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

Bertrand Kerautret[1]
(work in collaboration with Jacques-Olivier Lachaud[2], Mouhammad Said[2], and Miguel Colom[3])

[1]Université de Lorraine, LORIA CNRS, ADAGIo team , UMR 7503, Vandoeuvre-lès-Nancy, F-54506, France
[2]LAMA (UMR CNRS 5127), Université Savoie Mont Blanc, F-73376, France
[3]CMLA ENS Cachan, France

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

Overview of the presentation - Part II -

Overview of the presentation

Part I. Introduction to the Meaningful Scale Detection

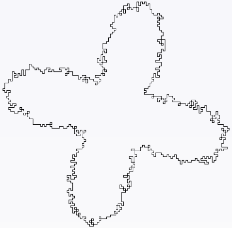
Noise induces generally the tuning of parameters

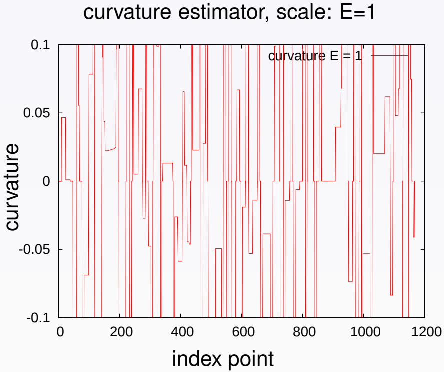
- Output quality/accuracy depends on the choice of the parameter.

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.

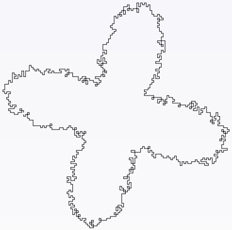


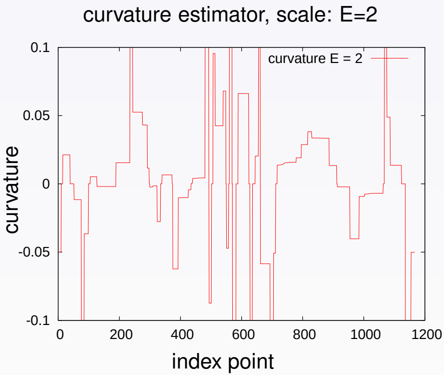


Part I. Introduction to the Meaningful Scale Detection

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.

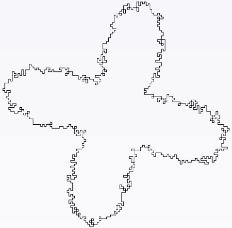


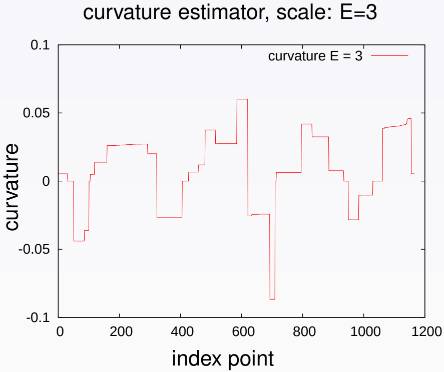


Part I. Introduction to the Meaningful Scale Detection

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.

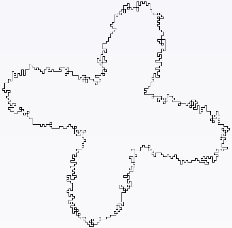


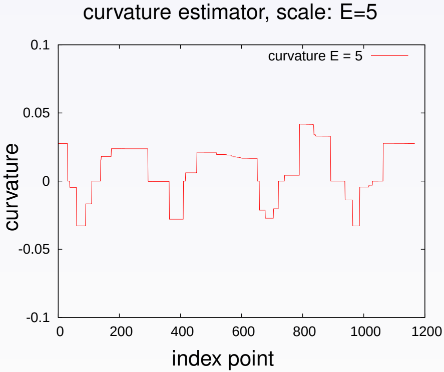


Part I. Introduction to the Meaningful Scale Detection

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.

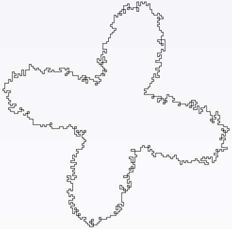


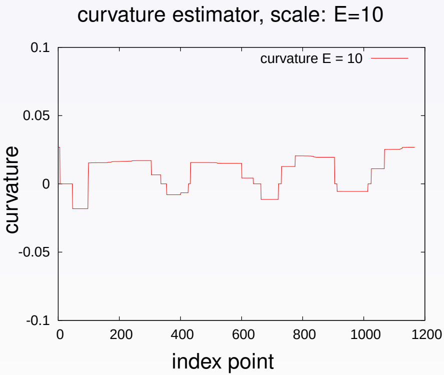


Part I. Introduction to the Meaningful Scale Detection

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.



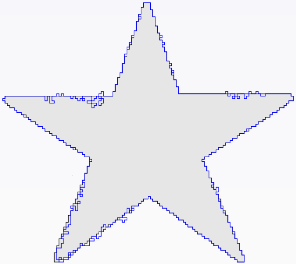


Part I. Introduction to the Meaningful Scale Detection

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 $\epsilon = 10$)

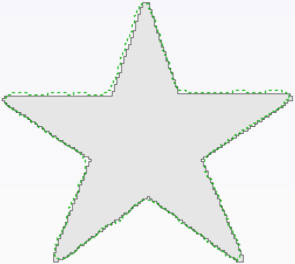


Part I. Introduction to the Meaningful Scale Detection

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 $\epsilon = 20$)

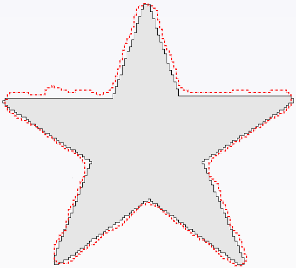


Part I. Introduction to the Meaningful Scale Detection

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 $\epsilon = 30$)

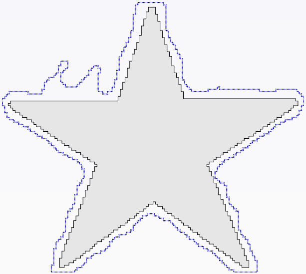


Part I. Introduction to the Meaningful Scale Detection

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 optimitization (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 optimitization (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 optimitization (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 optimitization (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 optimitization (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.

[Cao 03] $\epsilon = 1$

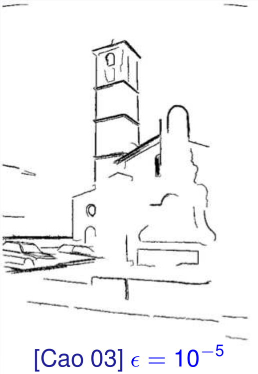
[Cao 03] $\epsilon = 10^{-5}$

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]

[Cao 03] $\epsilon = 1$



[Cao 03] $\epsilon = 10^{-5}$

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

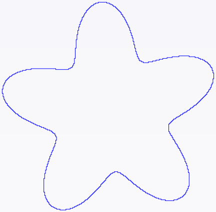
Main idea: Local shape analysis of digital contours

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.

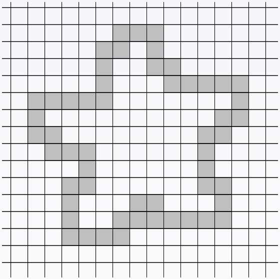


X

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.

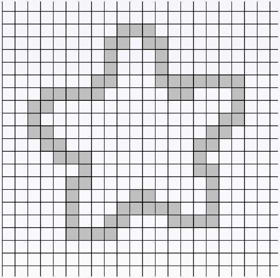


$Dig_{20}(X)$

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.

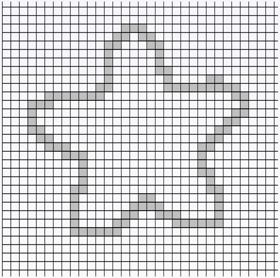


$Dig_{15}(X)$

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.

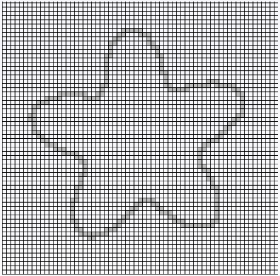


$Dig_{10}(X)$

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.

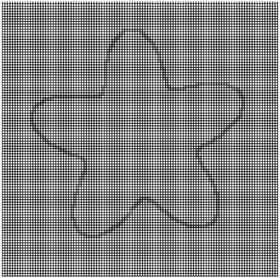


$Dig_5(X)$

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.

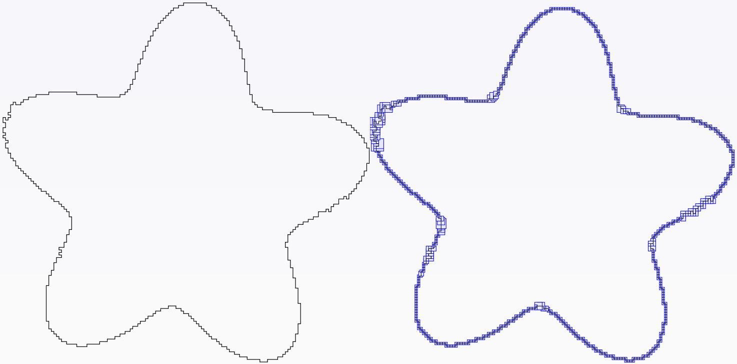


$Dig_3(X)$

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

Main idea: Local shape analysis of digital contours

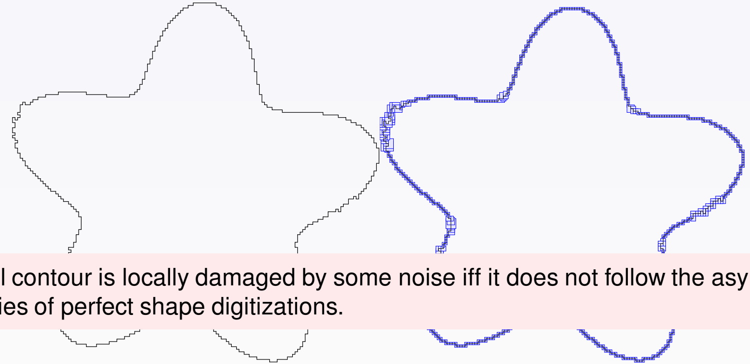
1. Exploit asymptotic properties of perfect shape digitizations.
2. Estimate them from a multiresolution decomposition of the input shape.
3. Compare them to determine a local meaningful scale.



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.

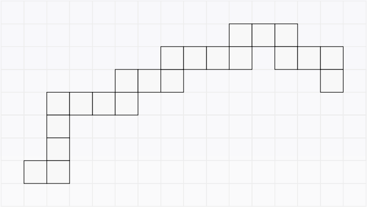
1.1 Main idea of the meaningful scale (1)

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 $\gcd(a, b) = 1$.



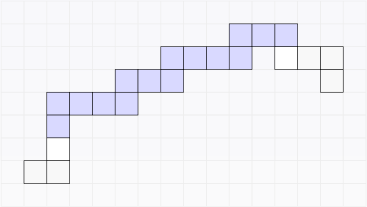
1.1 Main idea of the meaningful scale (1)

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 $\gcd(a, b) = 1$.



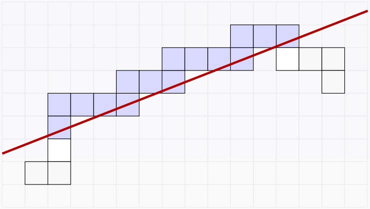
1.1 Main idea of the meaningful scale (1)

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 $\gcd(a, b) = 1$.



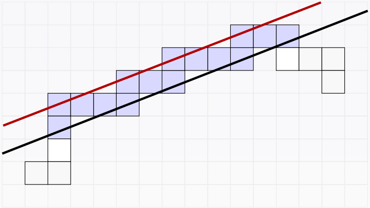
1.1 Main idea of the meaningful scale (1)

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)

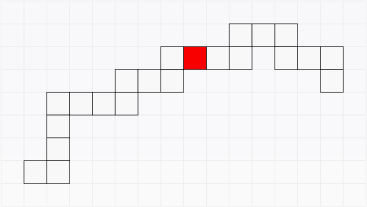
Concept based on the definition of discrete primitives

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)

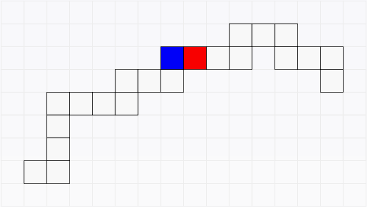
Concept based on the definition of discrete primitives

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)

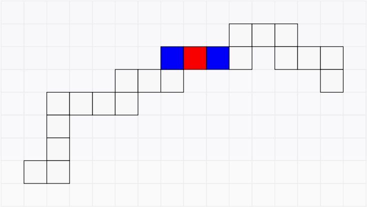
Concept based on the definition of discrete primitives

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)

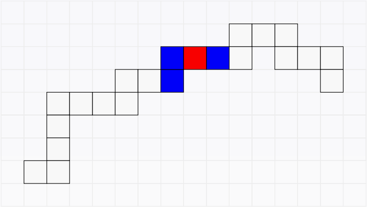
Concept based on the definition of discrete primitives

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)

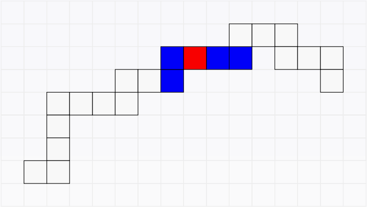
Concept based on the definition of discrete primitives

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)

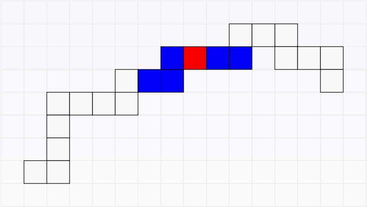
Concept based on the definition of discrete primitives

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)

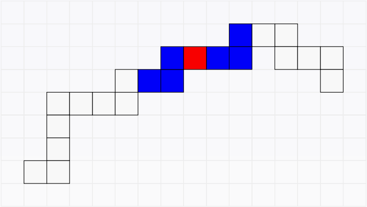
Concept based on the definition of discrete primitives

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)

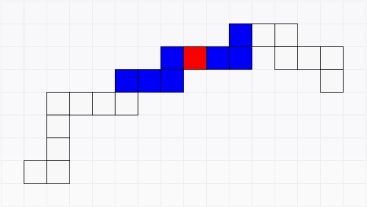
Concept based on the definition of discrete primitives

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)

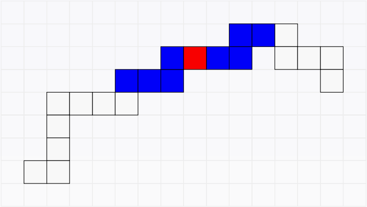
Concept based on the definition of discrete primitives

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)

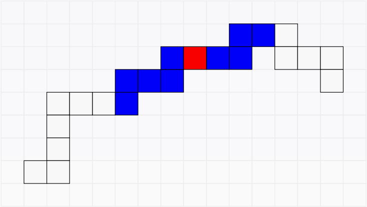
Concept based on the definition of discrete primitives

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)

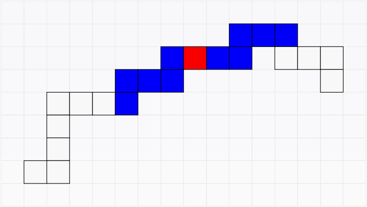
Concept based on the definition of discrete primitives

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)

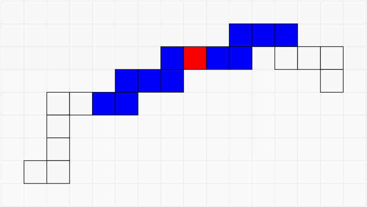
Concept based on the definition of discrete primitives

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)

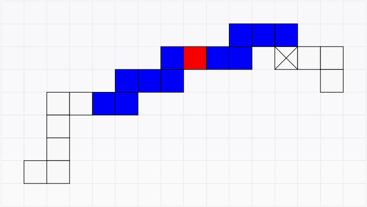
Concept based on the definition of discrete primitives

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)

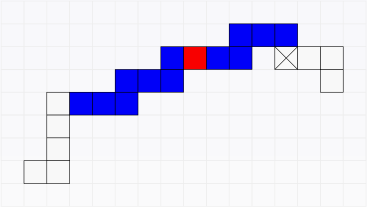
Concept based on the definition of discrete primitives

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)

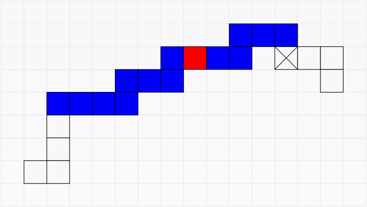
Concept based on the definition of discrete primitives

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)

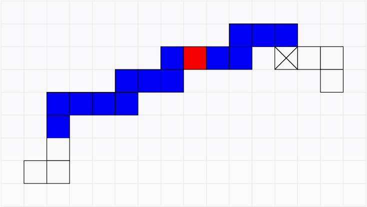
Concept based on the definition of discrete primitives

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)

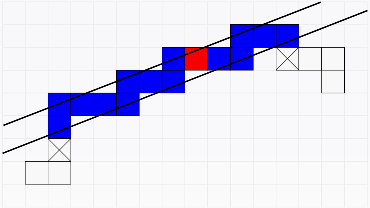
Concept based on the definition of discrete primitives

1. A standard Digital Straight Line (DSL):

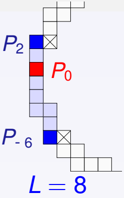
$$\{(x,y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\},$$

   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 $C \setminus M$



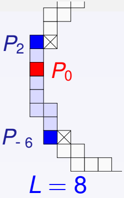
1.1 Main idea of the meaningful scale (2)



Main idea [Kerautret & Lachaud, 2012]

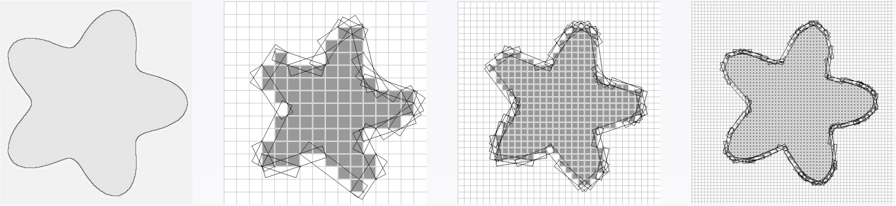
1. Exploit asymptotic properties of the Length (L) of maximal straight segments (valid on perfect shape digitizations).

1.1 Main idea of the meaningful scale (2)



Main idea [Kerautret & Lachaud, 2012]

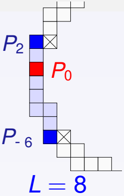
1. Exploit asymptotic properties of the Length (L) of maximal straight segments (valid on perfect shape digitizations).



X | $\mathrm{Dig}_2(X)$ | $\mathrm{Dig}_1(X)$ | $\mathrm{Dig}_{0,5}(X)$

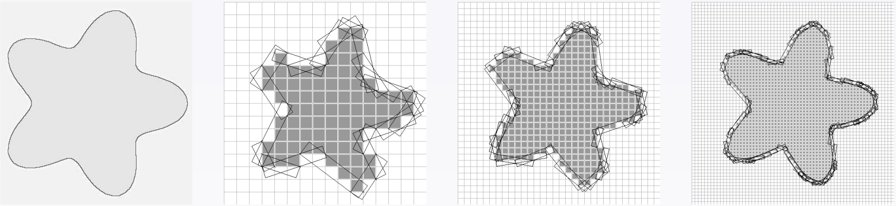
- X some simply connected compact shape of $\mathbb{R}^2$.
- $\mathrm{Dig}_h(X)$ = Gauss digitization of X with step h.

