ACPR 19 Tutorial

Digital Geometry in Pattern Recognition: Extracting Geometric Features with DGtal and Applications

– Part I –

Bertrand Kerautret[1] Jacques-Olivier Lachaud [2]

5th Asian Conference on Pattern Recognition
26 November, 2019, Auckland, New Zealand

[1] LIRIS (UMR CNRS 5205) Université de Lyon 2, F-69676, France

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

Overview of the presentation - Part I -

`https://kerautret.github.io/ACPR19-DGPRTutorial`

1. Motivation, Theory and Applications

Motivation

Digital Geometry

Study of shapes defined in a digital domain, generally images ($\mathbb{Z}^2$, $\mathbb{Z}^3$, $\ldots$) or sometimes regular lattices.

- 2D shapes = set of pixels = subsets of $\mathbb{Z}^2$

photo picture

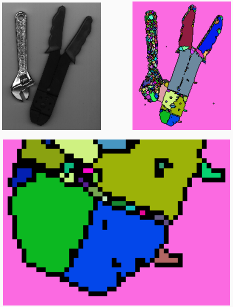
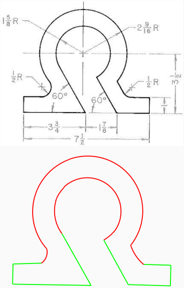


image segmentation



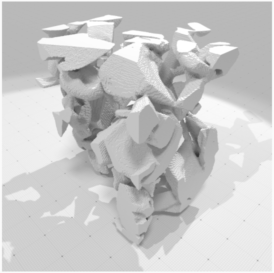
document analysis

Motivation

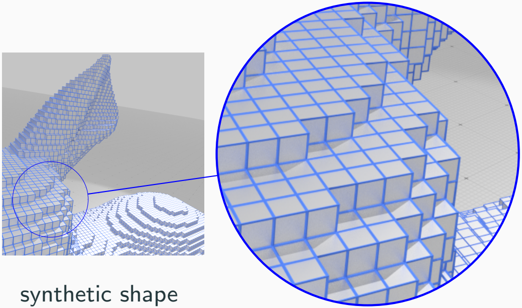
Digital Geometry

Study of shapes defined in a digital domain, generally images ($\mathbb{Z}^2$, $\mathbb{Z}^3$, $\ldots$) or sometimes regular lattices.

- 2D shapes $=$ set of pixels $=$ subsets of $\mathbb{Z}^2$
- 3D shapes $=$ set of voxels $=$ subsets of $\mathbb{Z}^3$



Micro-snow tomography

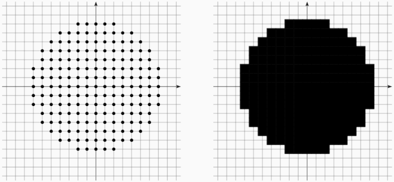


synthetic shape

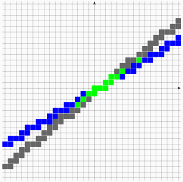
Motivation

Why a specific Digital Geometry ?

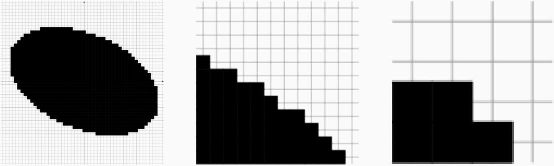
- geometry of pixels/voxels looks easy but is difficult for many reasons
- Euclidean definitions of connectedness, convexity, straight lines, differential geometric quantities fail



Convexity ?



Line Intersection ?



Infinitesimal differential geometry?

Applications require geometric tools

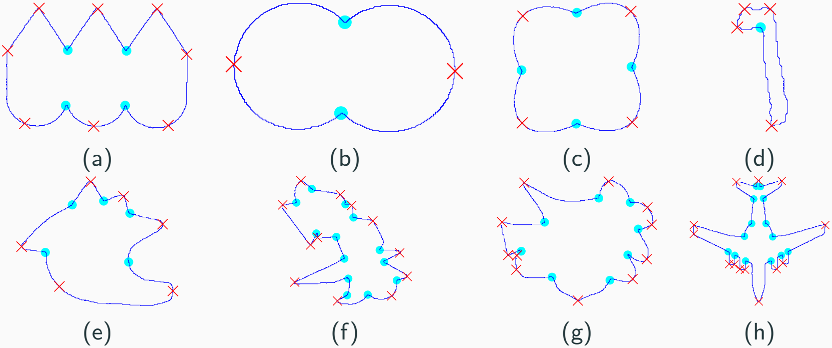
Classical image applications

- image restoration, noise identification/removal
- image segmentation with geometric priors
- shape matching, indexing
- precise shape measurements (biomedical and material imaging)

Desired geometric analysis

- identify linear or planar parts
- cut shape into convex / concave parts
- identify dominant points (high curvature) and inflexion points (perception)
- measure volume, perimeter, area, length, curvatures
- identify centerline of tubular objects
- compute skeleton, medial axis
- process shape geometry: remove noise, simplify, multi-scale decomposition

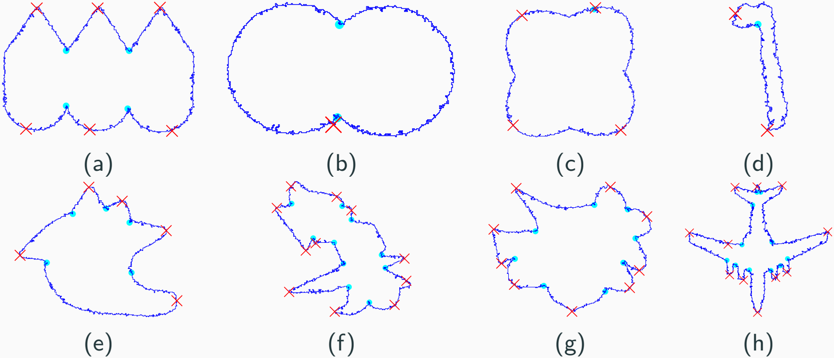
Applications where digital geometry is useful



Corner point detection

- digital contour tracking
- sound definition of digital straight segment
- stable and convergent digital curvature estimator

Applications where digital geometry is useful



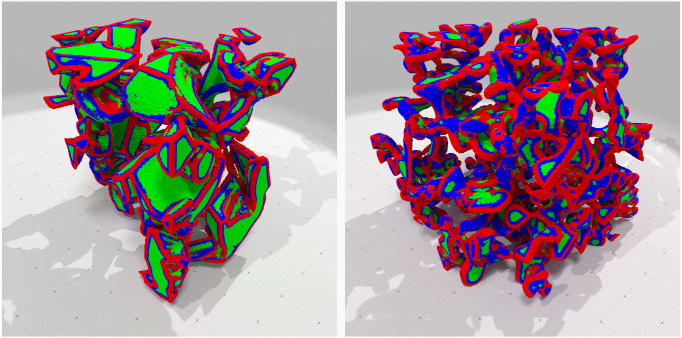
(a) (b) (c) (d)

(e) (f) (g) (h)

Corner point detection

- digital contour tracking
- sound definition of digital straight segment
- stable and convergent digital curvature estimator
- noise addressed with *thicker* digital straight segment

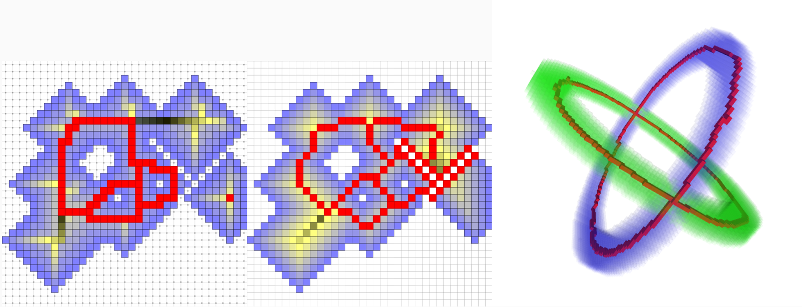
Applications where digital geometry is useful



3D shape feature extraction on snow micro-structures

- 3D micro-tomography of snow ⇒ binary 3D images
- digital topology ⇒ digital surface tracking
- extracting linear parts along axes plane *xy*, *xz*, *yz*
- theoretical asymptotic analysis of length wrt gridstep *h*
- identify features according to length of linear parts

Applications where digital geometry is useful



Topology identification and control, skeleton extraction

- consistent definitions of connectedness
- topological invariant (here homotopy)
- simple points preserve topology: very efficient topological control

Applications where digital geometry is useful

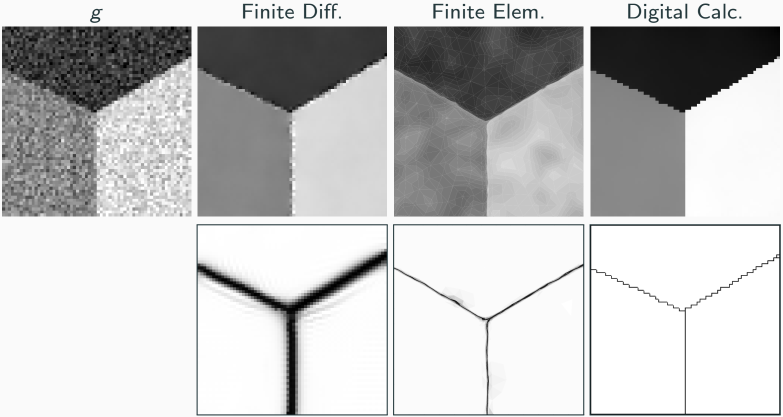


Image restoration, segmentation and inpainting

- most image processing task = variational formulation
- digital calculus = sound framework for variational problem in digital domain
- digital calculus formulation of Mumford-Shah model

Applications where digital geometry is useful

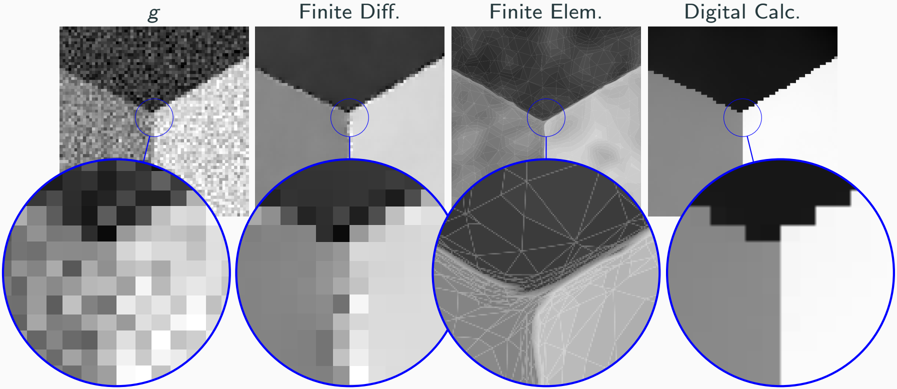


Image restoration, segmentation and inpainting

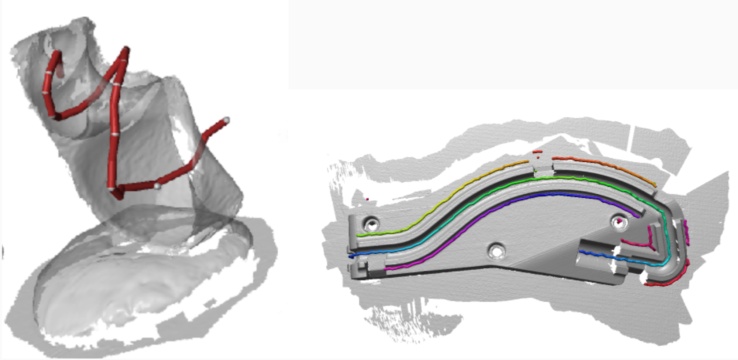
- most image processing task = variational formulation
- digital calculus = sound framework for variational problem in digital domain
- digital calculus formulation of Mumford-Shah model

Applications where digital geometry is useful

Generate 3D surface model from 3D labelled images

- surface tracking in 3D labelled partitions
- convergent normal vector estimation on interfaces
- discrete variational model to align digital surface with estimated normals

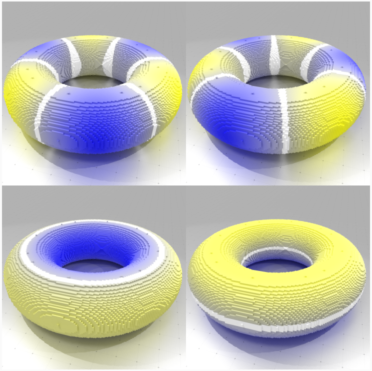
Applications where digital geometry is useful



Centerline extraction in arbitrary mesh / digital surfaces

- normal estimation on mesh / digital surfaces
- ray casting with 3D digital straight lines
- digital voting process

Applications where digital geometry is useful



Laplacian operator for shape analysis, simplification, matching

- convergent normal estimation on digital surfaces
- convergent surface integrals
- ⇒ pointwise convergent Laplacian operator
- provide eigenvalues/eigenvector analysis

Summary

Applications require sound theoretical foundations

- digital topology
    - contour tracking
    - topological invariants and simple points
    - digital surfaces
- geometric primitives
    - digital straight segments
    - digital planes
- convergent geometric estimators
    - tangent and normal estimation
    - surface integrals
- digital calculus
    - variational image and geometry processing
    - multiscale analysis

Summary

Applications require sound theoretical foundations

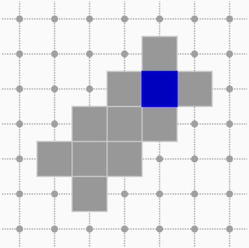
- **digital topology**
  - contour tracking
  - topological invariants and simple points
  - **digital surfaces**
- **geometric primitives**
  - **digital straightness**
  - digital planes
- **convergent geometric estimators**
  - **tangent** and normal **estimation**
  - curvatures estimation
  - surface integrals
- **digital calculus**
  - **variational image and geometry processing**
  - multiscale analysis

Main ingredients of digital geometry

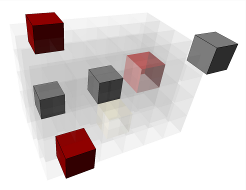
Topology: grid, adjacency, connectedness

- regular grid / lattice

2D discrete space



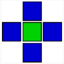
3D discrete space

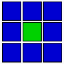


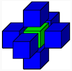
Main ingredients of digital geometry

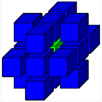
Topology: grid, adjacency, connectedness

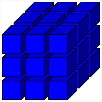
- regular grid / lattice
- 4-/8-adjacency (2D), 6-/18-/26-adjacencies (3), play dual roles


4-adj


8-adj


6-adj

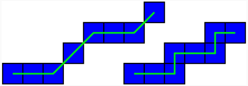

18-adj


26-adj

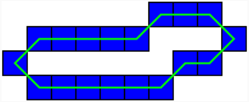
Main ingredients of digital geometry

Topology: grid, adjacency, connectedness

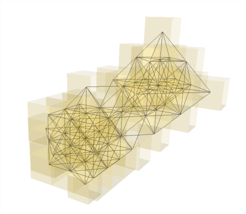
- regular grid / lattice
- 4-/8-adjacency (2D), 6-/18-/26-adjacencies (3), play dual roles
- curves, objects are related to adjacency pairs



8-Arc and 4-Arc



8-Curve

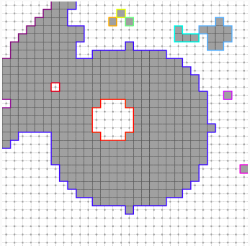


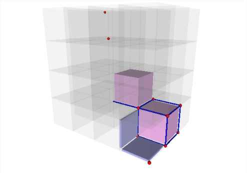
18-6-object

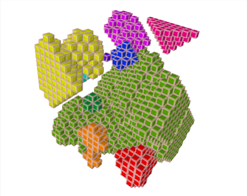
Main ingredients of digital geometry

Topology: grid, adjacency, connectedness

- regular grid / lattice
- 4-/8-adjacency (2D), 6-/18-/26-adjacencies (3), play dual roles
- curves, objects are related to adjacency pairs
- interpixel / cell topology, digital surfaces related to adjacency pairs
- sound definition of digital *d*-dimensional manifold



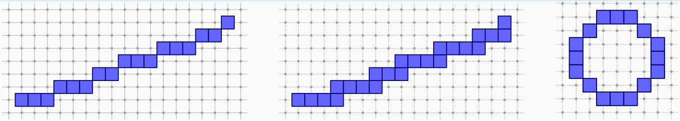




Main ingredients of digital geometry

Geometric primitives

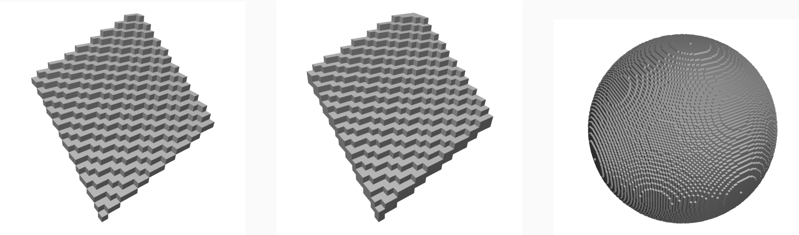
- 2D/3D naive or standard digital lines, circles.



