gipsa-lab

Grenoble | images | parole | signal | automatique | laboratoire

Digital geometry: digital objects analysis

Isabelle Sivignon, Yan Gérard

Gipsa-lab, Grenoble - LIMOS, Clermont-Ferrand

Digital sets/objects

Digital set

Set of points in $\mathbb{Z}^n$

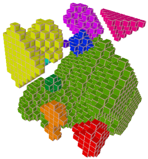
Digital sets/objects

Digital set

Set of points in $\mathbb{Z}^n$

Digital object

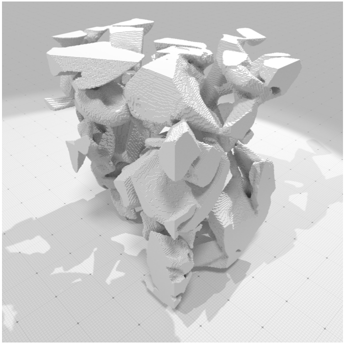
Set of points in $\mathbb{Z}^n$ + topology



What can we do from here ?

How to compute the *circularity*, *area* of these objects ?

How to compute the *curvature*, *local thickness* on this object ?



[Levallois 15, ANR digitalSnow, laboratoires LIRIS / LAMA / 3SR / MétéoFrance / CEN - CNRM GAME]

Plan

Outline

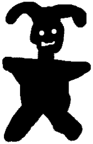
Plan

- Transformations
  - Distance Transform
  - Medial axis
  - Skeleton
- Measurements

Distance Transform

The problem

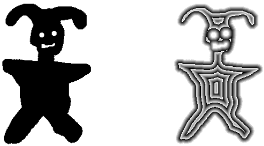
Given a digital set S, label each $p \in S$ with the distance to the closest point $q \notin S$.



Distance Transform

The problem

Given a digital set S, label each $p \in S$ with the distance to the closest point $q \notin S$.

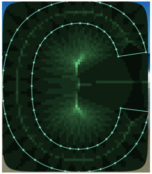


Applications

$\Rightarrow$ *signed distance field*

- Measures : thickness, differential operators on the boundary
- digital image processing : blurring effects, skeleton
- motion planning, pathfinding
- font smoothing, rendering

[http ://gamma.web.unc.edu/]



[Glyphy]

$\Rightarrow$ *distance between digital points ?*

Distance - first trial

Let p, q be two points in $\mathbb{Z}^n$.

What is the distance between p and q ?

Euclidean distance ?

$d_2(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$

$\Rightarrow$ not always an integer !... and we don't like roundings...

Distance - first trial

Let p, q be two points in $\mathbb{Z}^n$.

What is the distance between p and q ?

Euclidean distance ?

$d_2(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$

$\Rightarrow$ not always an integer !... and we don't like roundings...

d_1, d_∞ distances

$d_1(p,q) = \sum_{i=1}^{n} |p_i - q_i| \qquad d_\infty(p,q) = \max_{i=1..n} |p_i - q_i|$

$\Rightarrow$ integer values... but what are the balls like ?

Distance - first trial

Let p, q be two points in $\mathbb{Z}^n$.

What is the distance between p and q ?

Euclidean distance ?

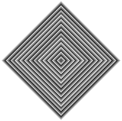
$d_2(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$

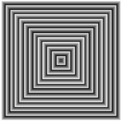
$\Rightarrow$ not always an integer !... and we don't like roundings...

d_1, d_∞ distances

$d_1(p,q) = \sum_{i=1}^{n} |p_i - q_i|$ $\quad$ $d_\infty(p,q) = \max_{i=1..n} |p_i - q_i|$

$\Rightarrow$ integer values... but what are the balls like ?





Chamfer distances

Idea [Montanari 68, Borgefors 84]

A finite set of displacements + a weight for each displacement.