1.1 Main idea of the meaningful scale (2)



Main idea [Kerautret & Lachaud, 2012]

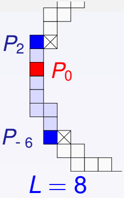
1. Exploit asymptotic properties of the Length (L) of maximal straight segments (valid on perfect shape digitizations).
2. They grow longer as h gets finer.



X $\quad$ $\mathrm{Dig}_2(X)$ $\quad$ $\mathrm{Dig}_1(X)$ $\quad$ $\mathrm{Dig}_{0,5}(X)$

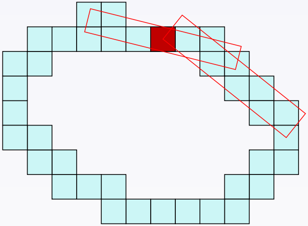
- X some simply connected compact shape of $\mathbb{R}^2$.
- $\mathrm{Dig}_h(X)$ = Gauss digitization of X with step h.

1.1 Main idea of the meaningful scale (2)

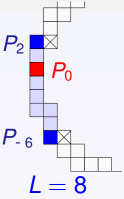


Main idea [Kerautret & Lachaud, 2012]

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.

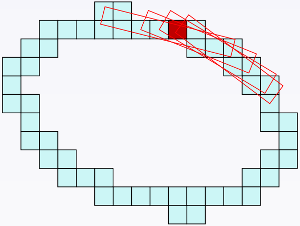


1.1 Main idea of the meaningful scale (2)

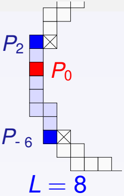


Main idea [Kerautret & Lachaud, 2012]

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.

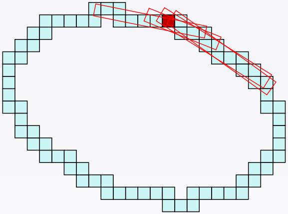


1.1 Main idea of the meaningful scale (2)

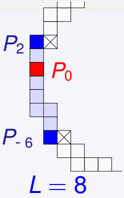


Main idea [Kerautret & Lachaud, 2012]

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.

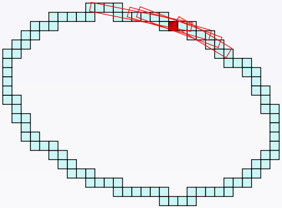


1.1 Main idea of the meaningful scale (2)

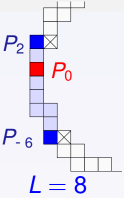


Main idea [Kerautret & Lachaud, 2012]

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.

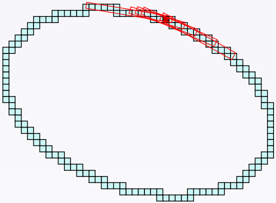


1.1 Main idea of the meaningful scale (2)



Main idea [Kerautret & Lachaud, 2012]

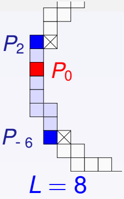
1. Exploit asymptotic properties of the Length (L) of maximal straight segments (valid on perfect shape digitizations).
2. They grow longer as h gets finer.

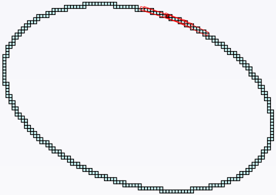


1.1 Main idea of the meaningful scale (2)

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.

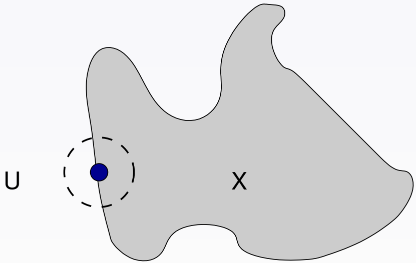




1.1 Main idea of the meaningful scale (3)

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 $p \in \partial X$,



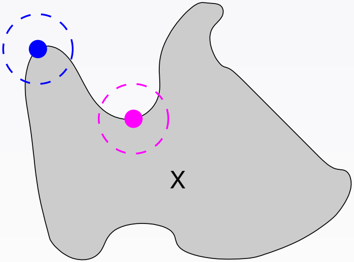
1.1 Main idea of the meaningful scale (3)

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 $\mathrm{Dig}_h(X)$ which cover p,

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

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



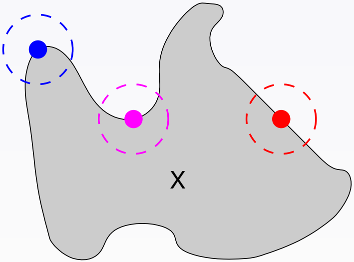
1.1 Main idea of the meaningful scale (3)

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 $\mathrm{Dig}_h(X)$ which cover p,

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

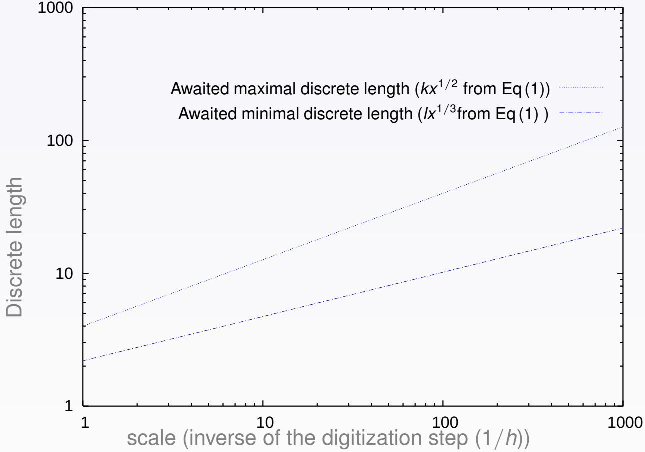
$$\partial X \cap U \text{ has null curvature, then } \Omega(1/h^{1}) \leq \quad L_j^h \quad \leq O(1/h^{1}) \qquad (2)$$



1.1 Main idea of the meaningful scale (4)

Previous theorem on asymptotic behaviour

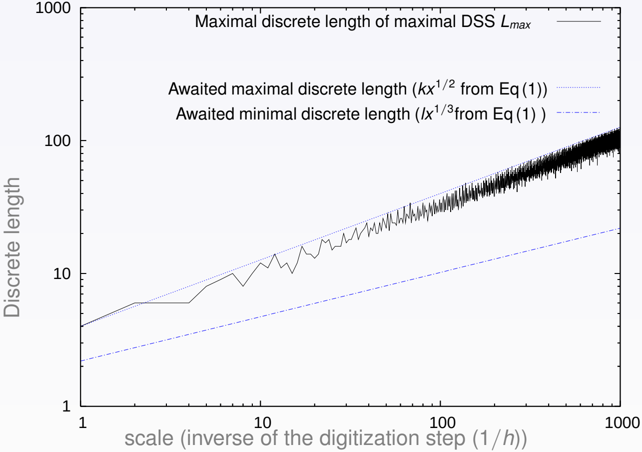
- Proved in [Lachaud 06].
- Can be checked experimentally:



1.1 Main idea of the meaningful scale (4)

Previous theorem on asymptotic behaviour

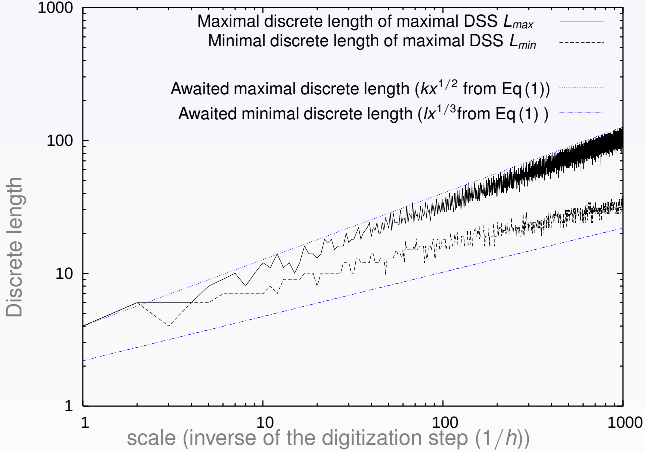
- Proved in [Lachaud 06].
- Can be checked experimentally:



1.1 Main idea of the meaningful scale (4)

Previous theorem on asymptotic behaviour

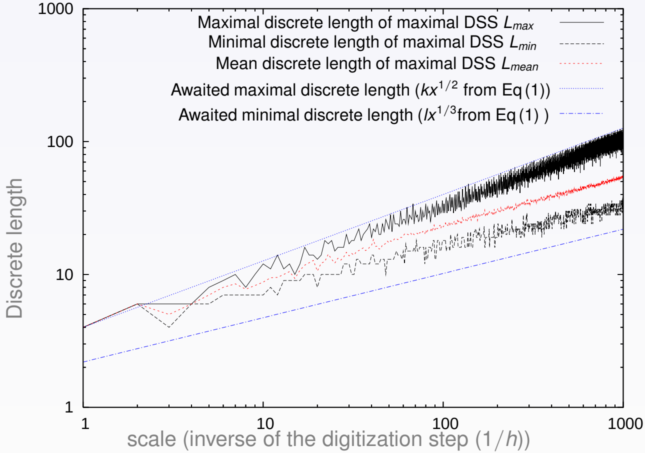
- Proved in [Lachaud 06].
- Can be checked experimentally:



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

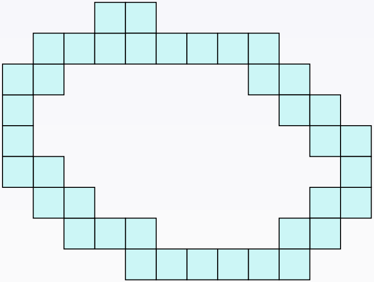
Some criticisms:

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

Asymptotic estimation with multiresolution

Some criticisms:

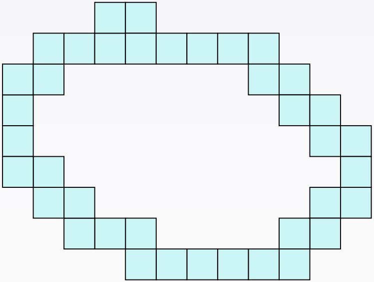
- Preceding theorem = digital properties of perfect shape.
- A technique for noise detection ?
- But: input is only a digital contour.



Asymptotic estimation with multiresolution

Some criticisms:

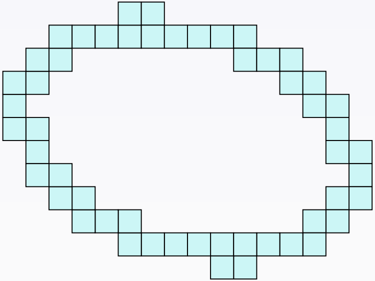
- Preceding theorem = digital properties of perfect shape.
- A technique for noise detection ?
- But: input is only a digital contour.
- finer and finer digitizations are not accessible.



Asymptotic estimation with multiresolution

Some criticisms:

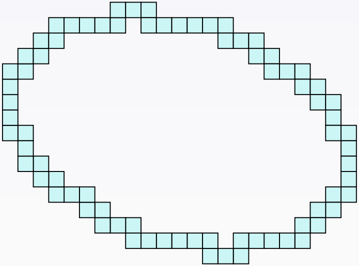
- Preceding theorem = digital properties of perfect shape.
- A technique for noise detection ?
- But: input is only a digital contour.
- finer and finer digitizations are not accessible.



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

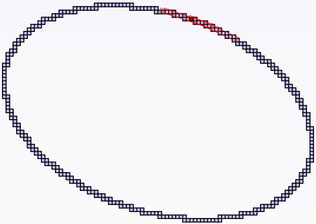
Asymptotic estimation with multiresolution

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

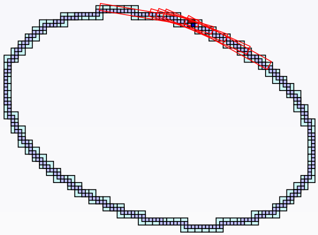
Asymptotic estimation with multiresolution

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

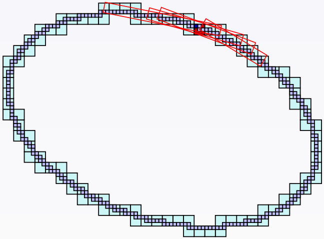
Asymptotic estimation with multiresolution

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

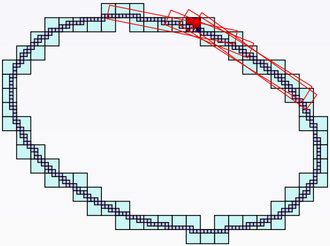
Asymptotic estimation with multiresolution

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

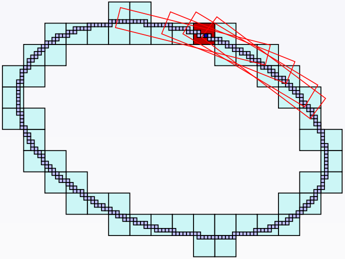
Asymptotic estimation with multiresolution

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

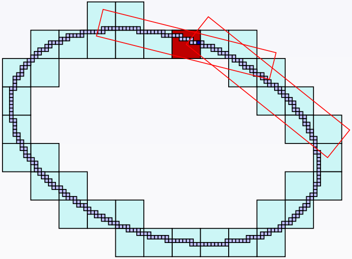
Asymptotic estimation with multiresolution

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

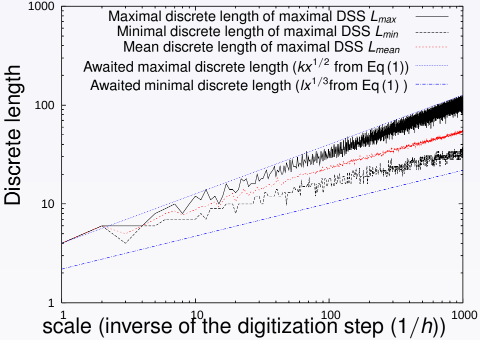
- coarser and coarser sampling of the contour
- measure digital lengths of maximal segments
- should also follow approximately asymptotic behavior



grid size 8

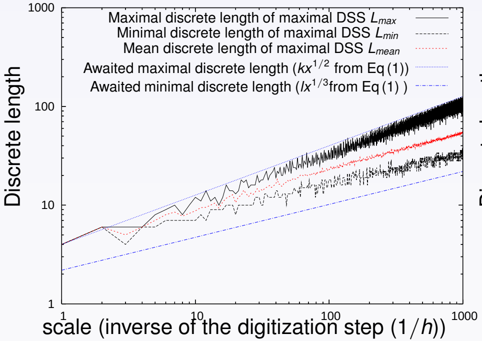
Preliminary experiments about reverse asymptotic behavior

Experiments of asymptotic behaviour

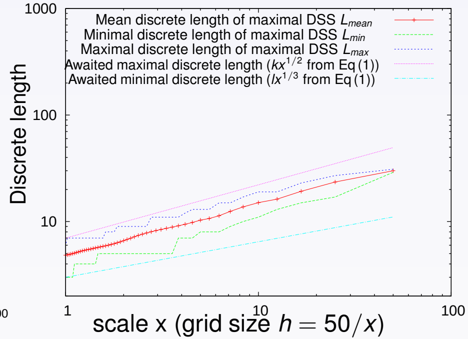


Preliminary experiments about reverse asymptotic behavior

Experiments of asymptotic behaviour



Experiments from subsampling



Subsampling a digital contour

Subsampling process:

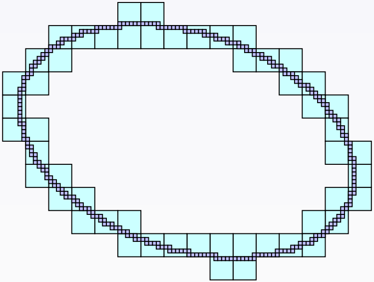
- Not spatial but applied to the sequence of points (x_j, y_j) of the contour C.
- All subsampled contours $\phi_i^{(x_0,y_0)}$ are computed for all shifts (x_0, y_0):
$(X_j, Y_j) = ((x_j - x_0) \div i, (y_j - y_0) \div i)$

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 $\phi_6^{0,0}$

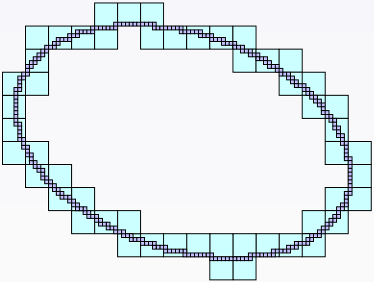


Subsampling a digital contour

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 $\phi_6^{1,0}$

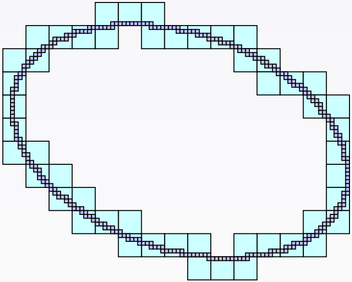


Subsampling a digital contour

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 $\phi_6^{2,0}$

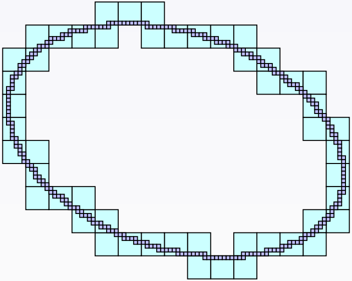


Subsampling a digital contour

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 $\phi_6^{3,0}$

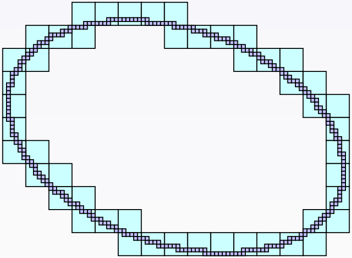


Subsampling a digital contour

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 $\phi_6^{3,1}$

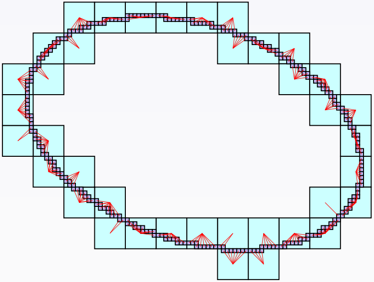


Subsampling a digital contour

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

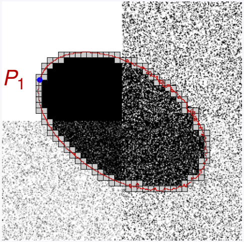
Multiscale profile $\mathcal{P}_n$ of a point P on a digital contour:

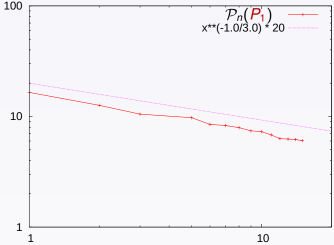
- $\mathcal{P}_n(P)$ = sequence $(\log i, \log(E(L^{h_i})))_{i=1..n}$, where:
  - E is the average operator.
  - L^{h_i} are the digital lengths of the maximal segments covering P for all subsampling $i \times i$.

1.2 Meaningful Scale Profile and Noise Detection

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

- $\mathcal{P}_n(P)$ = 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 $i \times i$.

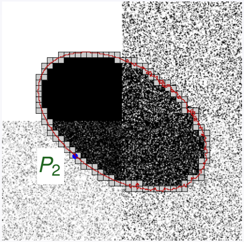


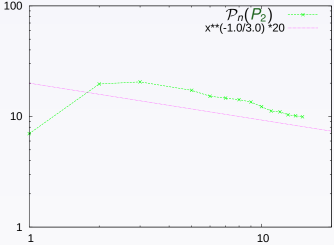


1.2 Meaningful Scale Profile and Noise Detection

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

- $\mathcal{P}_n(P)$ = 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 $i \times i$.

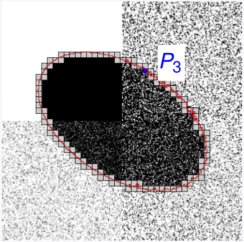


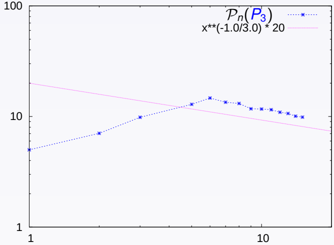


1.2 Meaningful Scale Profile and Noise Detection

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

- $\mathcal{P}_n(P)$ = 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 $i \times i$.

