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

Bertrand Kerautret[1]
(work in collaboration with Isabelle Debled-Rennesson[1], Jacques-Olivier Lachaud[2] and the DGtal Dev Team)

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

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

Overview of the presentation - Part I -

1. I. Short Overview of Digital Geometry Domain
   - 1.1 Origins and motivations
   - 1.2 Outline of main base definitions
   - 1.3 Main Actual Research Areas
2. 2. Geometric Estimator on Digital Contours
   - 2.1 Main primitives used to analyse digital contours
   - 2.2 Estimating curvature on (noisy) digital contours
   - 2.3 Application to contour representation
3. Presentation of the DGtal Library
   - 3.1 Short presentation of the library
   - 3.2 Extracting level sets contours with DGtal
   - 3.3 Example of geometric estimator

https://kerautret.github.io/ACCV2016DGPTutorial/

Overview of the presentation - Part I -

https://kerautret.github.io/ACCV2016DGPTutorial/

1. Short Overview of Digital Geometry Domain

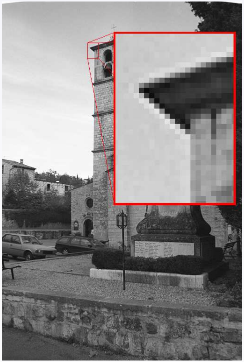
Research in Digital Geometry

- Digital objects composed of Pixels / Voxels.

1. Short Overview of Digital Geometry Domain

Research in Digital Geometry

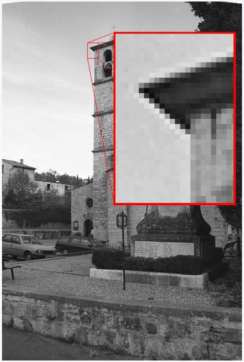
- Digital objects composed of Pixels / Voxels.

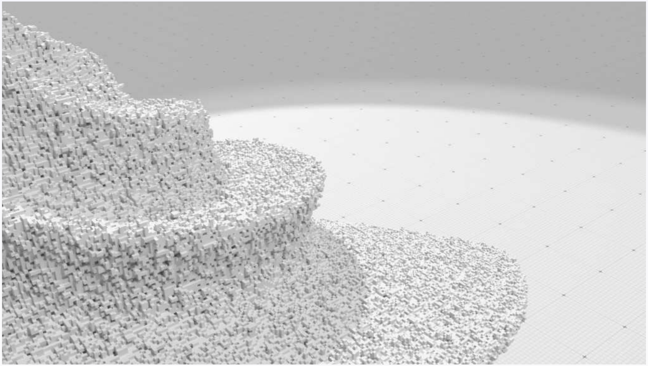


1. Short Overview of Digital Geometry Domain

Research in Digital Geometry

- Digital objects composed of Pixels / Voxels.

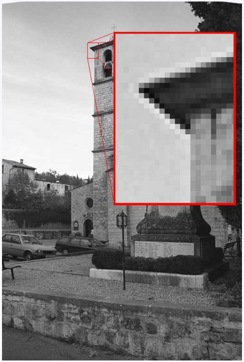
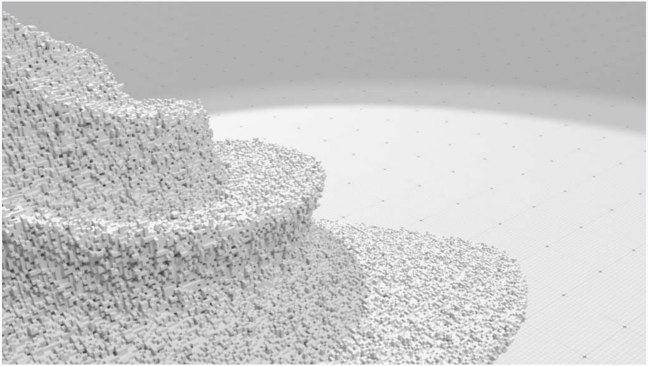




1. Short Overview of Digital Geometry Domain

Research in Digital Geometry

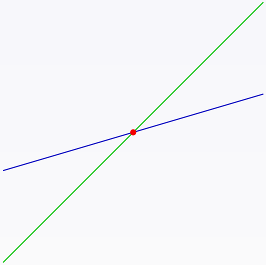
- Digital objects composed of Pixels / Voxels.
  $\Rightarrow$ Digital camera, Scanners, Medical Magnetic Resonance Imaging.

1. Short Overview of Digital Geometry Domain

Research in Digital Geometry

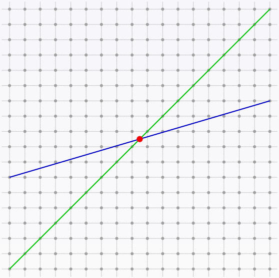
- Digital objects composed of Pixels / Voxels.
  $\Rightarrow$ Digital camera, Scanners, Medical Magnetic Resonance Imaging.
- Euclidean Geometry theorem are not valid in the Digital Space.



1. Short Overview of Digital Geometry Domain

Research in Digital Geometry

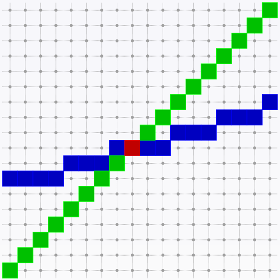
- Digital objects composed of Pixels / Voxels.
  $\Rightarrow$ Digital camera, Scanners, Medical Magnetic Resonance Imaging.
- Euclidean Geometry theorem are not valid in the Digital Space.



1. Short Overview of Digital Geometry Domain

Research in Digital Geometry

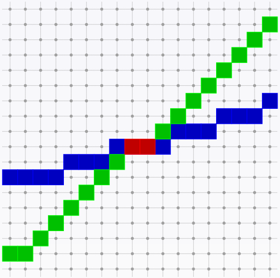
- Digital objects composed of Pixels / Voxels.
  ⇒ Digital camera, Scanners, Medical Magnetic Resonance Imaging.
- Euclidean Geometry theorem are not valid in the Digital Space.



1. Short Overview of Digital Geometry Domain

Research in Digital Geometry

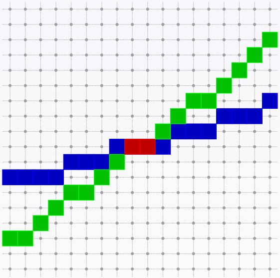
- Digital objects composed of Pixels / Voxels.
  $\Rightarrow$ Digital camera, Scanners, Medical Magnetic Resonance Imaging.
- Euclidean Geometry theorem are not valid in the Digital Space.



1. Short Overview of Digital Geometry Domain

Research in Digital Geometry

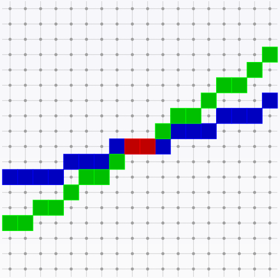
- Digital objects composed of Pixels / Voxels.
  $\Rightarrow$ Digital camera, Scanners, Medical Magnetic Resonance Imaging.
- Euclidean Geometry theorem are not valid in the Digital Space.



1. Short Overview of Digital Geometry Domain

Research in Digital Geometry

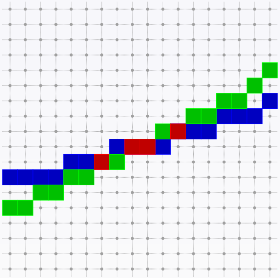
- Digital objects composed of Pixels / Voxels.
  ⇒ Digital camera, Scanners, Medical Magnetic Resonance Imaging.
- Euclidean Geometry theorem are not valid in the Digital Space.



1. Short Overview of Digital Geometry Domain

Research in Digital Geometry

- Digital objects composed of Pixels / Voxels.
  ⇒ Digital camera, Scanners, Medical Magnetic Resonance Imaging.
- Euclidean Geometry theorem are not valid in the Digital Space.



1.1 Origins and motivations

History

- 70's : A. Rosenfeld, G. Herman, E. Khalimsky
- Objective: $\Rightarrow$ Define a theoretical framework to transpose in $\mathbb{Z}^n$ the basic notion of Euclidean geometry.

1.1 Origins and motivations

History

- 70's : A. Rosenfeld, G. Herman, E. Khalimsky
- Objective: $\Rightarrow$ Define a theoretical framework to transpose in $\mathbb{Z}^n$ the basic notion of Euclidean geometry.

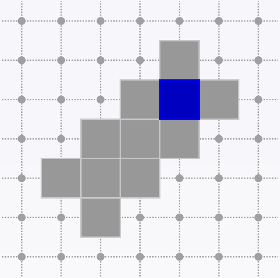
Digital Geometry

- Grid (representation of data).
- Topology.
- Basic Objects (points, straight lines, planes, ...).
- Adapted algorithmic process.

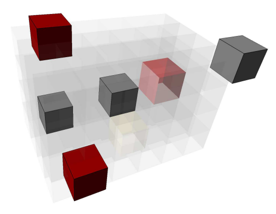
1.2 Outline of main base definitions

Regular Grid

2D discrete space



3D discrete space

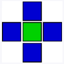


1.2 Outline of main base definitions

Regular Grid

Connexity

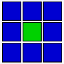
2d



4-connectivity

A and B of $\mathbb{Z}^2$ are 4-neighbour (or 4-adjacent) if :

$$|x_A - x_B| + |y_A - y_B| = 1$$



8-connectivity

A and B of $\mathbb{Z}^2$ are 8-neighbour (or 8-adjacent) if :

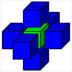
$$max(|x_A - x_B|, |y_A - y_B|) = 1$$

1.2 Outline of main base definitions

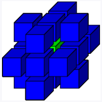
Regular Grid

Connexity

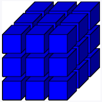
2d , 3d, nd



6-connectivity



18-connectivity



26-connectivity

α-connectivity or α-adjacency

1.2 Outline of main base definitions

Regular Grid

Connexity

2d , 3d, nd

Curves

$\mathcal{E} = \{p_i\}_{i=0..n}$: sequence of discrete points and a relation of *k*-adjacency

1.2 Outline of main base definitions

Regular Grid

Connexity

2d , 3d, nd

Curves

$\mathcal{E} = \{p_i\}_{i=0..n}$: sequence of discrete points and a relation of *k*-adjacency

- k-Arc: $\forall$ p_i of $\mathcal{E}$, p_i has exactly two k-neighbour points in $\mathcal{E}$, excepted p_0 and p_n called extremities of the arc.

1.2 Outline of main base definitions

Regular Grid

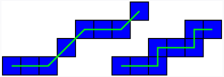
Connexity

2d , 3d, nd

Curves

$\mathcal{E} = \{p_i\}_{i=0..n}$: sequence of discrete points and a relation of *k*-adjacency

- k-Arc: $\forall$ p_i of $\mathcal{E}$, p_i has exactly two *k*-neighbour points in $\mathcal{E}$, excepted p_0 and p_n called extremities of the arc.



8-Arc and 4-Arc

1.2 Outline of main base definitions

Regular Grid

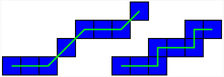
Connexity

2d , 3d, nd

Curves

$\mathcal{E} = \{p_i\}_{i=0..n}$: sequence of discrete points and a relation of k-adjacency

- k-Arc: $\forall$ p_i of $\mathcal{E}$, p_i has exactly two k-neighbour points in $\mathcal{E}$, excepted p_0 and p_n called extremities of the arc.
- k-Curve: $\mathcal{E}$ is a k-Arc and $p_0 = p_n$.



8-Arc and 4-Arc

1.2 Outline of main base definitions

Regular Grid

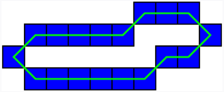
Connexity

2d , 3d, nd

Curves

$\mathcal{E} = \{p_i\}_{i=0..n}$: sequence of discrete points and a relation of k-adjacency

- k-Arc: $\forall$ p_i of $\mathcal{E}$, p_i has exactly two k-neighbour points in $\mathcal{E}$, excepted p_0 and p_n called extremities of the arc.
- k-Curve: $\mathcal{E}$ is a k-Arc and $p_0 = p_n$.

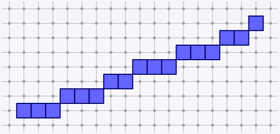
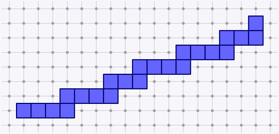
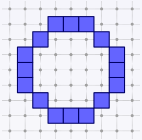


8-Curve

Digital Primitives

2D Main Primitives

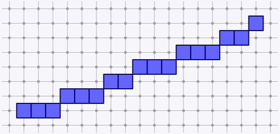
⇒ Naive, Standard digital lines, circle.

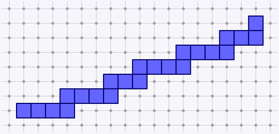




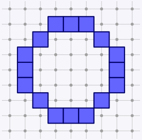
Digital Primitives

2D Main Primitives

⇒ Naive, Standard digital lines, circle.

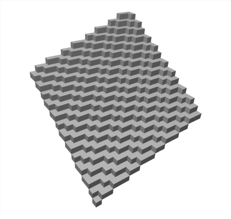


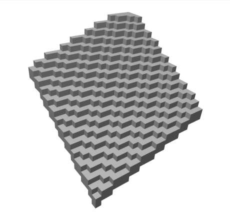


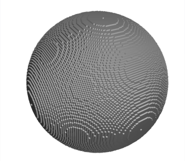


3D Main Primitives

⇒ Naive, Standard digital planes, sphere.







1.3 Main Actual Research Areas

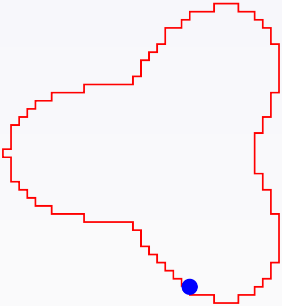
Geometric Estimators

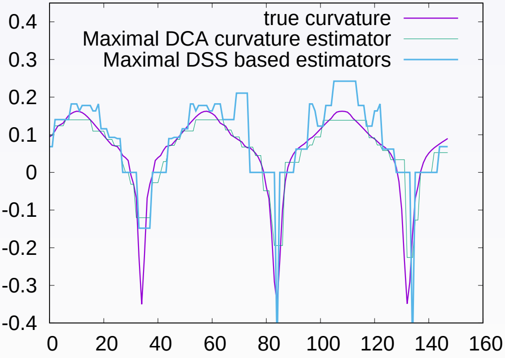
- 2D/3D estimators: lengths, normals, curvature.

1.3 Main Actual Research Areas

Geometric Estimators

- 2D/3D estimators: lengths, normals, curvature.





1.3 Main Actual Research Areas

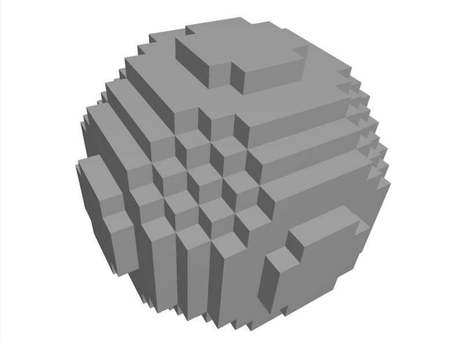
Geometric Estimators

- 2D/3D estimators: lengths, normals, curvature.

1.3 Main Actual Research Areas

Geometric Estimators

- 2D/3D estimators: lengths, normals, curvature.
- Convergence of estimators: finer and finer grids.

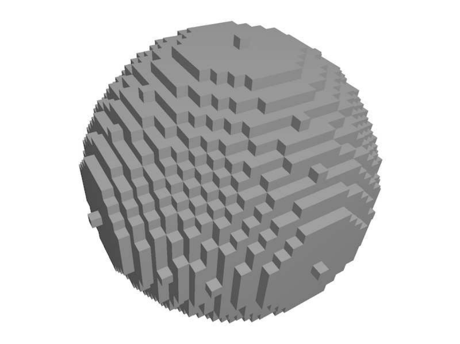


$gs = 2$

1.3 Main Actual Research Areas

Geometric Estimators

- 2D/3D estimators: lengths, normals, curvature.
- Convergence of estimators: finer and finer grids.

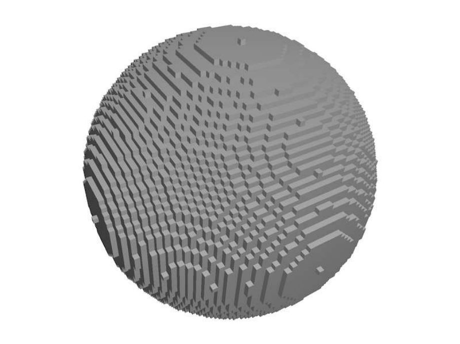


$gs = 1$

1.3 Main Actual Research Areas

Geometric Estimators

- 2D/3D estimators: lengths, normals, curvature.
- Convergence of estimators: finer and finer grids.

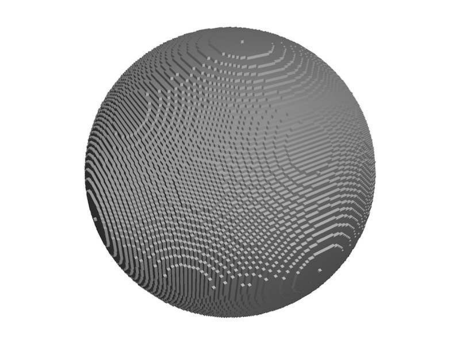


$gs = 0.5$

1.3 Main Actual Research Areas

Geometric Estimators

- 2D/3D estimators: lengths, normals, curvature.
- Convergence of estimators: finer and finer grids.