Main ingredients of digital geometry

Geometric primitives

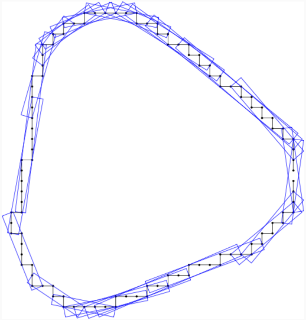
- 2D/3D naive or standard digital lines, circles.
- 3D naive or standard digital planes, spheres.

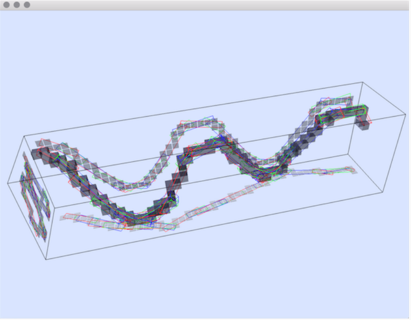


Main ingredients of digital geometry

Geometric primitives

- 2D/3D naive or standard digital lines, circles.
- 3D naive or standard digital planes, spheres.
- recognition algorithms for these primitives

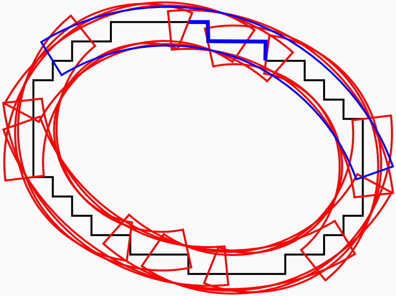


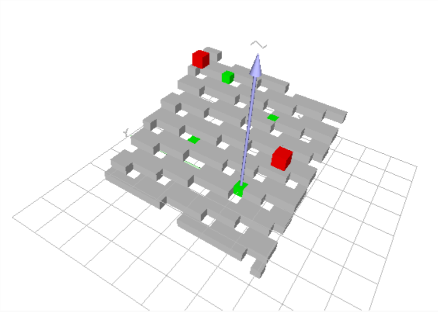


Main ingredients of digital geometry

Geometric primitives

- 2D/3D naive or standard digital lines, circles.
- 3D naive or standard digital planes, spheres.
- recognition algorithms for these primitives

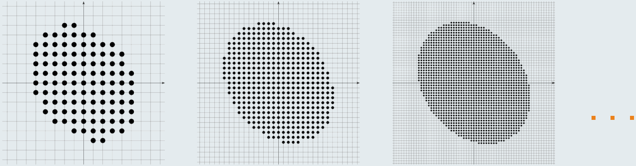




Main ingredients of digital geometry

Geometric estimators of area/volume/tangent/normals/curvatures

- multigrid convergence: the finer the sampling grid h, the better the geometric estimation



$\mathrm{Dig}_h(X)$ $\mathrm{Dig}_{h/2}(X)$ $\mathrm{Dig}_{h/4}(X)$ $\dots$

Main ingredients of digital geometry

Geometric estimators of area/volume/tangent/normals/curvatures

- multigrid convergence: the finer the sampling grid h, the better the geometric estimation
- multigrid convergent estimators of (speed as a function of h)

area/volume	pixel/voxel counting ($O(h)$ convex shapes, $O(h^{22/15})$ C^2-convex)
perimeter	minimum length polygon ($O(h^{4/3})$ convex shapes, $O(h)$ otherwise)
tangent 2D	max. digital straight segment ($O(h^{2/3})$ piecewise C^2 shapes), Voronoi Covariance Measure ($O(h^{2/3})$)
normal 3D	integral invariant ($O(h^{2/3})$), Voronoi Covariance Measure ($O(h^{2/3})$),
curvatures 2D/3D	integral invariant ($O(h^{1/3})$), corrected curvature measures ($O(h^{2/3})$)

Main ingredients of digital geometry

Geometric estimators of area/volume/tangent/normals/curvatures

- multigrid convergence: the finer the sampling grid h, the better the geometric estimation
- multigrid convergent estimators of (speed as a function of h)

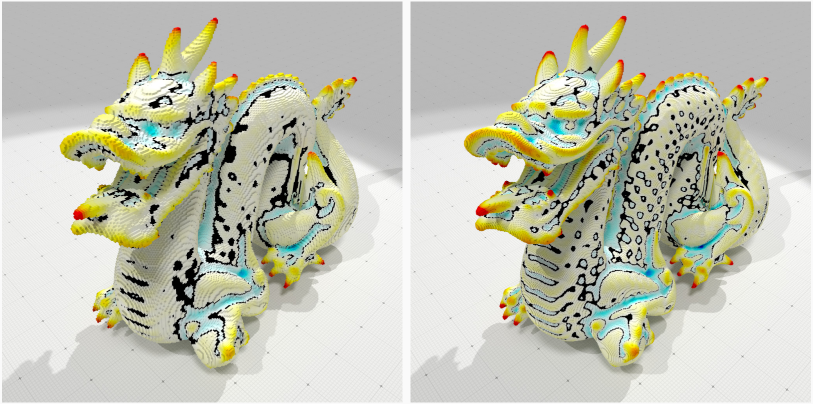
area/volume	pixel/voxel counting ($O(h)$ convex shapes, $O(h^{22/15})$ C^2-convex)
perimeter	minimum length polygon ($O(h^{4/3})$ convex shapes, $O(h)$ otherwise)
tangent 2D	max. digital straight segment ($O(h^{2/3})$ piecewise C^2 shapes), Voronoi Covariance Measure ($O(h^{2/3})$)
normal 3D	integral invariant ($O(h^{2/3})$), Voronoi Covariance Measure ($O(h^{2/3})$),
curvatures 2D/3D	integral invariant ($O(h^{1/3})$), corrected curvature measures ($O(h^{2/3})$)

All results presented in the tutorial were obtained from the DGtal library!

Main ingredients of digital geometry

Example of convergent curvature estimator

- Mean curvature estimation with corrected curvature measures

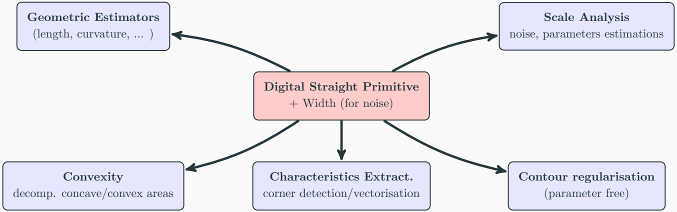


2. Geometry with Digital Straight lines

2. Geometry with Digital Straight Lines

Main primitive for 2D analysis:

- **D**igital **S**traight **S**egment recognition algorithms (DSS).
- Take the noise into account (parameters).
- Adapted locally to the shape (scale adjustment).



Geometric Estimators
(length, curvature, ...)

Scale Analysis
noise, parameters estimations

Digital Straight Primitive
+ Width (for noise)

Convexity
decomp. concave/convex areas

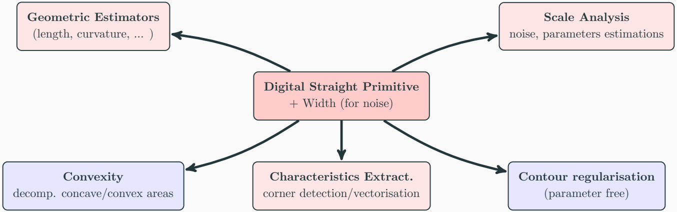
Characteristics Extract.
corner detection/vectorisation

Contour regularisation
(parameter free)

2. Geometry with Digital Straight Lines

Main primitive for 2D analysis:

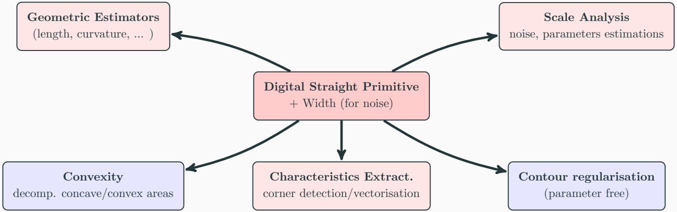
- **D**igital **S**traight **S**egment recognition algorithms (DSS).
- Take the noise into account (parameters).
- Adapted locally to the shape (scale adjustment).



2. Geometry with Digital Straight Lines

Main primitive for 2D analysis:

- **D**igital **S**traight **S**egment recognition algorithms (DSS).
- Take the noise into account (parameters).
- Adapted locally to the shape (scale adjustment).



Digital Straight Primitive
+ Width (for noise)

Geometric Estimators
(length, curvature, ...)

Scale Analysis
noise, parameters estimations

Convexity
decomp. concave/convex areas

Characteristics Extract.
corner detection/vectorisation

Contour regularisation
(parameter free)

Overview of Geometry with DSS:

- 2.1 Main idea of DSS recognition algorithms.
- 2.2 Adaptation to noise.
- 2.3 Applications examples: curvature, scale detection and vectorisation.

2.1 Main idea of DSS recognition algorithms

Arithmetic definition of digital line [Réveilles 91]

A digital line with parameters (a, b, μ) and arithmetical thickness ω is defined as the set of integer points (x, y) verifying :

$$\mu \leq ax - by < \mu + \omega$$

- a, b, μ, ω in $\mathbb{Z}$
- $gcd(a, b) = 1$, (b, a) main vector of the line
- noted $\mathcal{D}(a, b, \mu, \omega)$

2.1 Main idea of DSS recognition algorithms

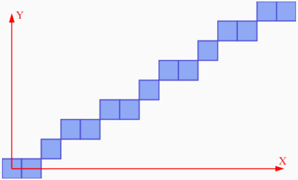
Arithmetic definition of digital line [Réveilles 91]

A digital line with parameters (a, b, μ) and arithmetical thickness ω is defined as the set of integer points (x, y) verifying :

$$\mu \leq ax - by < \mu + \omega$$

- $a,\ b,\ \mu,\ \omega$ in $\mathbb{Z}$
- $gcd(a, b) = 1$, (b, a) main vector of the line
- noted $\mathcal{D}(a, b, \mu, \omega)$

- if $\omega = max(|a|, |b|)$: $\mathcal{D}$ is 8-arc (naïve line).



$\mathcal{D}(5, 8, -1, 8)$

2.1 Main idea of DSS recognition algorithms

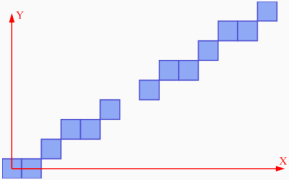
Arithmetic definition of digital line [Réveilles 91]

A digital line with parameters (a, b, μ) and arithmetical thickness ω is defined as the set of integer points (x, y) verifying :

$$\mu \leq ax - by < \mu + \omega$$

- $a,\ b,\ \mu,\ \omega$ in $\mathbb{Z}$
- $gcd(a, b) = 1$, (b, a) main vector of the line
- noted $\mathcal{D}(a, b, \mu, \omega)$

- if $\omega = max(|a|, |b|)$: $\mathcal{D}$ is 8-arc (naïve line).
- if $\omega < max(|a|, |b|)$: $\mathcal{D}$ is deconnected.



$\mathcal{D}(5, 8, -1, 8)$

2.1 Main idea of DSS recognition algorithms

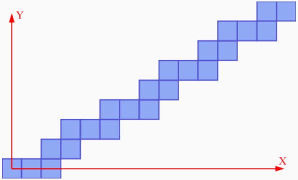
Arithmetic definition of digital line [Réveilles 91]

A digital line with parameters (a, b, μ) and arithmetical thickness ω is defined as the set of integer points (x, y) verifying :

$$\mu \leq ax - by < \mu + \omega$$

- a, b, μ, ω in $\mathbb{Z}$
- $gcd(a, b) = 1$, (b, a) main vector of the line
- noted $\mathcal{D}(a, b, \mu, \omega)$

- if $\omega = max(|a|, |b|)$: $\mathcal{D}$ is 8-arc (naïve line).
- if $\omega < max(|a|, |b|)$: $\mathcal{D}$ is deconnected.
- if $\omega = |a| + |b|$: $\mathcal{D}$ is 4-arc (standard line).



$\mathcal{D}(5, 8, -1, 8)$

2.1 Main idea of DSS recognition algorithms

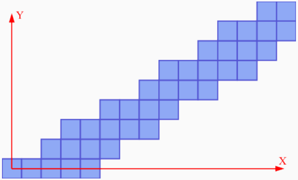
Arithmetic definition of digital line [Réveilles 91]

A digital line with parameters (a, b, μ) and arithmetical thickness ω is defined as the set of integer points (x, y) verifying :

$$\mu \leq ax - by < \mu + \omega$$

- a, b, μ, ω in $\mathbb{Z}$
- $gcd(a, b) = 1$, (b, a) main vector of the line
- noted $\mathcal{D}(a, b, \mu, \omega)$

- if $\omega = max(|a|, |b|)$: $\mathcal{D}$ is 8-arc (naïve line).
- if $\omega < max(|a|, |b|)$: $\mathcal{D}$ is deconnected.
- if $\omega = |a| + |b|$: $\mathcal{D}$ is 4-arc (standard line).
- if $\omega > |a| + |b|$: $\mathcal{D}$ is called a thick line.

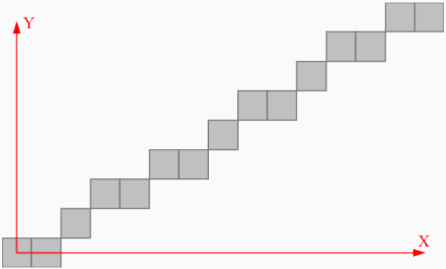


$\mathcal{D}(5, 8, -1, 8)$

2.1 Main idea of DSS recognition algorithms

Recognition Problem

- Goal: Recover the segment characteristics from a input sequence of points.
- Based on the remainder and periodicity detection.



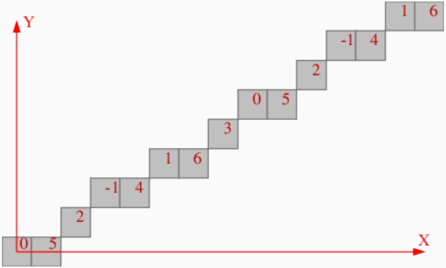
$\mathcal{D}(5, 8, -1, 13)$

2.1 Main idea of DSS recognition algorithms

Recognition Problem

- Goal: Recover the segment characteristics from a input sequence of points.
- Based on the remainder and periodicity detection.
- Remainder of a point M is defined as a function of $\mathcal{D}(a, b, \mu, \omega)$:

$$r_{\mathcal{D}}(M) = ax_M - by_M$$



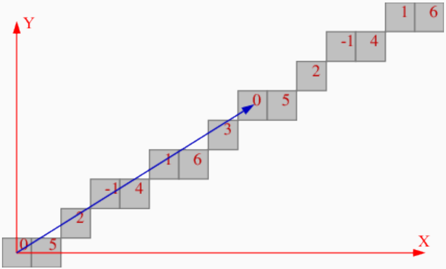
$\mathcal{D}(5, 8, -1, 13)$

2.1 Main idea of DSS recognition algorithms

Recognition Problem

- Goal: Recover the segment characteristics from a input sequence of points.
- Based on the remainder and periodicity detection.
- Remainder of a point M is defined as a function of $\mathcal{D}(a, b, \mu, \omega)$:

$$r_{\mathcal{D}}(M) = ax_M - by_M$$



$\mathcal{D}(5, 8, -1, 13)$

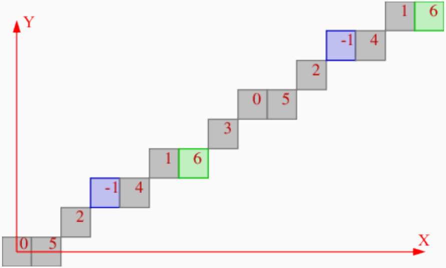
2.1 Main idea of DSS recognition algorithms

Recognition Problem

- Goal: Recover the segment characteristics from a input sequence of points.
- Based on the remainder and periodicity detection.
- Remainder of a point M is defined as a function of $\mathcal{D}(a, b, \mu, \omega)$:

$$r_{\mathcal{D}}(M) = ax_M - by_M$$

- Maintain the lower/upper leaning points.