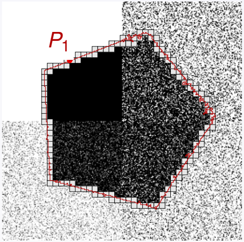


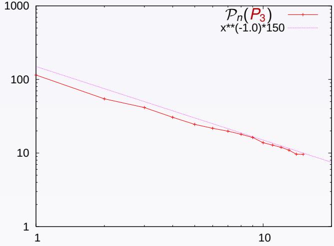


1.2 Meaningful Scale Profile and Noise Detection

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

- $\mathcal{P}_n(P)$ = 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 $i \times i$.

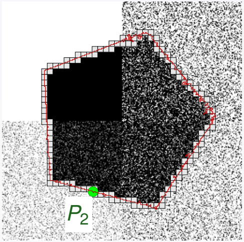


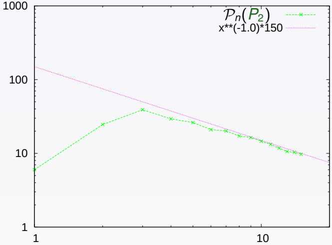


1.2 Meaningful Scale Profile and Noise Detection

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

- $\mathcal{P}_n(P)$ = 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 $i \times i$.

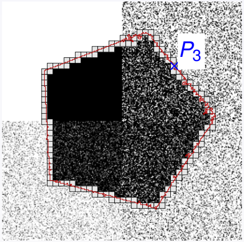


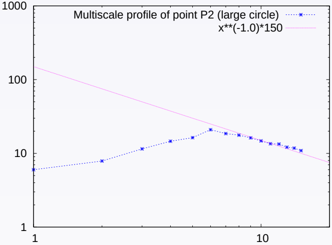


1.2 Meaningful Scale Profile and Noise Detection

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

- $\mathcal{P}_n(P)$ = 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 $i \times i$.

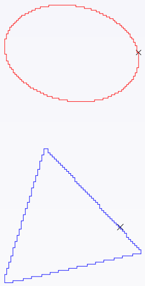


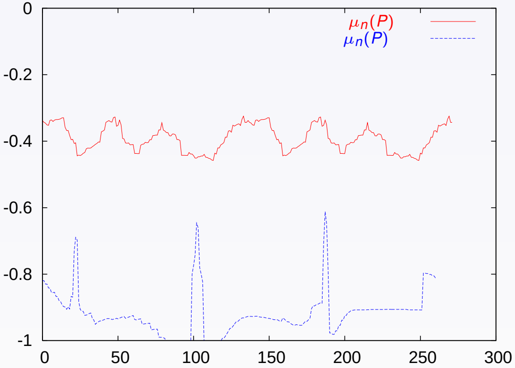


Local geometric evaluation with multi-scale profile

Ideal multiscale criterion: $\mu_n(P)$

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

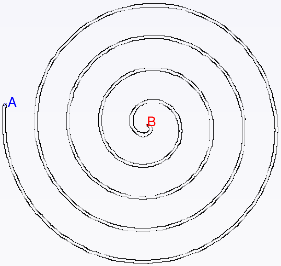


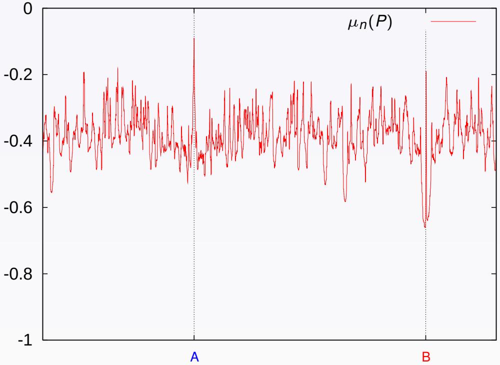


Local geometric evaluation with multi-scale profile

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

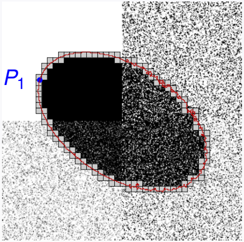
Meaningful scale:

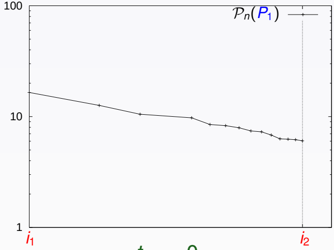
A meaningful scale of a multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$ is the pair (i_1, i_2) $1 \leq i_1 \leq i_2 \leq n$ such that for all i, $i_1 \leq i < i_2$,

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

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

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





$t_m = 0$

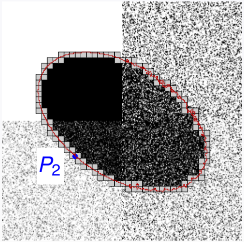
Local meaningful scale and noise detection

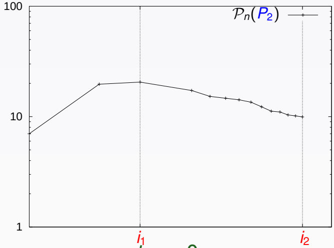
Meaningful scale:

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.





$t_m = 0$

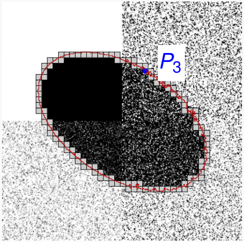
Local meaningful scale and noise detection

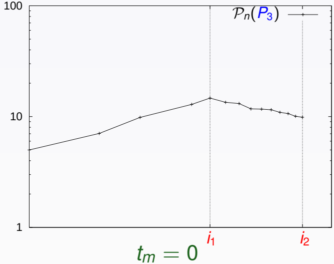
Meaningful scale:

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:

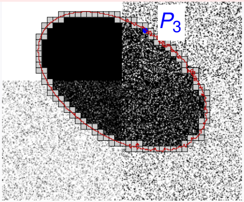
A meaningful scale of a multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$ is the pair (i_1, i_2) $1 \leq i_1 \leq i_2 \leq n$ such that for all i, $i_1 \leq i < i_2$,

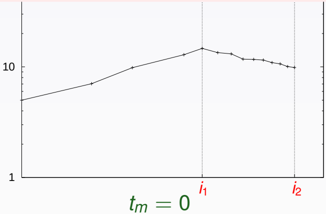
$$\frac{Y_{i+1} - Y_i}{X_{i+1} - X_i} \leq t_m,$$

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

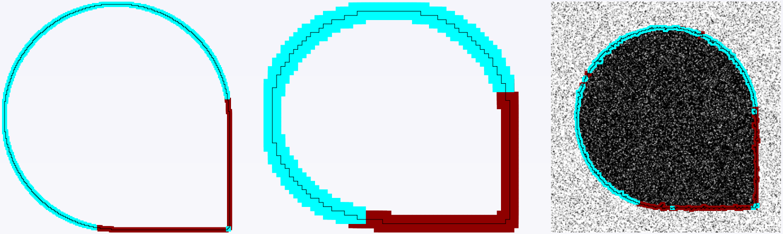
If (i_1, i_2) is the first meaningful scale at point P, the noise level at point P is $i_1 - 1$.





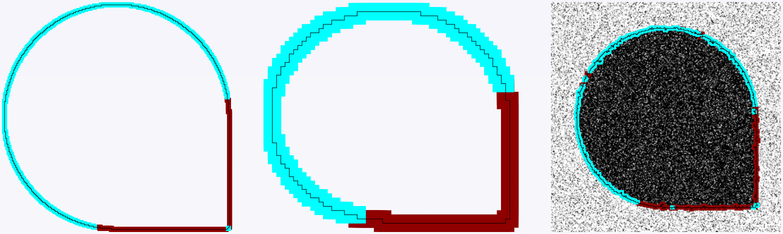
3. Experiments: flat and curved areas detection

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

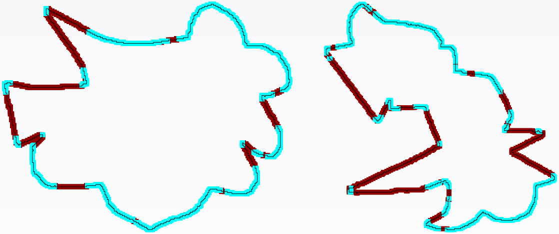


3. Experiments: flat and curved areas detection

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

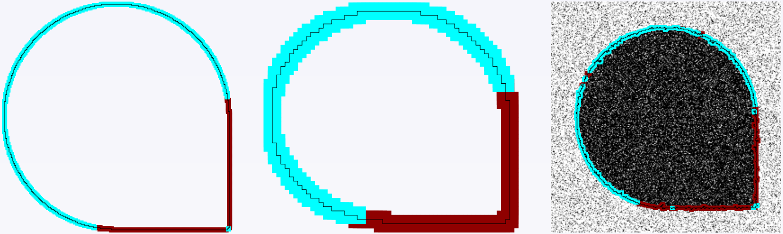


Test on real shapes:

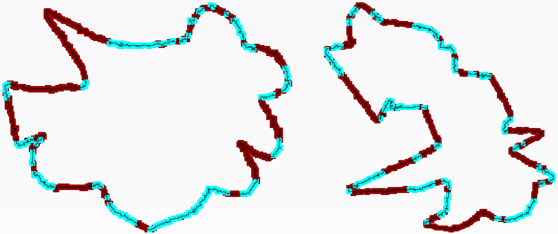


3. Experiments: flat and curved areas detection

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

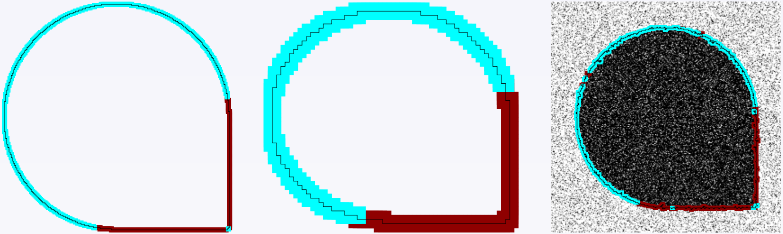


Test on damaged versions of these shapes:

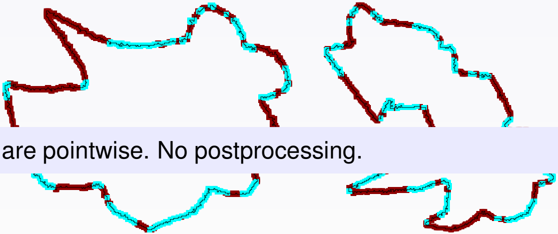


3. Experiments: flat and curved areas detection

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



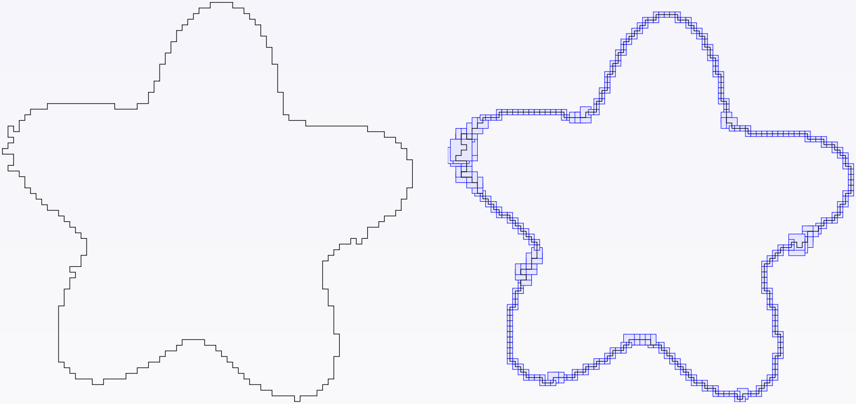
Test on damaged versions of these shapes:



All computations are pointwise. No postprocessing.

Experiments: local noise level detection

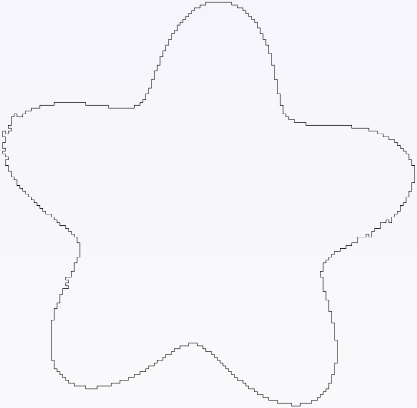
Flower with local noise

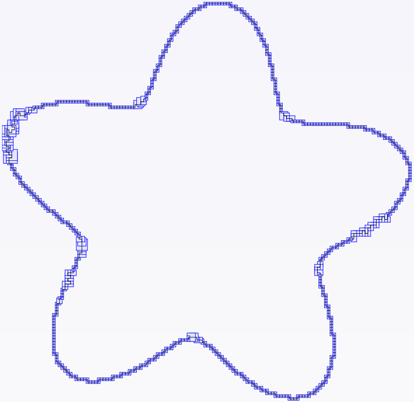


Local noise on resolution $R0$

Experiments: local noise level detection

Flower with local noise

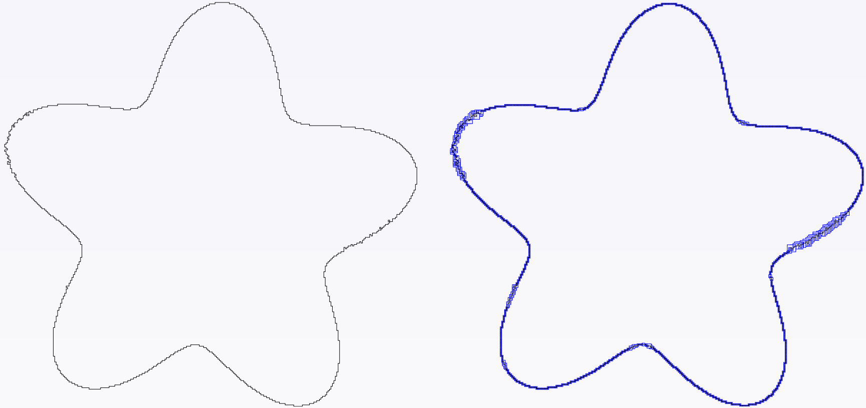




Local noise on resolution $R1$

Experiments: local noise level detection

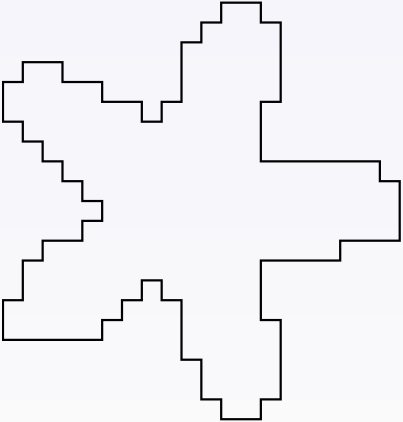
Flower with local noise

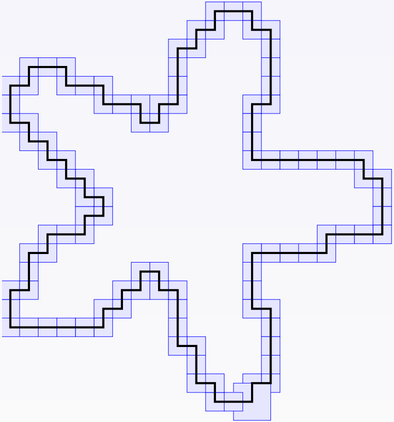


Local noise on resolution $R2$

Experiments: local noise level detection

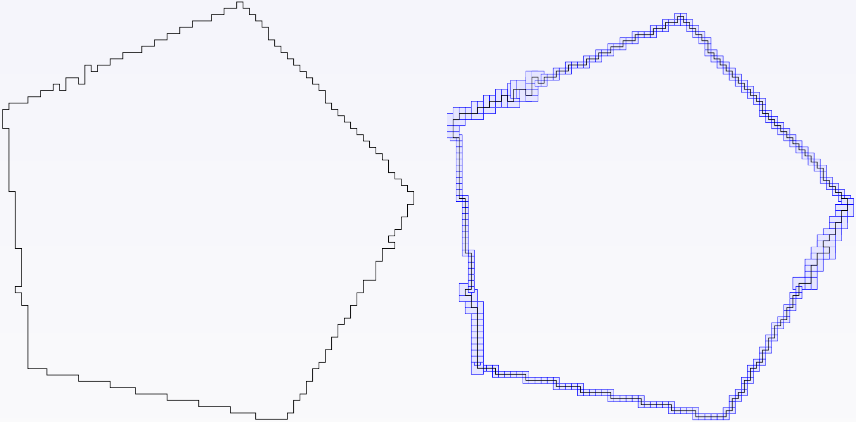
Tiny flower without noise





Local noise detection

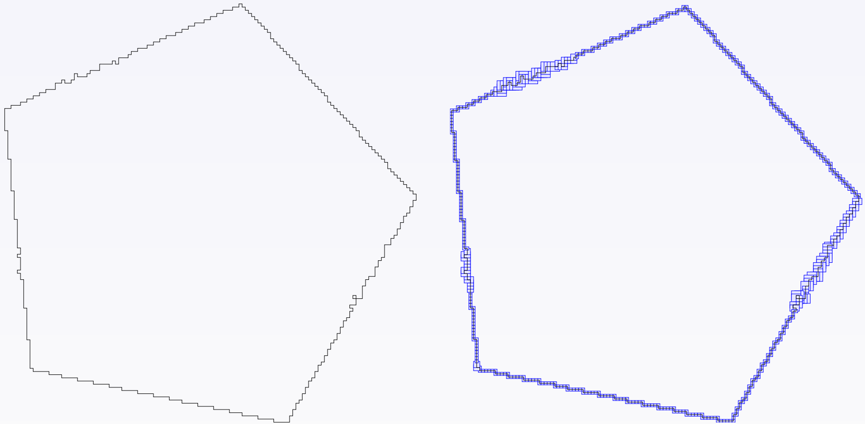
Polygon with local noise



Local noise on resolution $R0$

Local noise detection

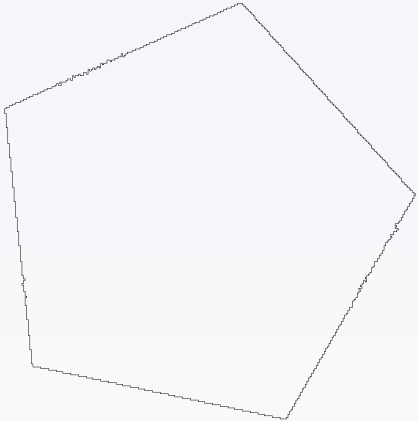
Polygon with local noise

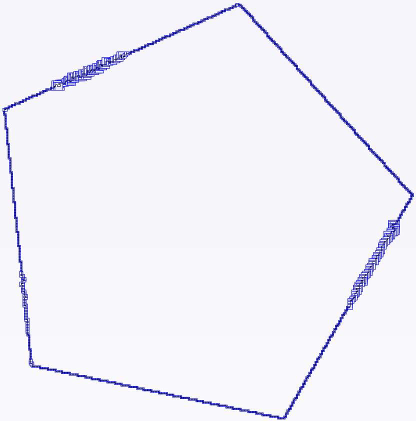


Local noise on resolution $R1$

Local noise detection

Polygon with local noise

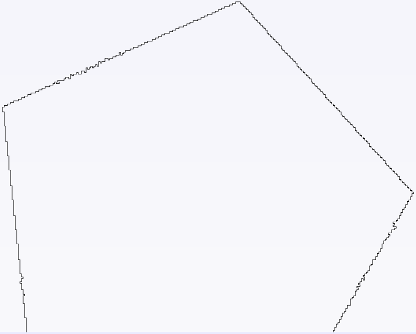


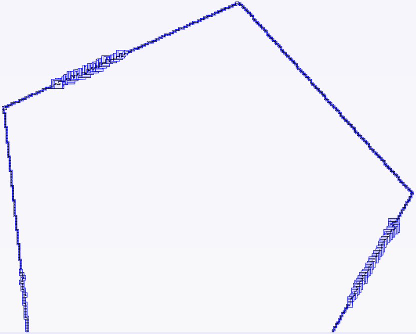


Local noise on resolution $R2$

Local noise detection

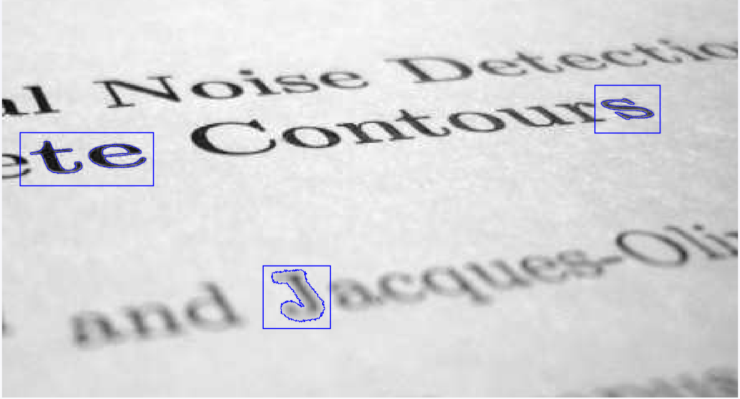
Polygon with local noise



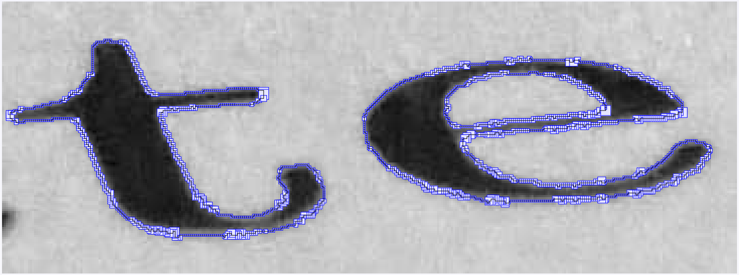


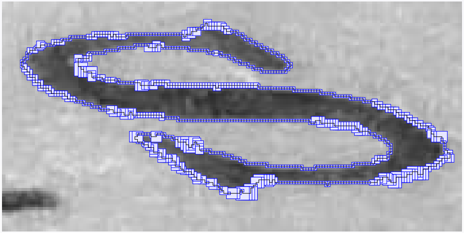
- Accuracy of noise detection independent of shape geometry, independent of shape resolution.
- Only one parameter : maximum level of subsampling (always 10 here).