$gs = 0.25$

1.3 Main Actual Research Areas

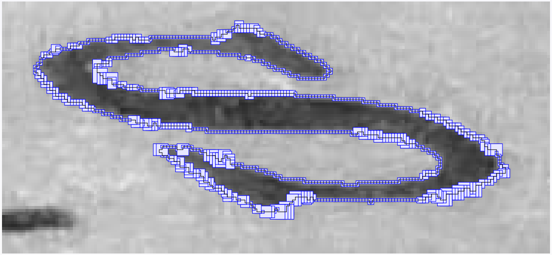
Geometric Estimators

- 2D/3D estimators: lengths, normals, curvature.
- Convergence of estimators: finer and finer grids.
- Estimator resistant to noise.

1.3 Main Actual Research Areas

Geometric Estimators

- 2D/3D estimators: lengths, normals, curvature.
- Convergence of estimators: finer and finer grids.
- Estimator resistant to noise.
- Scale estimation.



1.3 Main Actual Research Areas

Geometric Estimators

- 2D/3D estimators: lengths, normals, curvature.
- Convergence of estimators: finer and finer grids.
- Estimator resistant to noise.
- Scale estimation.

Other Reseacher Axis

- Shape representation.
- Digital and combinatorial tools for image segmentation and analysis.
- Geometric transforms (discrete objects, discrete model properties, digitization schemes, metrics...)
- Tomography, segmentation.

Overview of the presentation - Part I -

https://kerautret.github.io/ACCV2016DGPTutorial/

2. Geometric Estimator on Digital Contours

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. Geometric Estimator on Digital Contours

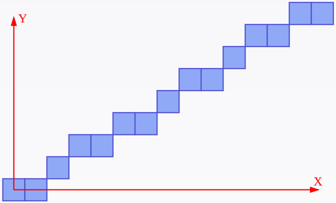
Arithmetic definition of digital line [Réveilles 91]

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).



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

2. Geometric Estimator on Digital Contours

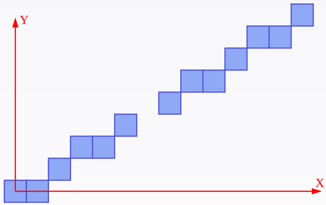
Arithmetic definition of digital line [Réveilles 91]

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.



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

2. Geometric Estimator on Digital Contours

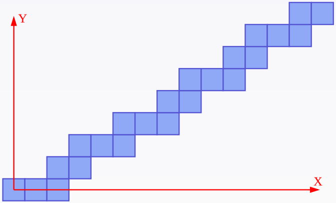
Arithmetic definition of digital line [Réveilles 91]

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. Geometric Estimator on Digital Contours

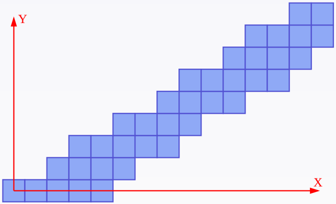
Arithmetic definition of digital line [Réveilles 91]

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.

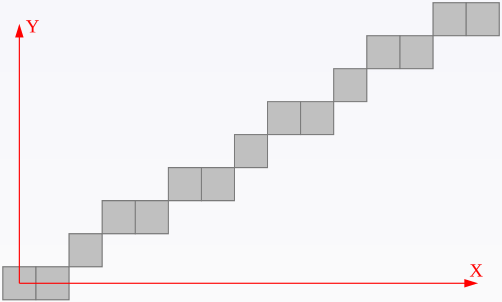


$\mathcal{D}(5, 8, -1, 8)$

2.1 Main primitives used to analyse digital contours

Recognition Problem

- Based on the remainder and periodicity detection.



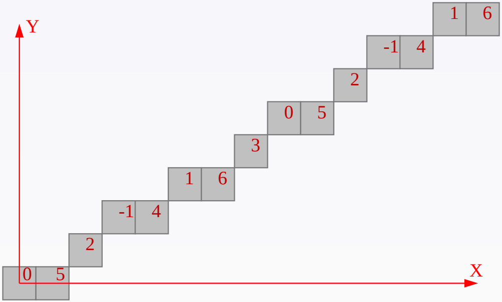
$\mathcal{D}(5, 8, -1, 13)$

2.1 Main primitives used to analyse digital contours

Recognition Problem

- Based on the remainder and periodicity detection.
- Remainder of a point M is defined as a function of $\mathcal{D}(a, b, \mu, \omega)$:

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



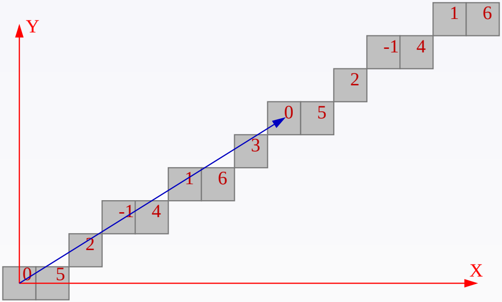
$\mathcal{D}(5, 8, -1, 13)$

2.1 Main primitives used to analyse digital contours

Recognition Problem

- Based on the remainder and periodicity detection.
- Remainder of a point M is defined as a function of $\mathcal{D}(a, b, \mu, \omega)$:

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



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

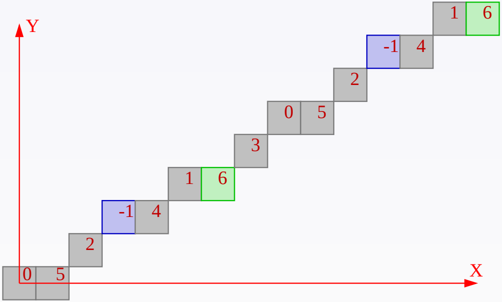
2.1 Main primitives used to analyse digital contours

Recognition Problem

- 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.



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

Linear and incremental recognition algorithm

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

- Compute remainder of new point *M*.

Linear and incremental recognition algorithm

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

- Compute remainder of new point M.
- From $r(M)$ update characteristics.

Growth of a recognized segment of a naïve line

Let be $\mathcal{S} = M_0M_1$ a recognized segment of $\mathcal{D}(a, b, \mu, max(|a|, |b|))$, M an added point to $\mathcal{S}$, $r_{\mathcal{D}}(M) = ax_M - by_M$:

Linear and incremental recognition algorithm

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

- Compute remainder of new point M.
- From $r(M)$ update characteristics.

Growth of a recognized segment of a naïve line

Let be $\mathcal{S} = M_0M_1$ a recognized segment of $\mathcal{D}(a, b, \mu, max(|a|, |b|))$, M an added point to $\mathcal{S}$, $r_{\mathcal{D}}(M) = ax_M - by_M$:

(*i*) $\mu \leq r_{\mathcal{D}}(M) < \mu + max(|a|, |b|)$: $M \in \mathcal{D}$,
$\mathcal{S} \cup \{M\}$ is a segment of $\mathcal{D}$,

Linear and incremental recognition algorithm

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

- Compute remainder of new point M.
- From $r(M)$ update characteristics.

Growth of a recognized segment of a naïve line

Let be $\mathcal{S} = M_0M_1$ a recognized segment of $\mathcal{D}(a, b, \mu, max(|a|, |b|))$, M an added point to $\mathcal{S}$, $r_{\mathcal{D}}(M) = ax_M - by_M$:

(*i*) $\mu \leq r_{\mathcal{D}}(M) < \mu + max(|a|, |b|)$: $M \in \mathcal{D}$,
$\mathcal{S} \cup \{M\}$ is a segment of $\mathcal{D}$,

(*iv*) $r_{\mathcal{D}}(M) < \mu - 1$ or $r_{\mathcal{D}}(M) > \mu + max(|a|, |b|)$: M is strongly exterior to $\mathcal{D}$,
$\mathcal{S} \cup \{M\}$ is not a segment of a naïve line.

Linear and incremental recognition algorithm

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

- Compute remainder of new point M.
- From $r(M)$ update characteristics.

Growth of a recognized segment of a naïve line

Let be $\mathcal{S} = M_0M_1$ a recognized segment of $\mathcal{D}(a, b, \mu, max(|a|, |b|))$, M an added point to $\mathcal{S}$, $r_{\mathcal{D}}(M) = ax_M - by_M$:

(i) $\mu \leq r_{\mathcal{D}}(M) < \mu + max(|a|, |b|)$: $M \in \mathcal{D}$,
$\mathcal{S} \cup \{M\}$ is a segment of $\mathcal{D}$,

(ii) $r_{\mathcal{D}}(M) = \mu + max(|a|, |b|)$: M is weakly exterior to $\mathcal{D}$,
$\mathcal{S} \cup \{M\}$ is a recognized segment of the naïve line whose slope is given by the vector $L_F M$,

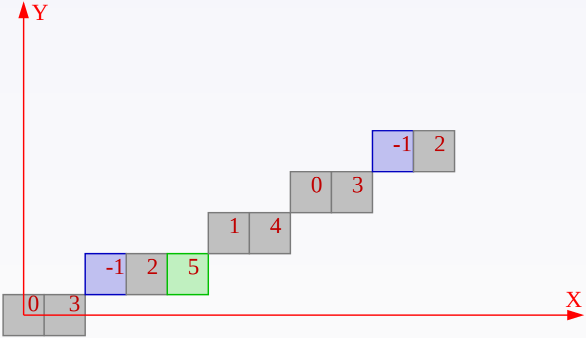
(iii) $r_{\mathcal{D}}(M) = \mu - 1$: M is weakly exterior to $\mathcal{D}$,
$\mathcal{S} \cup \{M\}$ is a recognized segment of the naïve line whose slope is given by the vector $U_F M$,

(iv) $r_{\mathcal{D}}(M) < \mu - 1$ or $r_{\mathcal{D}}(M) > \mu + max(|a|, |b|)$: M is strongly exterior to $\mathcal{D}$,
$\mathcal{S} \cup \{M\}$ is not a segment of a naïve line.

Linear and incremental recognition algorithm

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

- Compute remainder of new point M.
- From $r(M)$ update characteristics.

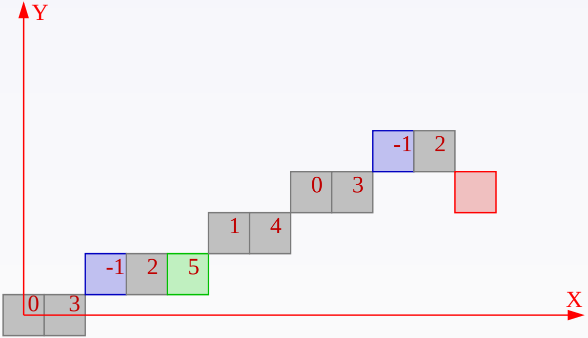


Recognized segment $\mathcal{S}$ of $\mathcal{D}_0(3, 7, -1, 7)$

Linear and incremental recognition algorithm

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

- Compute remainder of new point M.
- From $r(M)$ update characteristics.

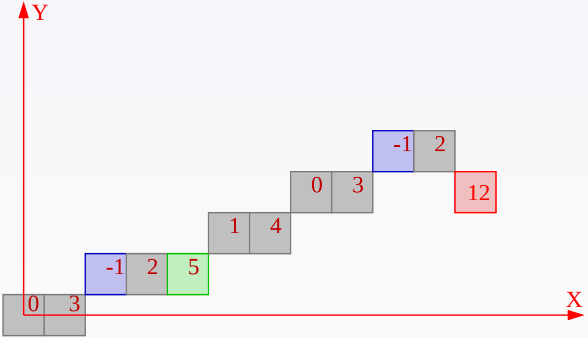


Recognized segment $\mathcal{S}$ of $\mathcal{D}_0(3, 7, -1, 7)$

Linear and incremental recognition algorithm

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

- Compute remainder of new point M.
- From $r(M)$ update characteristics.



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

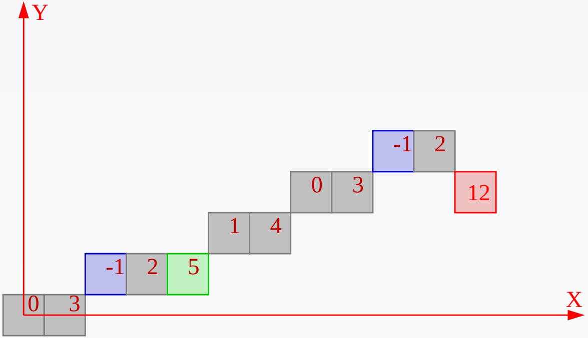
Linear and incremental recognition algorithm

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

- Compute remainder of new point M.
- From $r(M)$ update characteristics.

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 $\mathcal{D}_0(3, 7, -1, 7)$

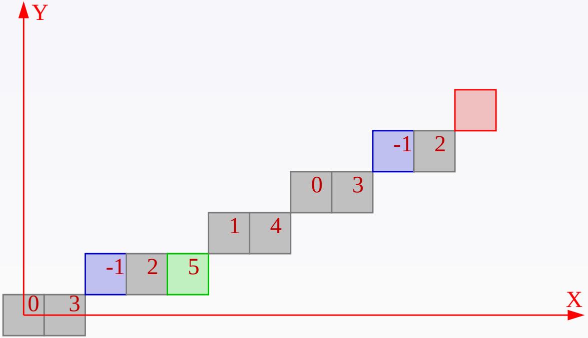
Linear and incremental recognition algorithm

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

- Compute remainder of new point M.
- From $r(M)$ update characteristics.

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 $\mathcal{D}_0(3, 7, -1, 7)$

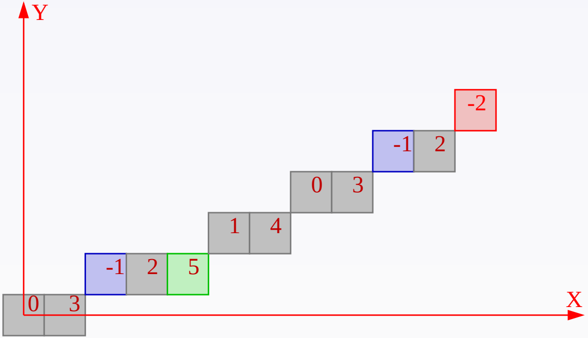
Linear and incremental recognition algorithm

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

- Compute remainder of new point M.
- From $r(M)$ update characteristics.

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 $\mathcal{D}_0(3, 7, -1, 7)$

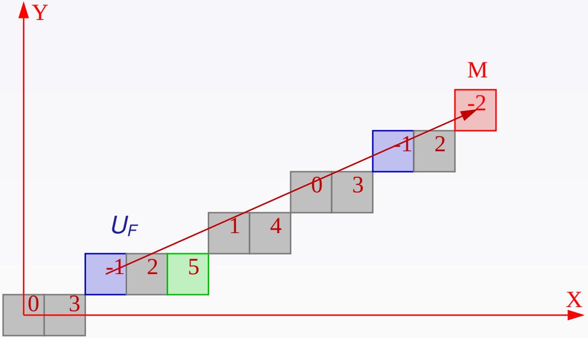
Linear and incremental recognition algorithm

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

- Compute remainder of new point M.
- From $r(M)$ update characteristics.