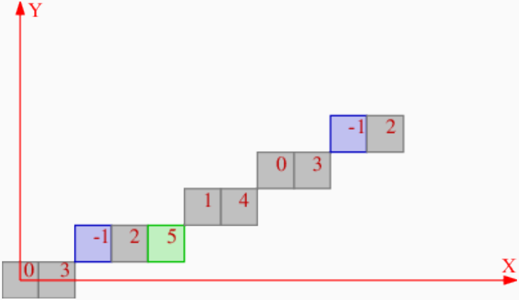


$\mathcal{D}(5, 8, -1, 13)$

2.1 Main idea of DSS recognition algorithms

Strategy of segment recognition ($\mathcal{S}$)

- Compute remainder of new point M.
- From $r(M)$ update characteristics.
- Update $\mathcal{S}$ parameters & leaning pts.

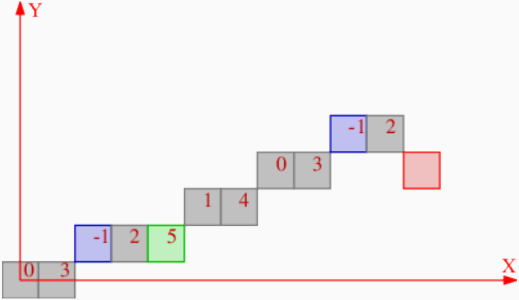


Recognized segment $\mathcal{S}$ of $\mathcal{D}_0(3,7,-1,7)$

2.1 Main idea of DSS recognition algorithms

Strategy of segment recognition ($\mathcal{S}$)

- Compute remainder of new point M.
- From $r(M)$ update characteristics.
- Update $\mathcal{S}$ parameters & leaning pts.

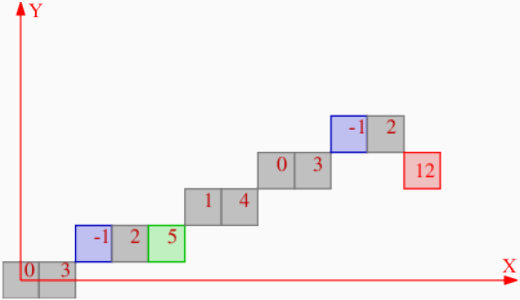


Recognized segment $\mathcal{S}$ of $\mathcal{D}_0(3, 7, -1, 7)$

2.1 Main idea of DSS recognition algorithms

Strategy of segment recognition ($\mathcal{S}$)

- Compute remainder of new point M.
- From $r(M)$ update characteristics.
- Update $\mathcal{S}$ parameters & leaning pts.



Recognized segment $\mathcal{S}$ of $\mathcal{D}_0(3, 7, -1, 7)$

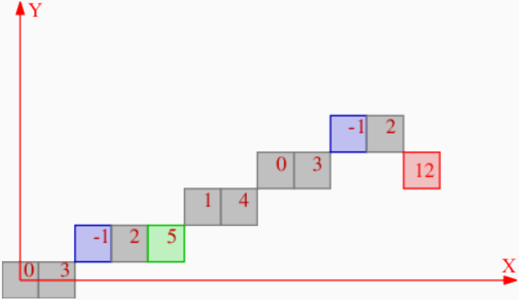
2.1 Main idea of DSS recognition algorithms

Strategy of segment recognition ($\mathcal{S}$)

- Compute remainder of new point M.
- From $r(M)$ update characteristics.
- Update $\mathcal{S}$ parameters & leaning pts.

Rules to update the characteristics of $\mathcal{S}$:

(*iv*) $r_{\mathcal{D}}(M) > \mu + max(|a|, |b|)$: M is strongly exterior to $\mathcal{D}$ and M not added to $\mathcal{S}$.



Recognized segment $\mathcal{S}$ of $\mathcal{D}_0(3, 7, -1, 7)$

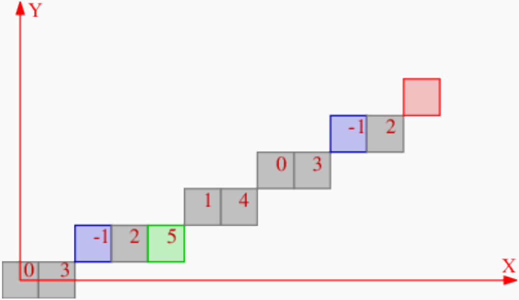
2.1 Main idea of DSS recognition algorithms

Strategy of segment recognition ($\mathcal{S}$)

- Compute remainder of new point M.
- From $r(M)$ update characteristics.
- Update $\mathcal{S}$ parameters & leaning pts.

Rules to update the characteristics of $\mathcal{S}$:

(*iv*) $r_{\mathcal{D}}(M) > \mu + max(|a|, |b|)$:
M is strongly exterior to $\mathcal{D}$ and M not added to $\mathcal{S}$.



Recognized segment $\mathcal{S}$ of $\mathcal{D}_0(3, 7, -1, 7)$

2.1 Main idea of DSS recognition algorithms

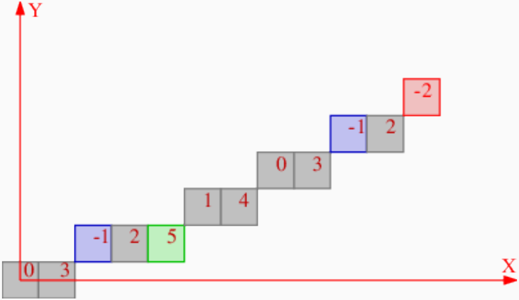
Strategy of segment recognition ($\mathcal{S}$)

- Compute remainder of new point M.
- From $r(M)$ update characteristics.
- Update $\mathcal{S}$ parameters & leaning pts.

Rules to update the characteristics of $\mathcal{S}$:

(*iii*) $r_{\mathcal{D}}(M) = \mu - 1$: M weakly exterior to $\mathcal{D}$,

M added to $\mathcal{S}$ and the slope is updated by the vector $U_F M$



Recognized segment $\mathcal{S}$ of $\mathcal{D}_0(3, 7, -1, 7)$

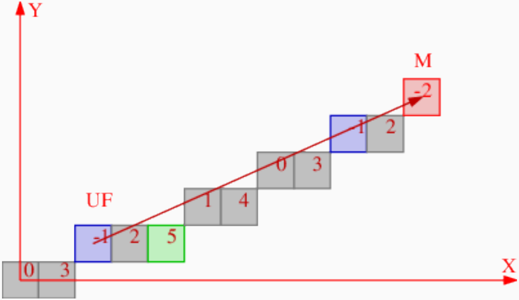
2.1 Main idea of DSS recognition algorithms

Strategy of segment recognition ($\mathcal{S}$)

- Compute remainder of new point M.
- From $r(M)$ update characteristics.
- Update $\mathcal{S}$ parameters & leaning pts.

Rules to update the characteristics of $\mathcal{S}$:

(iii) $r_{\mathcal{D}}(M) = \mu - 1$: M weakly exterior to $\mathcal{D}$,
M added to $\mathcal{S}$ and the slope is updated by the vector $U_F M$



Recognized segment $\mathcal{S}$ of $\mathcal{D}_0(3, 7, -1, 7)$

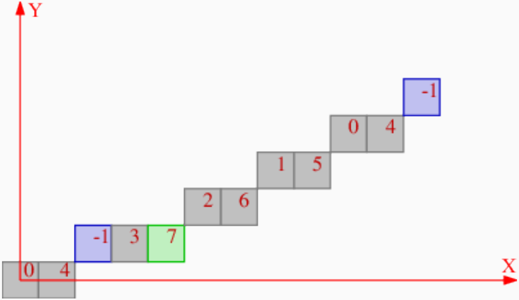
2.1 Main idea of DSS recognition algorithms

Strategy of segment recognition ($\mathcal{S}$)

- Compute remainder of new point M.
- From $r(M)$ update characteristics.
- Update $\mathcal{S}$ parameters & leaning pts.

Rules to update the characteristics of $\mathcal{S}$:

(iii) $r_{\mathcal{D}}(M) = \mu - 1$: M weakly exterior to $\mathcal{D}$,
M added to $\mathcal{S}$ and the slope is updated by the vector $U_F M$

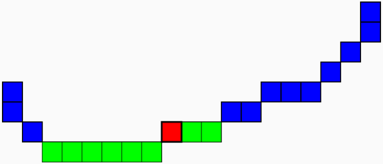


Recognized segment $\mathcal{S}$ of $\mathcal{D}_1(4, 9, -1, 9)$

2.1 Main idea of DSS recognition algorithms: maximal DSS

Primitive of Maximal Digital Straight Segment (MDSS)

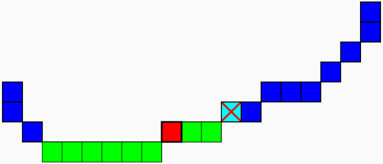
Let $\mathcal{C}$ be a digital curve, a segment of a naïve digital line is said maximal if it cannot be extended at the right and left hand sides on $\mathcal{C}$.



2.1 Main idea of DSS recognition algorithms: maximal DSS

Primitive of Maximal Digital Straight Segment (MDSS)

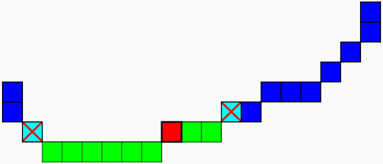
Let $\mathcal{C}$ be a digital curve, a segment of a naïve digital line is said maximal if it cannot be extended at the right and left hand sides on $\mathcal{C}$.



2.1 Main idea of DSS recognition algorithms: maximal DSS

Primitive of Maximal Digital Straight Segment (MDSS)

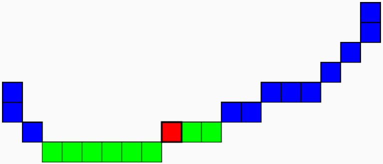
Let $\mathcal{C}$ be a digital curve, a segment of a naïve digital line is said maximal if it cannot be extended at the right and left hand sides on $\mathcal{C}$.



2.1 Main idea of DSS recognition algorithms: maximal DSS

Primitive of Maximal Digital Straight Segment (MDSS)

Let $\mathcal{C}$ be a digital curve, a segment of a naïve digital line is said maximal if it cannot be extended at the right and left hand sides on $\mathcal{C}$.



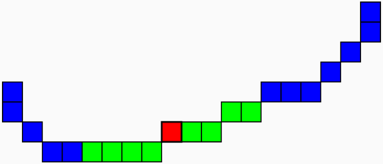
Sequence computation of maximal segments

Computable in linear type [Feschet and Tougne 99].

2.1 Main idea of DSS recognition algorithms: maximal DSS

Primitive of Maximal Digital Straight Segment (MDSS)

Let $\mathcal{C}$ be a digital curve, a segment of a naïve digital line is said maximal if it cannot be extended at the right and left hand sides on $\mathcal{C}$.



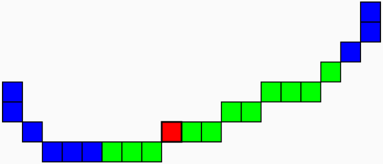
Sequence computation of maximal segments

Computable in linear type [Feschet and Tougne 99].

2.1 Main idea of DSS recognition algorithms: maximal DSS

Primitive of Maximal Digital Straight Segment (MDSS)

Let $\mathcal{C}$ be a digital curve, a segment of a naïve digital line is said maximal if it cannot be extended at the right and left hand sides on $\mathcal{C}$.



Sequence computation of maximal segments

Computable in linear type [Feschet and Tougne 99].

2.1 Main idea of DSS recognition algorithms: maximal DSS (2)

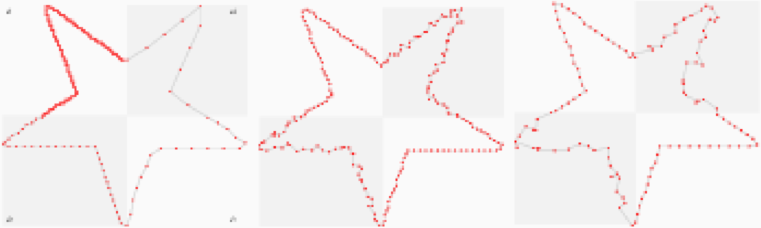
Advantage and limits of the MDSS

- + Gives a convergent technique to estimate geometric features like tangent, curvature.
- + Linear time algorithm.
- + Simple to implement and available in the DGtal Library.

2.1 Main idea of DSS recognition algorithms: maximal DSS (2)

Advantage and limits of the MDSS

- + Gives a convergent technique to estimate geometric features like tangent, curvature.
- + Linear time algorithm.
- + Simple to implement and available in the DGtal Library.
- - Limited to handle perfect digitized objects.
- - For real object it can be sensitive to noise.
- - Cannot process disconnected set of points.



2.2 Adaptation to noise

Primitive of Blurred Segment (or Alpha Thick Segments)

- Primitive able to handle noise.
- Can process disconnected set of points.
- No necessary integer coordinates.

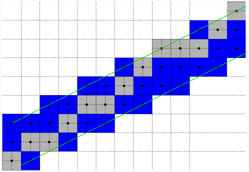
2.2 Adaptation to noise

Primitive of Blurred Segment (or Alpha Thick Segments)

- Primitive able to handle noise.
- Can process disconnected set of points.
- No necessary integer coordinates.

Overview

- Based on bounding line definition.



$\mathcal{D}(1, 2, -4, 6)$, bounding line of the sequence of grey points

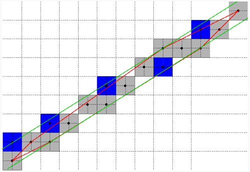
2.2 Adaptation to noise

Primitive of Blurred Segment (or Alpha Thick Segments)

- Primitive able to handle noise.
- Can process disconnected set of points.
- No necessary integer coordinates.

Overview

- Based on bounding line definition.
- Optimal Bounding line.



$\mathcal{D}(5, 8, -8, 11)$, optimal bounding line (width $\frac{10}{8} = 1.25$) of the sequence of grey points

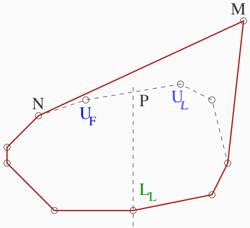
2.2 Adaptation to noise

Primitive of Blurred Segment (or Alpha Thick Segments)

- Primitive able to handle noise.
- Can process disconnected set of points.
- No necessary integer coordinates.

Overview

- Based on bounding line definition.
- Optimal Bounding line.
- Based on the convexhull computation.



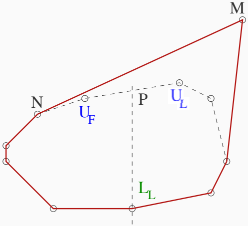
2.2 Adaptation to noise

Primitive of Blurred Segment (or Alpha Thick Segments)

- Primitive able to handle noise.
- Can process disconnected set of points.
- No necessary integer coordinates.

Overview

- Based on bounding line definition.
- Optimal Bounding line.
- Based on the convexhull computation.
- Thick param. to take noise into account.



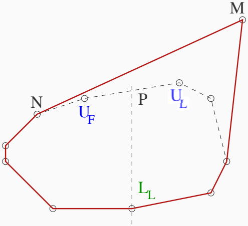
2.2 Adaptation to noise

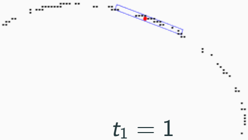
Primitive of Blurred Segment (or Alpha Thick Segments)

- Primitive able to handle noise.
- Can process disconnected set of points.
- No necessary integer coordinates.

Overview

- Based on bounding line definition.
- Optimal Bounding line.
- Based on the convexhull computation.
- Thick param. to take noise into account.





$t_1 = 1$

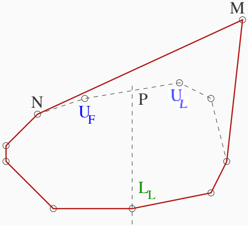
2.2 Adaptation to noise

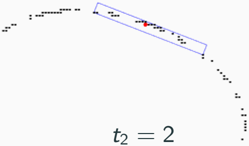
Primitive of Blurred Segment (or Alpha Thick Segments)

- Primitive able to handle noise.
- Can process disconnected set of points.
- No necessary integer coordinates.

