

Submitted to DGCI 2015



Faure, A., Buzer, L., and Feschet, F. (2009).

Tangential cover for thick digital curves.

Pattern Recognition, 42(10):2279 – 2287.



Deblé-Rennesson, I., Feschet, F., and Rouyer-Degli, J. (2006).

Optimal blurred segments decomposition of noisy shapes in linear time.

Computers & Graphics, 30(1).



Lachaud, J.

Digital shape analysis with maximal segments

In. *Applications of Discrete Geometry and Mathematical Morphology*. First International Workshop, WADGMM 2010, Istanbul, Turkey. pp. 14–27 (2010)



[Limare & Morel 2009] Limare N. and Morel, J-M (2009)

IPOL Project Presented at the CMLA Seminar

http://www.ipol.im/news/20091022_cmla/s5.html

CMLA ENS Cachan



[Stodden 09a] V. Stodden

The legal framework for reproducible scientific research: Licensing and copyright

Computing in Science & Engineering, vol. 11, no. 1, pp. 35–40, 2009.



[Stodden 09b] V. Stodden

Enabling reproducible research: Open licensing for scientific innovation

International Journal of Communications Law and Policy, Forthcoming, 2009.



[Donoho et al. 09] D. L. Donoho, A. Maleki, I. U. Rahman, M. Shahram, and V. Stodden

Reproducible research in computational harmonic analysis

Computing in Science & Engineering, vol. 11, no. 1, pp. 8–18, 2009.