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 $\mathcal{D}_0(3, 7, -1, 7)$

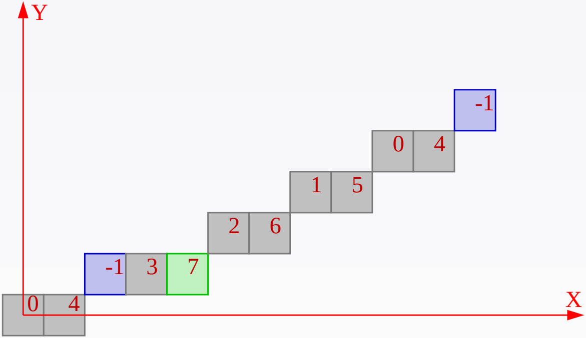
Linear and incremental recognition algorithm

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 $\mathcal{S}$:

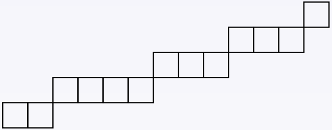
(*iii*) $r_{\mathcal{D}}(M) = \mu - 1$: M weakly exterior to $\mathcal{D}$, M added to $\mathcal{S}$ and the slope is updated by the vector $U_F M$



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

A recognition example

Input 4-arc to be recognized as straight segment

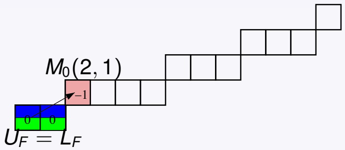


Initialisation : $a = 0$, $b = 1$, $\mu = 0$, $D_0(0, 1, 0, 1)$

$$0 \leq -y < 1$$

A recognition example

Input 4-arc to be recognized as straight segment



$$a_0 = 0,\ b_0 = 1,\ \mu_0 = 0,\ D_0(0,1,0,1)$$

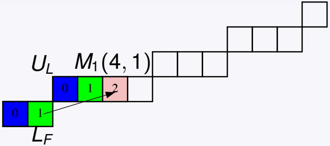
$$0 \leq -y < 1$$

$$r_0(M_0) = -1 \Rightarrow a_1 = 1,\ b_1 = 2,\ \mu_1 = 0,\ D_1(1,2,0,2)$$

$$0 \leq x - 2y < 2$$

A recognition example

Input 4-arc to be recognized as straight segment



$$a_1 = 1,\ b_1 = 2,\ \mu_1 = 0,\ D_1(1, 2, 0, 2)$$

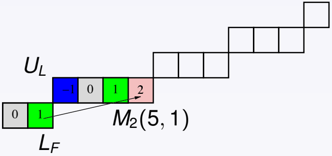
$$0 \leq x - 2y < 2$$

$$r_1(M_1) = 2 \Rightarrow a_2 = 1,\ b_2 = 3,\ \mu_2 = -1,\ D_2(1, 3, -1, 3)$$

$$-1 \leq x - 3y < 2$$

A recognition example

Input 4-arc to be recognized as straight segment



$$a_2 = 1,\ b_2 = 3,\ \mu_2 = -1,\ D_2(1,3,-1,3)$$

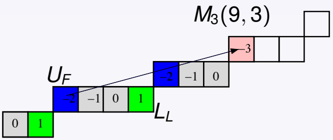
$$-1 \leq x - 3y < 2$$

$$r_2(M_2) = 2 \Rightarrow a_3 = 1,\ b_3 = 4,\ \mu_3 = -2,\ D_3(1,4,-2,4)$$

$$-2 \leq x - 4y < 2$$

A recognition example

Input 4-arc to be recognized as straight segment



$$a_3 = 1,\ b_3 = 4,\ \mu_3 = -2,\ D_3(1,4,-2,4)$$

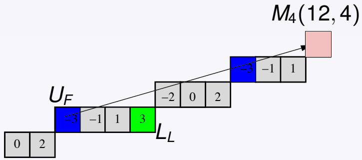
$$-2 \leq x - 4y < 2$$

$$r_3(M_3) = -3 \Rightarrow a_4 = 2,\ b_4 = 7,\ \mu_4 = -3,\ D_4(2,7,-3,7)$$

$$-3 \leq 2x - 7y < 4$$

A recognition example

Input 4-arc to be recognized as straight segment



$$a_4 = 2,\ b_4 = 7,\ \mu_4 = -3,\ D_4(2, 7, -3, 7)$$

$$-3 \leq 2x - 7y < 4$$

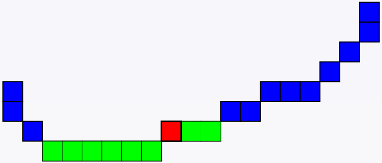
$$r_4(M_4) = -4 \Rightarrow a_5 = 3,\ b_5 = 10,\ \mu_5 = -4,\ D_5(3, 10, -4, 10)$$

$$-4 \leq 3x - 10y < 6$$

Segmentation of digital contour

Primitive of Maximal Digital Straight Segment (MDSS)

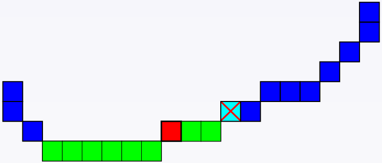
Let $\mathcal{C}$ be a digital curve, a segment of a naïve digital line is said maximal if it cannot be extended at the right and left hand sides on $\mathcal{C}$.



Segmentation of digital contour

Primitive of Maximal Digital Straight Segment (MDSS)

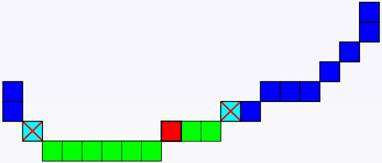
Let $\mathcal{C}$ be a digital curve, a segment of a naïve digital line is said maximal if it cannot be extended at the right and left hand sides on $\mathcal{C}$.



Segmentation of digital contour

Primitive of Maximal Digital Straight Segment (MDSS)

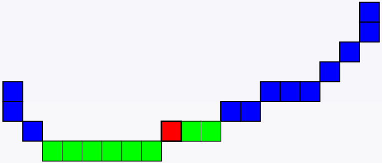
Let $\mathcal{C}$ be a digital curve, a segment of a naïve digital line is said maximal if it cannot be extended at the right and left hand sides on $\mathcal{C}$.



Segmentation of digital contour

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 $\mathcal{C}$.



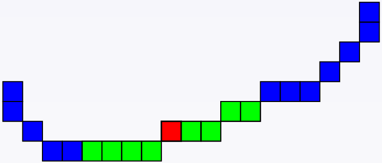
Sequence computation of maximal segments

Calculable en temps linéaire [Feschet and Tougne 99].

Segmentation of digital contour

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 $\mathcal{C}$.



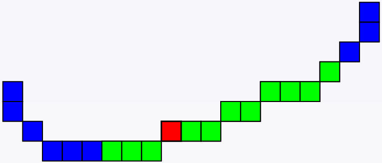
Sequence computation of maximal segments

Calculable en temps linéaire [Feschet and Tougne 99].

Segmentation of digital contour

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

Calculable en temps linéaire [Feschet and Tougne 99].

Segmentation of digital contour

Advantage and limits of the MDSS

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

Segmentation of digital contour

Advantage and limits of the MDSS

- + Gives a convergent primitive 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.

Primitive of Blurred Segments

Primitive of Blurred Segment (or Alpha Thick Segments)

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

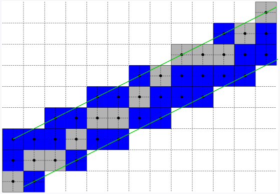
Primitive of Blurred Segments

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

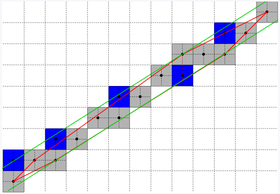
Primitive of Blurred Segments

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

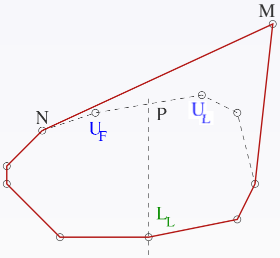
Primitive of Blurred Segments

Primitive of Blurred Segment (or Alpha Thick Segments)

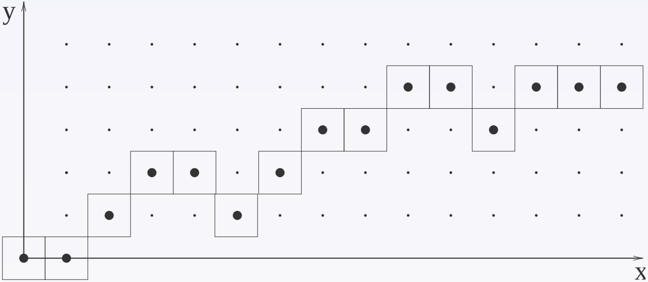
- Primitive able to handle noise.
- Can process disconnected set of points.
- No necessary integer coordinates.

Overview

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

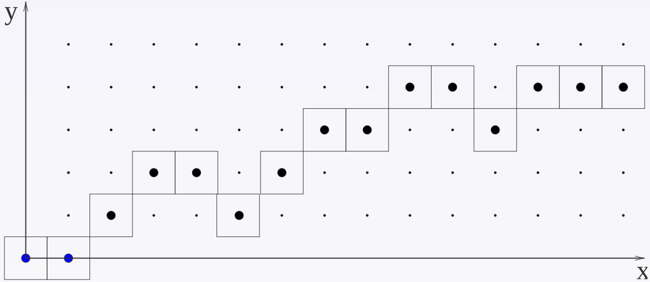


Blurred segment recognition



Sequence of pixels to recognize, $\nu = 2$

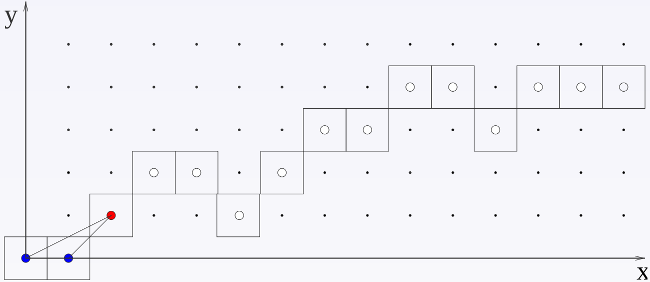
Blurred segment recognition



Sequence of pixels to recognize, $\nu = 2$

$\mathcal{D}_0(0, 1, 0, 1) : 0 \leq -y < 1$

Blurred segment recognition

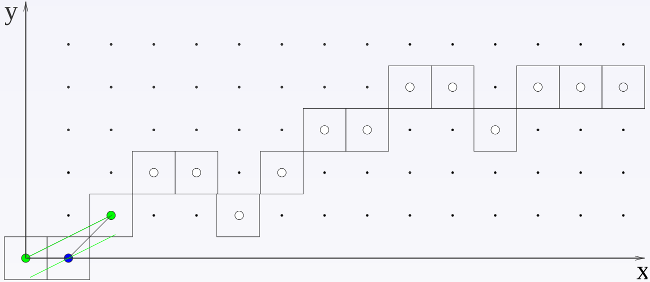


Sequence of pixels to recognize, $\nu = 2$

$\mathcal{D}_0(0, 1, 0, 1) : 0 \leq -y < 1$

Adding of the point M_3, $r_{\mathcal{D}_0}(M_3) = -1$

Blurred segment recognition



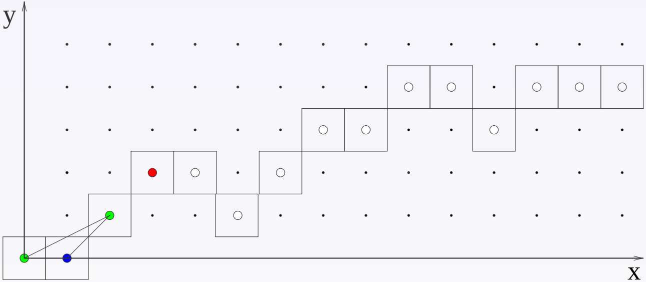
Sequence of pixels to recognize, $\nu = 2$

$\mathcal{D}_0(0,1,0,1) : 0 \leq -y < 1$

Adding of the point M_3, $r_{\mathcal{D}_0}(M_3) = -1$

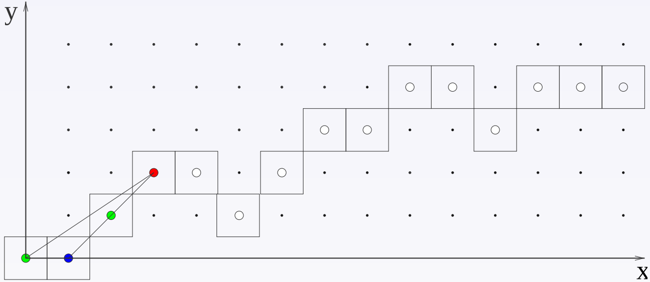
$\mathcal{D}_1(1,2,0,2) : 0 \leq x - 2y < 2$, $d_v = 0.5$

Blurred segment recognition



$$\mathcal{D}_1(1, 2, 0, 2) : 0 \leq x - 2y < 2$$

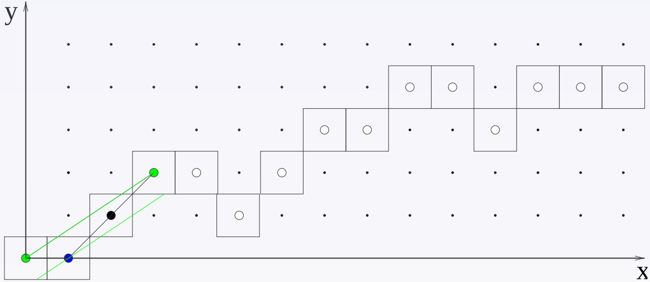
Blurred segment recognition



$$\mathcal{D}_1(1, 2, 0, 2) : 0 \leq x - 2y < 2$$

Adding of the point M_4, $r_{\mathcal{D}_1}(M_4) = -1$

Blurred segment recognition

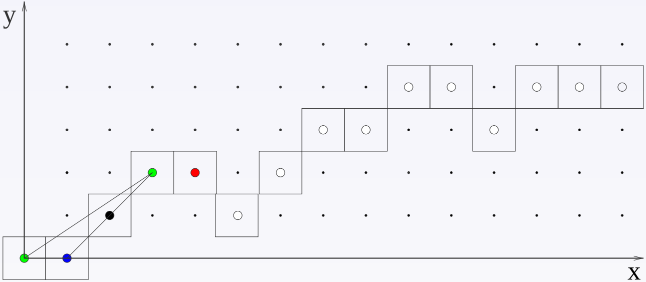


$\mathcal{D}_1(1, 2, 0, 2) : 0 \leq x - 2y < 2$

Adding of the point M_4, $r_{\mathcal{D}_1}(M_4) = -1$

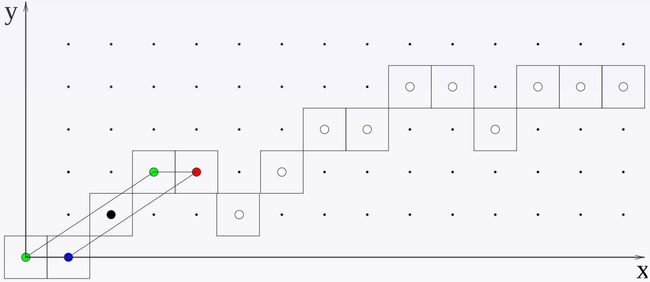
$\mathcal{D}_2(2, 3, 0, 3) : 0 \leq 2x - 3y < 3$, $d_v \simeq 0.66$

Blurred segment recognition



$$\mathcal{D}_2(2,3,0,3) : 0 \leq 2x - 3y < 3$$

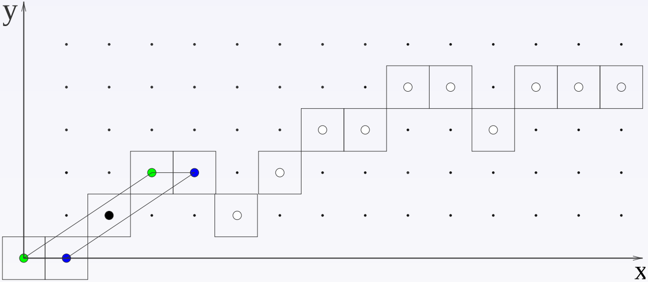
Blurred segment recognition



$\mathcal{D}_2(2, 3, 0, 3) : 0 \leq 2x - 3y < 3$

Adding of the point M_5, $r_{\mathcal{D}_2}(M_5) = 2$

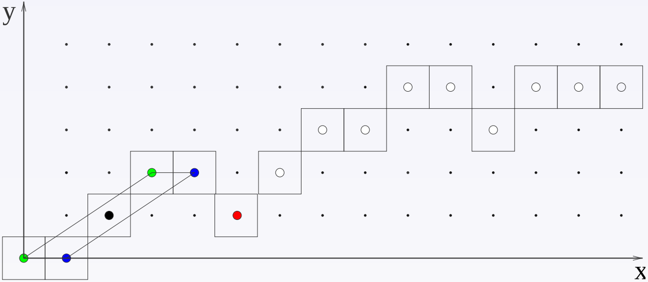
Blurred segment recognition



$\mathcal{D}_2(2, 3, 0, 3) : 0 \leq 2x - 3y < 3$

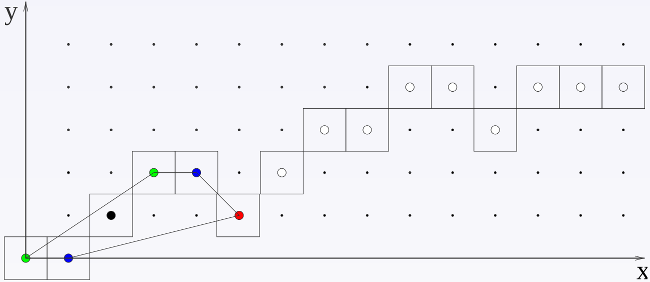
Adding of the point M_5, $r_{\mathcal{D}_2}(M_5) = 2$

Blurred segment recognition



$$\mathcal{D}_2(2,3,0,3) : 0 \leq 2x - 3y < 3$$

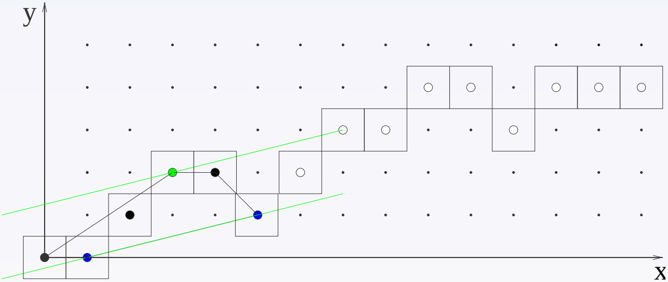
Blurred segment recognition



$\mathcal{D}_2(2, 3, 0, 3) : 0 \leq 2x - 3y < 3$

Adding of the point M_6, $r_{\mathcal{D}_2}(M_6) = 7$

Blurred segment recognition

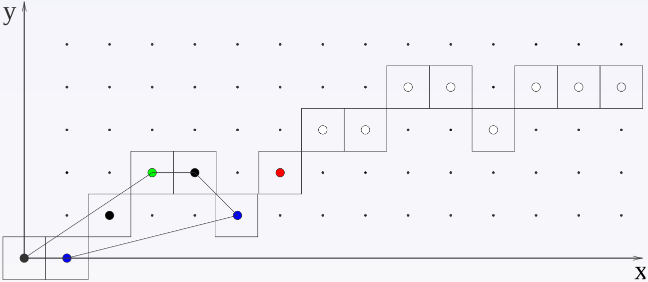


$\mathcal{D}_2(2, 3, 0, 3) : 0 \leq 2x - 3y < 3$

Adding of the point M_6, $r_{\mathcal{D}_2}(M_6) = 7$

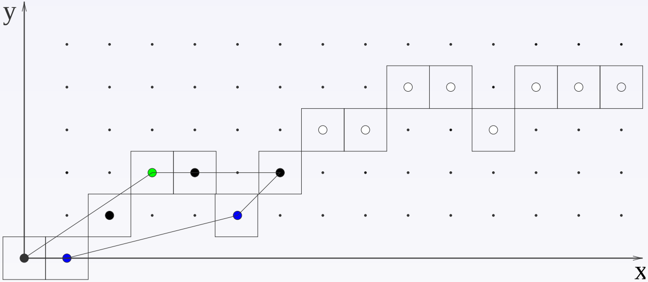
$\mathcal{D}_3(1, 4, -5, 7) : -5 \leq x - 4y < 2$, $d_v = 1.5$