Overview

- Based on bounding line definition.
- Optimal Bounding line.
- Based on the convexhull computation.
- Thick param. to take noise into account.





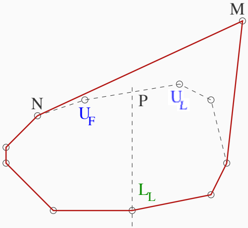
2.2 Adaptation to noise

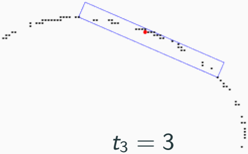
Primitive of Blurred Segment (or Alpha Thick Segments)

- Primitive able to handle noise.
- Can process disconnected set of points.
- No necessary integer coordinates.

Overview

- Based on bounding line definition.
- Optimal Bounding line.
- Based on the convexhull computation.
- Thick param. to take noise into account.





$t_3 = 3$

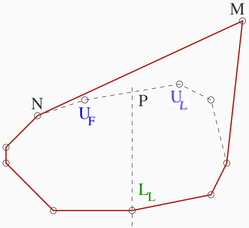
2.2 Adaptation to noise

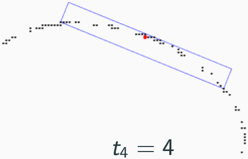
Primitive of Blurred Segment (or Alpha Thick Segments)

- Primitive able to handle noise.
- Can process disconnected set of points.
- No necessary integer coordinates.

Overview

- Based on bounding line definition.
- Optimal Bounding line.
- Based on the convexhull computation.
- Thick param. to take noise into account.





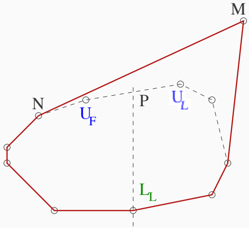
2.2 Adaptation to noise

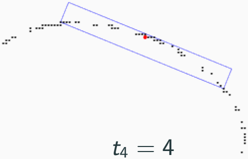
Primitive of Blurred Segment (or Alpha Thick Segments)

- Primitive able to handle noise.
- Can process disconnected set of points.
- No necessary integer coordinates.

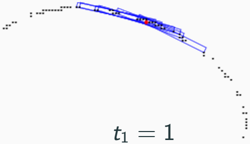
Overview

- Based on bounding line definition.
- Optimal Bounding line.
- Based on the convexhull computation.
- Thick param. to take noise into account.





$t_4 = 4$



$t_1 = 1$

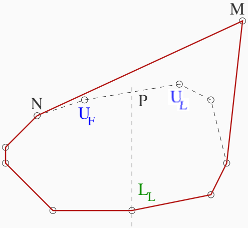
2.2 Adaptation to noise

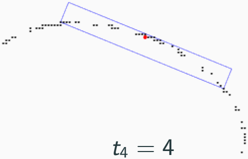
Primitive of Blurred Segment (or Alpha Thick Segments)

- Primitive able to handle noise.
- Can process disconnected set of points.
- No necessary integer coordinates.

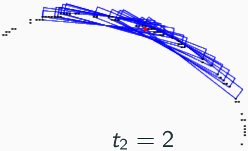
Overview

- Based on bounding line definition.
- Optimal Bounding line.
- Based on the convexhull computation.
- Thick param. to take noise into account.





$t_4 = 4$



$t_2 = 2$

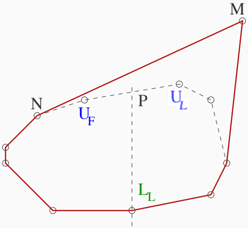
2.2 Adaptation to noise

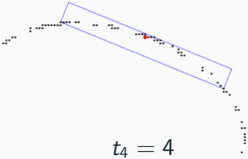
Primitive of Blurred Segment (or Alpha Thick Segments)

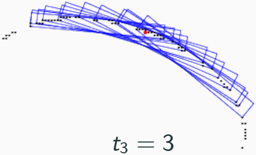
- Primitive able to handle noise.
- Can process disconnected set of points.
- No necessary integer coordinates.

Overview

- Based on bounding line definition.
- Optimal Bounding line.
- Based on the convexhull computation.
- Thick param. to take noise into account.




$t_4 = 4$


$t_3 = 3$

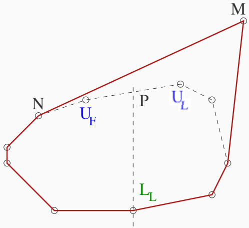
2.2 Adaptation to noise

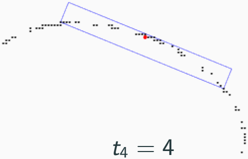
Primitive of Blurred Segment (or Alpha Thick Segments)

- Primitive able to handle noise.
- Can process disconnected set of points.
- No necessary integer coordinates.

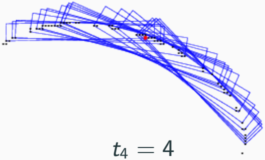
Overview

- Based on bounding line definition.
- Optimal Bounding line.
- Based on the convexhull computation.
- Thick param. to take noise into account.





$t_4 = 4$

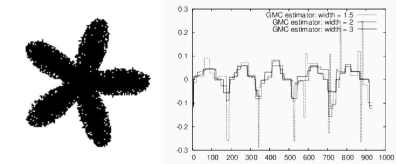


$t_4 = 4$

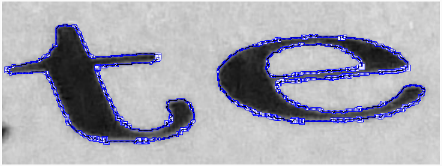
2.3 Applications of DSS

Overview of key applications:

- (1) Curvature estimator based on DSS.
- (2) Scale detection (noise).
- (3) Polygonalisation (arcs/segments).
- (4) Image vectorisation.



Curvature estimator (GMC)



noise estimator

Polygonalisation

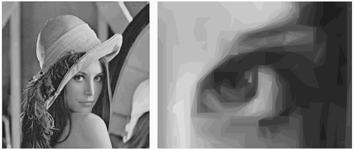


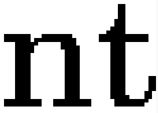
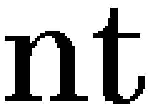
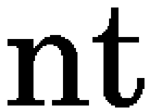
Image vectorisation

2.3 Applications of DSS: (1) Curvature estimator

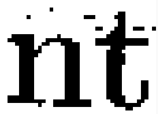
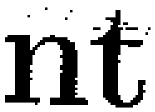
Objectives: [Kerautret and Lachaud 2009]

- Precise estimator even with low resolution.
- Adapted even on data with non perfect digitization or with noise.

Perfect Digitization

200 dpi	300 dpi	400 dpi
		

Result from printing and scan

200 dpi	300 dpi	400 dpi
		

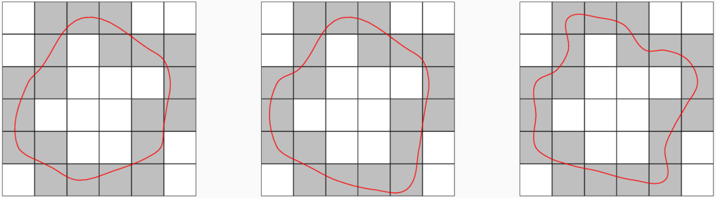
2.3 Applications of DSS: (1) Curvature estimator

Objectives: [Kerautret and Lachaud 2009]

- Precise estimator even with low resolution.
- Adapted even on data with non perfect digitization or with noise.

Main idea: (cf. deformable model)

- Take into account all the shapes having the same digitization.
- Retain the estimation corresponding to the shape having the highest probability (of lower energy).



2.3 Applications of DSS: (1) Curvature estimator

Objectives: [Kerautret and Lachaud 2009]

- Precise estimator even with low resolution.
- Adapted even on data with non perfect digitization or with noise.

Main idea: (cf. deformable model)

- Take into account all the shapes having the same digitization.
- Retain the estimation corresponding to the shape having the highest probability (of lower energy).

Realization:

- Best length estimator : minimize $\int ds$ [Sloboda *et al.* 98]

2.3 Applications of DSS: (1) Curvature estimator

Objectives: [Kerautret and Lachaud 2009]

- Precise estimator even with low resolution.
- Adapted even on data with non perfect digitization or with noise.

Main idea: (cf. deformable model)

- Take into account all the shapes having the same digitization.
- Retain the estimation corresponding to the shape having the highest probability (of lower energy).

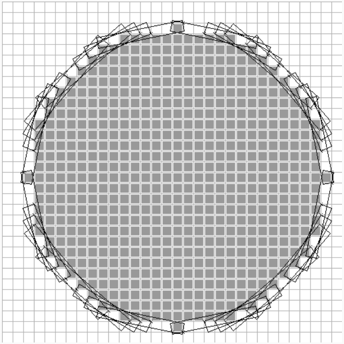
Realization:

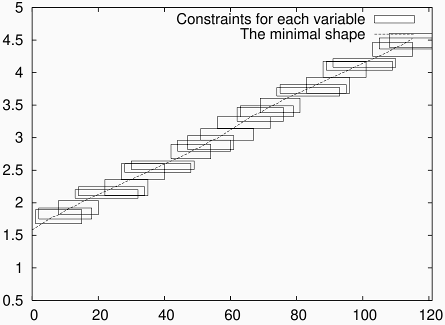
- Best length estimator : minimize $\int ds$ [Sloboda *et al.* 98]
- Best curvature estimator: minimize $\int \kappa^2 ds$
  $\Rightarrow$ Computed in the space of maximal segments (tangential cover).

2.3 Applications of DSS: (1) Curvature estimator

Examples of tangential cover with uncertainty on the slope

- Every maximal segment presents θ_{min}, θ_{max} values.
- For each surfel we can deduce the angle θ_{min} et θ_{max} of the tangent: $\min(\theta^i_{min})$ and $\max(\theta^i_{max})$.

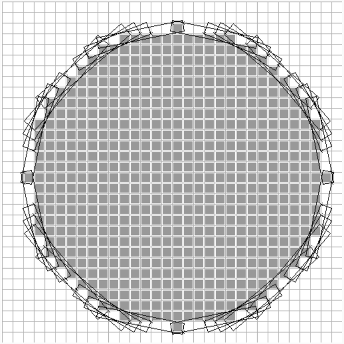


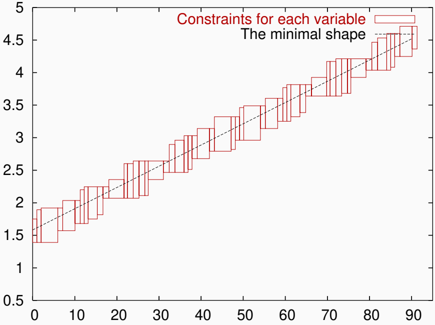


2.3 Applications of DSS: (1) Curvature estimator

Examples of tangential cover with uncertainty on the slope

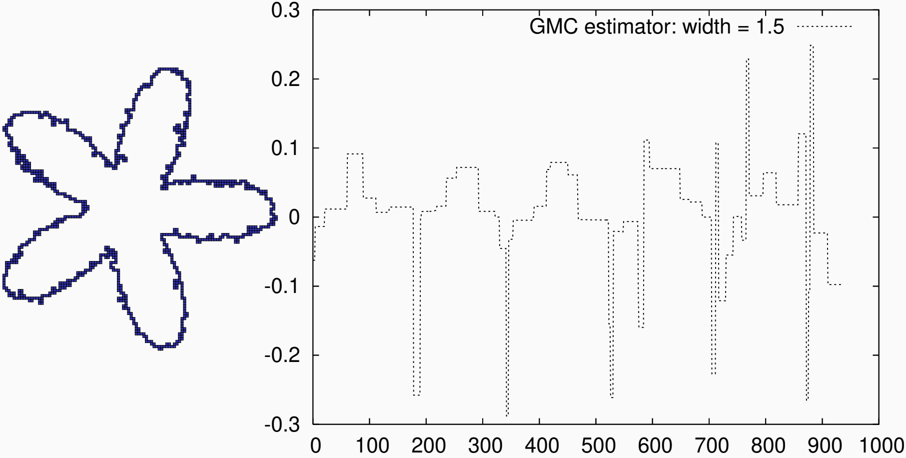
- Every maximal segment presents θ_{min}, θ_{max} values.
- For each surfel we can deduce the angle θ_{min} et θ_{max} of the tangent: $\min(\theta^i_{min})$ and $\max(\theta^i_{max})$.





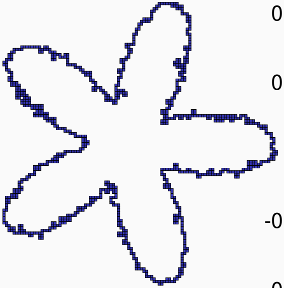
2.3 Applications of DSS: (1) Curvature estimator

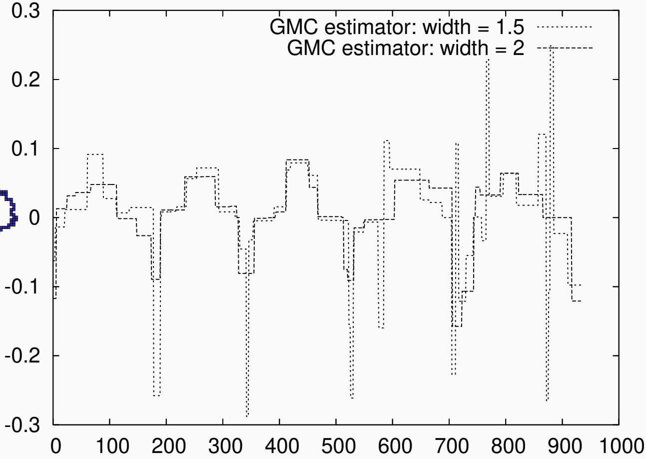
Results of curvature : on noisy contours



2.3 Applications of DSS: (1) Curvature estimator

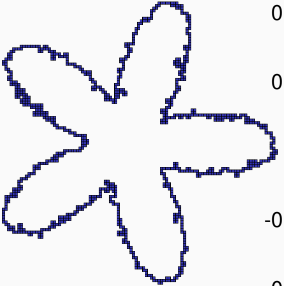
Results of curvature : on noisy contours

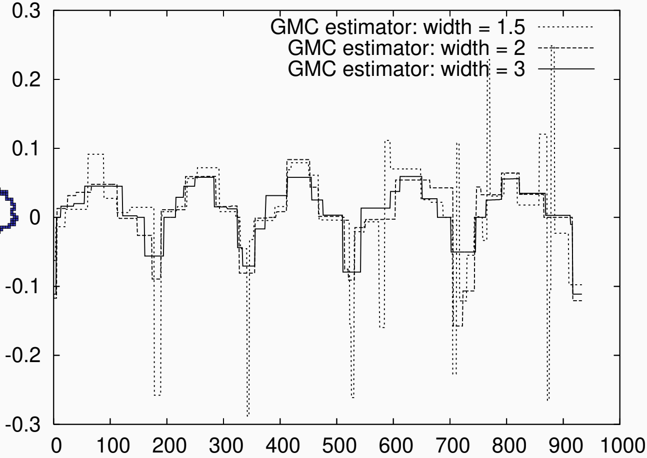




2.3 Applications of DSS: (1) Curvature estimator

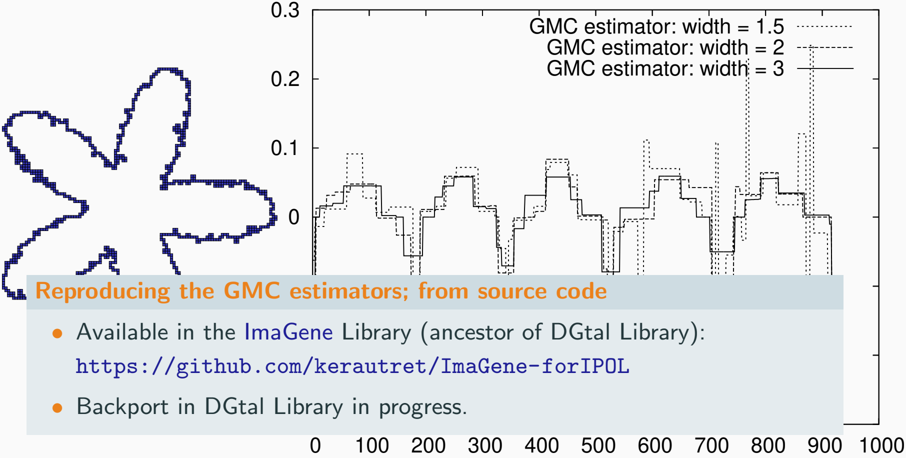
Results of curvature : on noisy contours





2.3 Applications of DSS: (1) Curvature estimator

Results of curvature : on noisy contours



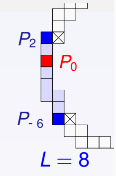
Reproducing the GMC estimators; from source code

- Available in the ImaGene Library (ancestor of DGtal Library): `https://github.com/kerautret/ImaGene-forIPOL`
- Backport in DGtal Library in progress.