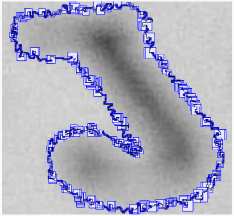
Noise detection on real images



Noise detection on real images

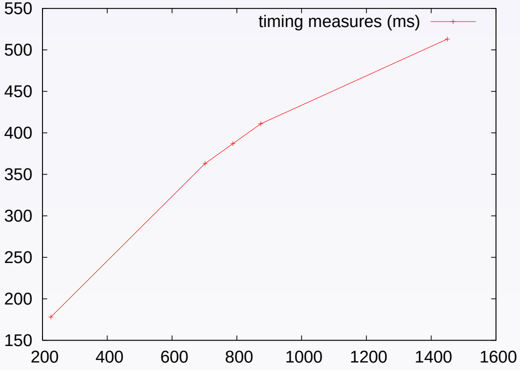






Timing measures

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

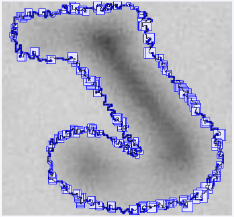


Obtained on a 2.4Ghz *Intel Core Duo*.

1.3. New applications: standard scale

Limits of the meaningful scale:

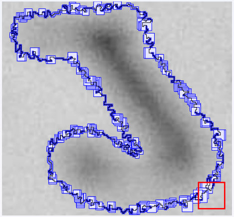
- The first significant scale.
- Not necessary the awaited scale.

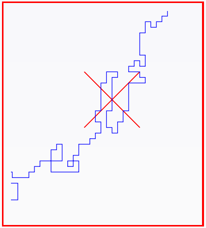


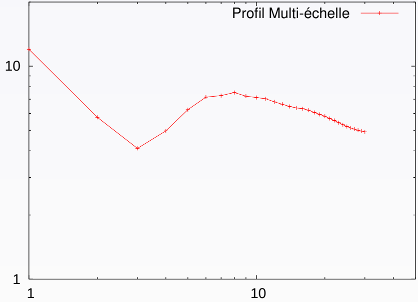
1.3. New applications: standard scale

Limits of the meaningful scale:

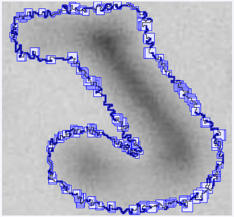
- The first significant scale.
- Not necessary the awaited scale.







1.3. New applications: standard scale



Limits of the meaningful scale:

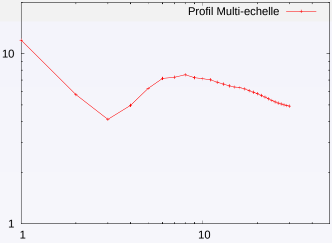
- The first significant scale.
- Not necessary the awaited scale.

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 $(i_1, i_2), (i_2, i_3), ..., (i_k, i_{k+1})$
- Each of one is associated to a linear regresssion starting from le scale i_{k+1} and which is not true for $i_k - 1$.

1.3. New applications: standard scale



Limits of the meaningful scale:

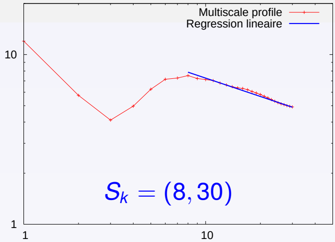
- The first significant scale.
- Not necessary the awaited scale.

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 $(i_1, i_2), (i_2, i_3), ..., (i_k, i_{k+1})$
- Each of one is associated to a linear regresssion starting from le scale i_{k+1} and which is not true for $i_k - 1$.

1.3. New applications: standard scale



Limits of the meaningful scale:

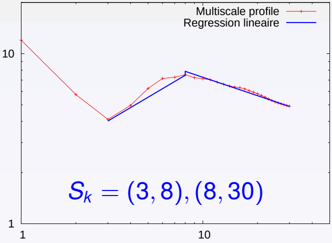
- The first significant scale.
- Not necessary the awaited scale.

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 $(i_1, i_2), (i_2, i_3), ..., (i_k, i_{k+1})$
- Each of one is associated to a linear regresssion starting from le scale i_{k+1} and which is not true for $i_k - 1$.

1.3. New applications: standard scale



Limits of the meaningful scale:

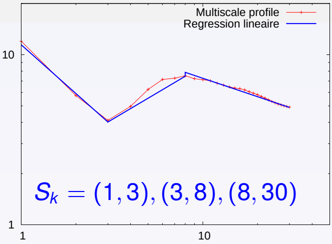
- The first significant scale.
- Not necessary the awaited scale.

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 $(i_1, i_2), (i_2, i_3), ..., (i_k, i_{k+1})$
- Each of one is associated to a linear regresssion starting from le scale i_{k+1} and which is not true for $i_k - 1$.

1.3. New applications: standard scale



Limits of the meaningful scale:

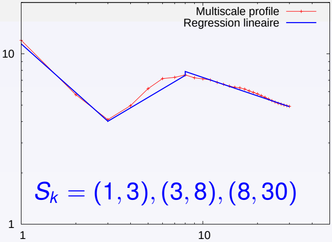
- The first significant scale.
- Not necessary the awaited scale.

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 $(i_1, i_2), (i_2, i_3), ..., (i_k, i_{k+1})$
- Each of one is associated to a linear regresssion starting from le 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 necessary the awaited scale.

Multicale profile decomposition:

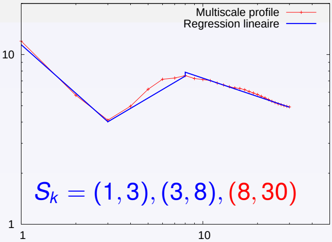
Decomposition of the multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$:

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

Standard scale

If we denote θ_{i_j} 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:
$\forall p, l \leq p < m, \theta_{i_p} < 0 \; and \; \theta_{i_{l-1}} \geq 0$

1.3. New applications: standard scale



Limits of the meaningful scale:

- The first significant scale.
- Not necessary the awaited scale.

Multicale profile decomposition:

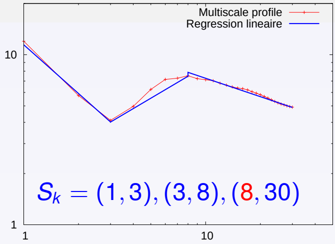
Decomposition of the multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$:

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

Standard scale

If we denote θ_{i_j} 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:
$\forall p, l \leq p < m, \theta_{i_p} < 0 \text{ and } \theta_{i_{l-1}} \geq 0$

1.3. New applications: standard scale



Limits of the meaningful scale:

- The first significant scale.
- Not necessary the awaited scale.

Multicale profile decomposition:

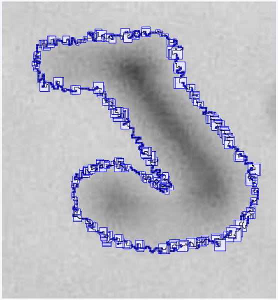
Decomposition of the multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$:

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

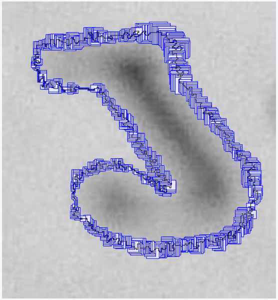
Standard scale

If we denote θ_{i_j} 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:
$\forall p, l \leq p < m, \theta_{i_p} < 0 \; and \; \theta_{i_{l-1}} \geq 0$

1.3. New applications: standard scale (results)



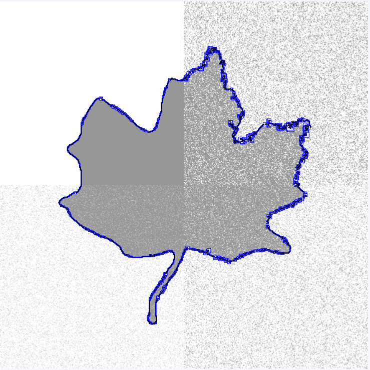
(a) *Meaningful Scale*



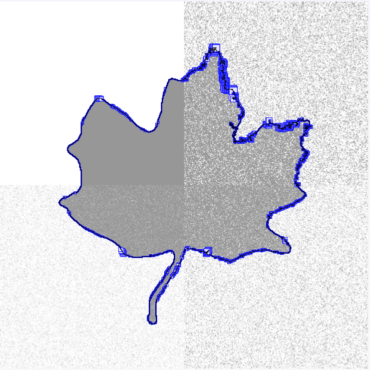
(b) *Standard Scale*

(c) median smoothing (E Std)

1.3. New applications: standard scale (results)

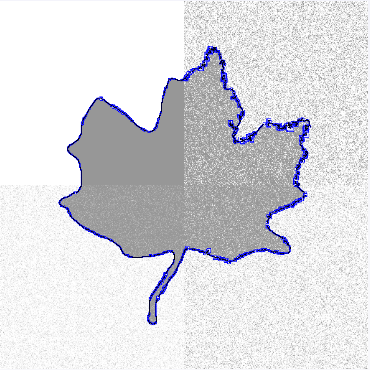


(a) *Meaningful Scale*
(max scale =30)

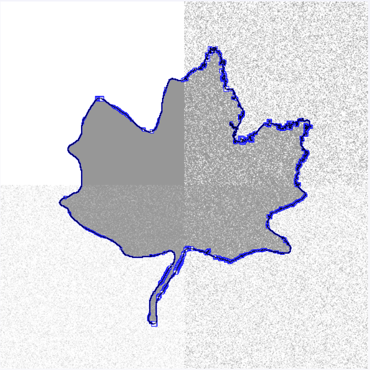


(b) *Standard Scale*
(max scale =30)

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

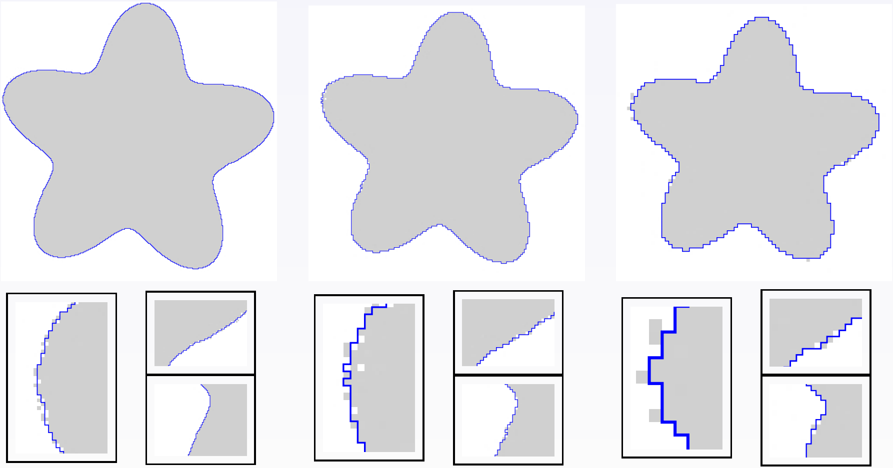
Noise removal on digital contours:

- 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:

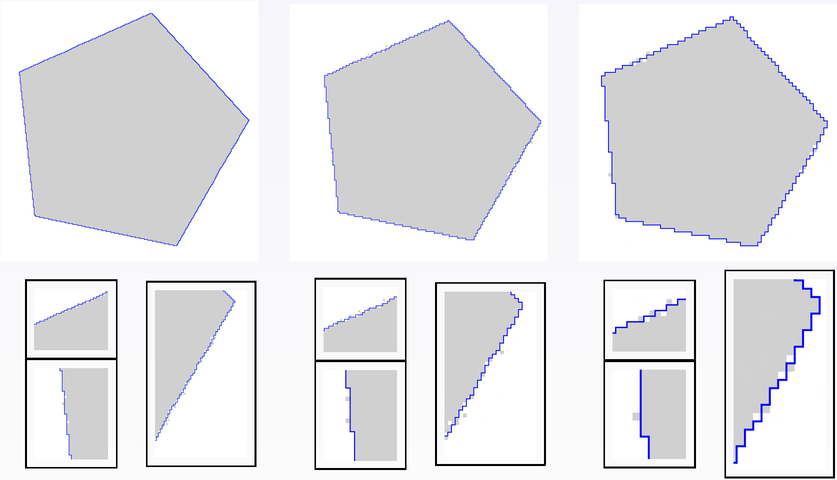
- Application of simple median filter.
- Filter size defined according to the noise level detected: $2\nu(P) + 1$



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

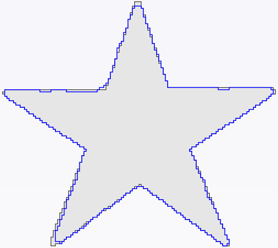
Noise removal on digital contours:

- 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: $2\nu(P) + 1$

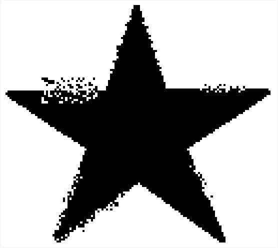


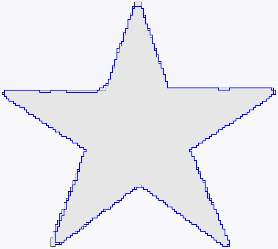
median filter
(meaningful scales)

1.3. New applications: noise removal

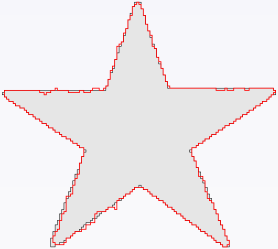
Noise removal on digital contours:

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





median filter
(meaningful scales)

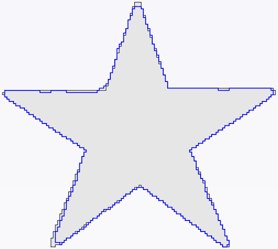


median filter size=2

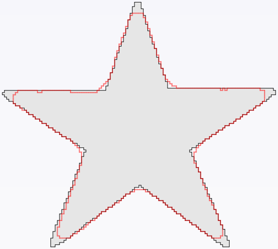
1.3. New applications: noise removal

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)

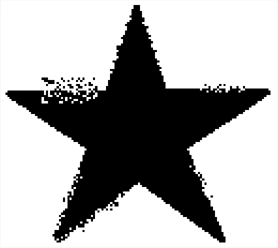


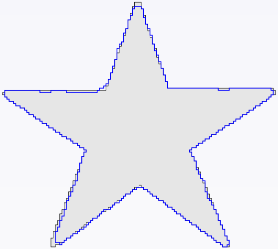
median filter size=6

1.3. New applications: noise removal

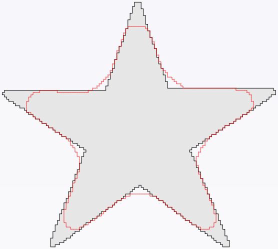
Noise removal on digital contours:

- 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

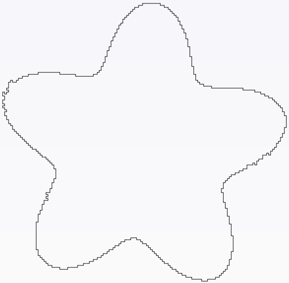
Application to geometric estimator:

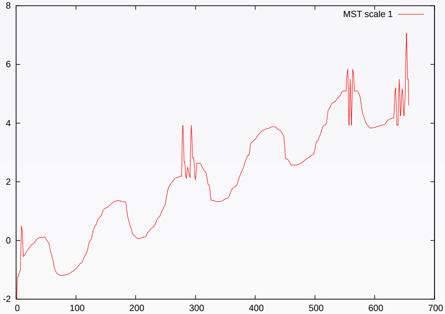
- Normal estimator $\lambda - MST$ [Lachaud07 etal].
- Principle: for each contour point C_k, we associate the normal computed from the meaningful scale level: $\phi^{0,0}_{\nu(C_k)+1}(C_k)$.