Blurred segment recognition



$$\mathcal{D}_3(1, 4, -5, 7) : -5 \leq x - 4y < 2$$

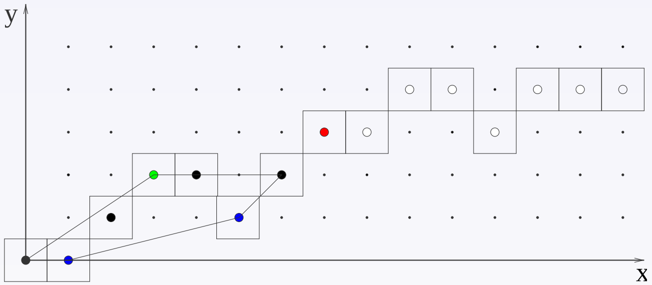
Blurred segment recognition



$\mathcal{D}_3(1, 4, -5, 7) : -5 \leq x - 4y < 2$

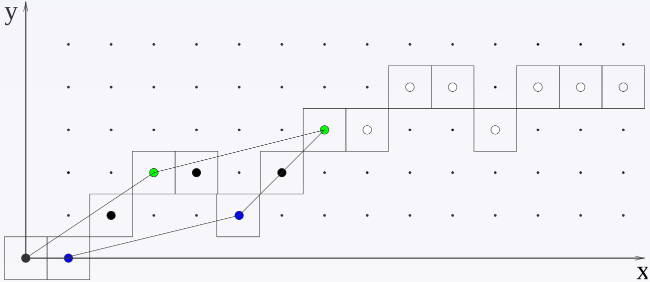
Adding of the point M_7, $r_{\mathcal{D}_3}(M_7) = 2$

Blurred segment recognition



$$\mathcal{D}_3(1, 4, -5, 7) : -5 \leq x - 4y < 2$$

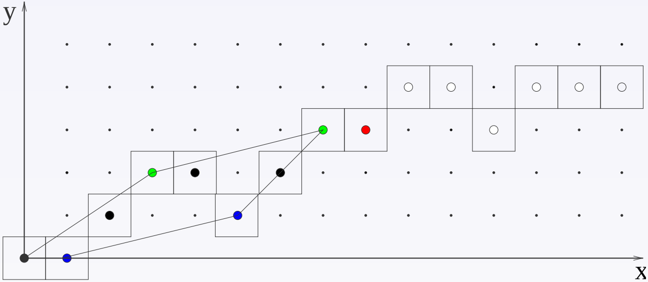
Blurred segment recognition



$$\mathcal{D}_3(1, 4, -5, 7) : -5 \leq x - 4y < 2$$

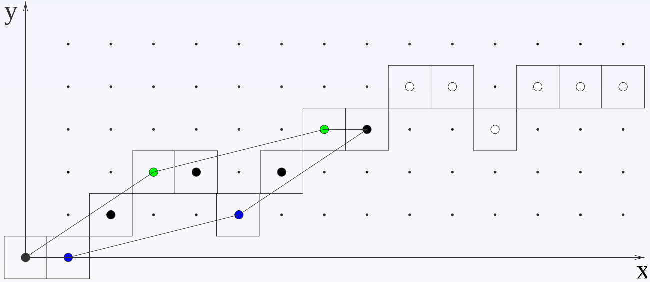
Adding of the point M_8, $r_{\mathcal{D}_3}(M_8) = -5$

Blurred segment recognition



$\mathcal{D}_3(1, 4, -5, 7) : -5 \leq x - 4y < 2$

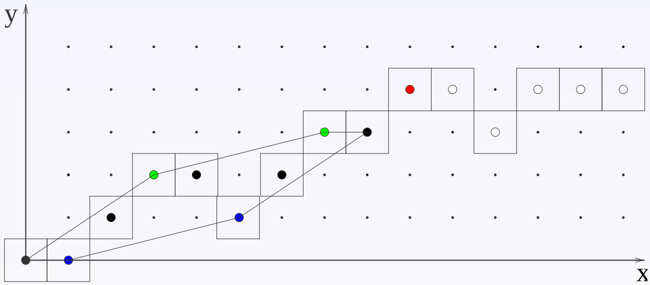
Blurred segment recognition



$\mathcal{D}_3(1, 4, -5, 7) : -5 \leq x - 4y < 2$

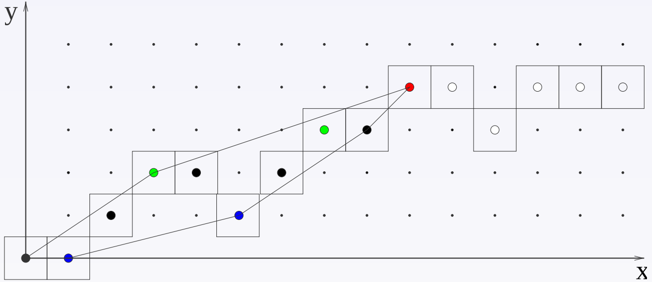
Adding of the point M_9, $r_{\mathcal{D}_3}(M_9) = -4$

Blurred segment recognition



$$\mathcal{D}_3(1, 4, -5, 7) : -5 \leq x - 4y < 2$$

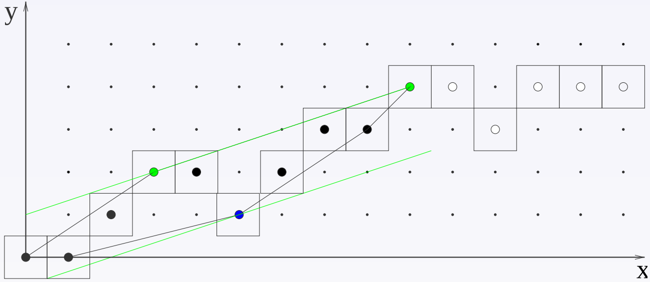
Blurred segment recognition



$\mathcal{D}_3(1, 4, -5, 7) : -5 \leq x - 4y < 2$

Adding of the point M_{10}, $r_{\mathcal{D}_3}(M_{10}) = -7$

Blurred segment recognition

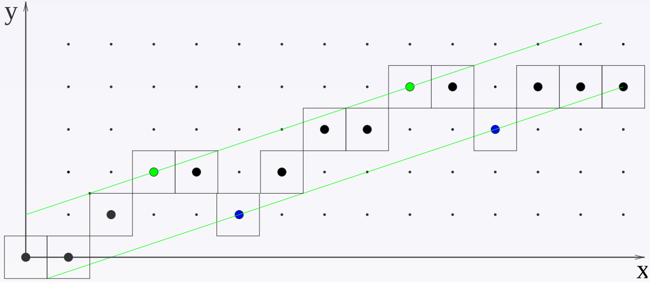


$\mathcal{D}_3(1, 4, -5, 7) : -5 \leq x - 4y < 2$

Adding of the point M_{10}, $r_{\mathcal{D}_3}(M_{10}) = -7$

$\mathcal{D}_4(1, 3, -3, 6) : -3 \leq x - 3y < 3$, $d_v \simeq 1.66$

Blurred segment recognition



Blurred segment of width 2 with $\mathcal{D}_4(1, 3, -3, 6)$ optimal bounding line

Base for geometric estimators

- Two main primitives which is a key point to estimate geometric features.
- Thanks to the blurred segment to handle noise.
- Example of such a method in the next section.

2.2 Estimating curvature on (noisy) digital contours

Context:

- Common with Jaques-Olivier Lachaud (LAMA, Chambéry).
- ANR Project GeoDIB (2006-2011): Digital Geometry on Noisy Objects.
- Aim: a precise estimator (stable) and noise resistant.

2.2 Estimating curvature on (noisy) digital contours

Context:

- Common with Jaques-Olivier Lachaud (LAMA, Chambéry).
- ANR Project GeoDIB (2006-2011): Digital Geometry on Noisy Objects.
- Aim: a precise estimator (stable) and noise resistant.

. . . under the conditions typical for digital image processing the curvature can rarely be estimated with a precision higher than 50% [Kovalevsky 01].

2.2 Estimating curvature on (noisy) digital contours

Different approach to estimate the curvature on digital objects

- Analytic definition: approximation with a parametric curve $(x(t), y(t))$,

$$\kappa = \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{\frac{3}{2}}}$$

2.2 Estimating curvature on (noisy) digital contours

Different approach to estimate the curvature on digital objects

- Analytic definition: approximation with a parametric curve $(x(t), y(t))$,

$$\kappa = \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{\frac{3}{2}}}$$

- Derivative of the Tangent θ:

$\theta[k]$: $\kappa = \theta * G'_\sigma$ [Vialard 96]

$\theta(s)$: $\kappa = \frac{d\theta^{\lambda-MST}}{ds}$ [Lachaud *et al.* 05]

2.2 Estimating curvature on (noisy) digital contours

Different approach to estimate the curvature on digital objects

- Analytic definition: approximation with a parametric curve $(x(t), y(t))$,

$$\kappa = \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{\frac{3}{2}}}$$

- Derivative of the Tangent θ:

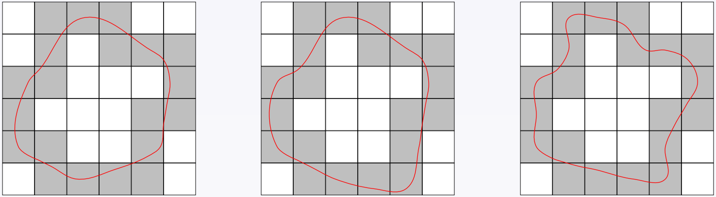
$$\theta[k]: \quad \kappa = \theta * G'_\sigma \qquad \text{[Vialard 96]}$$
$$\theta(s): \quad \kappa = \frac{d\theta^{\lambda-MST}}{ds} \qquad \text{[Lachaud } \textit{et al.} \text{ 05]}$$

- Estimation based on the osculating circle of radius r, $\kappa = \frac{1}{r}$
  - Link to the cord length and the radius [Coeurjolly *et al.* 01]
  - Circumscribed circle defined from half tangent [Coeurjolly, Svensson 03]
- Circle arc recognition r, $\kappa = \frac{1}{r}$ [Coeurjolly *et al.* 04].

GMC estimator: main idea

Evaluation criteria of the quality of a geometric estimator.

Infinity of shapes can correspond to a same digitization !



GMC estimator: main idea

Evaluation criteria of the quality of a geometric estimator.

Infinity of shapes can correspond to a same digitization !

Convergence of geometric estimators

Let G be a geometric measure, the estimator E_G is defined as multigrid-convergent towards G if and only if for all shapes $X \in \mathbb{F}$, there exits $h_X > 0$ such as :

$$\forall h, 0 < h < h_X, \|E_G(\mathrm{Dig}_h(X)) - G(X)\| \leq \tau(h),$$

where $\tau : \mathbb{R}^+ \rightarrow \mathbb{R}^{+*}$ has a limit value in 0 for $h = 0$,
and $\tau(h)$ defines the convergence speed of E_G near G.

GMC estimator: main idea

Evaluation criteria of the quality of a geometric estimator.

Infinity of shapes can correspond to a same digitization !

Convergence of geometric estimators

Let G be a geometric measure, the estimator E_G is defined as multigrid-convergent towards G if and only if for all shapes $X \in \mathbb{F}$, there exits $h_X > 0$ such as :

$$\forall h, 0 < h < h_X, \|E_G(\mathrm{Dig}_h(X)) - G(X)\| \leq \tau(h),$$

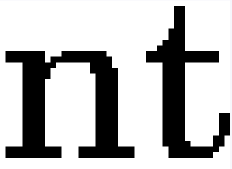
where $\tau : \mathbb{R}^+ \to \mathbb{R}^{+*}$ has a limit value in 0 for $h = 0$,
and $\tau(h)$ defines the convergence speed of E_G near G.

Several criticism:

- Precision only guarantee with a large resolution.
- Convergence property is only true for a perfect digitization.

GMC estimator: main idea

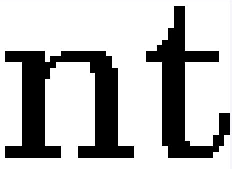
Example:

Perfect Digitization		
200 dpi	300 dpi	400 dpi
		

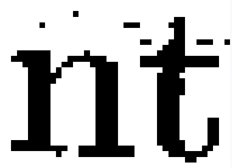
GMC estimator: main idea

Example:

Perfect Digitization

200 dpi	300 dpi	400 dpi
		

Result from printing and scan

200 dpi	300 dpi	400 dpi
		

GMC estimator: main idea

Objectives: [Kerautret and Lachaud 2009]

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

GMC estimator: main idea

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).

GMC estimator: main idea

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]

GMC estimator: main idea

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).

Tangential cover and tangent space (1)

Definition: maximal segments

Maximal segments which can not be extended on the right or left.
Predicate $S(i, j)$: set of points of index i to j constituting a digital segment.

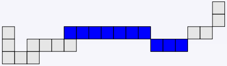
$$S(i, j) \wedge \neg S(i, j+1) \wedge \neg S(i-1, j)$$

Tangential cover and tangent space (1)

Definition: maximal segments

Maximal segments which can not be extended on the right or left.
Predicate $S(i,j)$: set of points of index i to j constituting a digital segment.

$$S(i,j) \wedge \neg S(i,j+1) \wedge \neg S(i-1,j)$$

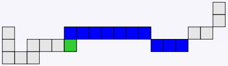


Tangential cover and tangent space (1)

Definition: maximal segments

Maximal segments which can not be extended on the right or left.
Predicate $S(i,j)$: set of points of index i to j constituting a digital segment.

$$S(i,j) \wedge \neg S(i,j+1) \wedge \neg S(i-1,j)$$

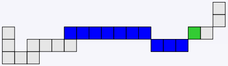


Tangential cover and tangent space (1)

Definition: maximal segments

Maximal segments which can not be extended on the right or left.
Predicate $S(i, j)$: set of points of index i to j constituting a digital segment.

$$S(i, j) \wedge \neg S(i, j+1) \wedge \neg S(i-1, j)$$

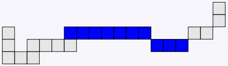


Tangential cover and tangent space (1)

Definition: maximal segments

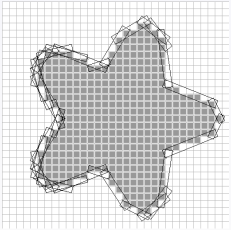
Maximal segments which can not be extended on the right or left.
Predicate $S(i,j)$: set of points of index i to j constituting a digital segment.

$$S(i,j) \wedge \neg S(i,j+1) \wedge \neg S(i-1,j)$$



Tangential Cover

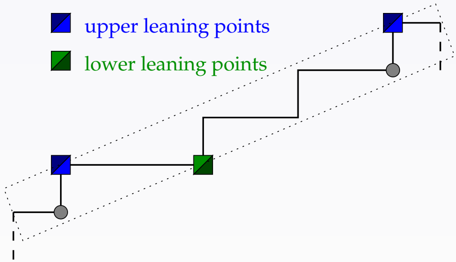
Set of maximal segments defined along the digital contour.
$\Rightarrow$ used to estimate the tangent of a contour [Lachaud *et al.* 05].



Tangential cover and tangent space (2)

Estimation of the tangent:

- Estimation of the tangent slope is defined in an given uncertainty.
- Defined from the upper and lower leaning points.



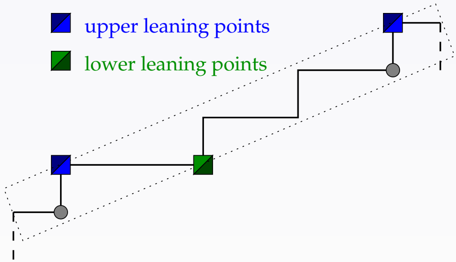
Tangential cover and tangent space (2)

Estimation of the tangent:

- Estimation of the tangent slope is defined in an given uncertainty.
- Defined from the upper and lower leaning points.
- For a maximal segment of characteristics (a, b), we have a:

$$p_{min} = \frac{a}{b} - \frac{1}{b}$$

$$p_{max} = \frac{a}{b} + \frac{1}{b}$$

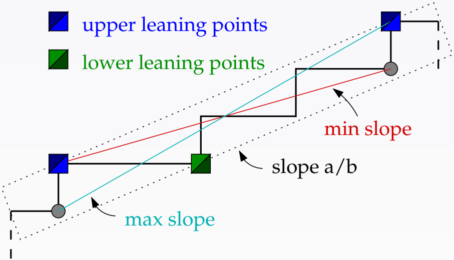


Tangential cover and tangent space (2)

Estimation of the tangent:

- Estimation of the tangent slope is defined in an given uncertainty.
- Defined from the upper and lower leaning points.
- For a maximal segment of characteristics (a, b), we have a:

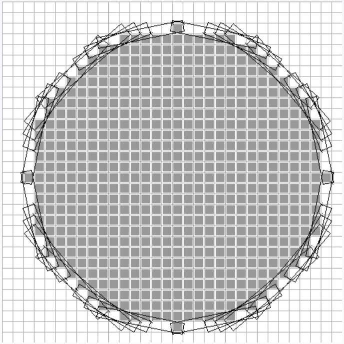
$$p_{min} = \frac{a}{b} - \frac{1}{b}$$
$$p_{max} = \frac{a}{b} + \frac{1}{b}$$

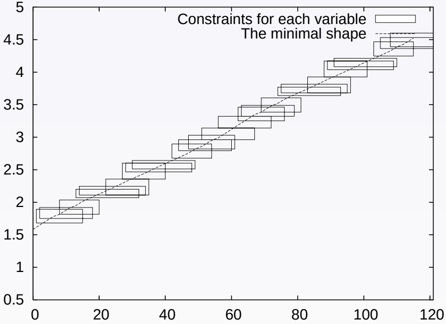


Tangential cover and tangent space (3)

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 $\max(\theta^i_{max})$.