2.3 Applications of DSS: (2) Scale detection

Main idea [Kerautret & Lachaud, 2012]

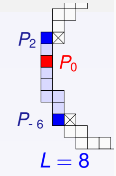
1. Exploit asymptotic properties of the Length (L) of maximal straight segments (valid on perfect shape digitizations).

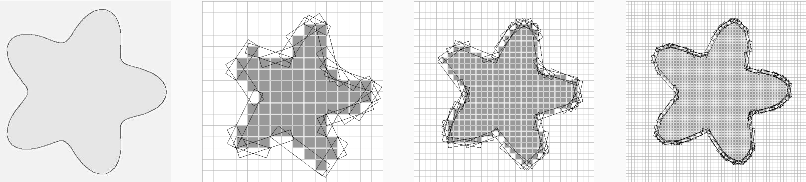


2.3 Applications of DSS: (2) Scale detection

Main idea [Kerautret & Lachaud, 2012]

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



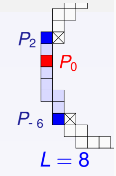

X $\mathrm{Dig}_2(X)$ $\mathrm{Dig}_1(X)$ $\mathrm{Dig}_{0,5}(X)$

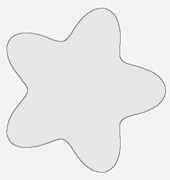
- X some simply connected compact shape of $\mathbb{R}^2$.
- $\mathrm{Dig}_h(X)$ = Gauss digitization of X with step h.

2.3 Applications of DSS: (2) Scale detection

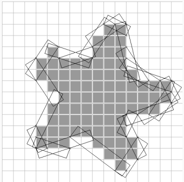
Main idea [Kerautret & Lachaud, 2012]

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.

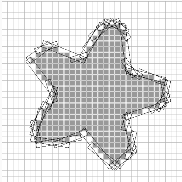




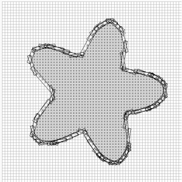
X



$\mathrm{Dig}_2(X)$



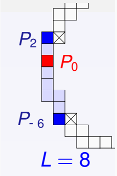
$\mathrm{Dig}_1(X)$



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

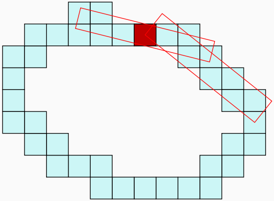
- X some simply connected compact shape of $\mathbb{R}^2$.
- $\mathrm{Dig}_h(X) =$ Gauss digitization of X with step h.

2.3 Applications of DSS: (2) Scale detection



Main idea [Kerautret & Lachaud, 2012]

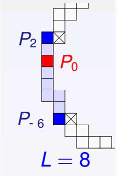
1. Exploit asymptotic properties of the Length (L) of maximal straight segments (valid on perfect shape digitizations).
2. They grow longer as h gets finer.

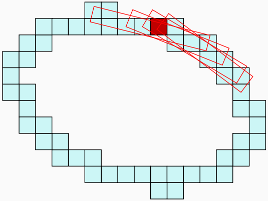


2.3 Applications of DSS: (2) Scale detection

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.

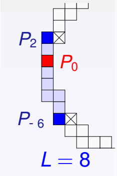


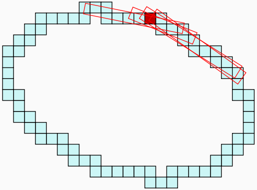


2.3 Applications of DSS: (2) Scale detection

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.

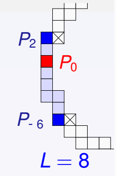


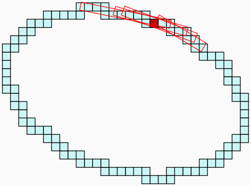


2.3 Applications of DSS: (2) Scale detection

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.

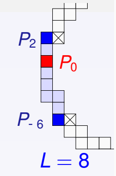


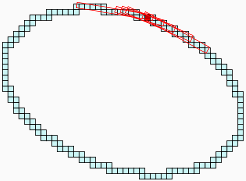


2.3 Applications of DSS: (2) Scale detection

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.

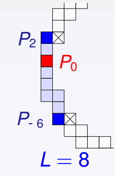


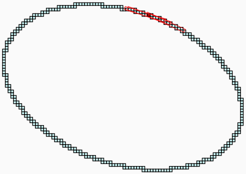


2.3 Applications of DSS: (2) Scale detection

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.

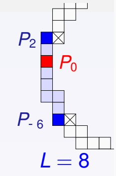




2.3 Applications of DSS: (2) Scale detection

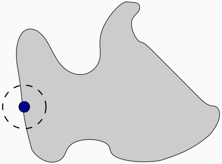
Main idea [Kerautret & Lachaud, 2012]

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.



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

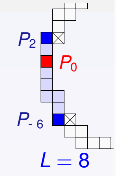
- X simply connected shape in R^2 with piecewise C^3 boundary ∂X,
- U an open connected neighborhood of $p \in \partial X$,



2.3 Applications of DSS: (2) Scale detection

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.

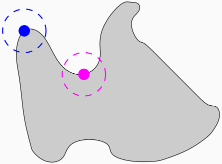


Theorem [Lachaud 06]: asymptotic behavior of the length of maximal segments

- 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,

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

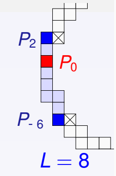
$$\partial X \cap U \text{ has null curvature, then } \Omega(1/h) \leq \quad L_j^h \quad \leq O(1/h^{1}) \tag{2}$$



2.3 Applications of DSS: (2) Scale detection

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.

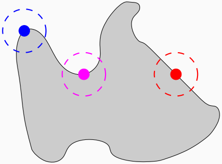


Theorem [Lachaud 06]: asymptotic behavior of the length of maximal segments

- 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,

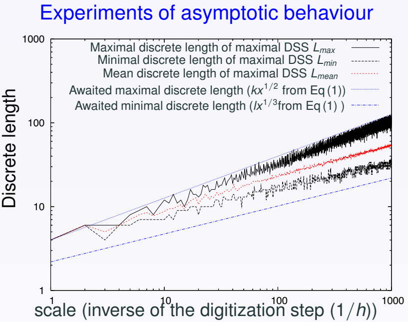
$$\partial X \cap U \text{ convex or concave, then } \Omega(1/h^{1/3}) \leq \quad L_j^h \quad \leq O(1/h^{1/2}) \qquad (1)$$

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



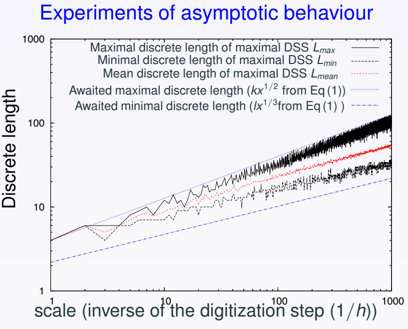
2.3 Applications of DSS: (2) Scale detection

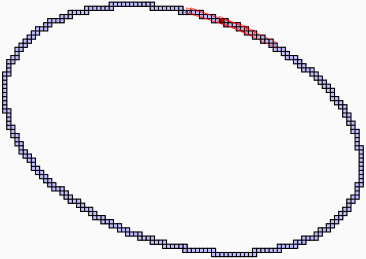
Experiments about reverse asymptotic behavior:



2.3 Applications of DSS: (2) Scale detection

Experiments about reverse asymptotic behavior:

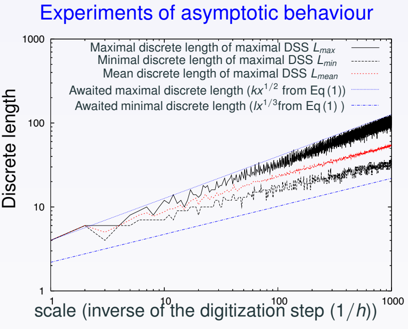


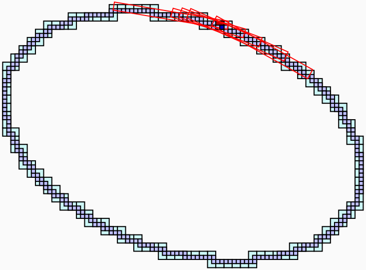


grid size 1

2.3 Applications of DSS: (2) Scale detection

Experiments about reverse asymptotic behavior:

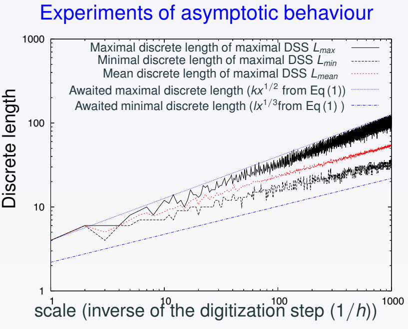


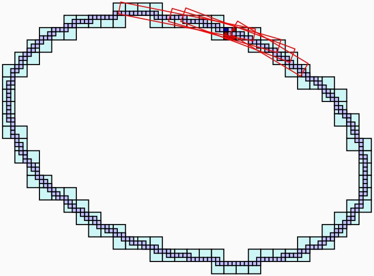


grid size 2

2.3 Applications of DSS: (2) Scale detection

Experiments about reverse asymptotic behavior:

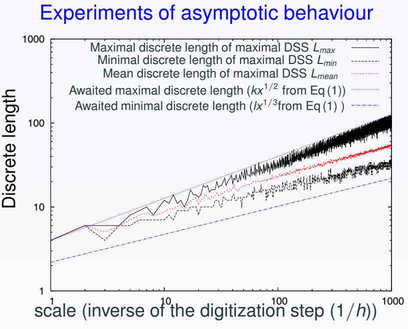


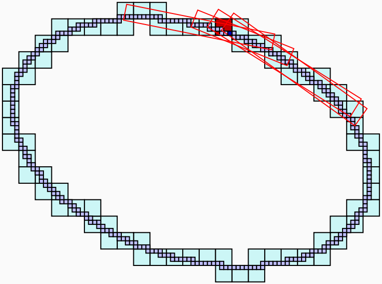


grid size 3

2.3 Applications of DSS: (2) Scale detection

Experiments about reverse asymptotic behavior:

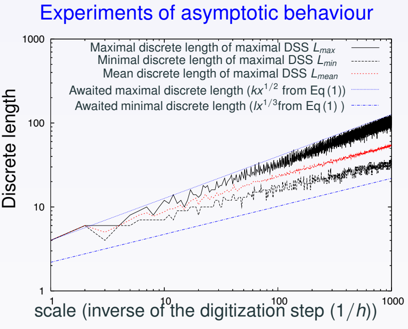


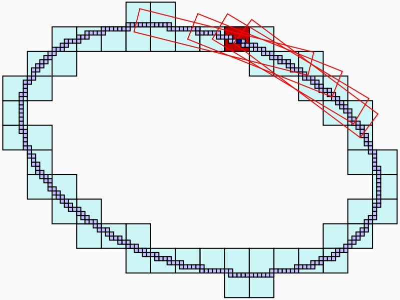


grid size 4

2.3 Applications of DSS: (2) Scale detection

Experiments about reverse asymptotic behavior:

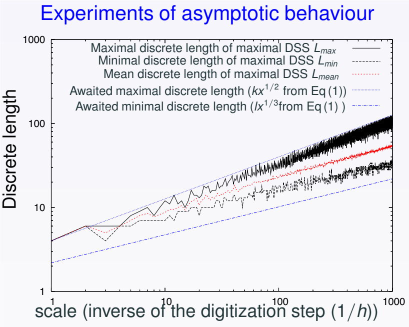


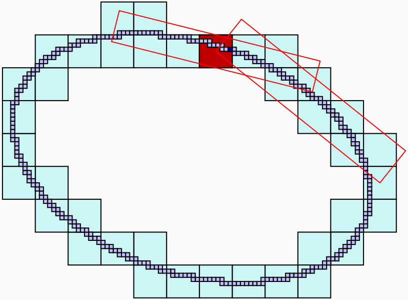


grid size 6

2.3 Applications of DSS: (2) Scale detection

Experiments about reverse asymptotic behavior:

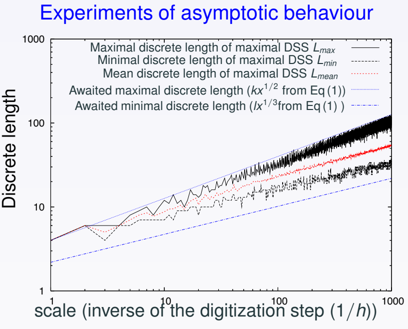




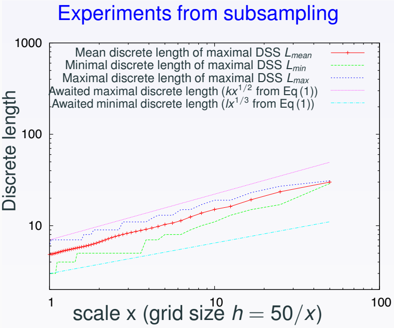
grid size 8

2.3 Applications of DSS: (2) Scale detection

Experiments about reverse asymptotic behavior:



Experiments of asymptotic behaviour



Experiments from subsampling

Local meaningful scale and noise detection

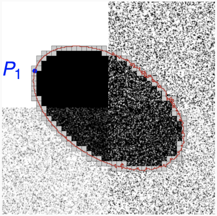
Meaningful scale:

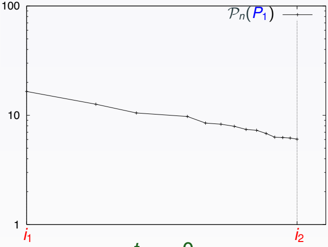
A meaningful scale of a multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$ is the pair (i_1, i_2) $1 \leq i_1 \leq i_2 \leq n$ such that for all i, $i_1 \leq i < i_2$,

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

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

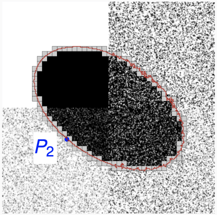
Meaningful scale:

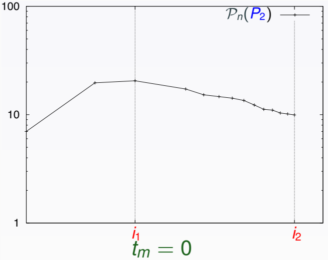
A meaningful scale of a multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$ is the pair (i_1, i_2) $1 \leq i_1 \leq i_2 \leq n$ such that for all i, $i_1 \leq i < i_2$,

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

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

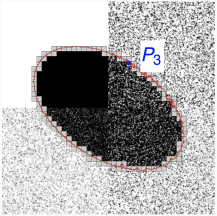
Meaningful scale:

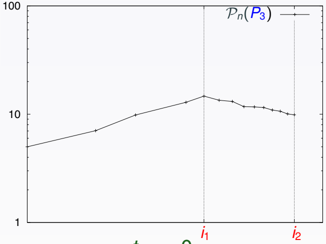
A meaningful scale of a multi-scale profile $(X_i, Y_i)_{1 \leq i \leq n}$ is the pair (i_1, i_2) $1 \leq i_1 \leq i_2 \leq n$ such that for all i, $i_1 \leq i < i_2$,