1.3. New applications: noise removal

Application to geometric estimator:

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

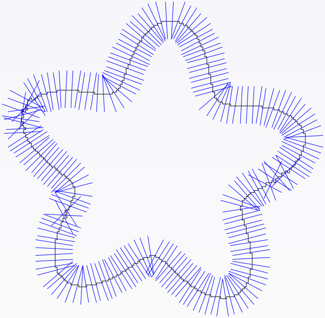


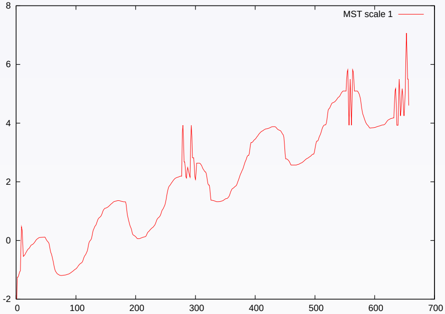


1.3. New applications: noise removal

Application to geometric estimator:

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

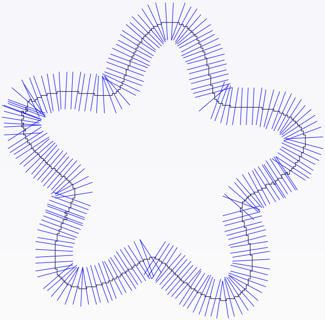


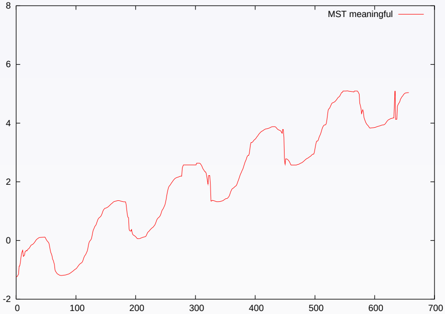


1.3. New applications: noise removal

Application to geometric estimator:

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

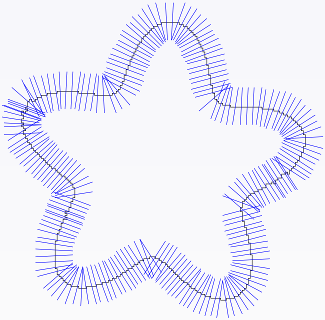


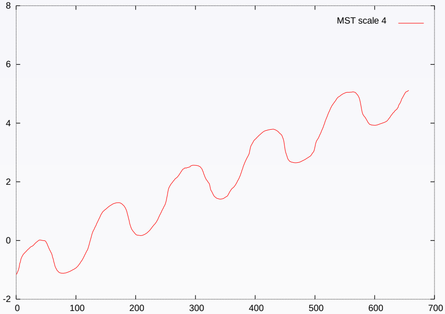


1.3. New applications: noise removal

Application to geometric estimator:

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

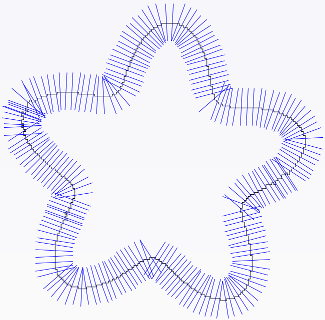


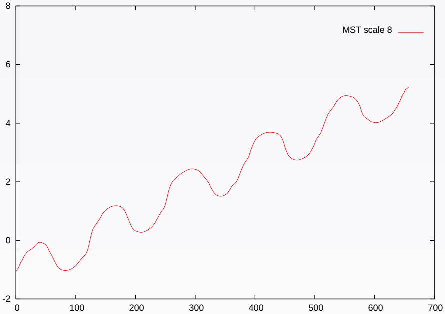


1.3. New applications: noise removal

Application to geometric estimator:

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

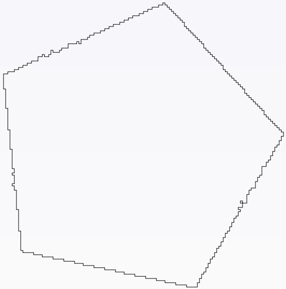


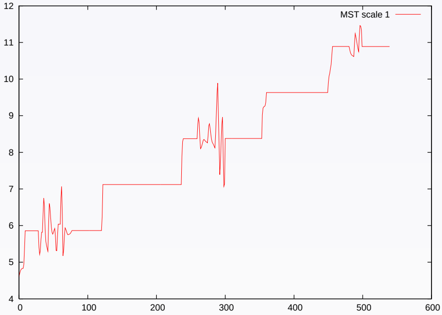


1.3. New applications: noise removal

Application to geometric estimator:

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

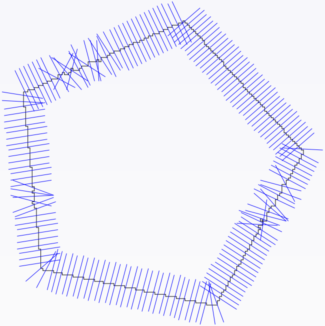


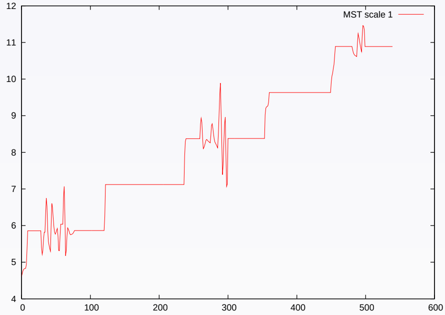


1.3. New applications: noise removal

Application to geometric estimator:

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

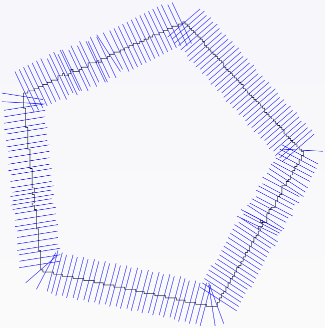


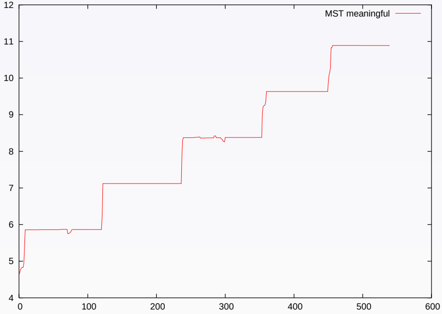


1.3. New applications: noise removal

Application to geometric estimator:

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

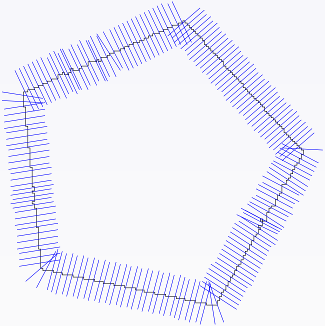


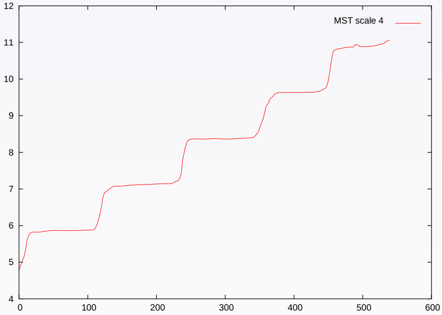


1.3. New applications: noise removal

Application to geometric estimator:

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

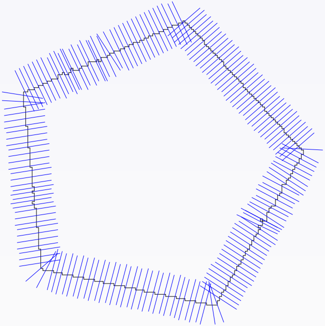


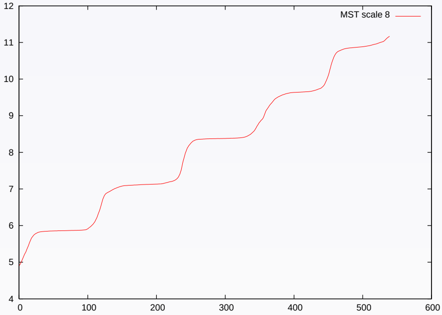


1.3. New applications: noise removal

Application to geometric estimator:

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

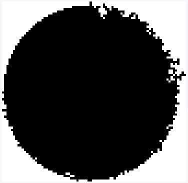




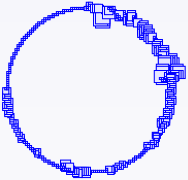
1.3. Other application examples

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

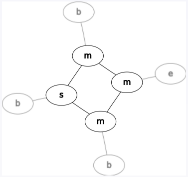
- Use the meaningful boxes as input to irregular isothetic grid.
- Reeb graph.



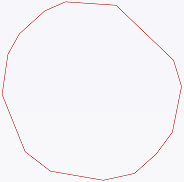
source



meaningful scale



reeb graph



polygonal representation

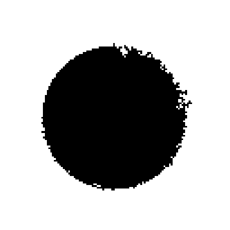
1.3. Other application examples

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

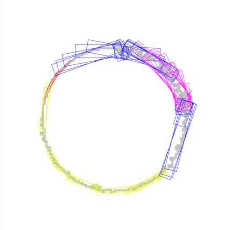
- Use the meaningful boxes as input to irregular isothetic grid.
- Reeb graph.

Adaptative tangential cover [Ngo et al., 15]

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



source

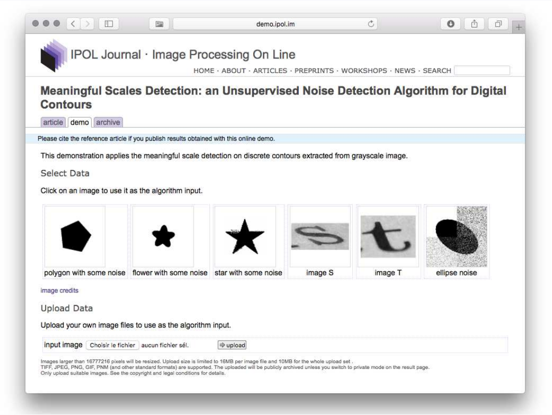


adaptative tangential cover

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).



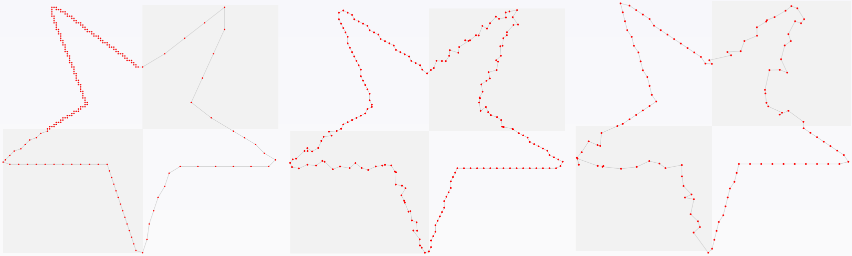
1.3 Online demonstration and limitations

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

2.1 Extension to the Meaningful Thickness

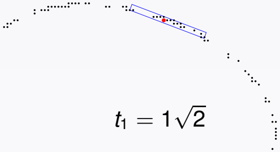
Main idea

- 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

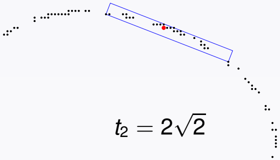


$t_1 = 1\sqrt{2}$

2.1 Extension to the Meaningful Thickness

Main idea

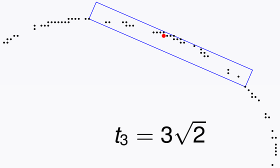
- Use another primitive to process set of points of the Euclidean plane.
- alpha-thick segments [Debled-Rennesson et al., 2006, Faure et al., 2009]:
  - Defined with a thick parameter: t



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

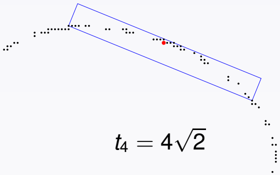


$t_3 = 3\sqrt{2}$

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

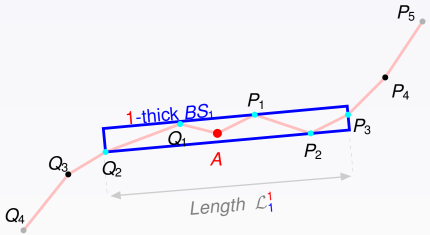


$t_4 = 4\sqrt{2}$

2.1 Extension to the Meaningful Thickness

Main idea

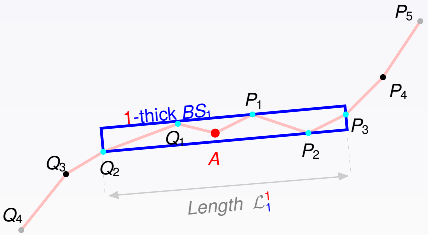
- Use another primitive to process set of points of the Euclidean plane.
- alpha-thick segments [Debled-Rennesson et al., 2006, Faure et al., 2009]:
  - Defined with a thick parameter: t
  - maximal isothetic thickness of the convex hull.
    $\Rightarrow (P_1, Q_1, Q_2, P_2, P_3)$



2.1 Extension to the Meaningful Thickness

Main idea

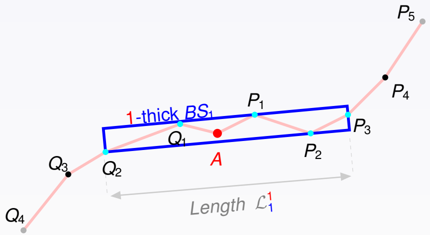
- Use another primitive to process set of points of the Euclidean plane.
- alpha-thick segments [Debled-Rennesson et al., 2006, Faure et al., 2009]:
  - Defined with a thick parameter: t
  - maximal isothetic thickness of the convex hull.
    $\Rightarrow (P_1, Q_1, Q_2, P_2, P_3)$
- **Maximal** alpha-thick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

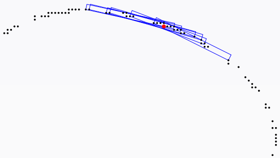
	expected slope	
plot	curved part	flat part
$(\log(t_i), \log(\max_j \mathcal{L}_j^{t_i}/t_i))$	$\approx -\frac{1}{2}$	≈ -1
$(\log(t_i), \log(\min_j \mathcal{L}_j^{t_i}/t_i))$	$\approx -\frac{1}{3}$	≈ -1

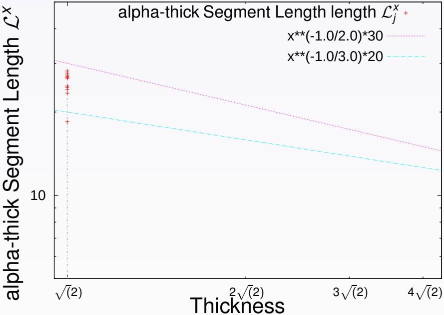
Thickness asymptotic behaviour

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:

	expected slope	
plot	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



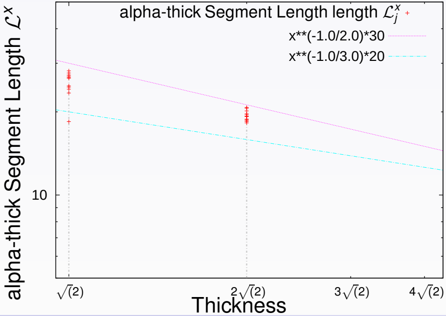


Thickness asymptotic behaviour

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:

	expected slope	
plot	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

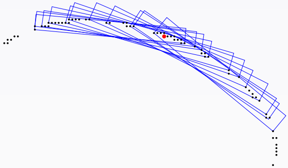


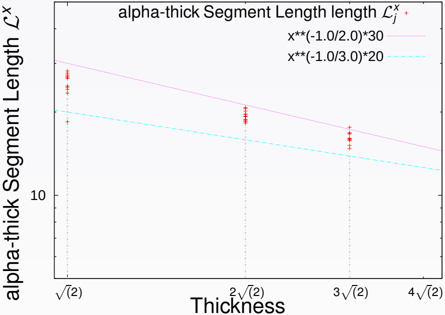
Thickness asymptotic behaviour

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:

	expected slope	
plot	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



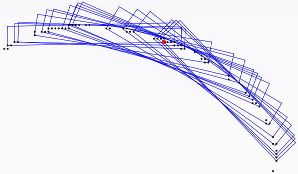


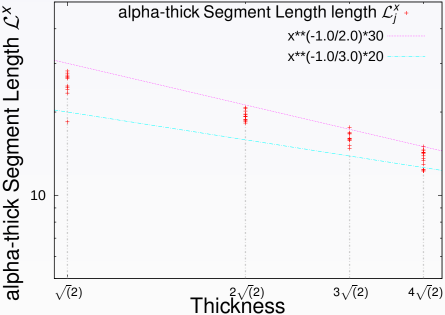
Thickness asymptotic behaviour

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:

	expected slope	
plot	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



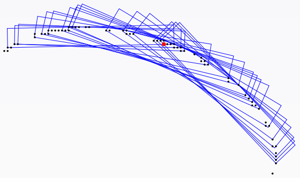


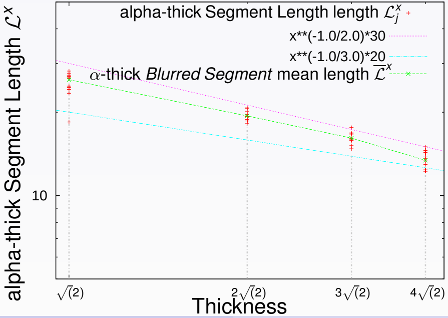
Thickness asymptotic behaviour

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:

	expected slope	
plot	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



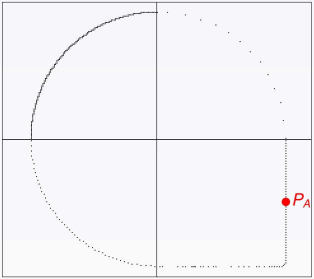


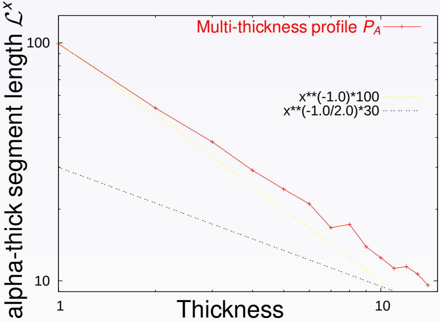
2.2 Meaningful Thickness profile

Multi-thickness Profile

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

Example obtained from a shape with different samplings:



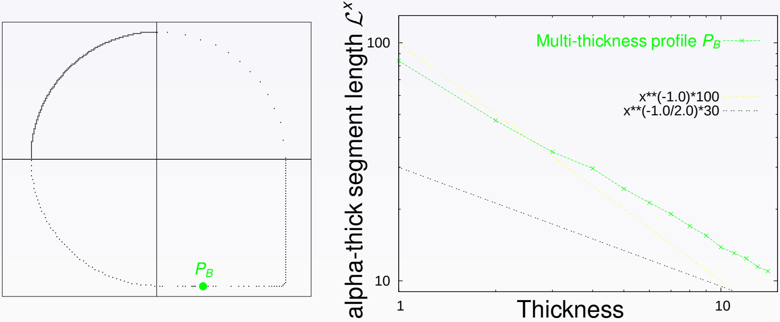


2.2 Meaningful Thickness profile

Multi-thickness Profile

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

Example obtained from a shape with different samplings:

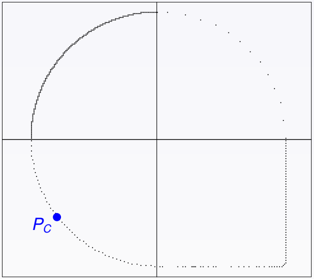


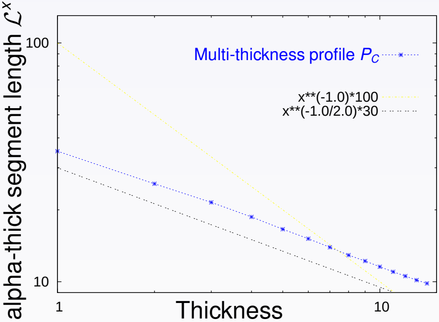
2.2 Meaningful Thickness profile

Multi-thickness Profile

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

Example obtained from a shape with different samplings:



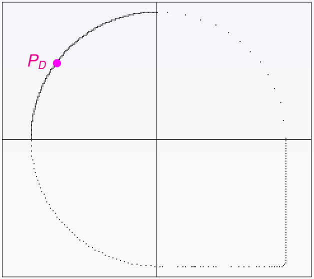


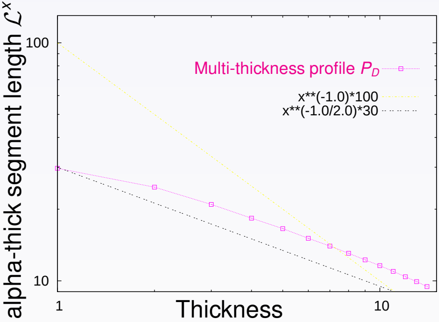
2.2 Meaningful Thickness profile

Multi-thickness Profile

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

Example obtained from a shape with different samplings:



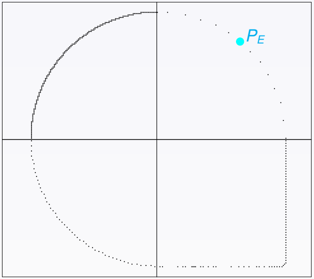


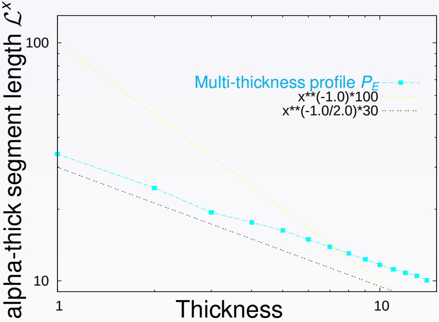
2.2 Meaningful Thickness profile

Multi-thickness Profile

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

Example obtained from a shape with different samplings:



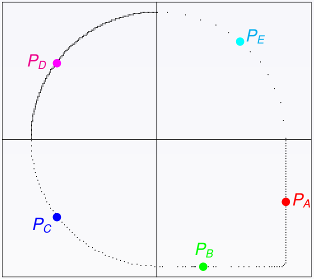


2.2 Meaningful Thickness profile

Multi-thickness Profile

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

Example obtained from a shape with different samplings:



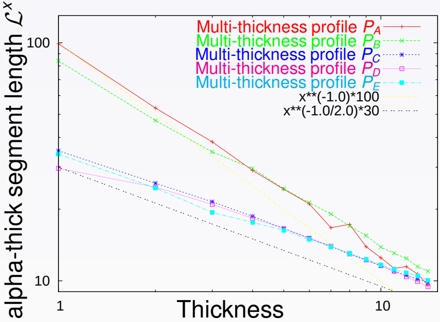
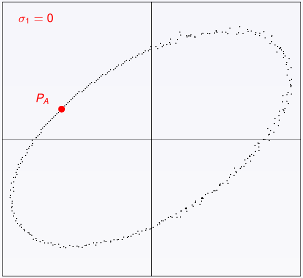


Illustration of multi-thickness profile (2)

Example obtained by adding noise:



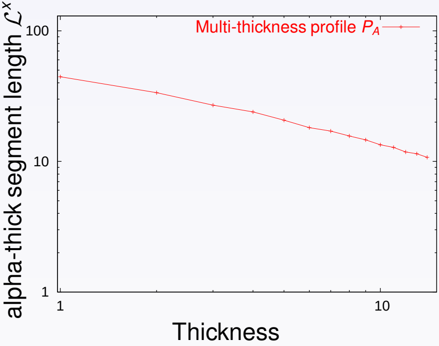


Illustration of multi-thickness profile (2)

Example obtained by adding noise:

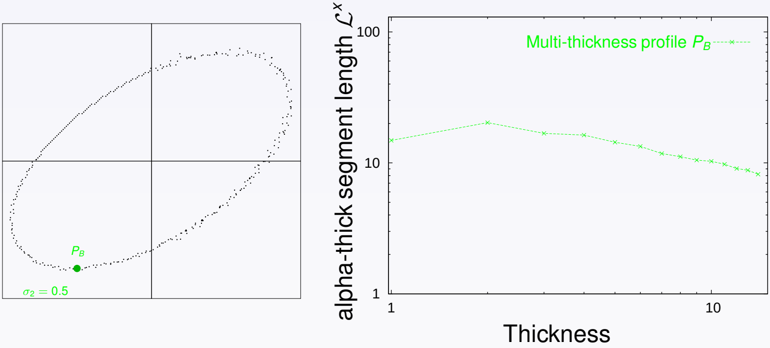
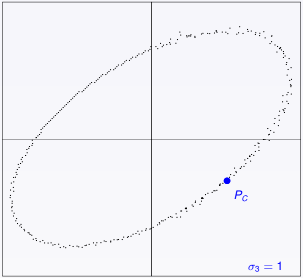


Illustration of multi-thickness profile (2)

Example obtained by adding noise:



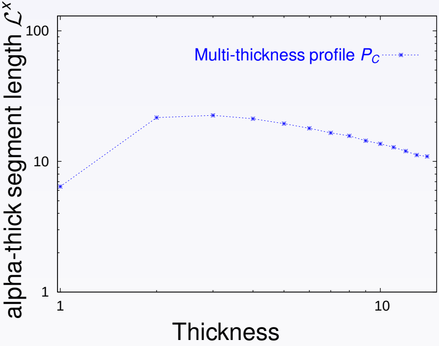
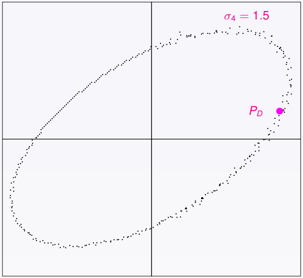


Illustration of multi-thickness profile (2)

Example obtained by adding noise:



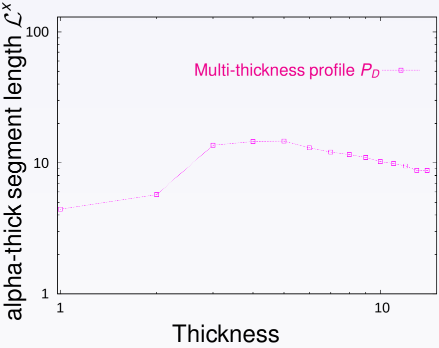
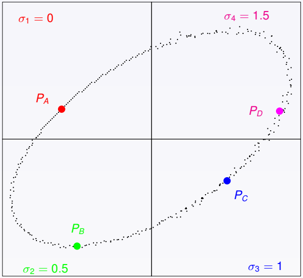


Illustration of multi-thickness profile (2)

Example obtained by adding noise:



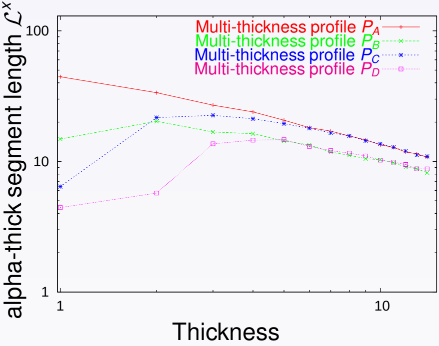
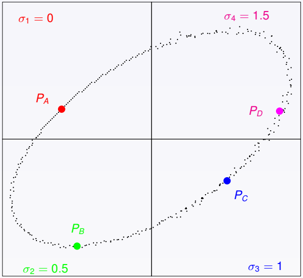
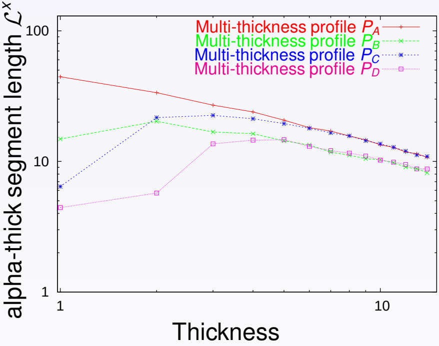


Illustration of multi-thickness profile (2)

Example obtained by adding noise:





$\Rightarrow$ Define a noise threshold T_m to discriminate the curved and noisy zone.

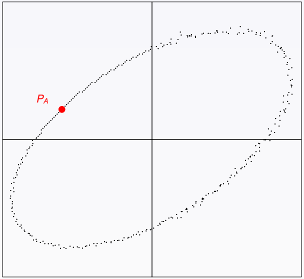
Noise detection and Meaningful Thickness

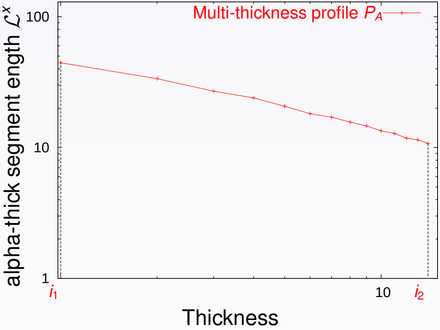
Meaningful thickness

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.





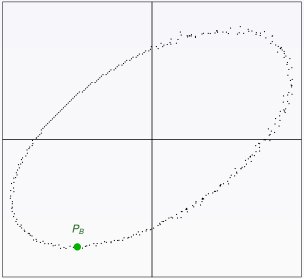
Noise detection and Meaningful Thickness

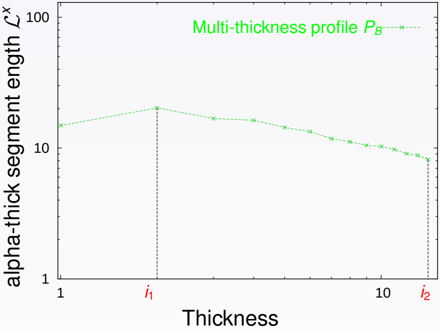
Meaningful thickness

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.





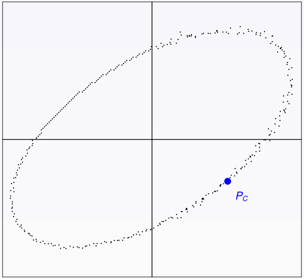
Noise detection and Meaningful Thickness

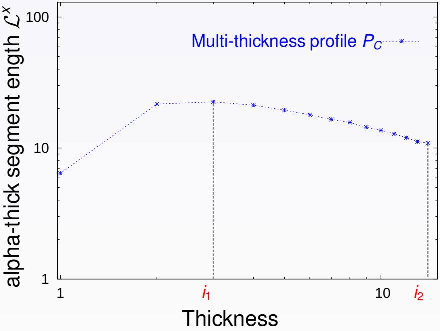
Meaningful thickness

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.





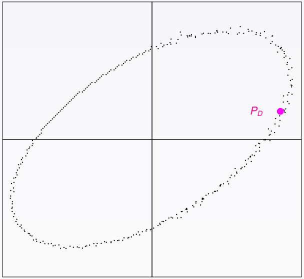
Noise detection and Meaningful Thickness

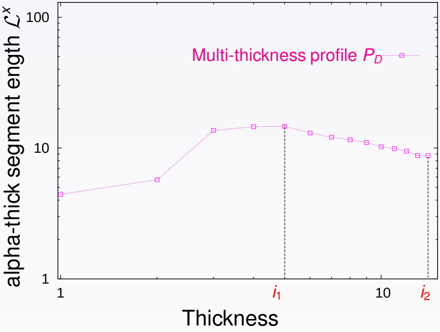
Meaningful thickness

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.





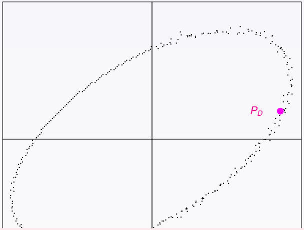
Noise detection and Meaningful Thickness

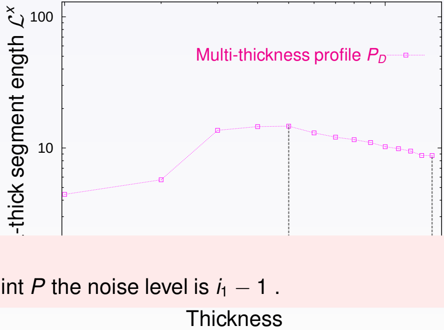
Meaningful thickness

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.

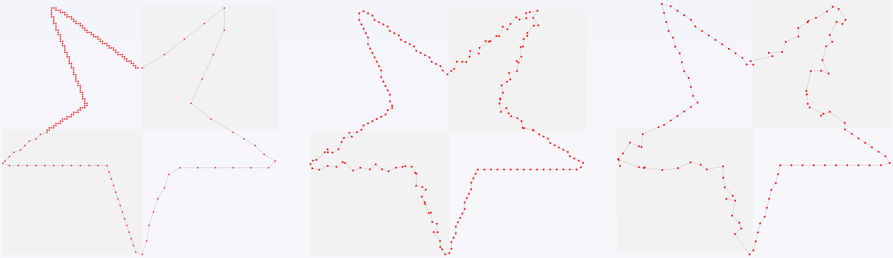




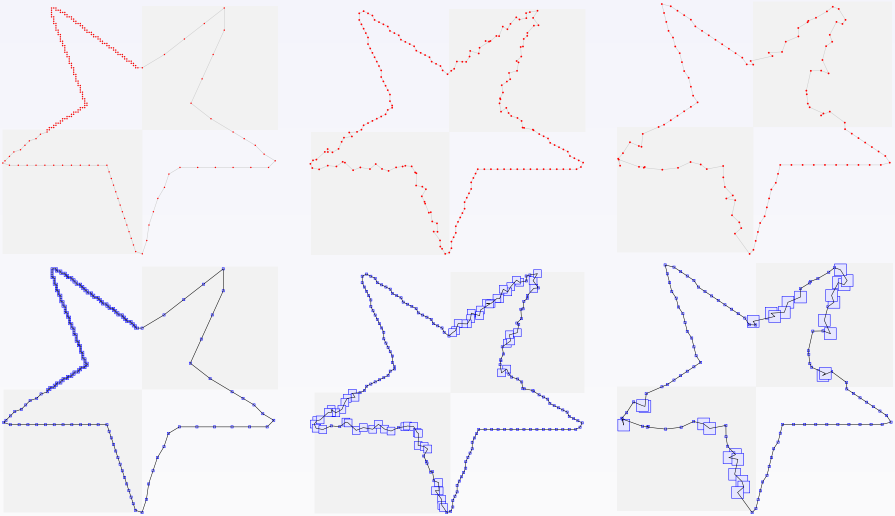
Noise level

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

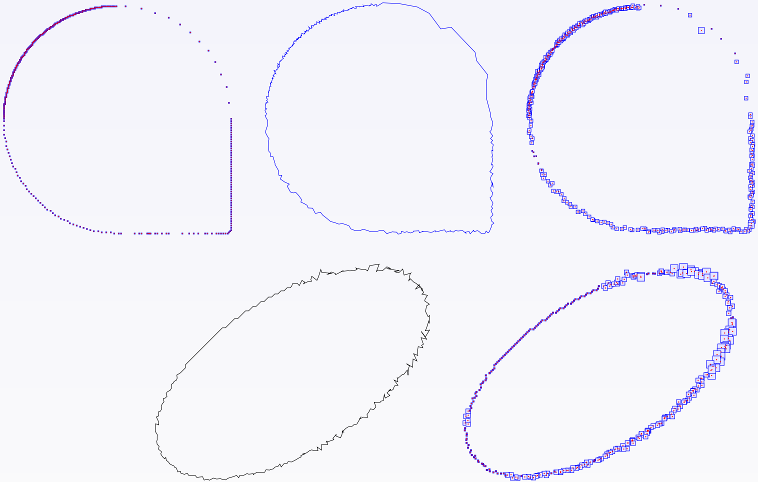
Experiments on polygonal shapes (1)



Experiments on polygonal shapes (1)

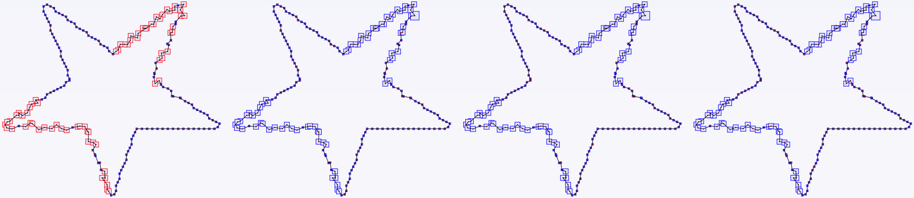


Experiments on polygonal shapes (2)



Stability from intern parameters

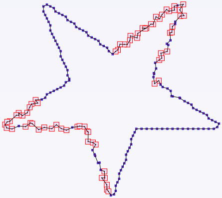
- Maximal thickness t_{max}



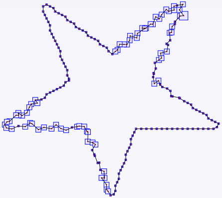
$t_{max} = 5\sqrt{2}$ $t_{max} = 10\sqrt{2}$ $t_{max} = 15\sqrt{2}$ $t_{max} = 20\sqrt{2}$

Stability from intern parameters

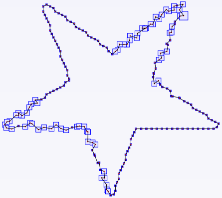
- Maximal thickness t_{max}



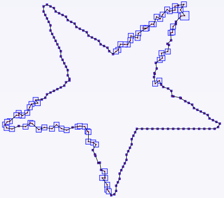
$t_{max} = 5\sqrt{2}$



$t_{max} = 10\sqrt{2}$

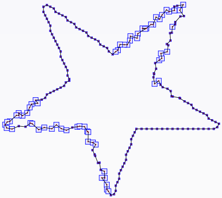


$t_{max} = 15\sqrt{2}$

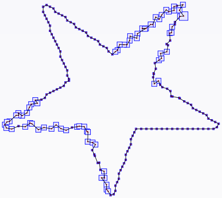


$t_{max} = 20\sqrt{2}$

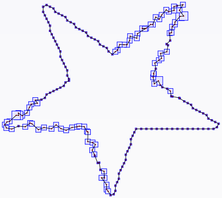
- Noise threshold T_m



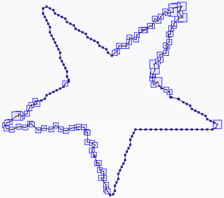
$T_m = 0.2$



$T_m = 0.0$

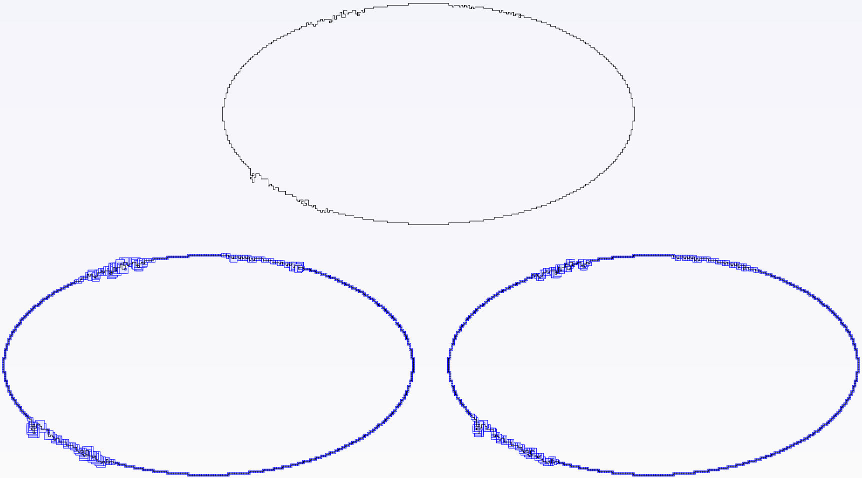


$T_m = -0.2$



$T_m = -0.4$

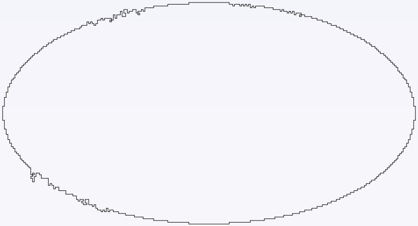
Comparison with the Meaningful Scales [Kerautret & Lachaud, 2012]