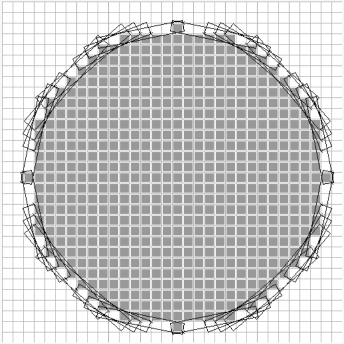


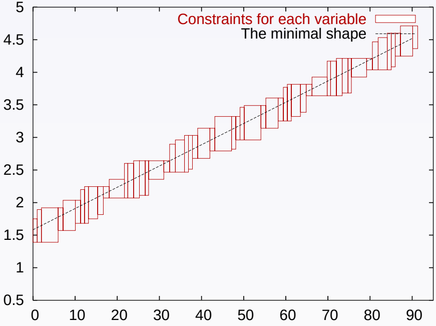


Tangential cover and tangent space (3)

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 $\max(\theta^i_{max})$.





Tangential cover and tangent space (3): adaptation to noisy data

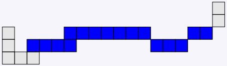
Maximal Blurred Segments

- Algorithm of recognition [Debled06 *et al.*].
- Blurred segments which can not be extended.

Tangential cover and tangent space (3): adaptation to noisy data

Maximal Blurred Segments

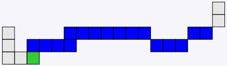
- Algorithm of recognition [Debled06 *et al.*].
- Blurred segments which can not be extended.



Tangential cover and tangent space (3): adaptation to noisy data

Maximal Blurred Segments

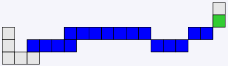
- Algorithm of recognition [Debled06 *et al.*].
- Blurred segments which can not be extended.



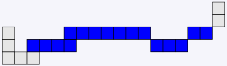
Tangential cover and tangent space (3): adaptation to noisy data

Maximal Blurred Segments

- Algorithm of recognition [Debled06 *et al.*].
- Blurred segments which can not be extended.



Tangential cover and tangent space (3): adaptation to noisy data



Maximal Blurred Segments

- Algorithm of recognition [Debled06 *et al.*].
- Blurred segments which can not be extended.

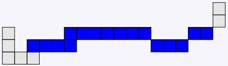
Recognition Algorithm:

- Hypothesis: points added by increasing x value.
- Can be removed by using the equivalent Alpha-Thick segments (see next section).

Tangential cover and tangent space (3): adaptation to noisy data

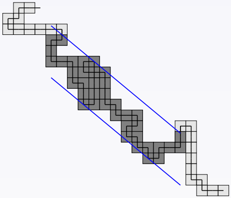
Maximal Blurred Segments

- Algorithm of recognition [Debled06 *et al.*].
- Blurred segments which can not be extended.



Recognition Algorithm:

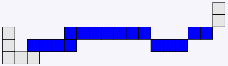
- Hypothesis: points added by increasing x value.
- Can be removed by using the equivalent Alpha-Thick segments (see next section).



Tangential cover and tangent space (3): adaptation to noisy data

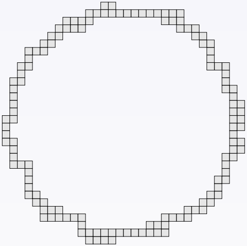
Maximal Blurred Segments

- Algorithm of recognition [Debled06 *et al.*].
- Blurred segments which can not be extended.



Recognition Algorithm:

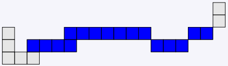
- Hypothesis: points added by increasing x value.
- Can be removed by using the equivalent Alpha-Thick segments (see next section).



Tangential cover and tangent space (3): adaptation to noisy data

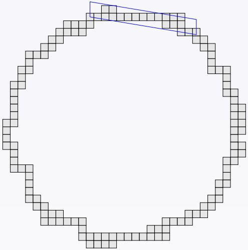
Maximal Blurred Segments

- Algorithm of recognition [Debled06 *et al.*].
- Blurred segments which can not be extended.



Recognition Algorithm:

- Hypothesis: points added by increasing x value.
- Can be removed by using the equivalent Alpha-Thick segments (see next section).

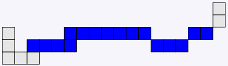


width 2

Tangential cover and tangent space (3): adaptation to noisy data

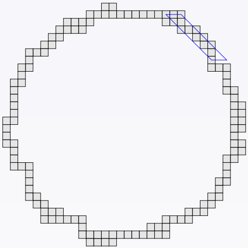
Maximal Blurred Segments

- Algorithm of recognition [Debled06 *et al.*].
- Blurred segments which can not be extended.



Recognition Algorithm:

- Hypothesis: points added by increasing x value.
- Can be removed by using the equivalent Alpha-Thick segments (see next section).

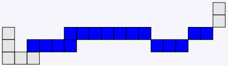


width 2

Tangential cover and tangent space (3): adaptation to noisy data

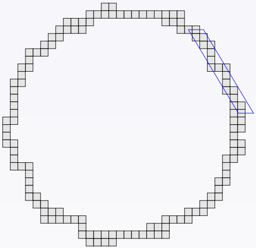
Maximal Blurred Segments

- Algorithm of recognition [Debled06 *et al.*].
- Blurred segments which can not be extended.



Recognition Algorithm:

- Hypothesis: points added by increasing x value.
- Can be removed by using the equivalent Alpha-Thick segments (see next section).

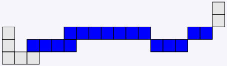


width 2

Tangential cover and tangent space (3): adaptation to noisy data

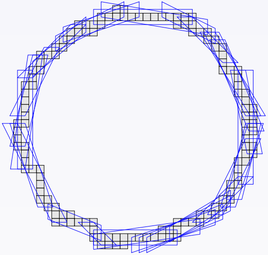
Maximal Blurred Segments

- Algorithm of recognition [Debled06 *et al.*].
- Blurred segments which can not be extended.



Recognition Algorithm:

- Hypothesis: points added by increasing x value.
- Can be removed by using the equivalent Alpha-Thick segments (see next section).

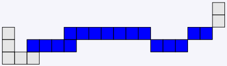


width 2

Tangential cover and tangent space (3): adaptation to noisy data

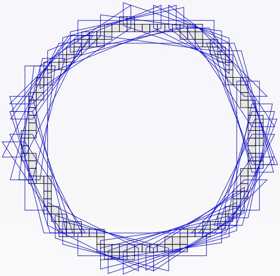
Maximal Blurred Segments

- Algorithm of recognition [Debled06 *et al.*].
- Blurred segments which can not be extended.



Recognition Algorithm:

- Hypothesis: points added by increasing *x* value.
- Can be removed by using the equivalent Alpha-Thick segments (see next section).

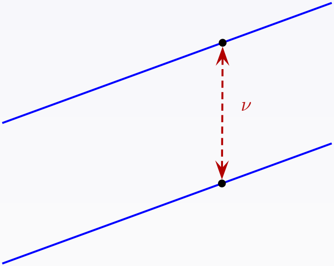


width 3

Tangential cover and tangent space (3): adaptation to noisy data

Interval of the tangent directions: $R = [\theta_{min}, \theta_{max}]$

- Depends of the width ν of the blurred segment.

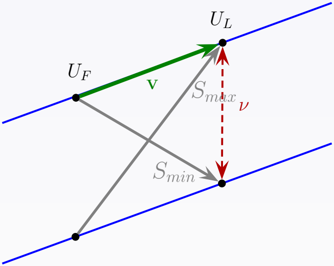


Tangential cover and tangent space (3): adaptation to noisy data

Interval of the tangent directions: $R = [\theta_{min}, \theta_{max}]$

- Depends of the width ν of the blurred segment.
- Defined from the vector $V(v_x, v_y)$ from the leaning points of the convex hull:

$$R = \left[\tan^{-1}\left(\frac{v_y + \nu}{v_x}\right), \tan^{-1}\left(\frac{v_y - \nu}{v_x}\right)\right]$$



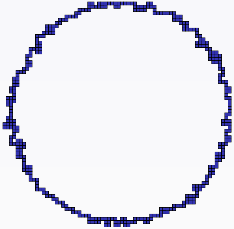
Tangential cover and tangent space (3): adaptation to noisy data

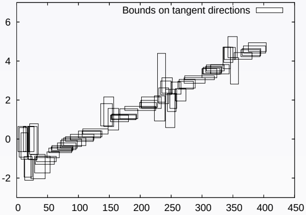
Interval of the tangent directions: $R = [\theta_{min}, \theta_{max}]$

- Depends of the width ν of the blurred segment.
- Defined from the vector $V(v_x, v_y)$ from the leaning points of the convex hull:

$$R = \left[\tan^{-1}\left(\frac{v_y + \nu}{v_x}\right), \tan^{-1}\left(\frac{v_y - \nu}{v_x}\right)\right]$$

Example:





Tangential cover and tangent space (3): adaptation to noisy data

Computation of values θ_{min} and θ_{max} for the common parts

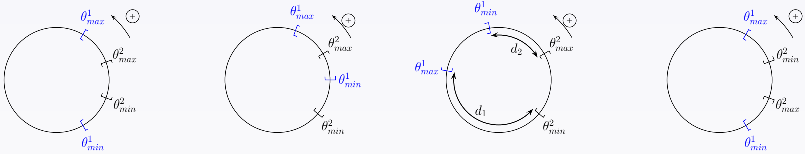
- Previous solution: $min(\theta^i_{min})$ et $max(\theta^i_{max})$
  $\Rightarrow$ not always coherent with intervals of size larger than π.
- Process by successively merging intervals defined from different configurations.

Tangential cover and tangent space (3): adaptation to noisy data

Computation of values θ_{min} and θ_{max} for the common parts

- Previous solution: $min(\theta^i_{min})$ et $max(\theta^i_{max})$
  $\Rightarrow$ not always coherent with intervals of size larger than π.
- Process by successively merging intervals defined from different configurations.

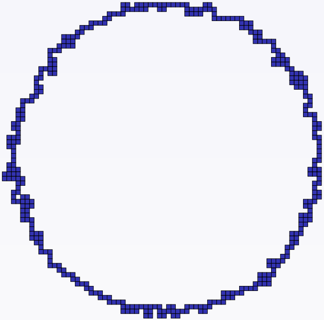
Examples of configurations for the merging step of $I_1 = [\theta^1_{min}, \theta^1_{max}]$ and $I_2 = [\theta^2_{min}, \theta^2_{max}]$:



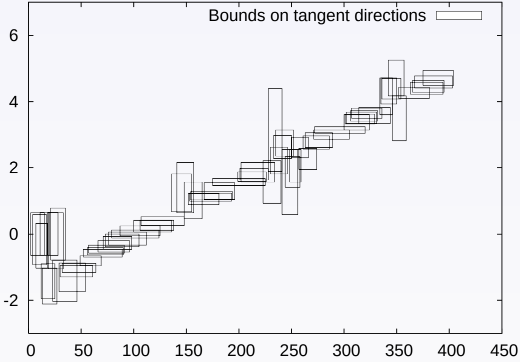
(a)	(b)	(c)	(d)
$I = [\theta^1_{min}, \theta^1_{max}]$	$I = [\theta^2_{min}, \theta^1_{max}]$	$I = [\theta^2_{min}, \theta^1_{max}]$	$I = [0, 2\pi]$

Tangential cover and tangent space (3): adaptation to noisy data

Example:



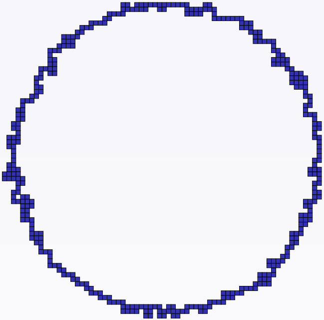
(a)



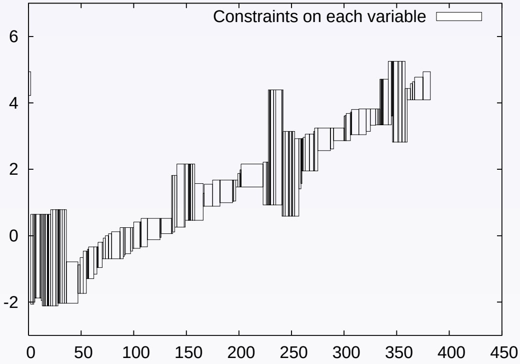
(b)

Tangential cover and tangent space (3): adaptation to noisy data

Example:



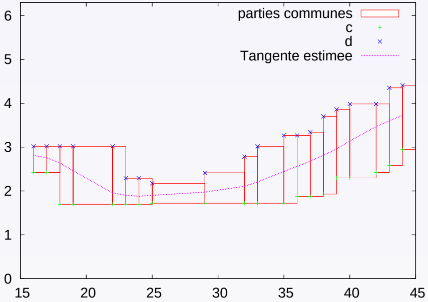
(a)



(b)

Curvature computation from optimization (1)

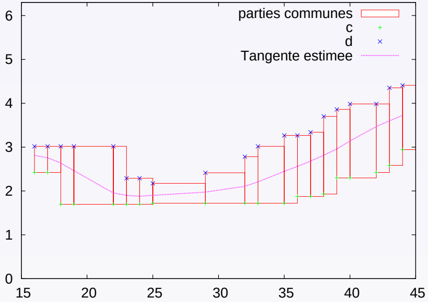
Problem formulation



- x_i: curvilinear abscissa
- c_i, d_i : direction constraints min/max.
- y_i estimated tangent in x_i
- $\forall i \;\; c_i \leq y_i \leq d_i$
- tangent $\theta(s)$ = piecewise linear reconstruction (x_i, y_i)

Curvature computation from optimization (1)

Problem formulation



- x_i: curvilinear abscissa
- c_i, d_i : direction constraints min/max.
- y_i estimated tangent in x_i
- $\forall i \;\; c_i \leq y_i \leq d_i$
- tangent $\theta(s) =$ piecewise linear reconstruction (x_i, y_i)

Curvature

- Distance between x_i and x_{i+1} estimated by the $\lambda - MST$ estimator [Lachaud *et al.* 05].
- Each segment $(x_i, y_i) - (x_{i+1}, y_{i+1})$ are an arc circle in the euclidean plane of curvature equals to the slope.
- Reconstructed curve is C^1 and piecewise C^∞.