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

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

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

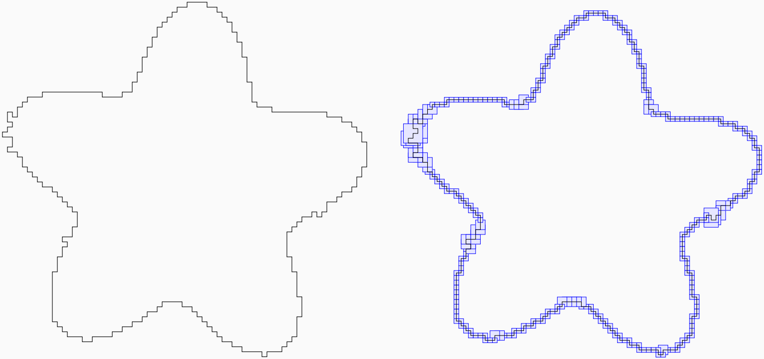




$t_m = 0$

Experiments: local noise level detection

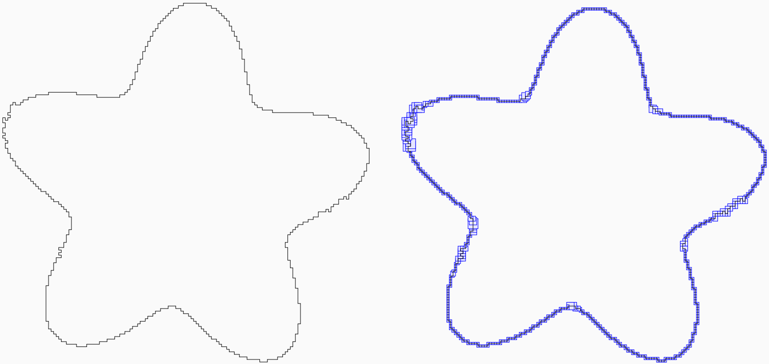
Flower with local noise



Local noise on resolution $R0$

Experiments: local noise level detection

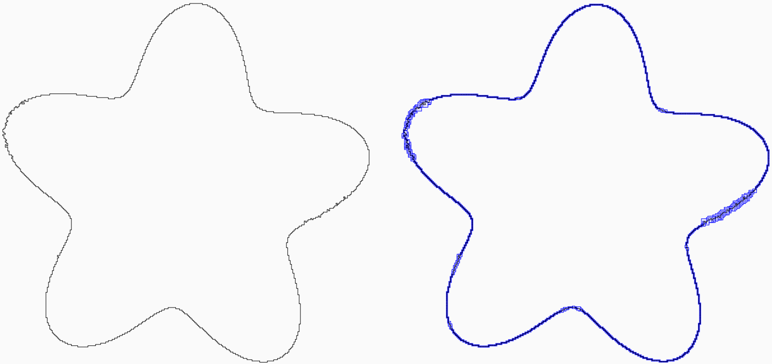
Flower with local noise



Local noise on resolution $R1$

Experiments: local noise level detection

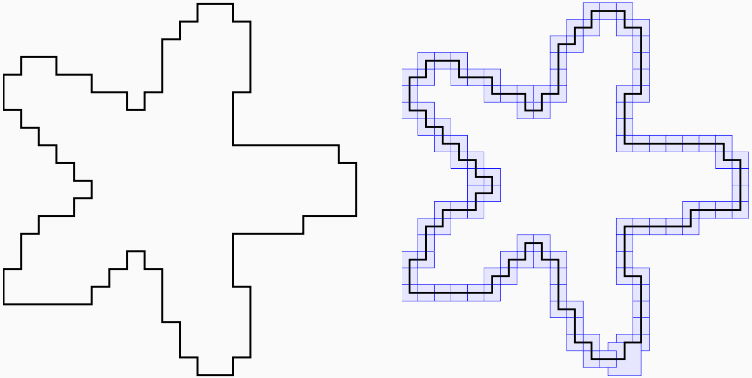
Flower with local noise



Local noise on resolution $R2$

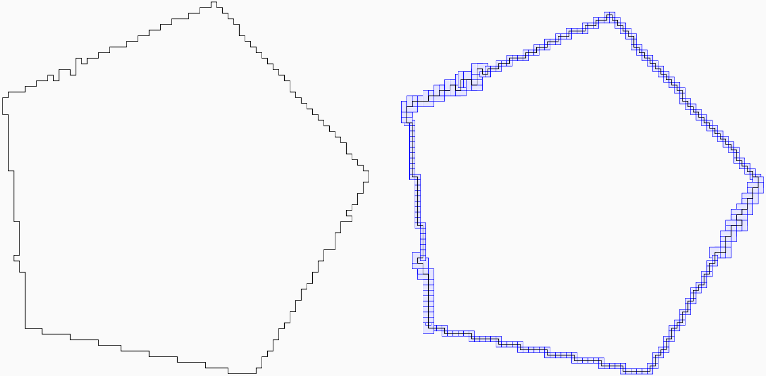
Experiments: local noise level detection

Tiny flower without noise



Local noise detection

Polygon with local noise



Local noise on resolution $R0$

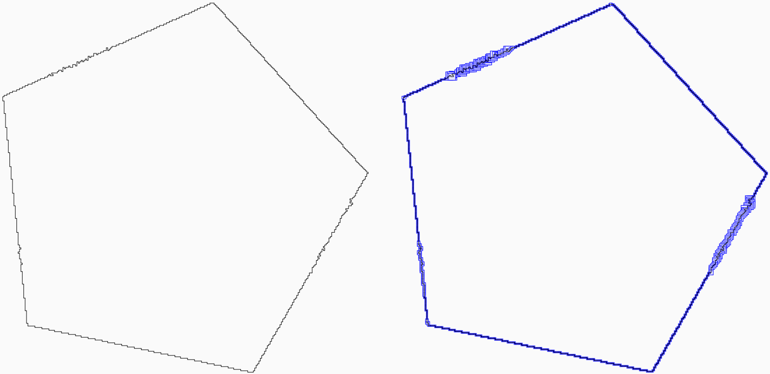
Local noise detection

Polygon with local noise

Local noise on resolution $R1$

Local noise detection

Polygon with local noise



Local noise on resolution $R2$

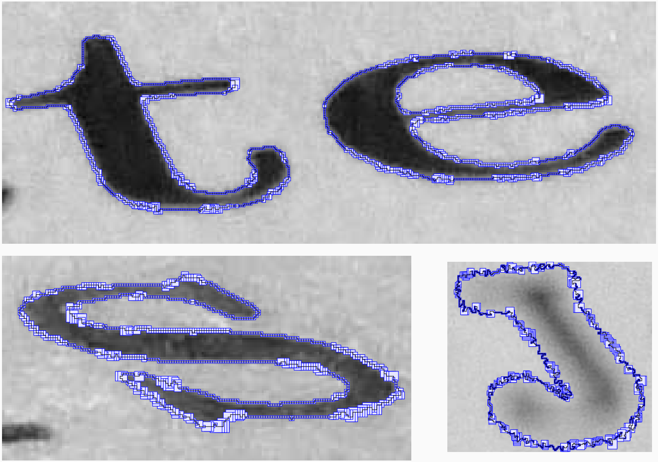
Local noise detection

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

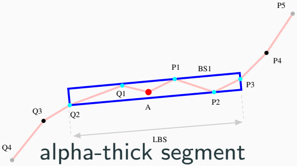
Noise detection on real images



2.3 Applications of DSS: (2) Scale detection

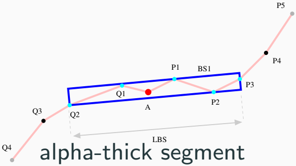
Multi-thickness Profile

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



alpha-thick segment

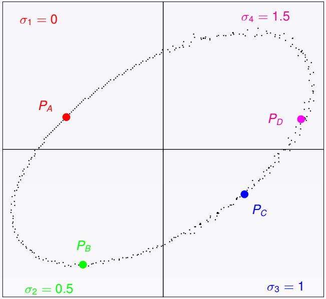
2.3 Applications of DSS: (2) Scale detection

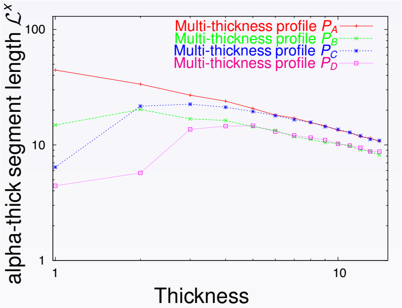


alpha-thick segment

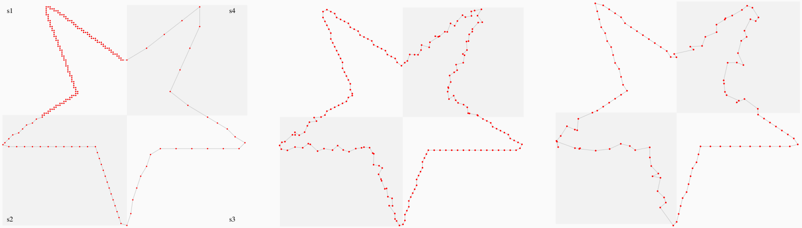
Multi-thickness Profile

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

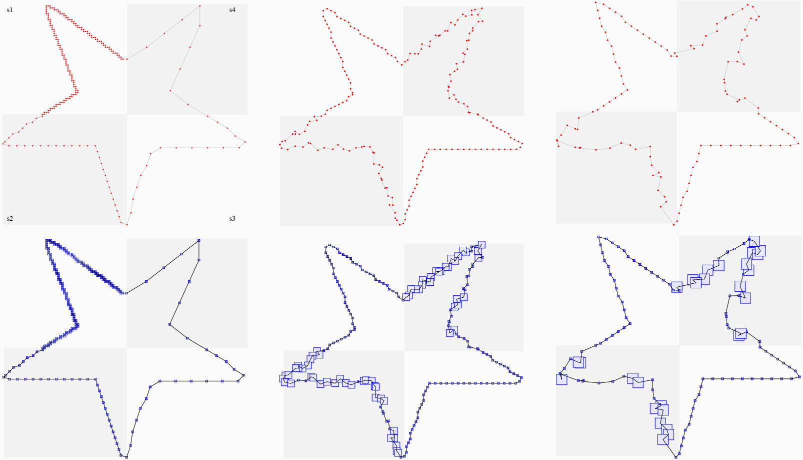




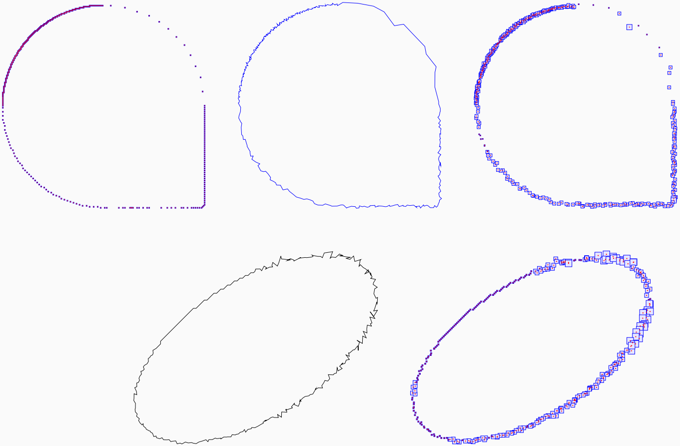
Experiments on polygonal shapes (1)



Experiments on polygonal shapes (1)



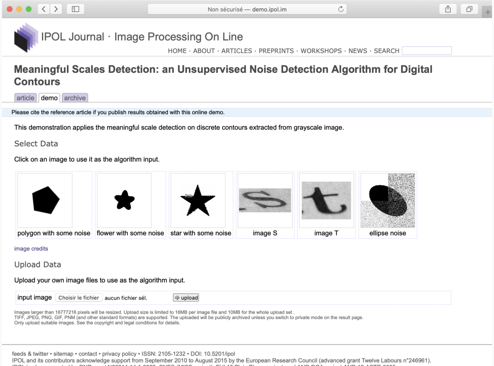
Experiments on polygonal shapes (2)



Reproduction of the results

Online demonstration available on IPOL: [Kerautret & Lachaud, 2014]

- The algorithm can be tested online: `http://www.ipol.im/pub/art/2014/75/`
- IPOL article with source code (based on the ImaGene Library).
- Reproducible in DGtal (with Alpha-Thick Segments), see examples of tutorial.



IPOL Journal · Image Processing On Line

Meaningful Scales Detection: an Unsupervised Noise Detection Algorithm for Digital Contours

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

This demonstration applies the meaningful scale detection on discrete contours extracted from grayscale image.

Select Data

Click on an image to use it as the algorithm input.

Upload Data

Upload your own image files to use as the algorithm input.

input image Choisir le fichier aucun fichier sél. upload

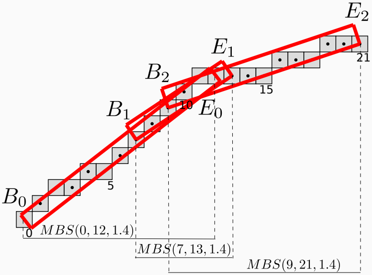
Images larger than 16777216 pixels will be resized. Upload size is limited to 16MB per image file and 10MB for the whole upload set .
TIFF, JPEG, PNG, GIF, PNM (and other standard formats) are supported. The uploaded will be publicly archived unless you switch to private mode on the result page.
Only upload suitable images. See the copyright and legal conditions for details.

feeds & twitter • sitemap • contact • privacy policy • ISSN: 2105-1232 • DOI: 10.5201/ipol
IPOL and its contributors acknowledge support from September 2010 to August 2015 by the European Research Council (advanced grant Twelve Labours n°246961).

2.3 Applications of DSS: (3) Image vectorisation

Selection of polygonalisation algorithms [Kerautret *et al.* 17]

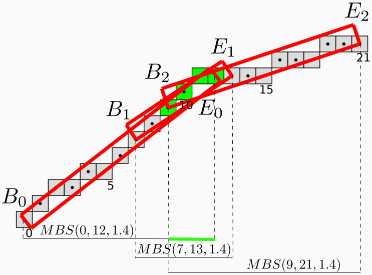
- Dominant points based polygonalization (DPP) [Nguyen 11].



2.3 Applications of DSS: (3) Image vectorisation

Selection of polygonalisation algorithms [Kerautret *et al.* 17]

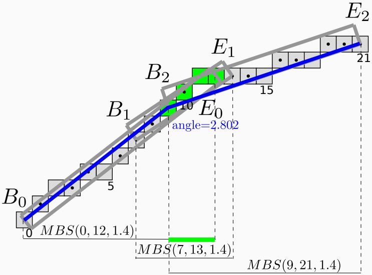
- Dominant points based polygonalization (DPP) [Nguyen 11].



2.3 Applications of DSS: (3) Image vectorisation

Selection of polygonalisation algorithms [Kerautret *et al.* 17]

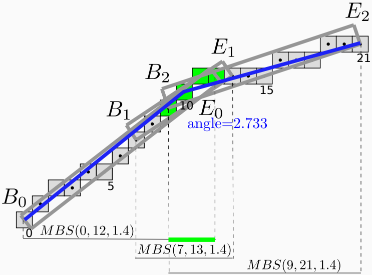
- Dominant points based polygonalization (DPP) [Nguyen 11].



2.3 Applications of DSS: (3) Image vectorisation

Selection of polygonalisation algorithms [Kerautret *et al.* 17]

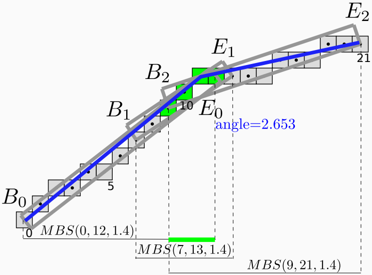
- Dominant points based polygonalization (DPP) [Nguyen 11].



2.3 Applications of DSS: (3) Image vectorisation

Selection of polygonalisation algorithms [Kerautret *et al.* 17]

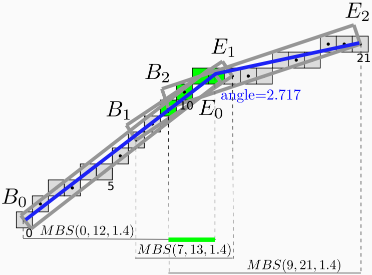
- Dominant points based polygonalization (DPP) [Nguyen 11].



2.3 Applications of DSS: (3) Image vectorisation

Selection of polygonalisation algorithms [Kerautret *et al.* 17]

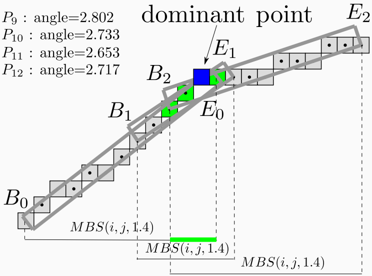
- Dominant points based polygonalization (DPP) [Nguyen 11].



2.3 Applications of DSS: (3) Image vectorisation

Selection of polygonalisation algorithms [Kerautret *et al.* 17]

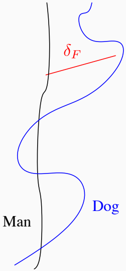
- Dominant points based polygonalization (DPP) [Nguyen 11].



2.3 Applications of DSS: (3) Image vectorisation

Selection of polygonalisation algorithms [Kerautret *et al.* 17]

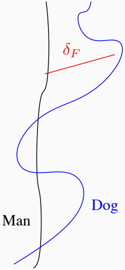
- Dominant points based polygonalization (DPP) [Nguyen 11].
- Polygonalization from Frechet dist. (minimal leash length) [Sivignon 11].