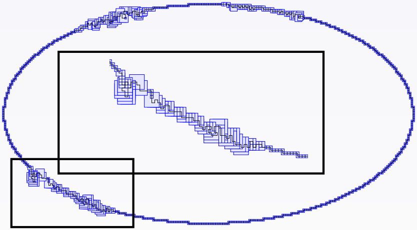


Meaningful scale

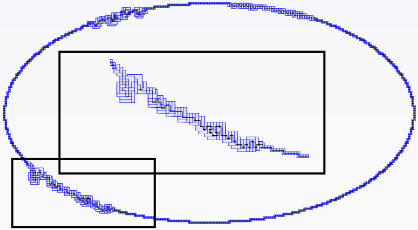
Meaningful thickness

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



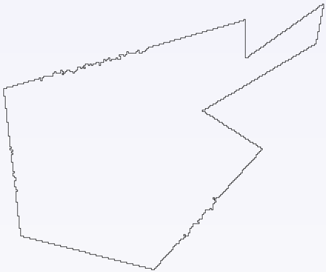


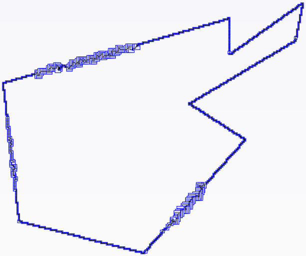
Meaningful scale



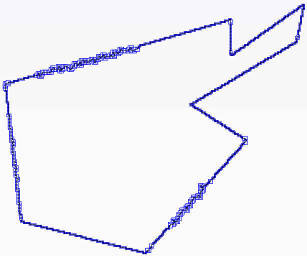
Meaningful thickness

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



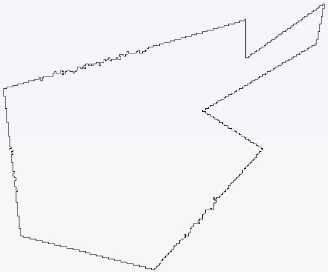


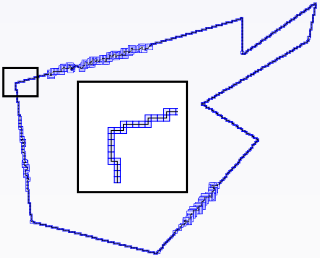
Meaningful scale



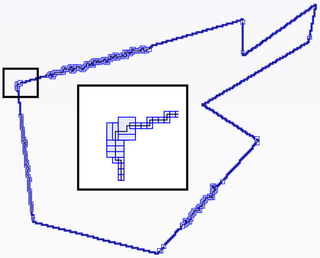
Meaningful thickness

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



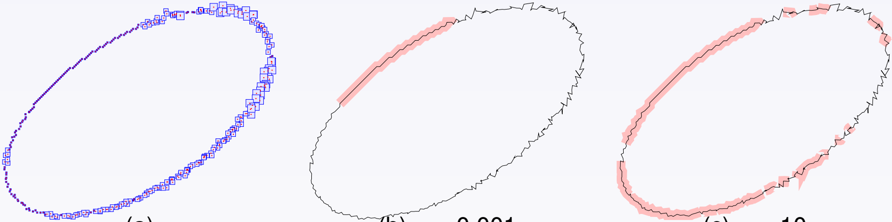


Meaningful scale



Meaningful thickness

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



(a) (b) $\epsilon = 0.001$ [Cao 03] (c) $\epsilon = 10$ [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

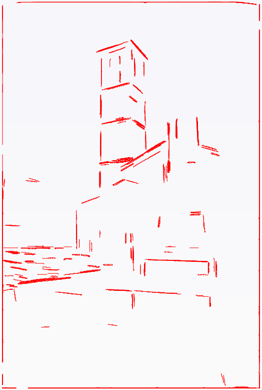
2.3 Simple applications (1)

Meaningful contour extraction from image level sets:

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

source

contour from level set

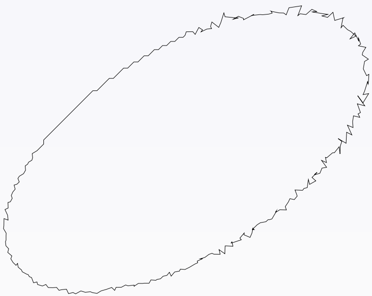


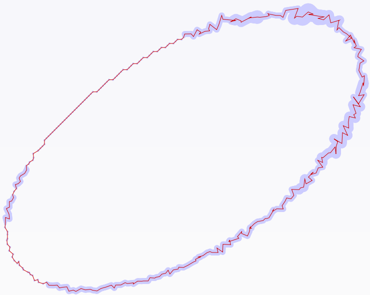
meaningful straight parts

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.



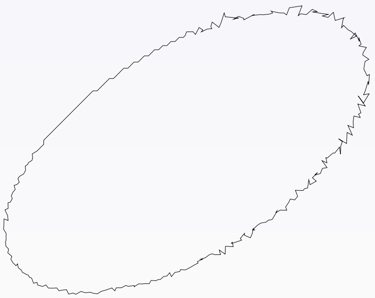


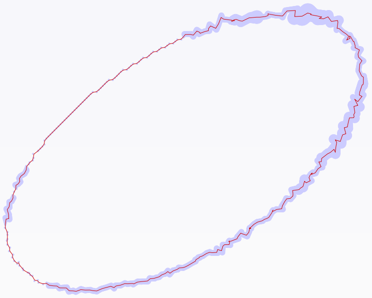
iteration 1

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.



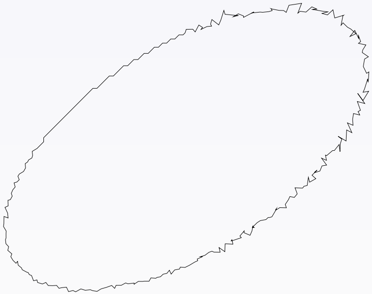


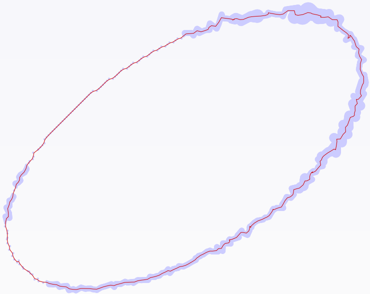
iteration 2

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.



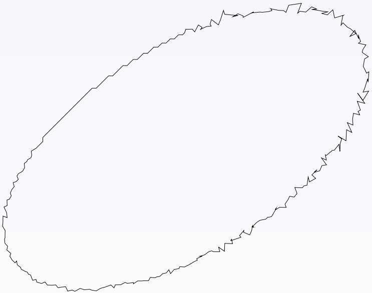


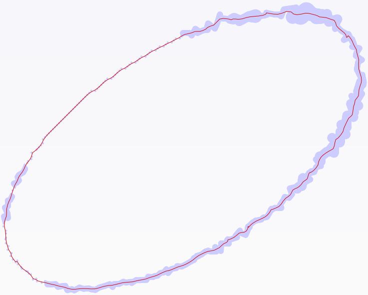
iteration 3

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.



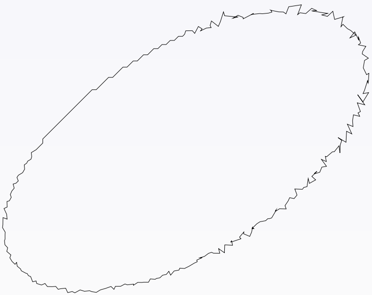


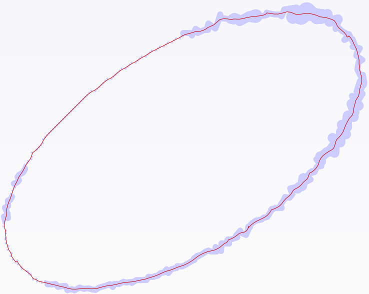
iteration 4

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.



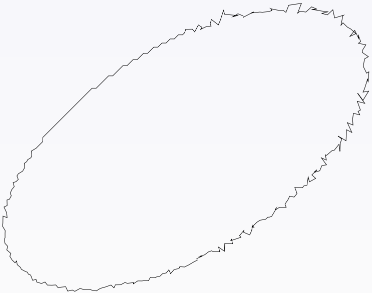


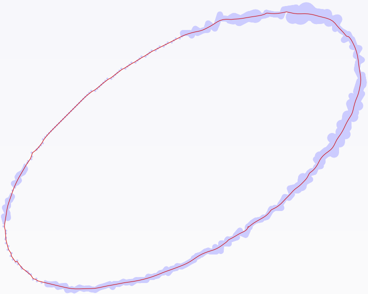
iteration 5

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.



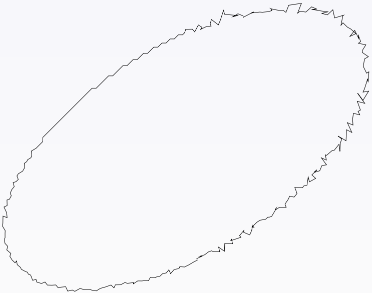


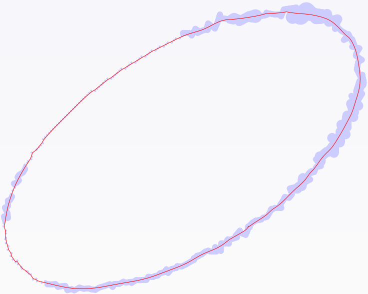
iteration 10

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.



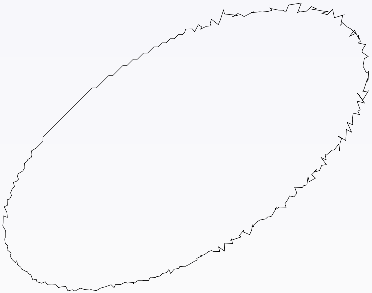


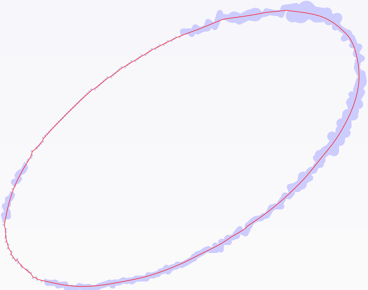
iteration 20

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.



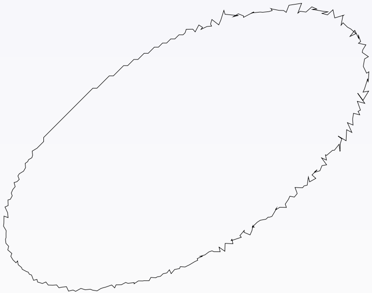


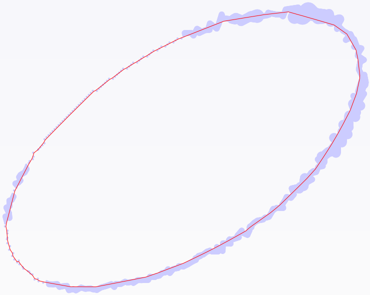
iteration 50

2.3 Simple applications (2)

Polygon denoising

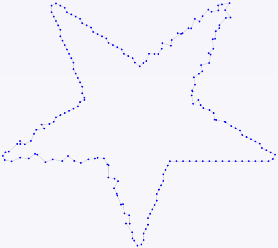
- Applying an iterative process on contour points P_i.
- Each points are moved:
  - by a weighted average of its two neighbors.
  - from constraint defined from meaningful thickness.
- Constraints are also defined between polygon vertex by linear interpolation.



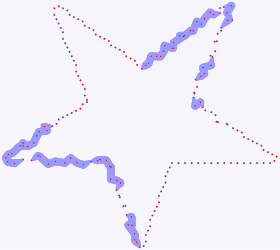


iteration 500

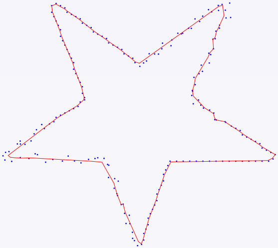
2.3 Simple applications (2)



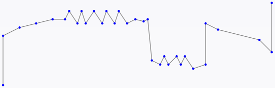
(d) source contour



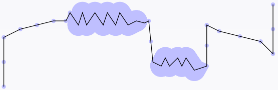
(e) meaningful thickness constraint



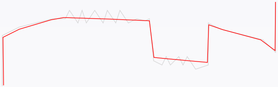
(f) resulting reconstruction



(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 easely exploit it to define geometric esitmators.

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

Alpha Thick Segment

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

(see file: tuto4_compATS.cpp)

```
// types definition
typedef Z2i::Point Point
typedef  AlphaThickSegmentComputer<Point> AlphaThickSegmentComputer2D;
// Construction of the AlphaThickSegmentComputer from a maximal thickness = 5
AlphaThickSegmentComputer2D aComputer(5);

// Initialisation from the point of index 30:
aComputer.init(contour.begin()+30);

while(aComputer.extendFront()){
}
```

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

Alpha Thick Segment

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

(see file: tuto4_compATS.cpp)

```
// types definition
typedef Z2i::Point Point
typedef  AlphaThickSegmentComputer<Point> AlphaThickSegmentComputer2D;
// Construction of the AlphaThickSegmentComputer from a maximal thickness = 5
AlphaThickSegmentComputer2D aComputer(5);

// Initialisation from the point of index 30:
aComputer.init(contour.begin()+30);

while(aComputer.extendFront()){
}
```

with Board2D display:

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

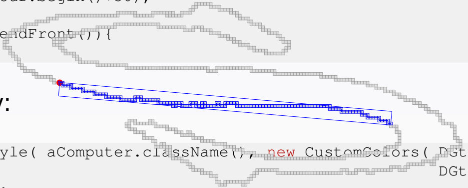
First step: Alpha-Thick Tangential Cover computation (1)

Alpha Thick Segment

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

(see file: tuto4_compATS.cpp)

```
// types definition
typedef Z2i::Point Point
typedef  AlphaThickSegmentComputer<Point> AlphaThickSegmentComputer2D;
// Construction of the AlphaThickSegmentComputer from a maximal thickness = 5
AlphaThickSegmentComputer2D aComputer(5);

// Initialisation from the point of index 30:
aComputer.init(contour.begin()+30);

while(aComputer.extendFront()){
}
```



with Board2D display:

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

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

Tangential Cover

- Commonly used to define geometric estimator: length, tangent, curvature.
  $\Rightarrow$ ex: $\lambda - MST$ 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.
  $\Rightarrow$ ex: $\lambda - MST$ 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.
  $\Rightarrow$ ex: $\lambda - MST$ 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.
  $\Rightarrow$ ex: $\lambda - MST$ 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)

```
// to get the functions firstMaximalSegment(), lastMaximalSegment() and
     nextMaximalSegment()
#include "DGtal/geometry/curves/SegmentComputerUtils.h"

// Get the first  and last AlphaThickSegment
AlphaThickSegmentComputer2D aComputer(5);
firstMaximalSegment(aComputer, contour.begin()+30, contour.begin(), contour.end());
AlphaThickSegmentComputer2D first (aComputer);
lastMaximalSegment(aComputer, contour.begin()+30, contour.begin(), contour.end());
AlphaThickSegmentComputer2D last (aComputer);

// And iterate to display all the set of maximal segments
while(first.end() != last.end()){
  aBoard << first;
  nextMaximalSegment(first, contour.end());
}

aBoard << first;
```

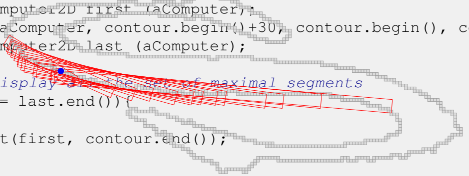
First step: Alpha-Thick Tangential Cover computation (2)

Tangential Cover

- Commonly used to define geometric estimator: length, tangent, curvature.
  $\Rightarrow$ ex: $\lambda - MST$ 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)

```
// to get the functions firstMaximalSegment(), lastMaximalSegment() and
     nextMaximalSegment()
#include "DGtal/geometry/curves/SegmentComputerUtils.h"

// Get the first  and last AlphaThickSegment
AlphaThickSegmentComputer2D aComputer(5);
firstMaximalSegment(aComputer, contour.begin()+30, contour.begin(), contour.end());
AlphaThickSegmentComputer2D first (aComputer);
lastMaximalSegment(aComputer, contour.begin()+30, contour.begin(), contour.end());
AlphaThickSegmentComputer2D last (aComputer);

// And iterate to display all the set of maximal segments
while(first.end() != last.end()){
  aBoard << first;
  nextMaximalSegment(first, contour.end());
}

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.

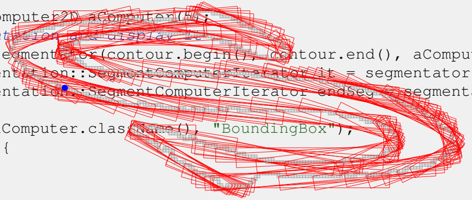
```
#include "DGtal/geometry/curves/SaturatedSegmentation.h"
...
typedef AlphaThickSegmentComputer<Z2i::Point> AlphaThickSegmentComputer2D;
typedef  SaturatedSegmentation<AlphaThickSegmentComputer2D> AlphaSegmentation;
...
AlphaThickSegmentComputer2D aComputer(5);
// Apply the segmentation and display it
AlphaSegmentation segmentator(contour.begin(), contour.end(), aComputer);
typename AlphaSegmentation::SegmentComputerIterator it = segmentator.begin();
typename AlphaSegmentation::SegmentComputerIterator endSeg = segmentator.end();

aBoard << SetMode(aComputer.className(), "BoundingBox");
while(it != endSeg){
  aBoard << *it;
  ++it;
}
```

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.

```
#include "DGtal/geometry/curves/SaturatedSegmentation.h"
...
typedef AlphaThickSegmentComputer<Z2i::Point> AlphaThickSegmentComputer2D;
typedef  SaturatedSegmentation<AlphaThickSegmentComputer2D> AlphaSegmentation;
...
AlphaThickSegmentComputer2D aComputer(5);
// Apply the segmentation and display it
AlphaSegmentation segmentator(contour.begin(), contour.end(), aComputer);
typename AlphaSegmentation::SegmentComputerIterator it = segmentator.begin();
typename AlphaSegmentation::SegmentComputerIterator endSeg = segmentator.end();

aBoard << SetMode(aComputer.className(), "BoundingBox");
while(it != endSeg){
  aBoard << *it;
  ++it;
}
```



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`)

```
// To represent the multiscale profile
#include "DGtal/math/Profile.h"
...
Profile<> sp;
sp.init(5);
...
// compute all the segments covering a point (for each thickness):
for(unsigned int thickness = 1; thickness<=5; thickness++){
  AlphaThickSegmentComputer2D aComputer(thickness);
  firstMaximalSegment(aComputer, contour.begin()+index, contour.begin(), contour.end());
  AlphaThickSegmentComputer2D first (aComputer);
  lastMaximalSegment(aComputer, contour.begin()+index, contour.begin(), contour.end());
  AlphaThickSegmentComputer2D last (aComputer);
  ...
}
```

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)

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

Second step: Extracting the Meaningful Thickness

Main steps of the Meaningful Thickness algorithm

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

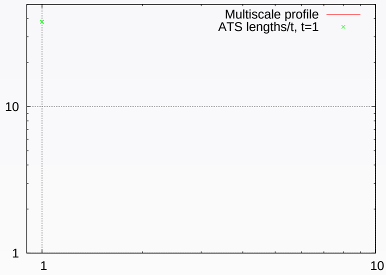
```
std::vector<double> x,y;
s.getProfile(x, y);
trace.info() << "Noise␣level:␣␣<<␣": "␣<<␣sp.noiseLevel()␣<<␣std::endl;
```

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.

```
std::vector<double> x,y;
s.getProfile(x, y);
trace.info() << "Noise␣level:␣␣<<␣": "␣<<␣sp.noiseLevel()␣<<␣std::endl;
```

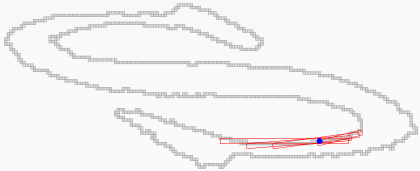


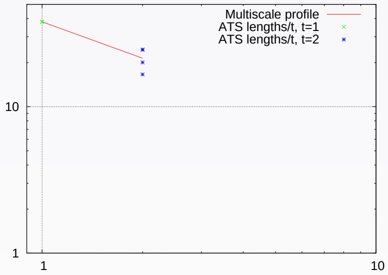
Second step: Extracting the Meaningful Thickness

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.