Curvature computation from optimization (2)

Minimization $\int \kappa^2 ds$:

$$J[(y_i)] = \int \kappa^2 ds = \sum_i \left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right)^2 (x_{i+1} - x_i) = \sum_i \frac{(y_{i+1} - y_i)^2}{(x_{i+1} - x_i)}$$

Minimization

Minimizing $J[(y_i)]$ with $\forall i \;\; c_i \leq y_i \leq d_i$

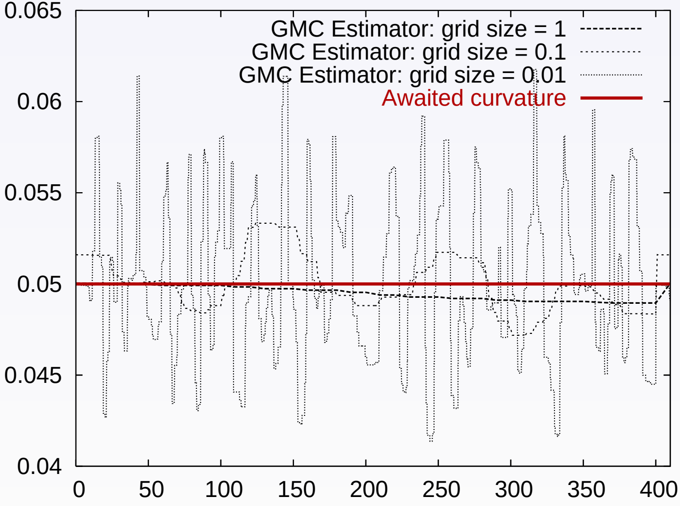
Number of variable to be optimized: $O(N^{\frac{2}{3}})$ [Devieilleville05]

Implementation

Gradient descent: y_i put on the segment $(x_{i-1}, y_{i-1}) - (x_{i+1}, y_{i+1})$

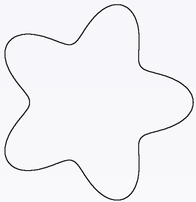
Results of curvature estimation

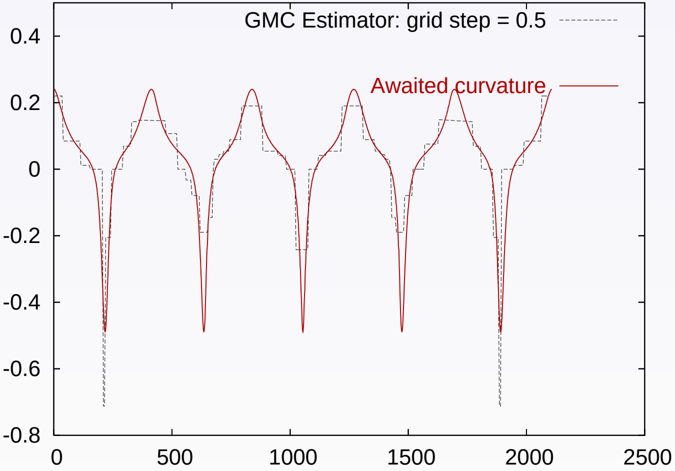
Application on non noisy shape: circle $R = 20$



Results of curvature estimation (2)

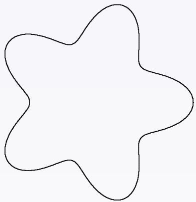
Application on non noisy shape: flower with 5 petals

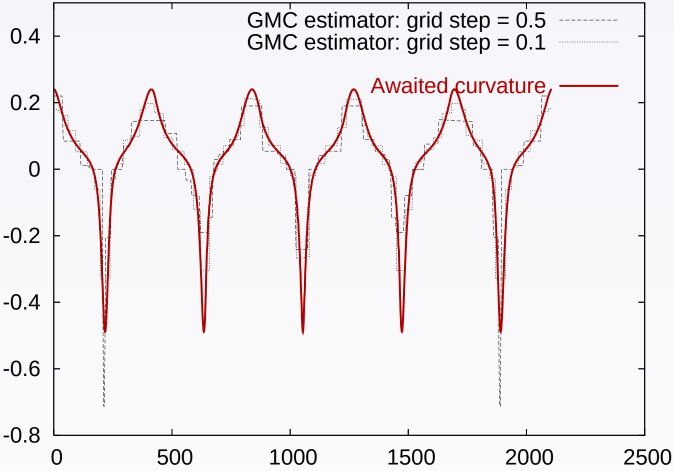




Results of curvature estimation (2)

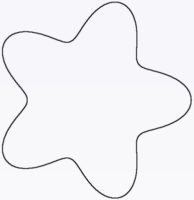
Application on non noisy shape: flower with 5 petals

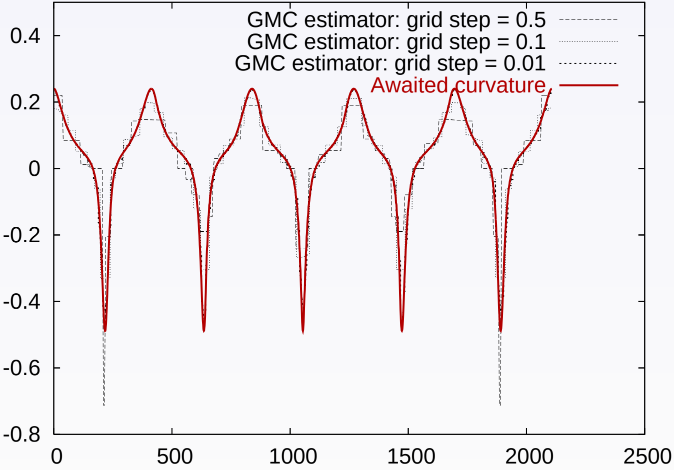




Results of curvature estimation (2)

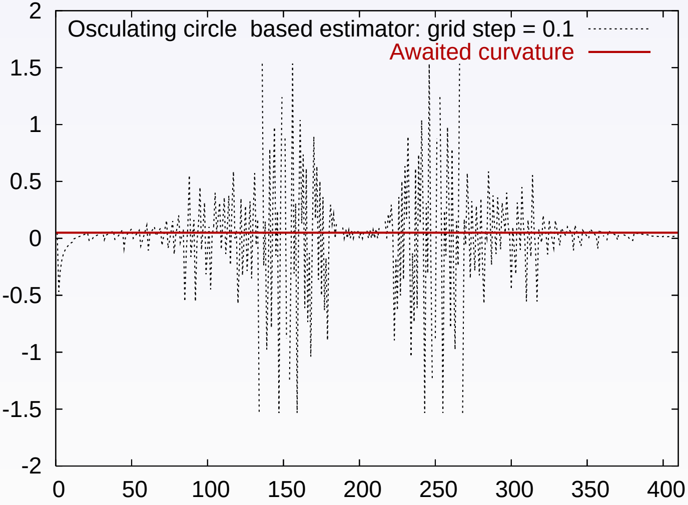
Application on non noisy shape: flower with 5 petals





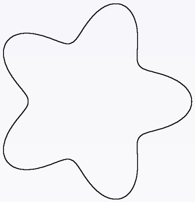
Results of curvature estimation (3): comparisons

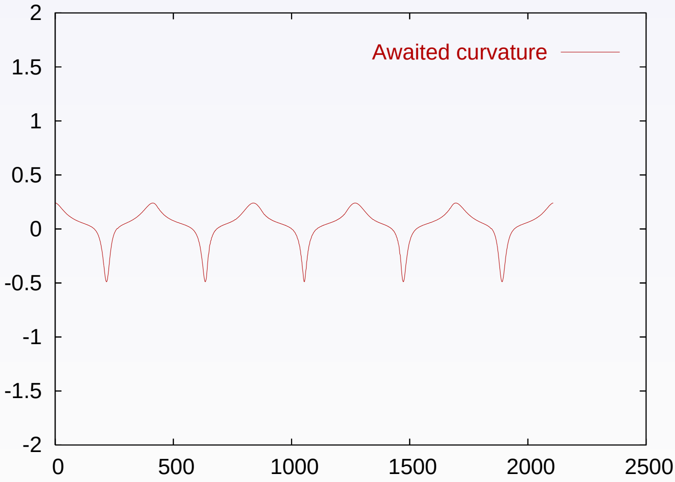
Method based on osculating circle estimation [Coeurjolly *et al.* 01]
(Comparaison on a circle of radius 20, grid size = 0.1)



Results of curvature estimation (3): comparisons

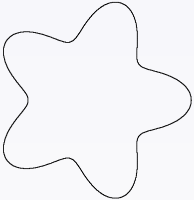
Method based on osculating circle estimation [Coeurjolly *et al.* 01]
(Comparisons on the flower shape with 5 petals, grid size = 0.1)

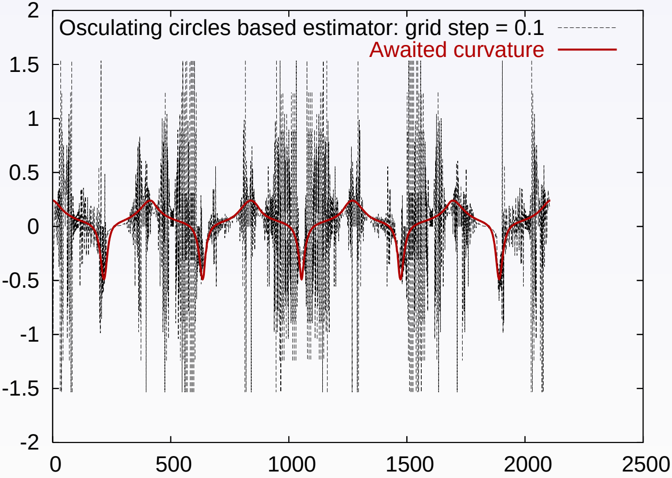




Results of curvature estimation (3): comparisons

Method based on osculating circle estimation [Coeurjolly *et al.* 01]
(Comparisons on the flower shape with 5 petals, grid size = 0.1)





Results of curvature estimation (3): on noisy contours

Generating noisy contours

Principle: [Kanungo96]

Based on the document model degradation:

- Degradation defined according the distance contour.
- Pixels are changed according a Gaussian distribution.
- Extract the connected components.

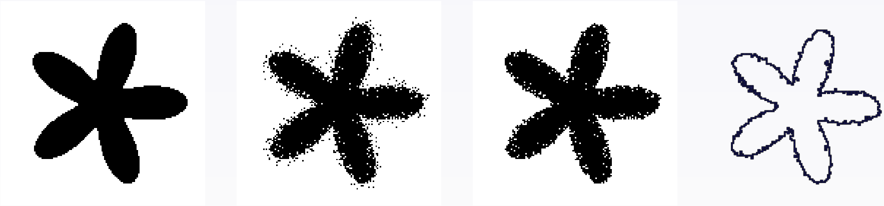
Results of curvature estimation (3): on noisy contours

Generating noisy contours

Principle: [Kanungo96]

Based on the document model degradation:

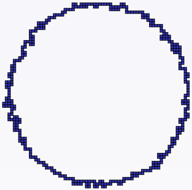
- Degradation defined according the distance contour.
- Pixels are changed according a Gaussian distribution.
- Extract the connected components.

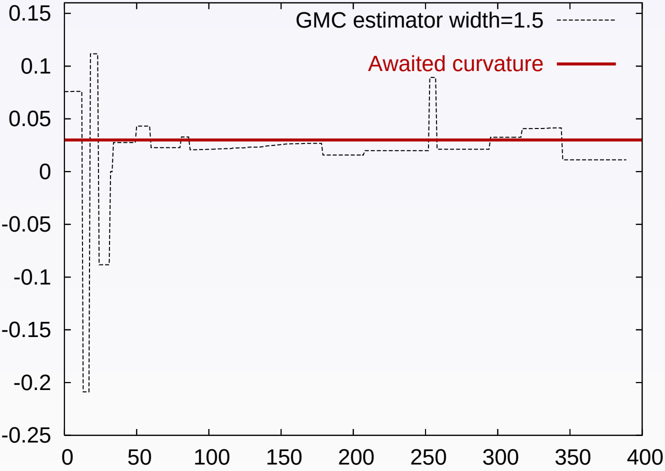


(a) (b) (c) (d)

Results of curvature estimation (3): on noisy contours

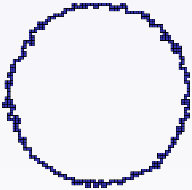
Results in a noisy circle

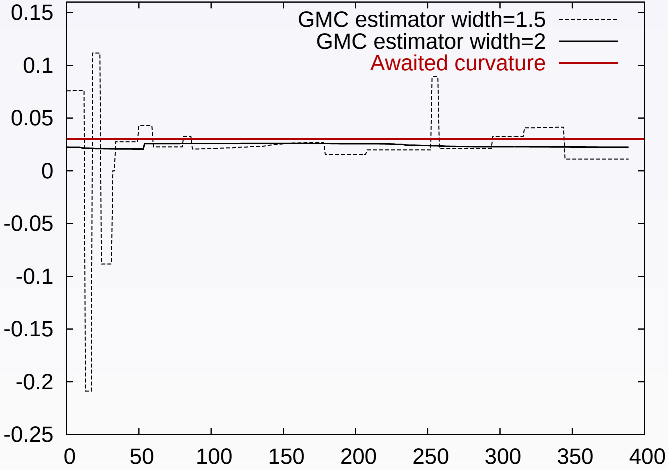




Results of curvature estimation (3): on noisy contours

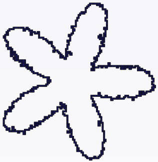
Results in a noisy circle

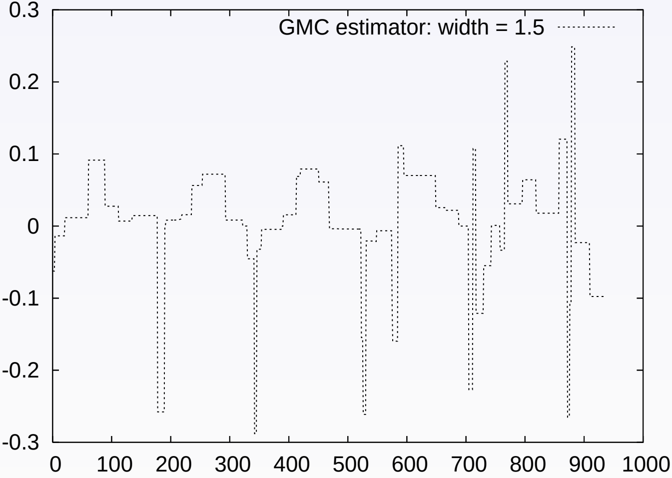




Results of curvature estimation (3): on noisy contours

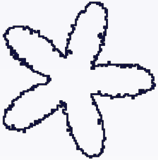
Results in a noisy flower

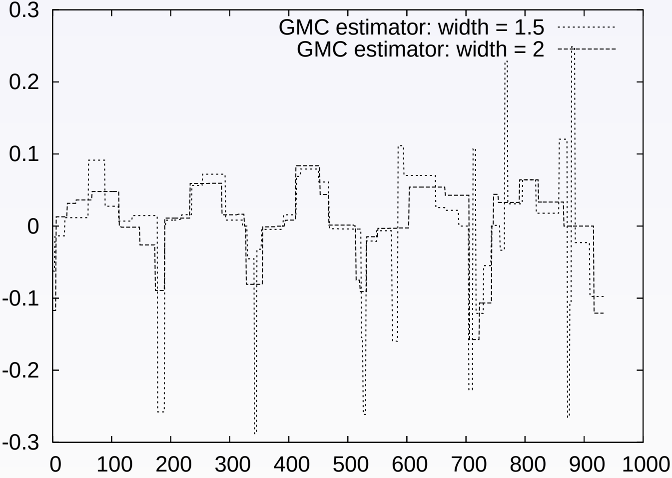




Results of curvature estimation (3): on noisy contours

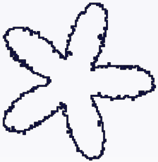
Results in a noisy flower

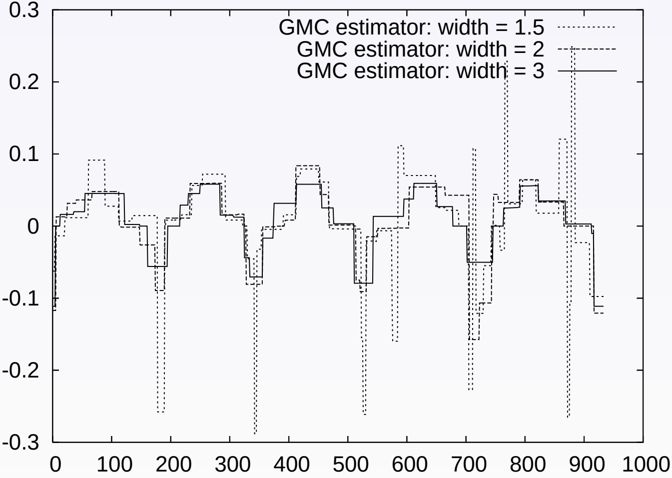




Results of curvature estimation (3): on noisy contours

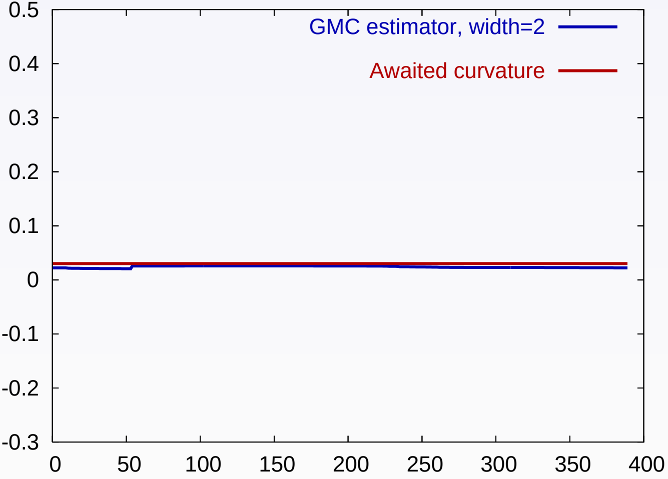
Results in a noisy flower





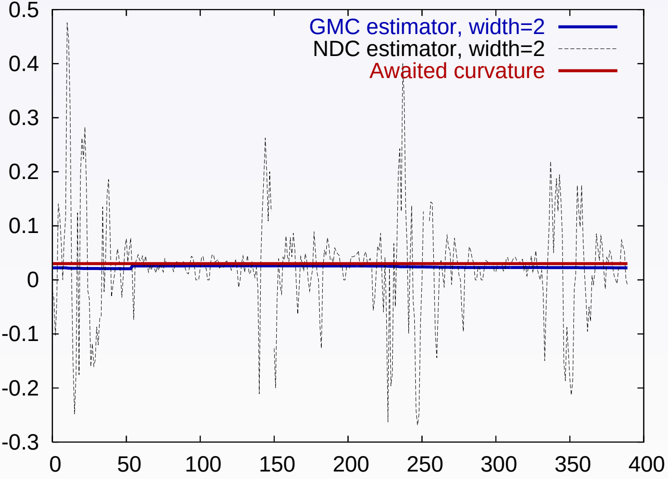
Results of curvature estimation (4): on noisy contours (comparisons)

Comparisons of GMC with the estimator based on the osculating circles and blurred segments (NDC) [Nguyen, Debled 07]
noisy circle



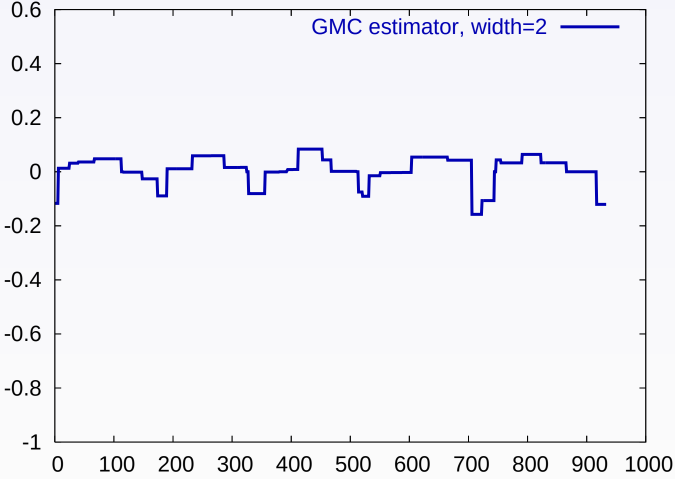
Results of curvature estimation (4): on noisy contours (comparisons)

Comparisons of GMC with the estimator based on the osculating circles and blurred segments (NDC) [Nguyen, Debled 07]
noisy circle



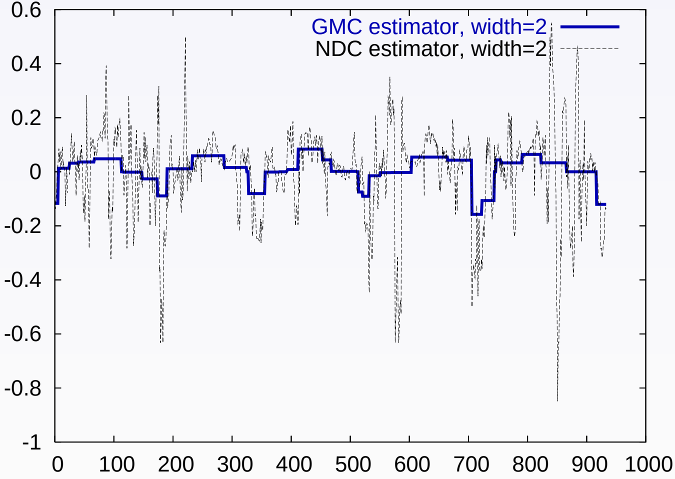
Results of curvature estimation (4): on noisy contours (comparisons)

Comparisons of GMC with the estimator based on the osculating circles and blurred segments (NDC) [Nguyen, Debled 07]
noisy flower



Results of curvature estimation (4): on noisy contours (comparisons)

Comparisons of GMC with the estimator based on the osculating circles and blurred segments (NDC) [Nguyen, Debled 07]
noisy flower



Reproducing the GMC estimators

From source code

- Available in the ImaGene Library (ancestor of DGtal Library): `https://gforge.liris.cnrs.fr/projects/imagene`
- Version adapted to noise: `http://kerrecherche.iutsd.uhp-nancy.fr/Ker`
- Backport in DGtal Library in progress.