2.3 Applications of DSS: (3) Image vectorisation

Selection of polygonalisation algorithms [Kerautret *et al.* 17]

- Dominant points based polygonalization (DPP) [Nguyen 11].
- Polygonalization from Frechet dist. (minimal leash length) [Sivignon 11].

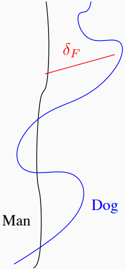


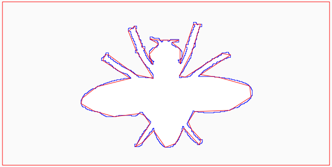
$\epsilon = 1$

2.3 Applications of DSS: (3) Image vectorisation

Selection of polygonalisation algorithms [Kerautret *et al.* 17]

- Dominant points based polygonalization (DPP) [Nguyen 11].
- Polygonalization from Frechet dist. (minimal leash length) [Sivignon 11].



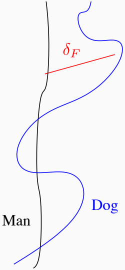


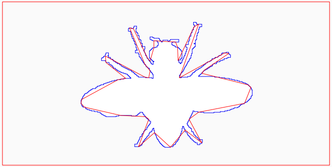
$\epsilon = 5$

2.3 Applications of DSS: (3) Image vectorisation

Selection of polygonalisation algorithms [Kerautret *et al.* 17]

- Dominant points based polygonalization (DPP) [Nguyen 11].
- Polygonalization from Frechet dist. (minimal leash length) [Sivignon 11].



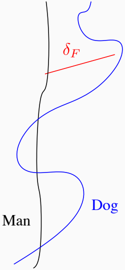


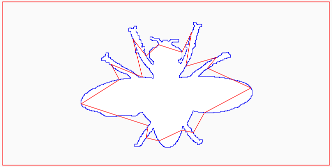
$\epsilon = 10$

2.3 Applications of DSS: (3) Image vectorisation

Selection of polygonalisation algorithms [Kerautret *et al.* 17]

- Dominant points based polygonalization (DPP) [Nguyen 11].
- Polygonalization from Frechet dist. (minimal leash length) [Sivignon 11].



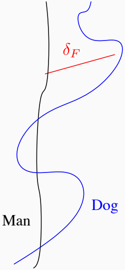


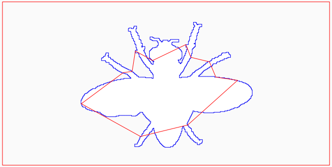
$\epsilon = 25$

2.3 Applications of DSS: (3) Image vectorisation

Selection of polygonalisation algorithms [Kerautret *et al.* 17]

- Dominant points based polygonalization (DPP) [Nguyen 11].
- Polygonalization from Frechet dist. (minimal leash length) [Sivignon 11].



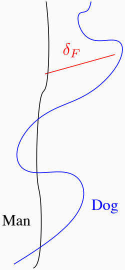


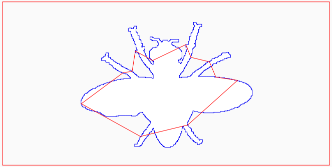
$\epsilon = 50$

2.3 Applications of DSS: (3) Image vectorisation

Selection of polygonalisation algorithms [Kerautret *et al.* 17]

- Dominant points based polygonalization (DPP) [Nguyen 11].
- Polygonalization from Frechet dist. (minimal leash length) [Sivignon 11].
- Extract from local maxima from curvature (GMC or other).



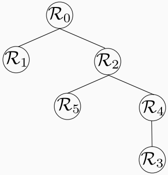


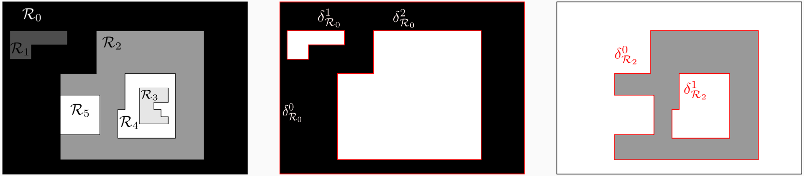
$\epsilon = 50$

2.3 Applications of DSS: (3) Image vectorisation

Component Tree representation [Najman & Couprie 06]

- Constructed from the intensity thresholds.
- Starting from the root (full image),
- to the node representing the included regions.

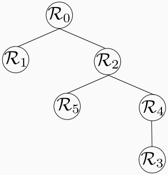




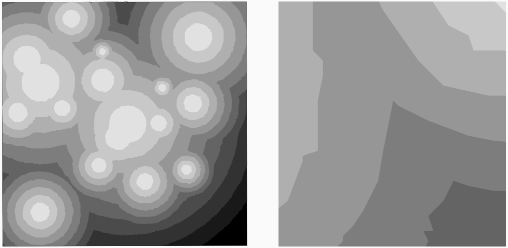
2.3 Applications of DSS: (3) Image vectorisation

Component Tree representation [Najman & Couprie 06]

- Constructed from the intensity thresholds.
- Starting from the root (full image),
- to the node representing the included regions.



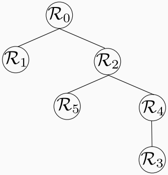
Representation from component tree



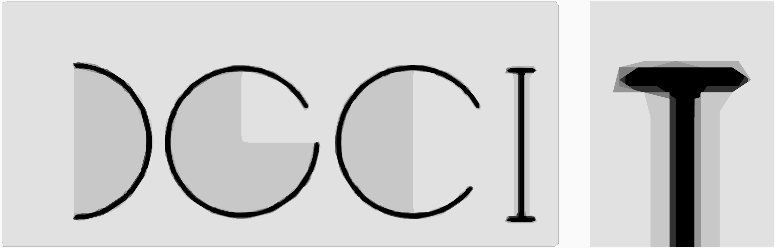
2.3 Applications of DSS: (3) Image vectorisation

Component Tree representation [Najman & Couprie 06]

- Constructed from the intensity thresholds.
- Starting from the root (full image),
- to the node representing the included regions.



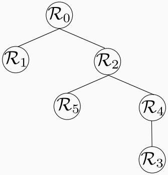
Representation from component tree



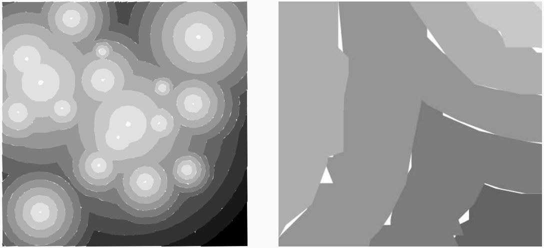
2.3 Applications of DSS: (3) Image vectorisation

Component Tree representation [Najman & Couprie 06]

- Constructed from the intensity thresholds.
- Starting from the root (full image),
- to the node representing the included regions.



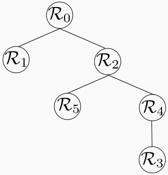
Representation using simple filling



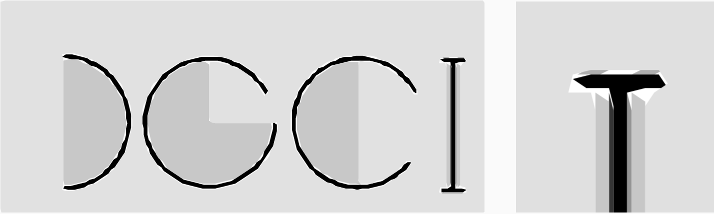
2.3 Applications of DSS: (3) Image vectorisation

Component Tree representation [Najman & Couprie 06]

- Constructed from the intensity thresholds.
- Starting from the root (full image),
- to the node representing the included regions.



Representation using simple filling

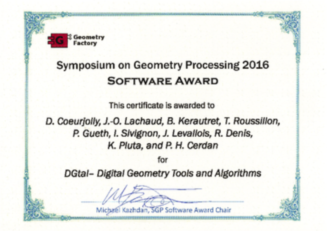


3. DGtal Library Overview

3.1 Short presentation of the library

Origin/evolution: (`www.dgtal.org`)

- DGtal: Digital Geometry tools and Algorithms
- Mainly a French initiative from the Discrete Geometry communauty.
- Born 10 year ago during the IWCIA workshop (end of november 2009) 🎂
- C++ based library: work (and tested) on *Linux*, *MacOS* and *Windows*.
- Current version 1.0 (from March 2019).
- SGP Software Award at the Symposium on Geometry Processing:



Geometry Factory

Symposium on Geometry Processing 2016

SOFTWARE AWARD

This certificate is awarded to

D. Coeurjolly, J.-O. Lachaud, B. Kerautret, T. Roussillon, P. Gueth, I. Sivignon, J. Levallois, R. Denis, K. Pluta, and P. H. Cerdan

for

DGtal– Digital Geometry Tools and Algorithms

Michael Kazhdan, SGP Software Award Chair

3.1 Short presentation of the library

Origin/evolution: (`www.dgtal.org`)

- DGtal: Digital Geometry tools and Algorithms
- Mainly a French initiative from the Discrete Geometry communauty.
- Born 10 year ago during the IWCIA workshop (end of november 2009) 🎂
- C++ based library: work (and tested) on *Linux*, *MacOS* and *Windows*.
- Current version 1.0 (from March 2019).
- SGP Software Award at the Symposium on Geometry Processing:

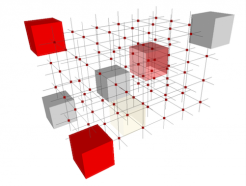
Main Objectives:

- Gathers in a unified setting many data structures and algorithms.
- For the discrete geometry communauty and related (digital topology, image processing, discrete geometry, arithmetic).
- It makes easier the appropriation of our tools for neophytes.
- Simplify comparisons of new methods with already existing approaches.
- Simplifies the construction of demonstration tools.

3.1 Short presentation of the library (2)

Main actual packages:

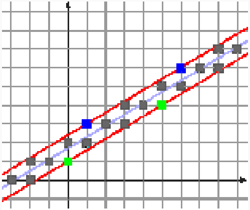
- Kernel package: number types, digital space, domain



3.1 Short presentation of the library (2)

Main actual packages:

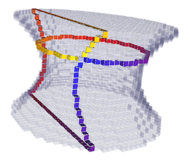
- Kernel package: number types, digital space, domain
- Arithmetic package: standard arithmetic computations
  ⇒ greatest common divisor, Bézout vectors, continued fractions, convergent, intersection of integer half-spaces



3.1 Short presentation of the library (2)

Main actual packages:

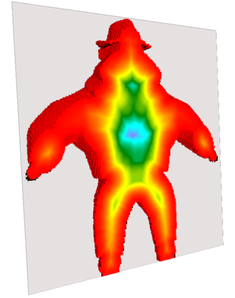
- Kernel package: number types, digital space, domain
- Arithmetic package: standard arithmetic computations
- Topology package: classic topology tools
  $\Rightarrow$ Rosenfeld oriented tools, cartesian cellular topology, digital surface topology (Herman), tools to extract connected component, simple points,...



3.1 Short presentation of the library (2)

Main actual packages:

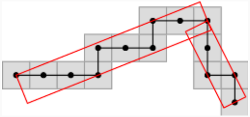
- Kernel package: number types, digital space, domain
- Arithmetic package: standard arithmetic computations
- Topology package: classic topology tools
- Geometry package: geometric estimators 2D/3D: ⇒ length, normal curvature estimators, 3D transform...



3.1 Short presentation of the library (2)

Main actual packages:

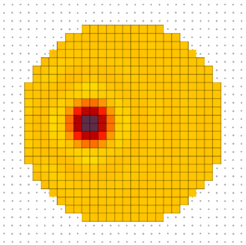
- Kernel package: number types, digital space, domain
- Arithmetic package: standard arithmetic computations
- Topology package: classic topology tools
- Geometry package: geometric estimators 2D/3D: ⇒ length, normal curvature estimators, 3D transform...



3.1 Short presentation of the library (2)

Main actual packages:

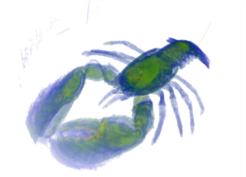
- Kernel package: number types, digital space, domain
- Arithmetic package: standard arithmetic computations
- Topology package: classic topology tools
- Geometry package: geometric estimators 2D/3D:
- DEC package: Discrete exterior calculus:
  ⇒ provides an easy and efficient way to describe linear operator over various structure



3.1 Short presentation of the library (2)

Main actual packages:

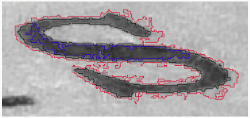
- Kernel package: number types, digital space, domain
- Arithmetic package: standard arithmetic computations
- Topology package: classic topology tools
- Geometry package: geometric estimators 2D/3D:
- DEC package: Discrete exterior calculus:
- Board & Viewer package: import/export image and visualization:
  ⇒ interactive and non interactive viewer 2d/3d...



3.1 Short presentation of the library (2)

Main actual packages:

- Kernel package: number types, digital space, domain
- Arithmetic package: standard arithmetic computations
- Topology package: classic topology tools
- Geometry package: geometric estimators 2D/3D:
- DEC package: Discrete exterior calculus:
- Board & Viewer package: import/export image and visualization:
- Image package: implement image model and data-structures.



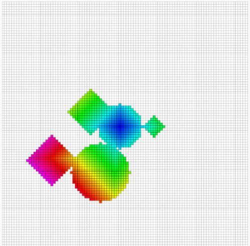
3.1 Short presentation of the library (2)

Main actual packages:

- Kernel package: number types, digital space, domain
- Arithmetic package: standard arithmetic computations
- Topology package: classic topology tools
- Geometry package: geometric estimators 2D/3D:
- DEC package: Discrete exterior calculus:
- Board & Viewer package: import/export image and visualization:
- Image package: implement image model and data-structures.
- Shape package: shape related concepts, models and algorithms.
  ⇒ generic framework and tools to construct multigrid shapes in DGtal

3.1 Short presentation of the library (2)

Main actual packages:

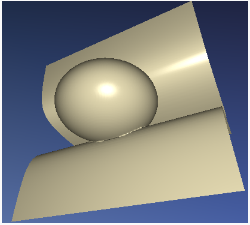


- Kernel package: number types, digital space, domain
- Arithmetic package: standard arithmetic computations
- Topology package: classic topology tools
- Geometry package: geometric estimators 2D/3D:
- DEC package: Discrete exterior calculus:
- Board & Viewer package: import/export image and visualization:
- Image package: implement image model and data-structures.
- Shape package: shape related concepts, models and algorithms.
- Graph package: gathers concepts and classes related to graphs.
  ⇒ with wrappers to boost::graph

3.1 Short presentation of the library (2)

Main actual packages:

- Kernel package: number types, digital space, domain
- Arithmetic package: standard arithmetic computations
- Topology package: classic topology tools
- Geometry package: geometric estimators 2D/3D:
- DEC package: Discrete exterior calculus:
- Board & Viewer package: import/export image and visualization:
- Image package: implement image model and data-structures.
- Shape package: shape related concepts, models and algorithms.
- Graph package: gathers concepts and classes related to graphs.
- Math package: various mathematical subpackages.



3.1 Short presentation of the library (3)

Library organization and details:

- Three main projects:
    - Main DGtal library (`https://github.com/DGtal-team/DGtal`).
    - DGtal-Tools project: contains tools based on DGtal (`https://github.com/DGtal-team/DGtal-Tools`).
    - DGtal-Tools-contrib: contains tools using DGtal. (`https://github.com/DGtal-team/DGtalTools-contrib`)

3.1 Short presentation of the library (3)

Library organization and details:

- Three main projects:
    - Main DGtal library (`https://github.com/DGtal-team/DGtal`).
    - DGtal-Tools project: contains tools based on DGtal (`https://github.com/DGtal-team/DGtal-Tools`).
    - DGtal-Tools-contrib: contains tools using DGtal. (`https://github.com/DGtal-team/DGtalTools-contrib`)
- CMake oriented compilation.

3.1 Short presentation of the library (3)

Library organization and details:

- Three main projects:
  - Main DGtal library (`https://github.com/DGtal-team/DGtal`).
  - DGtal-Tools project: contains tools based on DGtal (`https://github.com/DGtal-team/DGtal-Tools`).
  - DGtal-Tools-contrib: contains tools using DGtal. (`https://github.com/DGtal-team/DGtalTools-contrib`)
- CMake oriented compilation.
- Boost dependancies, and (optionals) LibQGLViewer, ITK, CGal,CAIRO, Eigen, GMP,...