```
std::vector<double> x,y;
s.getProfile(x, y);
trace.info() << "Noise␣level:␣␣<<␣": "␣<<␣sp.noiseLevel()␣<<␣std::endl;
```



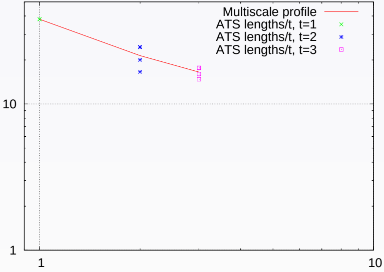


Second step: Extracting the Meaningful Thickness

Main steps of the Meaningful Thickness algorithm

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

```
std::vector<double> x,y;
s.getProfile(x, y);
trace.info() << "Noise␣level:␣␣<<␣": "␣<<␣sp.noiseLevel()␣<<␣std::endl;
```

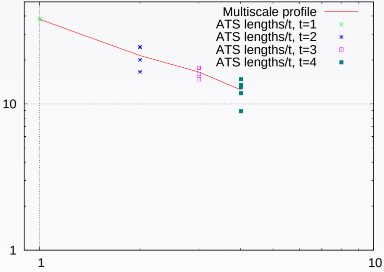


Second step: Extracting the Meaningful Thickness

Main steps of the Meaningful Thickness algorithm

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

```
std::vector<double> x,y;
s.getProfile(x, y);
trace.info() << "Noise level:  << ": " << sp.noiseLevel() << std::endl;
```

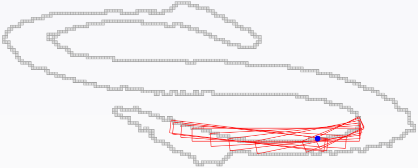


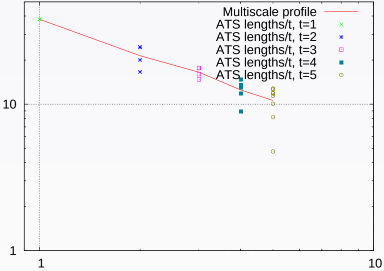
Second step: Extracting the Meaningful Thickness

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.

```
std::vector<double> x,y;
s.getProfile(x, y);
trace.info() << "Noise␣level:␣␣<<␣": "␣<<␣sp.noiseLevel()␣<<␣std::endl;
```



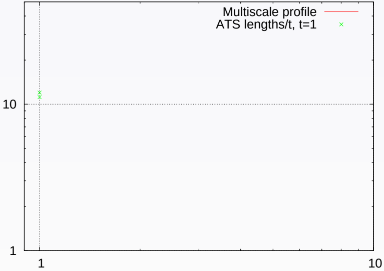


Second step: Extracting the Meaningful Thickness

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.

```
std::vector<double> x,y;
s.getProfile(x, y);
trace.info() << "Noise␣level:␣␣<<␣": "␣<<␣sp.noiseLevel()␣<<␣std::endl;
```

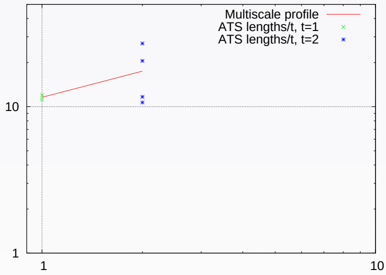


Second step: Extracting the Meaningful Thickness

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.

```
std::vector<double> x,y;
s.getProfile(x, y);
trace.info() << "Noise level:  << ": " << sp.noiseLevel() << std::endl;
```

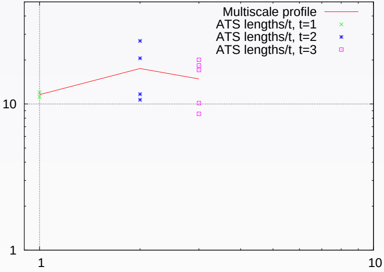


Second step: Extracting the Meaningful Thickness

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.

```
std::vector<double> x,y;
s.getProfile(x, y);
trace.info() << "Noise␣level:␣␣<<␣": "␣<<␣sp.noiseLevel()␣<<␣std::endl;
```

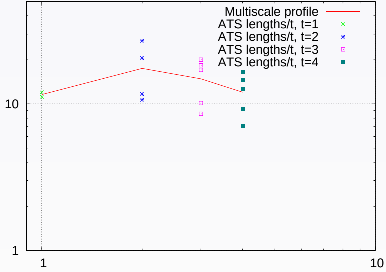


Second step: Extracting the Meaningful Thickness

Main steps of the Meaningful Thickness algorithm

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

```
std::vector<double> x,y;
s.getProfile(x, y);
trace.info() << "Noise level:  << ": " << sp.noiseLevel() << std::endl;
```

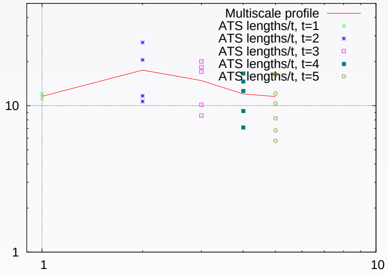


Second step: Extracting the Meaningful Thickness

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.

```
std::vector<double> x,y;
s.getProfile(x, y);
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.
  $\Rightarrow$ similary 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)

```
std::vector<Point>::const_iterator itInit = contour.begin();
for(unsigned int thickness = 1; thickness <= scaleMax; thickness++){
  AlphaThickSegmentComputer2D aComputer(thickness);
  AlphaSegmentation segmentator(contour.begin(), contour.end(), aComputer);
  typename AlphaSegmentation::SegmentComputerIterator it = segmentator.begin();
  typename AlphaSegmentation::SegmentComputerIterator endSeg = segmentator.end();
  for( ; it != endSeg; ++it){
    trace.info() << ".";
    AlphaThickSegmentComputer2D seg(*it);
    double lengthSegment = seg.getSegmentLength();
    std::vector<Point>::const_iterator itS = seg.begin();
    for ( ; itS != seg.end(); ++itS) {
      unsigned int i = std::distance(itInit, itS);
      vectProfiles[i].addValue(thickness-1, lengthSegment/thickness);
    }
  }
}
```

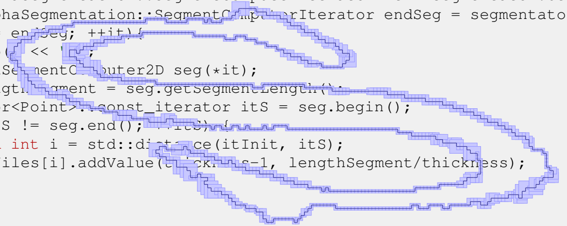
Second step: Extracting the Meaningful Thickness (2)

Meaningful thickness computation on all points

- Compute the full tangential cover from SaturatedSegmentation.
  $\Rightarrow$ 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)

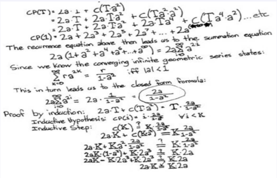
```
std::vector<Point>::const_iterator itInit = contour.begin();
for(unsigned int thickness = 1; thickness <= scaleMax; thickness++){
  AlphaThickSegmentComputer2D aComputer(thickness);
  AlphaSegmentation segmentator(contour.begin(), contour.end(), aComputer);
  typename AlphaSegmentation::SegmentComputerIterator it = segmentator.begin();
  typename AlphaSegmentation::SegmentComputerIterator endSeg = segmentator.end();
  for( ; it != endSeg; ++it){
    trace.info() << ".";
    AlphaThickSegmentComputer2D seg(*it);
    double lengthSegment = seg.getSegmentLength();
    std::vector<Point>::const_iterator itS = seg.begin();
    for ( ; itS != seg.end(); ++itS){
      unsigned int i = std::distance(itInit, itS);
      vectProfiles[i].addValue(thickness-1, lengthSegment/thickness);
    }
  }
}
```



Overview of the presentation

3. Highlighting Reproducible Research

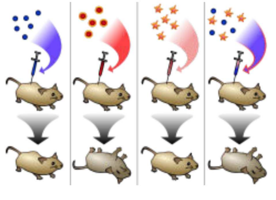
3. Highlighting Reproducible Research (1)



Reproducible research in sciences:

- *Theoretical scientists* share demonstrations;

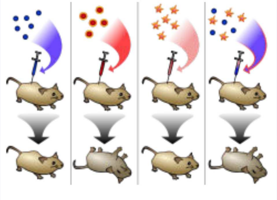
3. Highlighting Reproducible Research (1)



Reproducible research in sciences:

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

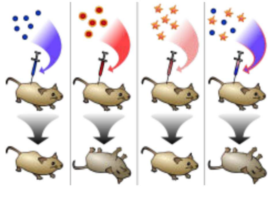
3. Highlighting Reproducible Research (1)



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... ?*

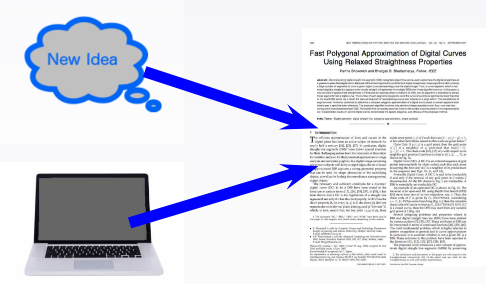
Computer Science:

- 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.



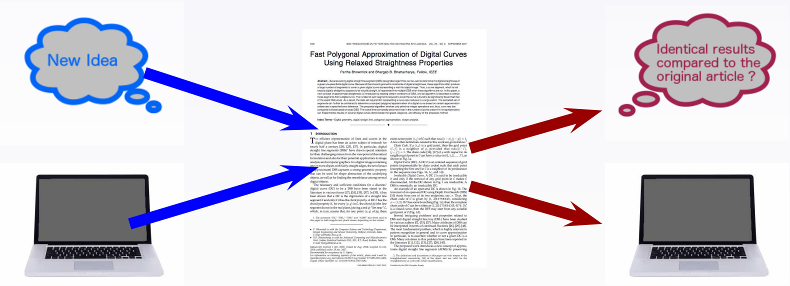
3. Highlighting Reproducible Research (2)

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

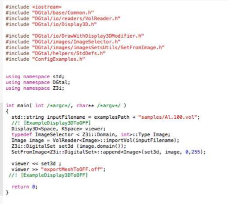
- ⊕ A real added value for the publication;
- ⊕ increases the impact/comparisons;
- ⊖ software is not really acknowledged;
- ⊖ important effort (documentation, tests, user maintenance).

```
#include <iostream>
#include "DGtal/base/Common.h"
#include "DGtal/io/readers/VolReader.h"
#include "DGtal/io/Display3D.h"

#include "DGtal/io/DrawWithDisplay3DModifier.h"
#include "DGtal/images/ImageSelector.h"
#include "DGtal/images/imagesSetsUtils/SetFromImage.h"
#include "DGtal/helpers/StdDefs.h"
#include "ConfigExamples.h"


using namespace std;
using namespace DGtal;
using namespace Z3i;


int main( int /*argc*/, char** /*argv*/ )
{
  std::string inputFilename = examplesPath + "samples/Al.100.vol";
  //! [ExampleDisplay3DToOFF]
  Display3D<Space, KSpace> viewer;
  typedef ImageSelector < Z3i::Domain, int>::Type Image;
  Image image = VolReader<Image>::importVol(inputFilename);
  Z3i::DigitalSet set3d (image.domain());
  SetFromImage<Z3i::DigitalSet>::append<Image>(set3d, image, 0,255);

  viewer << set3d ;
  viewer >> "exportMeshToOFF.off";
 //! [ExampleDisplay3DToOFF]

  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).



```
#include <iostream>
#include "DGtal/base/Common.h"
#include "DGtal/io/readers/VolReader.h"
#include "DGtal/io/Display3D.h"

#include "DGtal/io/DrawWithDisplay3DModifier.h"
#include "DGtal/images/ImageSelector.h"
#include "DGtal/images/imagesSetsUtils/SetFromImage.h"
#include "DGtal/helpers/StdDefs.h"
#include "ConfigExamples.h"


using namespace std;
using namespace DGtal;
using namespace Z3i;


int main( int /*argc*/, char** /*argv*/ )
{
  std::string inputFilename = examplesPath + "samples/Al.100.vol";
  //! [ExampleDisplay3DToOFF]
  Display3D<Space, KSpace> viewer;
  typedef ImageSelector < Z3i::Domain, int>::Type Image;
  Image image = VolReader<Image>::importVol(inputFilename);
  Z3i::DigitalSet set3d (image.domain());
  SetFromImage<Z3i::DigitalSet>::append<Image>(set3d, image, 0,255);

  viewer << set3d ;
  viewer >> "exportMeshToOFF.off";
 //! [ExampleDisplay3DToOFF]

  return 0;
}
```

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).

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;
  $\Rightarrow$ 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**.

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:

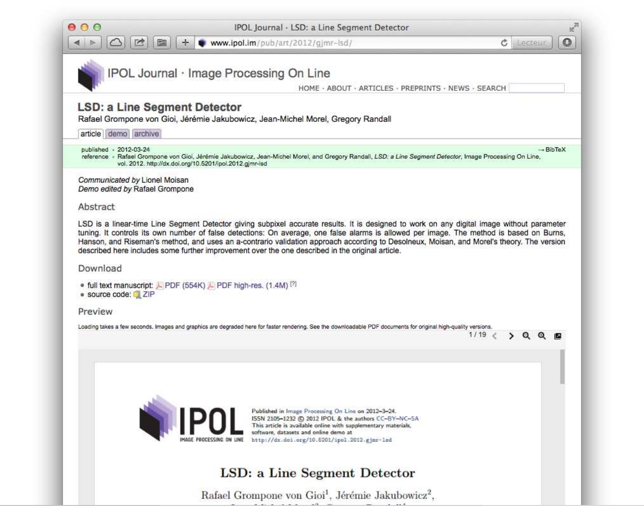
- Follows the guideline on reproducible research topics;
- reproducible research standard [Stodden 09a] [Stodden 09b];
- answer to credibility crisis in scientific computation (as pointed out by Donoho [Donoho *et al.* 09]).

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.

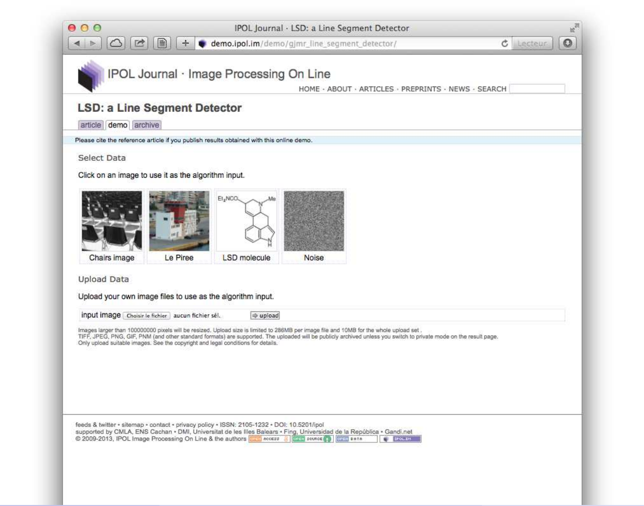
3.1 Image Processing On Line Presentation: principle and current form

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



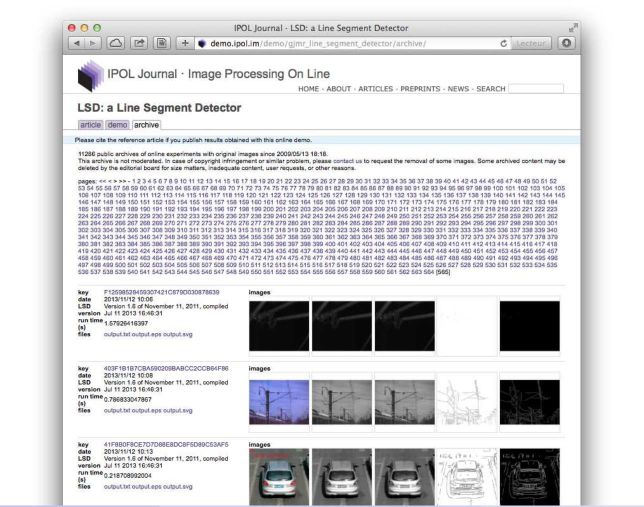
3.1 Image Processing On Line Presentation: principle and current form

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



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.



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.

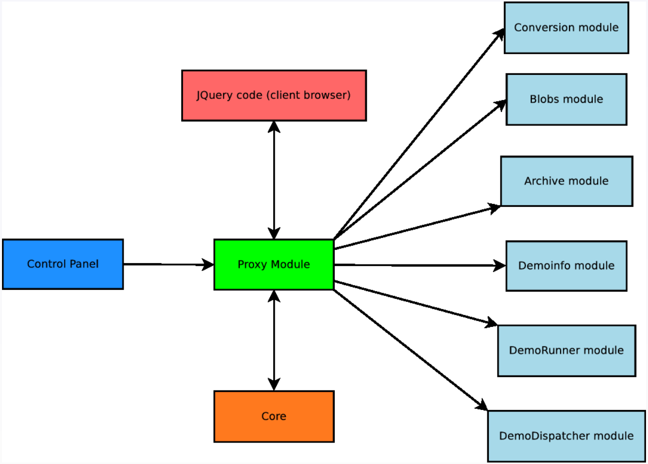
Software point of view:

- 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)

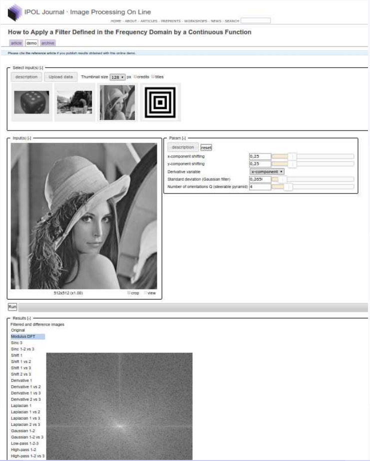
- Based on micro-services to allow scalability and fast development.
- Better than previous architecture for distributed algorithm computing.



3.2 Structure of an IPOL demonstration (2)

Preview of new interface

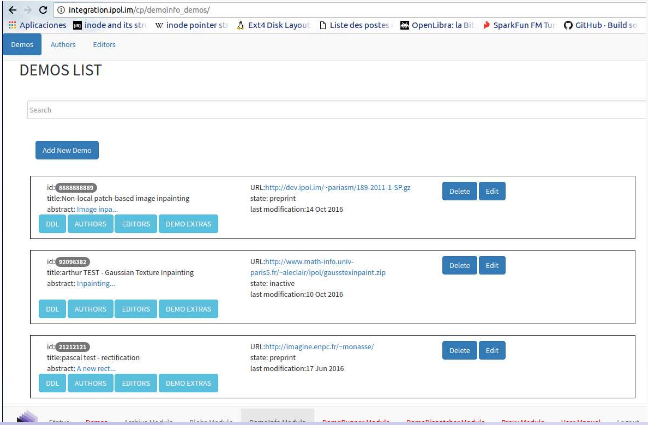
- Web interface more ergonomic and simple..
- Example of the demo of a published article:
  `http://ipolcore.ipol.im/demo/clientApp/demo.html?id=116`



3.2 Structure of an IPOL demonstration (3)

New Demo Control Panel

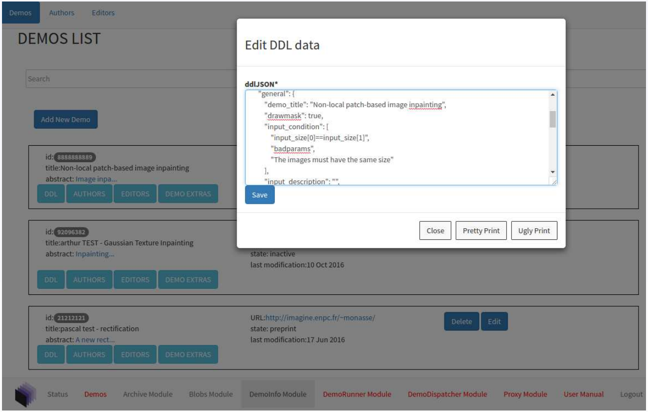
- Replace previous demo source code.
- The IPOL Editors create the demo using an internal web tool called: "Control Panel"



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



Thanks for your attention !

Keraudret, 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/ipol.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 Debled-Rennesson, B. Kerautret
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.

Debled-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.