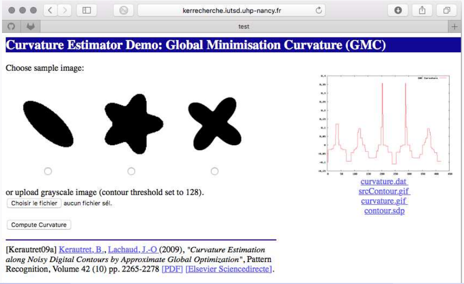
Reproducing the GMC estimators

From source code

- Available in the ImaGene Library (ancestor of DGtal Library): `https://gforge.liris.cnrs.fr/projects/imagene`
- Version adapted to noise: `http://kerrecherche.iutsd.uhp-nancy.fr/Ker`
- Backport in DGtal Library in progress.

From online demonstration

`http://kerrecherche.iutsd.uhp-nancy.fr/gmc/`



2.3 Application to contour representation: Corner Detection

(Common work with J.O Lachaud and B. Naegel [Kerautret08])

Exploitation of the GMC estimator:

- Stability, no oscillation in the estimation.

2.3 Application to contour representation: Corner Detection

(Common work with J.O Lachaud and B. Naegel [Kerautret08])

Exploitation of the GMC estimator:

- Stability, no oscillation in the estimation.
- Easy to analyse.

2.3 Application to contour representation: Corner Detection

(Common work with J.O Lachaud and B. Naegel [Kerautret08])

Exploitation of the GMC estimator:

- Stability, no oscillation in the estimation.
- Easy to analyse.
- Only one important parameter defined from the width of the blurred segments.

2.3 Application to contour representation: Corner Detection

(Common work with J.O Lachaud and B. Naegel [Kerautret08])

Exploitation of the GMC estimator:

- Stability, no oscillation in the estimation.
- Easy to analyse.
- Only one important parameter defined from the width of the blurred segments.

Simple algorithm of corner detection:

1. Computation of the curvature $\kappa(p_i)$ for each contour point with the GMC estimator with a widh ν.

2.3 Application to contour representation: Corner Detection

(Common work with J.O Lachaud and B. Naegel [Kerautret08])

Exploitation of the GMC estimator:

- Stability, no oscillation in the estimation.
- Easy to analyse.
- Only one important parameter defined from the width of the blurred segments.

Simple algorithm of corner detection:

1. Computation of the curvature $\kappa(p_i)$ for each contour point with the GMC estimator with a widh ν.
2. Detection of all regions with maximal curvature defined by a set of points:

$$R_k = \{(p_i)_{i\in[a,b]} \mid \forall i, (\kappa(p_i) = \kappa(p_a)) \wedge (\kappa(p_{a-1}) < \kappa(p_a)) \wedge \\ (\kappa(p_{b+1}) < \kappa(p_b)) \wedge (\kappa(p_a) > 0)\}$$

2.3 Application to contour representation: Corner Detection

(Common work with J.O Lachaud and B. Naegel [Kerautret08])

Exploitation of the GMC estimator:

- Stability, no oscillation in the estimation.
- Easy to analyse.
- Only one important parameter defined from the width of the blurred segments.

Simple algorithm of corner detection:

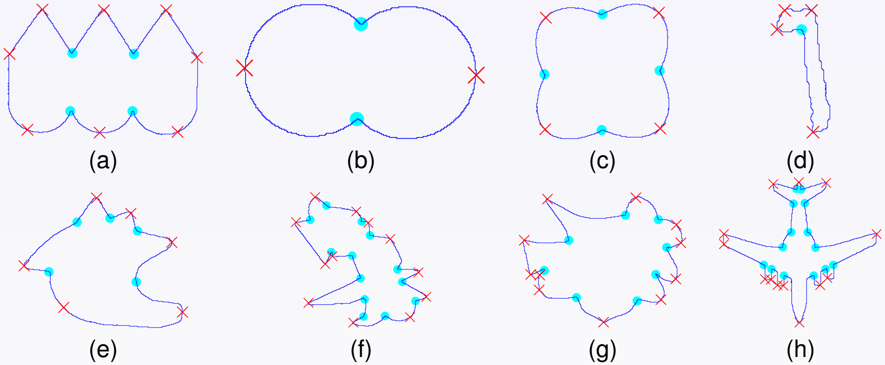
1. Computation of the curvature $\kappa(p_i)$ for each contour point with the GMC estimator with a widh ν.
2. Detection of all regions with maximal curvature defined by a set of points:

$$R_k = \{(p_i)_{i\in[a,b]} \mid \forall i, (\kappa(p_i) = \kappa(p_a)) \wedge (\kappa(p_{a-1}) < \kappa(p_a)) \wedge (\kappa(p_{b+1}) < \kappa(p_b)) \wedge (\kappa(p_a) > 0)\}$$

3. For each region R_k mark the point $p_{(a+b)/2}$ as a corner.

Corner detection: results

Results on classic test shapes:

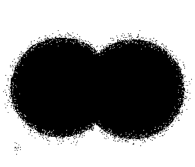


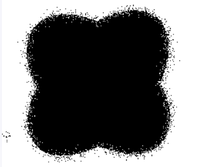
(a) (b) (c) (d) (e) (f) (g) (h)

Corner detection: results

Rajout de bruit sur les formes initiales:


(a)


(b)

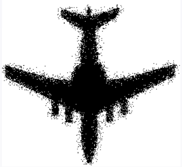

(c)


(d)


(e)

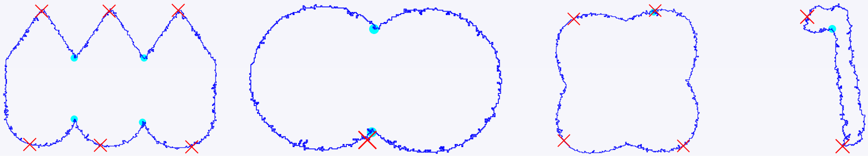

(f)


(g)

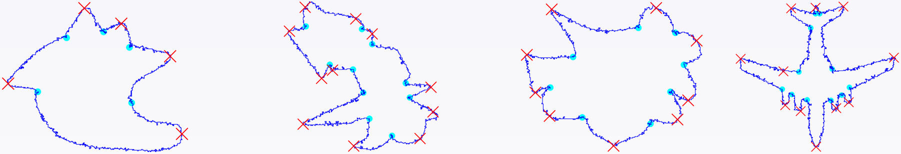

(h)

Corner detection: results

Results obtained with the GMC estimator with width $\nu = 4$:



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



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

Corner detection: comparisons

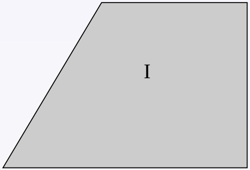
Comparisons with morphological approach: [Chang *et al.* 07]

Principle of the BAP detector (Base Angle Point):

- Application of the “top-hat” operator:

$$C(I) = I \backslash \gamma_{B_\lambda}(I),$$

where $\gamma_B(X) = \cup_i \{B_i \mid B_i \subseteq X\}$.



Corner detection: comparisons

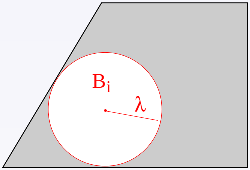
Comparisons with morphological approach: [Chang *et al.* 07]

Principle of the BAP detector (Base Angle Point):

- Application of the "top-hat" operator:

$$C(I) = I \backslash \gamma_{B_\lambda}(I),$$

where $\gamma_B(X) = \cup_i \{B_i \mid B_i \subseteq X\}$.



Corner detection: comparisons

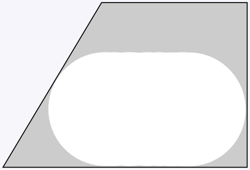
Comparisons with morphological approach: [Chang *et al.* 07]

Principle of the BAP detector (Base Angle Point):

- Application of the “top-hat” operator:

$$C(I) = I \backslash \gamma_{B_\lambda}(I),$$

where $\gamma_B(X) = \cup_i \{B_i \mid B_i \subseteq X\}$.



Corner detection: comparisons

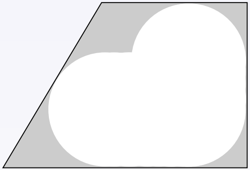
Comparisons with morphological approach: [Chang *et al.* 07]

Principle of the BAP detector (Base Angle Point):

- Application of the "top-hat" operator:

$$C(I) = I \backslash \gamma_{B_\lambda}(I),$$

where $\gamma_B(X) = \cup_i \{B_i \mid B_i \subseteq X\}$.



Corner detection: comparisons

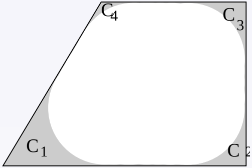
Comparisons with morphological approach: [Chang *et al.* 07]

Principle of the BAP detector (Base Angle Point):

- Application of the "top-hat" operator:

$$C(I) = I \setminus \gamma_{B_\lambda}(I),$$

where $\gamma_B(X) = \cup_i \{B_i \mid B_i \subseteq X\}$.



Corner detection: comparisons

Comparisons with morphological approach: [Chang *et al.* 07]

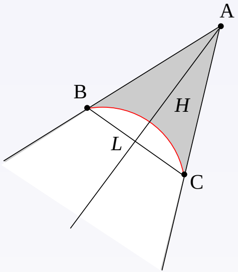
Principle of the BAP detector (Base Angle Point):

- Application of the "top-hat" operator:

$$C(I) = I \backslash \gamma_{B_\lambda}(I),$$

where $\gamma_B(X) = \cup_i \{B_i \mid B_i \subseteq X\}$.

- Analyse of the residual areas.
- Filter to remove non significant residual areas.



Corner detection: comparisons

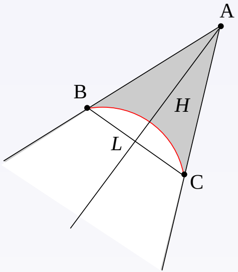
Comparisons with morphological approach: [Chang *et al.* 07]

Principle of the BAP detector (Base Angle Point):

- Application of the "top-hat" operator:

$$C(I) = I \backslash \gamma_{B_\lambda}(I),$$

  where $\gamma_B(X) = \cup_i \{B_i \mid B_i \subseteq X\}$.
- Analyse of the residual areas.
- Filter to remove non significant residual areas.



Main drawbacks:

- Noise sensibility.
- Three parameters (λ, L, H) not easy to define for noisy shapes.

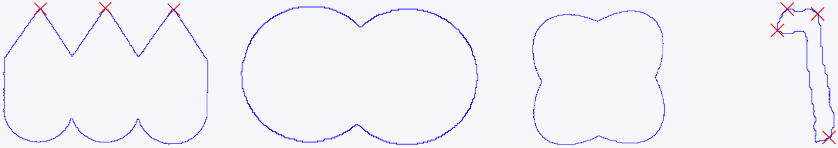
Corner detection: comparisons (2)

Comparisons with morphological approach: [Chang *et al.* 07]

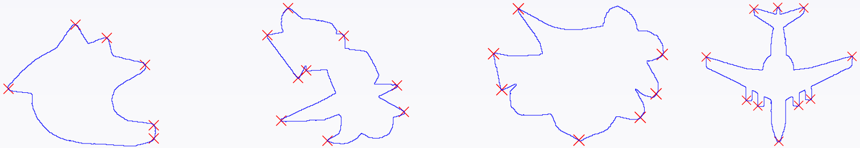
Corner detection: comparisons (2)

Comparisons with morphological approach: [Chang *et al.* 07]

Results obtained on non noisy images with the parameters: $(\lambda, L, H) = (12, 2, 1)$



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

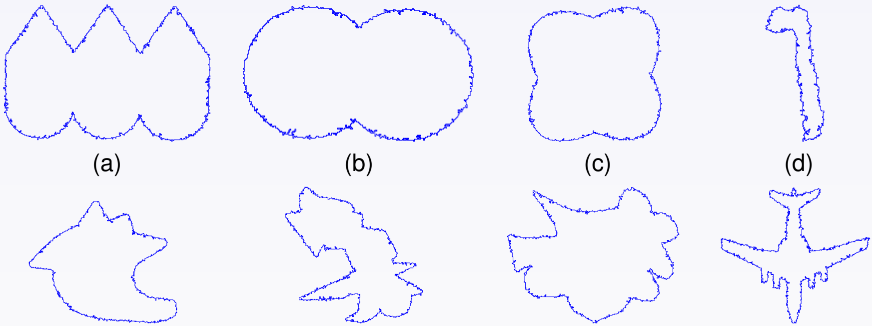


(e) (f) (g) (h)

Corner detection: comparisons (3)

Comparisons with morphological approach: [Chang *et al.* 07]

Results obtained on noisy images with parameters: $(\lambda, L, H) = (12, 2, 1)$



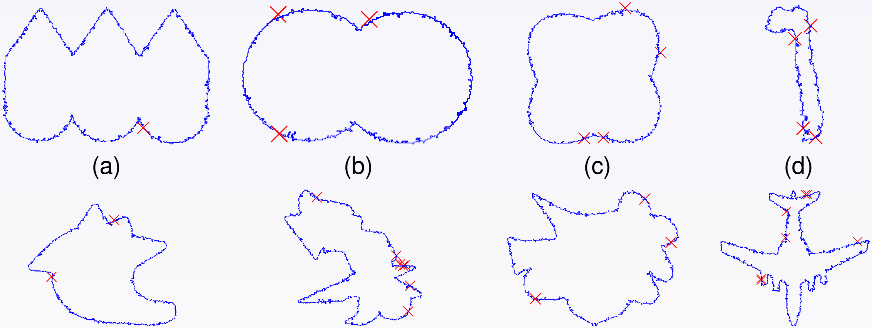
(a) (b) (c) (d)

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

Corner detection: comparisons (3)

Comparisons with morphological approach: [Chang *et al.* 07]

Results obtained on noisy images with parameters: $(\lambda, L, H) = (3, 1, 1)$

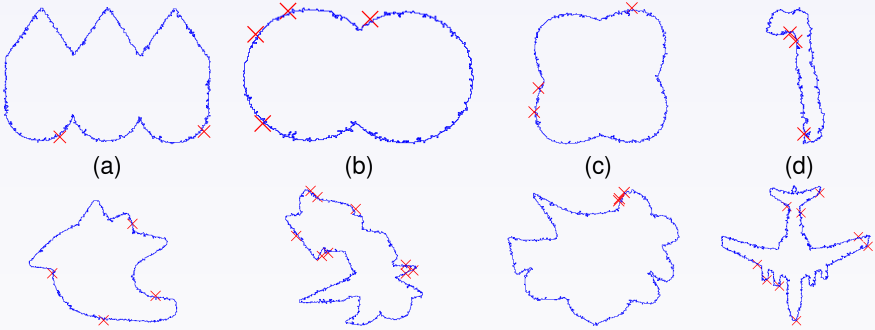


(a) (b) (c) (d) (e) (f) (g) (h)

Corner detection: comparisons (3)

Comparisons with morphological approach: [Chang *et al.* 07]

Results obtained on noisy images with parameters: $(\lambda, L, H) = (2, 2, 1)$



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

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

2.3 Application to contour representation: Arcs/Segments

Digital contour representation:

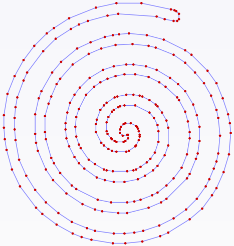
- Numerous methods are based on the linear structures (see for instance: [Feschet 2010] or [Sivignon 2011]).
- Reconstruction with circular arcs.