3.1 Short presentation of the library (3)

Library organization and details:

- Three main projects:
  - Main DGtal library (`https://github.com/DGtal-team/DGtal`).
  - DGtal-Tools project: contains tools based on DGtal (`https://github.com/DGtal-team/DGtal-Tools`).
  - DGtal-Tools-contrib: contains tools using DGtal. (`https://github.com/DGtal-team/DGtalTools-contrib`)
- CMake oriented compilation.
- Boost dependancies, and (optionals) LibQGLViewer, ITK, CGal,CAIRO, Eigen, GMP,...

Programming principle:

- Generic Programming.
- Concept, models of concepts and concept checking.

⇒ C++ with template programming

3.1 Short presentation of the library (4)

First example, see: `https://github.com/kerautret/ACPR19-DGPRTutorial`

- Example to read input contour.
- Display the digital contour.
- Export the visualization.

3.1 Short presentation of the library (4)

First example, see: `https://github.com/kerautret/ACPR19-DGPRTutorial`

- Example to read input contour.
- Display the digital contour.
- Export the visualization.

(see file: tuto1_baseDGtal.cpp)

```
#include "DGtal/base/Common.h"
#include "DGtal/helpers/StdDefs.h"
// To use the reading of input points:
#include "DGtal/io/readers/PointListReader.h"

// To display graphics elements
#include "DGtal/io/boards/Board2D.h"
...
typedef Z2i::Point Point;
std::vector<Point> contour = PointListReader<Point>::getPointsFromFile("
     contour.sdp");

//Displaying the input read contour:
Board2D aBoard;
for (auto&& p :contour) { aBoard << p; }
aBoard.saveEPS("res.eps");
```

3.1 Short presentation of the library (4)

First example, see: `https://github.com/kerautret/ACPR19-DGPRTutorial`

- Example to read input contour.
- Display the digital contour.
- Export the visualization.

(see file: tuto1_baseDGtal.cpp)

```
#include "DGtal/base/Common.h"
#include "DGtal/helpers/StdDefs.h"
// To use the reading of input points:
#include "DGtal/io/readers/PointListReader.h"

// To display graphics elements
#include "DGtal/io/boards/Board2D.h"
...
typedef Z2i::Point Point;
std::vector<Point> contour = PointListReader<Point>::getPointsFromFile("
        contour.sdp");

//Displaying the input read contour:
Board2D aBoard;
for (auto&& p :contour) { aBoard << p; }
aBoard.saveEPS("res.eps");
```

3.2 Extracting level sets contours with DGtal

Second tutorial exercise (see `tuto2_LSC/README.md`)

Three main steps in DGtal:

- Create a Khalimsky space:

(see file: tuto2_LSC.cpp)

```
Z2i::KSpace ks;
ks.init(image.domain().lowerBound(),
        image.domain().upperBound(), false);
```

3.2 Extracting level sets contours with DGtal

Second tutorial exercise (see `tuto2_LSC/README.md`)

Three main steps in DGtal:

- Create a Khalimsky space:

(see file: tuto2_LSC.cpp)

```
Z2i::KSpace ks;
ks.init(image.domain().lowerBound(),
        image.domain().upperBound(), false);
```

- Extract a set of pixel of the image:

```
  Z2i::DigitalSet set (image.domain());
  SetFromImage<Z2i::DigitalSet>::append(set, image, 0, 108);
```

3.2 Extracting level sets contours with DGtal

Second tutorial exercise (see `tuto2_LSC/README.md`)

Three main steps in DGtal:

- Create a Khalimsky space:

(see file: tuto2_LSC.cpp)

```
Z2i::KSpace ks;
ks.init(image.domain().lowerBound(),
        image.domain().upperBound(), false);
```

- Extract a set of pixel of the image:

```
  Z2i::DigitalSet set (image.domain());
  SetFromImage<Z2i::DigitalSet>::append(set, image, 0, 108);
```

- Track intergrid Cell and display them from Freman Chains objects:

```
SurfelAdjacency<2> sAdj(true);
std::vector<std::vector<Z2i::Point>> vCnt;
Surfaces<Z2i::KSpace>::extractAllPointContours4C(vCnt, ks, set, sAdj);
  ...
for (const auto &c: vCnt)
    FreemanChain<int> fc (c);
    ...
```

3.2 Extracting level sets contours with DGtal

Second tutorial exercise (see `tuto2_LSC/README.md`)

Three main steps in DGtal:

- Create a Khalimsky space:

(see file: tuto2_LSC.cpp)

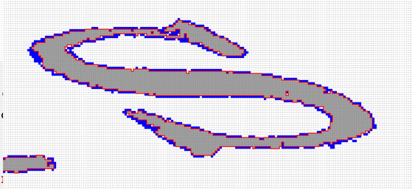
```
Z2i::KSpace ks;
ks.init(image.domain().lowerBound(),
        image.domain().upperBound(), false);
```

- Extract a set of pixel of the image:

```
  Z2i::DigitalSet set (image.domain());
  SetFromImage<Z2i::DigitalSet>::append(set, image, 0, 108);
```

- Track intergrid Cell   objects:

```
SurfelAdjacency<2>
std::vector<std::v
Surfaces<Z2i::KSpa                                       set, sAdj);
  ...
for (const auto &c
    FreemanChain<i
    ...
```



3.3 Example of geometric estimator

Third tutorial exercise (see `tuto3_curvatures/README.md`)

Computing curvature with DCA estimator [Roussillon & Lachaud 11].

⇒ Based on Digital Circular Arcs.

3.3 Example of geometric estimator

Third tutorial exercise (see `tuto3_curvatures/README.md`)

Computing curvature with DCA estimator [Roussillon & Lachaud 11].

⇒ Based on Digital Circular Arcs.

- Defines types for Range and Iterator on input curve:

(see file: tuto3_curvatures.cpp)

```
typedef GridCurve<>::IncidentPointsRange Range;
typedef Range::ConstIterator ClassicIterator;
Range r = curve.getIncidentPointsRange();
std::vector<double> estimations;
```

3.3 Example of geometric estimator

Third tutorial exercise (see `tuto3_curvatures/README.md`)

Computing curvature with DCA estimator [Roussillon & Lachaud 11].

⇒ Based on Digital Circular Arcs.

- Defines types for Range and Iterator on input curve:

(see file: tuto3_curvatures.cpp)

```
typedef GridCurve<>::IncidentPointsRange Range;
typedef Range::ConstIterator ClassicIterator;
Range r = curve.getIncidentPointsRange();
std::vector<double> estimations;
```

- Construct estimator and apply it:

```
SegmentComputer sc;
SCEstimator sce;
CurvatureEstimator estimator(sc, sce);
...

estimator.init( 1, r.begin(), r.end() );
estimator.eval( r.begin(), r.end(),
                std::back_inserter(estimations) );
```

3.3 Example of geometric estimator

Third tutorial exercise

Computing curvature w ud 11].

⇒ Based on Digital Cir

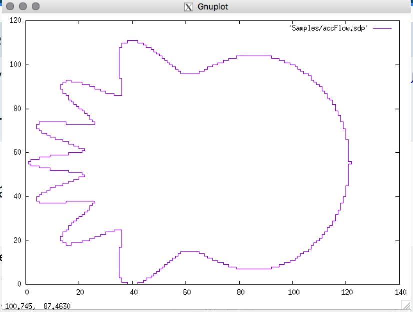
- Defines types for Ra

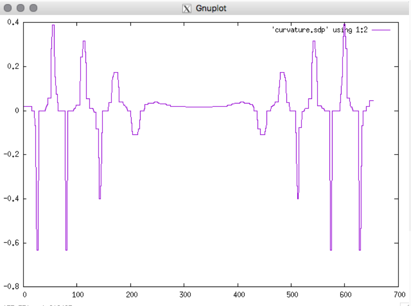
```
typedef GridCurve
typedef Range::C
Range r = curve.
std::vector<double> estimations;
```

- Construct estimator

```
SegmentComputer
SCEstimator sce;
CurvatureEstimat
...

estimator.init(
estimator.eval(
              s
```





4. Practical session: Hands on DGtal

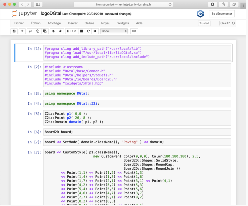
4. Practical session: Hands on DGtal

Pratical installation/exercices

Visit Github page: https://kerautret.github.io/ACPR19-DGPRTutorial

Test DGtal online with Jupyter notebook

- http://ker.iutsd.univ-lorraine.fr/notebook
- Login: use password: admin;123



```
In [1]: #pragma cling add_library_path("/usr/local/lib")
        #pragma cling load("/usr/local/lib/libDGtal.so")
        #pragma cling add_include_path("/usr/local/include")

In [2]: #include <iostream>
        #include "DGtal/base/Common.h"
        #include "DGtal/helpers/StdDefs.h"
        #include "DGtal/io/boards/Board2D.h"
        #include "xwidgets/xhtml.hpp"

In [3]: using namespace DGtal;

In [4]: using namespace DGtal::Z2i;

In [5]: Z2i::Point p1( 0,0 );
        Z2i::Point p2( 26, 8 );
        Z2i::Domain domain( p1, p2 );

In [6]: Board2D board;

In [7]: board << SetMode( domain.className(), "Paving" ) << domain;

In [7]: board << CustomStyle( p1.className(),
                              new CustomPen( Color(0,0,0), Color(180,180,180), 2.5,
                                             Board2D::Shape::SolidStyle,
                                             Board2D::Shape::RoundCap,
                                             Board2D::Shape::RoundJoin ))
                 << Point(1,1) << Point(1,2) << Point(1,3)
                 << Point(1,4) << Point(1,5) << Point(1,6)
                 << Point(1,7) << Point(2,1) << Point(3,1) << Point(4,1)
                 << Point(4,2) << Point(5,2) << Point(5,3)
                 << Point(5,4) << Point(5,5) << Point(5,6)
                 << Point(4,6) << Point(4,7) << Point(3,7)
                 << Point(2,7) << Point(9,1) << Point(9,2)
                 << Point(8,2) << Point(8,3)
```

Thanks for your attention !

Réveilles 91 J.-P. Réveilles,
Géométrie discrète, calcul en nombres entiers et algorithmique.
Thèse d'état, Université Louis Pasteur, Strasbourg, 1991.

[Feschet and Tougne 99] Feschet, Fabien and Tougne, Laure
Optimal time computation of the tangent of a discrete curve: Application to the curvature
International Conference on Discrete Geometry for Computer Imagery
Springer LNCS pp. 31–40 1999

[Debled06 *et al.*] Debled-Rennesson, I.; Feschet, F.; Rouyer-Degli
Optimal Blurred Segments Decomposition of Noisy Shapes in Linear Times
Comp. & Graphics **30** (2006) 30–36

[Kovalevsky 01] Kovalevsky, V.
Curvature in Digital 2D Images
International Journal of Pattern Recognition and Artificial Intelligence
2001, 15, 1183-1200

[Vialard 96] Vialard, Anne
Chemin Euclidiens: Un modèle de représentation des contours discrets,
Université de Bordeaux 1, 1996

Kerautret, B. and Lachaud, J.-O. (2012).
Meaningful Scales Detection along Digital Contours for Unsupervised Local Noise Estimation detection.
IEEE. Trans. on PAMI, in press (10.1109/TPAMI.2012.38).

Bertrand Kerautret, and Jacques-Olivier Lachaud,
Meaningful Scales Detection: an Unsupervised Noise Detection Algorithm for Digital Contours,
Image Processing On Line, 4 (2014), pp. 98–115.
http://dx.doi.org/10.5201/ipol.2014.75

- 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).
- [Lachaud *et al.* 05]] Lachaud, J.-O. and Vialard, A. and de Vieilleville, F
  Analysis and comparative evaluation of discrete tangent estimators
  Proc. Int. Conf DGCI'2005 Volume 3429 of LNCS (Springer), 140–251 2005.
- [Coeurjolly *et al.* 2001] Coeurjolly, D., Miguet, S., Tougne, L.
  Discrete curvature based on osculating circle estimation.
  In proc. Int. workshop Visual Form, Volume 2059 of LNCS (Springer), 303–312 2001.
- [Coeurjolly, Svensson 2003] Coeurjolly, David and Svensson, Stina
  Discrete curvature based on osculating circle estimation.
  Image Analysis, 247–254 2003.

[Coeurjolly *et al.* 2004] Coeurjolly, David and Gérard, Yan and Reveilles, J.-P. and Tougne, Laure
An elementary algorithm for digital arc segmentation
Discrete Applied Mathematics, 138(1): 31–50 2004.

[Kerautret and Lachaud 2009] Kerautret, B., Lachaud, J.O.
Curvature estimation along noisy digital contours by approximate global optimization.
Pattern Recognition 42(10), 2265 – 2278 (2009)

[Kerautret *et al.*2017] Kerautret, B., Ngo, P, Kenmochi, Y., Vacavant A.
Greyscale Image Vectorization from Geometric Digital Contour Representations
20th International Conference on Discrete Geometry for Computer Imagery Volume 10502 of LNCS (Springer), 319-331 2017.

Nguyen, T.P., Debled-Rennesson, I.:
Arc segmentation in linear time.
In: Computer Analysis of Images and Patterns - 14th International Conference, CAIP 2011, Seville, Spain, August 29-31, 2011, Proceedings, Part I. (2011) 84–92

Sivignon, I.
In: A Near-Linear Time Guaranteed Algorithm for Digital Curve Simplification under the Fréchet Distance. Springer (2011) 333–345

Najman, L., Couprie, M.:
Building the component tree in quasi-linear time.
Trans. Img. Proc. **15** (2006) 3531–3539

[Sloboda *et al.* 98] F. Sloboda, B. Zatko, J. Stoer
On approximation of planar one-dimensional continua
Proceedings of Advances in Digital and Computational Geometry, 113–160 2004.

[De Vieilleville *et al.* 05] de Vieilleville, F. and Lachaud, J.-O. and Feschet, F.
Maximal digital straight segments and convergence of discrete geometric estimators
In proc. of Scandinavian Conf SCIA, Volume 3540 of LNCS (Springer), 988–1003 2005.

[Kanungo 96] Kanungo, T
Document Degradation Models and a Methodology for Degradation Model Validation,
Phd Thesis, University of Washington, 1996

[Nguyen, Debled *et al.* 07] Nguyen, T., Debled-Rennesson, I.
Curvature estimation in noisy curves.
In proc. of Int Conf CAIP, Volume 4673 of LNCS (Springer), 474-481 2007.

[Kerautret *et al.* 08] Kerautret, B. and Lachaud, J.-O. and Naegel, B.
Curvature based corner detector for discrete, noisy and multi-scale contours
International Journal of Shape Modeling 14(2): 127–145, 2008

[Chang *et al.* 07] Chang, X., Gao, L., Li, Y.
Corner detection based on morphological disk element.
In: Proceedings of the 2007 American Control Conference, IEEE (2007) 1994-1999

[Feschet 2010] Feschet, F.
Multiscale analysis from 1d parametric geometric decomposition of shapes.
In: IEEE (ed.) Int. Conf. on Pattern Recognition. pp. 2102–2105 (2010)

[Sivignon 2011] I. Sivignon: A Near-Linear Time Guaranteed Algorithm for Digital Curve Simplification under the Fréchet Distance.
In: Proc of DGCI 2011. pp. 333–345 (2011)

[Kerautret *et al.* 11] Kerautret, B. and Lachaud, J. O. and Nguyen, T. P.
Circular arc reconstruction of digital contours with chosen Hausdorff error,
In proc. of DGCI, Volume 6607 of LNCS (Springer), 250-262 2011.

[Malgouyres *et al.*2008] Malgouyres, R., Brunet, F., Fourey, S.: Binomial convolutions and derivatives estimation from noisy discretizations. In: Proc. DGCI. pp. 370–379 (2008)

[NguyenDebled10] T. P. Nguyen et I. Debled-Rennesson
Arc Segmentation in Linear Time,
In proc. of Int Conf CAIP, Volume 6854 of LNCS (Springer), 84-92 2007.

[T.P. Nguyen a2010] Nguyen, T.P.
Etude des courbes discrètes: applications en analyse d'images.
Ph.D. thesis, Nancy University - LORIA (2010), (in french)

[Roussillon & Lachaud 11] Tristan Roussillon, Jacques-Olivier Lachaud
Accurate Curvature Estimation along Digital Contours with Maximal Digital Circular Arcs.
IWCIA 2011: 43-55