2.3 Application to contour representation: Arcs/Segments

Digital contour representation:

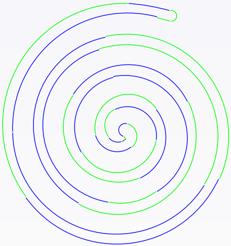
- Numerous methods are based on the linear structures (see for instance: [Feschet 2010] or [Sivignon 2011]).
- Reconstruction with circular arcs.
- More compact representation for numerous shapes.

Curvature based polygon



$\delta_H = 2.018$
288 segments

circular arcs representation

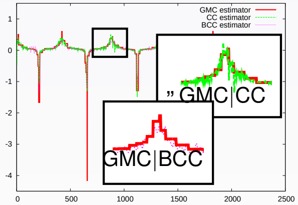


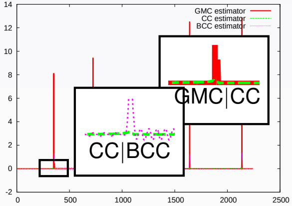
$\delta_H = 2.011$
27 arcs, 1 segment

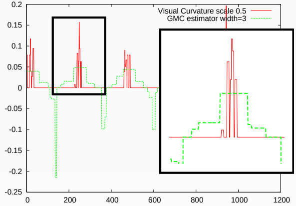
2.3 Application to contour representation: Arcs/Segments (2)

Main ideas [Kerautret11]

- Exploit recent curvature estimators being robust to noise [Kerautret and Lachaud 2009, Malgouyres *et al.* 2008].
- Based on a simple algorithm of type *split and merged*.



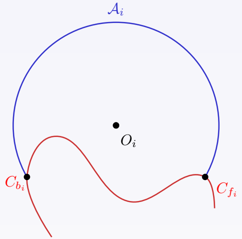




2.3 Application to contour representation: Arcs/Segments (2)

Main ideas [Kerautret11]

- Exploit recent curvature estimators being robust to noise [Kerautret and Lachaud 2009, Malgouyres *et al.* 2008].
- Based on a simple algorithm of type *split and merged*.
- Approximation error defined from an Hausdorff distance approximation δ_H between arc and contour.

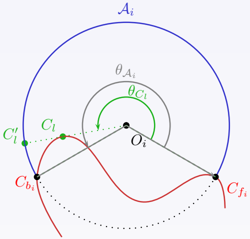


$$\delta_H(\mathcal{A}_i, \mathcal{C}_i) = \max\{\max_{b \in \mathcal{C}_i}\{\min_{a \in \mathcal{A}_i} d(a,b)\}, \max_{a \in \mathcal{A}_i}\{\min_{b \in \mathcal{C}_i} d(a,b)\}\}$$

2.3 Application to contour representation: Arcs/Segments (2)

Main ideas [Kerautret11]

- Exploit recent curvature estimators being robust to noise [Kerautret and Lachaud 2009, Malgouyres *et al.* 2008].
- Based on a simple algorithm of type *split and merged*.
- Approximation error defined from an Hausdorff distance approximation δ_H between arc and contour.

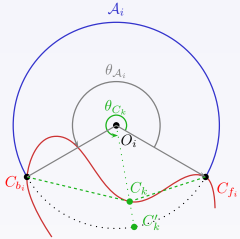


$$\delta_H(\mathcal{A}_i, \mathcal{C}_i) = \max\{\max_{b\in\mathcal{C}_i}\{\min_{a\in\mathcal{A}_i} d(a,b)\}, \max_{a\in\mathcal{A}_i}\{\min_{b\in\mathcal{C}_i} d(a,b)\}\}$$

2.3 Application to contour representation: Arcs/Segments (2)

Main ideas [Kerautret11]

- Exploit recent curvature estimators being robust to noise [Kerautret and Lachaud 2009, Malgouyres *et al.* 2008].
- Based on a simple algorithm of type *split and merged*.
- Approximation error defined from an Hausdorff distance approximation δ_H between arc and contour.

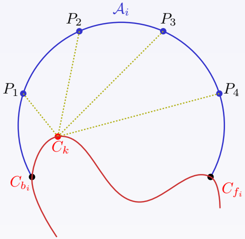


$$\delta_H(\mathcal{A}_i, \mathcal{C}_i) = \max\{\max_{b \in \mathcal{C}_i}\{\min_{a \in \mathcal{A}_i} d(a, b)\}, \max_{a \in \mathcal{A}_i}\{\min_{b \in \mathcal{C}_i} d(a, b)\}\}$$

2.3 Application to contour representation: Arcs/Segments (2)

Main ideas [Kerautret11]

- Exploit recent curvature estimators being robust to noise [Kerautret and Lachaud 2009, Malgouyres *et al.* 2008].
- Based on a simple algorithm of type *split and merged*.
- Approximation error defined from an Hausdorff distance approximation δ_H between arc and contour.

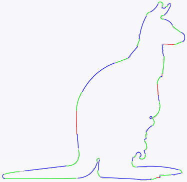


$$\delta_H(\mathcal{A}_i, \mathcal{C}_i) = \max\{\max_{b \in \mathcal{C}_i}\{\min_{a \in \mathcal{A}_i} d(a,b)\}, \max_{a \in \mathcal{A}_i}\{\min_{b \in \mathcal{C}_i} d(a,b)\}\}$$

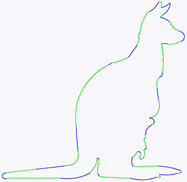
2.3 Application to contour representation: Arcs/Segments (3)

Results and comparisons by using other curvatures estimators [Malgouyres *et al.* 2008] and by usin other method based on the tangent space representation [NguyenDebled10].

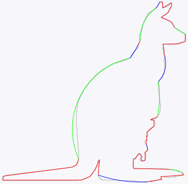
scale=2



GMC: $\overline{A} = 53$, $\overline{S} = 5$
184 ms. $\delta_H = 2.06539$



BCC $\overline{A} = 53$, $\overline{S} = 2$
961 ms. $\delta_H = \mathbf{1.99648}$



NASR $\overline{A} = 10$, $\overline{S} = 40$
115 ms. $\delta_H = 8.81655$

2.3 Application to contour representation: Arcs/Segments (3)

Results and comparisons by using other curvatures estimators [Malgouyres *et al.* 2008] and by usin other method based on the tangent space representation [NguyenDebled10].

scale=4

GMC: $\overline{A} = 31$, $\overline{S} = 4$ 261 ms. $\delta_H = \mathbf{3.94931}$

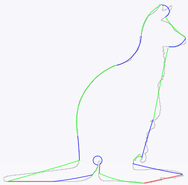
BCC $\overline{A} = 33$, $\overline{S} = 9$ 2990 ms. $\delta_H = 6.11619$

NASR $\overline{A} = 1$, $\overline{S} = 34$ 138 ms. $\delta_H = 9.84886$

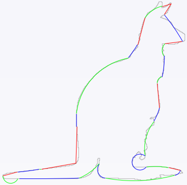
2.3 Application to contour representation: Arcs/Segments (3)

Results and comparisons by using other curvatures estimators [Malgouyres *et al.* 2008] and by usin other method based on the tangent space representation [NguyenDebled10].

scale=10



GMC: $\overline{A} = 17$, $\overline{S} = 3$
464 ms. $\delta_H = \mathbf{10.2849}$

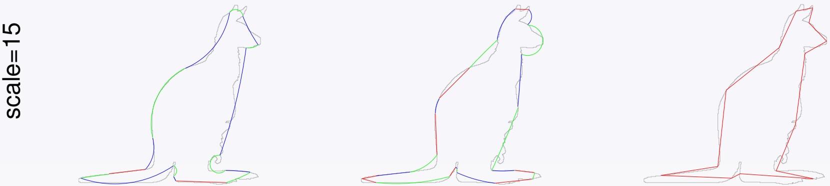


BCC $\overline{A} = 20$, $\overline{S} = 9$
15596 ms. $\delta_H = 10.5289$

NASR $\overline{A} = 1$, $\overline{S} = 16$
171 ms. $\delta_H = 19.6977$

2.3 Application to contour representation: Arcs/Segments (3)

Results and comparisons by using other curvatures estimators [Malgouyres *et al.* 2008] and by usin other method based on the tangent space representation [NguyenDebled10].



GMC: $\overline{A} = 14$, $\overline{S} = 2$ 619 ms. $\delta_H =$ **17.2402**

BCC $\overline{A} = 15$, $\overline{S} = 8$ 33420 ms. $\delta_H = 17.7125$

NASR $\overline{A} = 0$, $\overline{S} = 14$ 190 ms. $\delta_H = 32.8938$

Reproducing the curvature based contour representation

From source code

- Available in the KerUtils repository (depends of ImaGene Libraray) :
- `http://kerrecherche.iutsd.uhp-nancy.fr/Ker`
- ImaGene: `https://gforge.liris.cnrs.fr/projects/imagene`

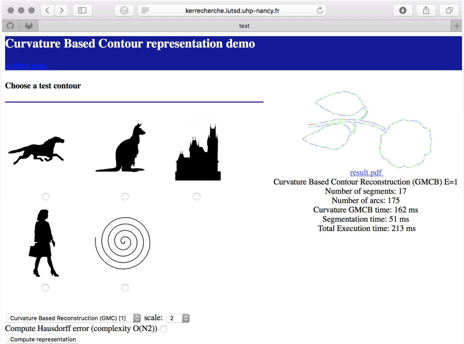
Reproducing the curvature based contour representation

From source code

- Available in the KerUtils repository (depends of ImaGene Libraray) :
- http://kerrecherche.iutsd.uhp-nancy.fr/Ker
- ImaGene: https://gforge.liris.cnrs.fr/projects/imagene

From online demonstration

http://kerrecherche.iutsd.uhp-nancy.fr/CBContours/index.php



Overview of the presentation - Part I -

https://kerautret.github.io/ACCV2016DGPTutorial/

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 during the IWCIA workshop in 2009.
- C++ based library: work (and tested) on *Linux*, *MacOS* and *Windows*.
- Current version 0.9.2 (0.9.3 in preparation: end 2016).
- Receive the SGP Software Award at the Symposium on Geometry Processing: http://awards.geometryprocessing.org

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 during the IWCIA workshop in 2009.
- C++ based library: work (and tested) on *Linux*, *MacOS* and *Windows*.
- Current version 0.9.2 (0.9.3 in preparation: end 2016).
- Receive the SGP Software Award at the Symposium on Geometry Processing: http://awards.geometryprocessing.org

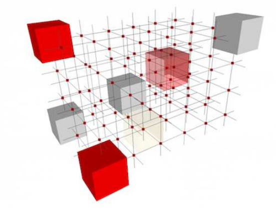
Main Objectives:

- 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:

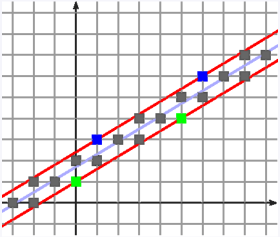
- Kernel package: number types, digital space, domain



3.1 Short presentation of the library (2)

Main actual packages:

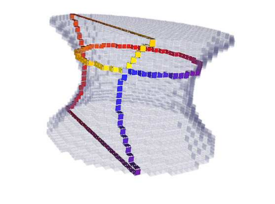
- Kernel package: number types, digital space, domain
- Arithmetic package: standard arithmetic computations
  $\Rightarrow$ greatest common divisor, Bézout vectors, continued fractions, convergent, intersection of integer half-spaces



3.1 Short presentation of the library (2)

Main actual packages:

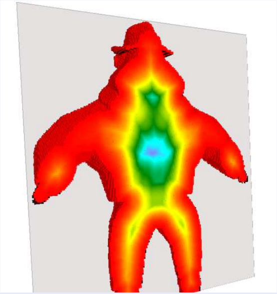
- Kernel package: number types, digital space, domain
- Arithmetic package: standard arithmetic computations
- Topology package: classic topology tools
  $\Rightarrow$ Rosenfeld oriented tools, cartesian cellular topology, digital surface topology (Herman), tools to extract connected component, identifying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



3.1 Short presentation of the library (2)

Main actual packages:

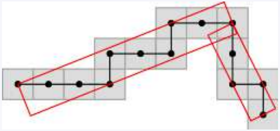
- Kernel package: number types, digital space, domain
- Arithmetic package: standard arithmetic computations
- Topology package: classic topology tools
- Geometry package: geometric estimators 2D/3D:
  ⇒ length, normal curvature estimators, 3D transform...



3.1 Short presentation of the library (2)

Main actual packages:

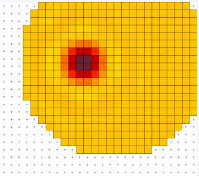
- **Kernel** package: number types, digital space, domain
- **Arithmetic** package: standard arithmetic computations
- **Topology** package: classic topology tools
- **Geometry** package: geometric estimators 2D/3D:
  $\Rightarrow$ length, normal curvature estimators, 3D transform...



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:
  $\Rightarrow$ provides an easy and efficient way to describe linear operator over various structure



3.1 Short presentation of the library (2)

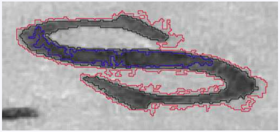
Main actual packages:

- 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:

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)

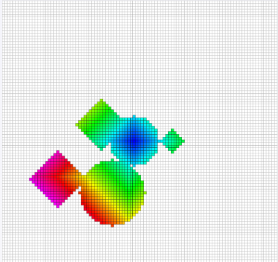
Main actual packages:

- 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.
  $\Rightarrow$ generic framework and tools to construct multigrid shapes in DGtal

3.1 Short presentation of the library (2)

Main actual packages:

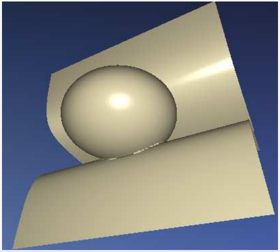
- Kernel package: number types, digital space, domain
- Arithmetic package: standard arithmetic computations
- Topology package: classic topology tools
- Geometry package: geometric estimators 2D/3D:
- DEC package: Discrete exterior calculus:
- Board & Viewer package: import/export image and visualization:



- 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.
  $\Rightarrow$ 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.

$\Rightarrow$ C++ with template programming

3.1 Short presentation of the library (4)

First example (see: `https://github.com/kerautret/ACCV2016DGPTutorial`)

- 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/ACCV2016DGPTutorial`)

- 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/ACCV2016DGPTutorial`)

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

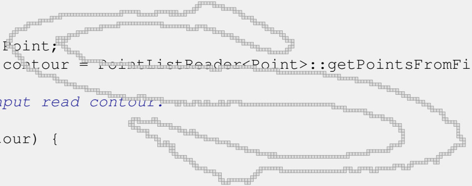
(see file: tuto1_baseDGtal.cpp)

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

- 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:

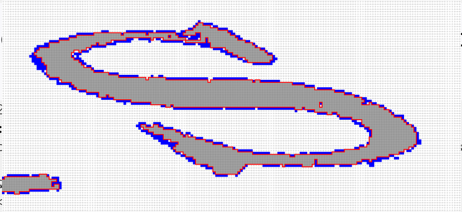
(see file: tuto2_LSC.cpp)

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

- Extract a set of pixel of the image:

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

- Track intergrid C ... bjects:



```
SurfelAdjacency<2
std::vector<std::
Surfaces<Z2i::KSp                                         Adj);
  ...
for (const auto &
    FreemanChain<
    ...
```

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].
$\Rightarrow$ 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)

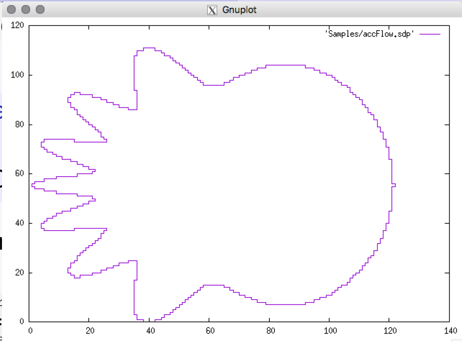
```
    typedef GridCurve<>::IncidentPointsRange Range;
    typedef Range::ConstIterator ClassicIterator;
    Range r = curve.getIncidentPointsRange();
    std::vector<double> estimations;
```

- 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 ge



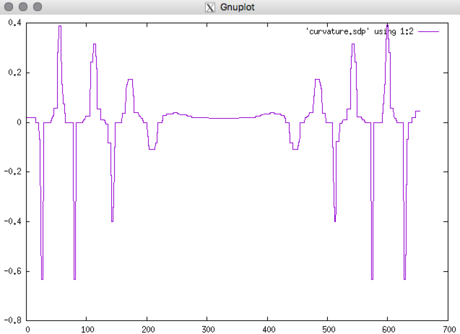
Third tutorial exercise

Computing curvature 11].
⇒ Based on Digital C

- Defines types fo

```
typedef GridCur
typedef Range::
Range r = curve
std::vector<dou
```

- Construct estima




```
SegmentComputer
SCEstimator sce
CurvatureEstima
...

estimator.init(
estimator.eval(                                  ons) );
```

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

[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)

[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