Displacements

	a	
a	0	a
	a	

$a = 1 \Rightarrow d_1$

b	a	b
a	0	a
b	a	b

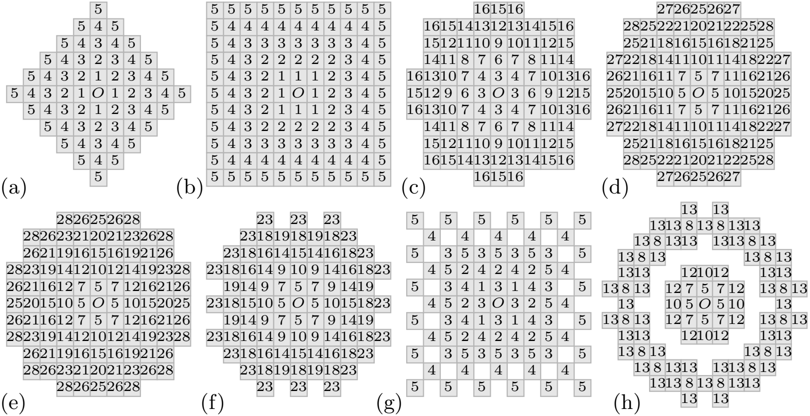
$a = b = 1 \Rightarrow d_\infty$

$2b$	c	$2a$	c	$2b$
c	b	a	b	c
$2a$	a	0	a	$2a$
c	b	a	b	c
$2b$	c	$2a$	c	$2b$

Distance

Distance between p and q = weight of the "lighter" path using only prescribed displacements.

How to set the weights ?



[Géométrie discrète et images numériques, ouvrage collectif, 2007]

Some conditions on the weights

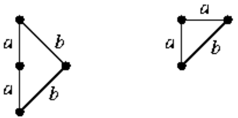
Axioms

1. $d(p,q) \geq 0, d(p,q) = 0 \Leftrightarrow p = q$ (positive, definite)
2. $d(p,q) = d(q,p)$ (symmetric)
3. $\forall r \in E,\ d(p,q) \leq d(p,r) + d(r,q)$ (triangular inequality)

Conditions

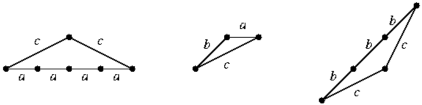
3×3

$0 < a \leq b \leq 2a$



5×5

$0 < 2a \leq c \leq a + b$
$3b \leq 2c$



Other integer distances

Other path-based distance

- neighbourhood sequences : weights and displacements vary at each step [Rosenfeld 68, Strand 07, Normand et al. 2013]

"Integer" Euclidean distance

- store the vector $\vec{pq}$
- squared Euclidean distance

Pros and cons

distance	exact	isotropic	storage	DT
path-based distances	✗	✗	✓	✓
vector	✓	✓	✗	✓
squared Euclidean	✓	✓	✗	✓

Distance transform with path-based distances

First idea

weighted graph representation + shortest path algorithm (Dikjstra for instance)

Distance transform with path-based distances

First idea

weighted graph representation + shortest path algorithm (Dikjstra for instance)

Second idea

Decompose the mask into several sub-masks + raster scan for each sub-mask $\Rightarrow$ propagate the min values

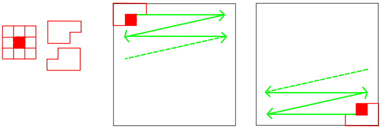
Distance transform with path-based distances

Second idea

Decompose the mask into several sub-masks + raster scan for each sub-mask $\Rightarrow$ propagate the min values

Example with 3×3 chamfer distance [Rosenfeld 66, Montanari 68]

$$DT(i,j) = \min_{(k,l) \in mask} (DT(i+k, j+l) + weight(k,l))$$



Initialisation :

$$DT(i,j) = 0 \quad si \quad (i,j) \notin S$$
$$DT(i,j) = +\infty \quad si \quad (i,j) \in S$$

Complexity : $\mathcal{O}(m.N^n)$

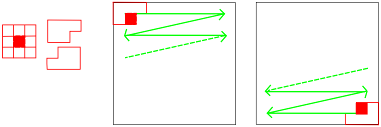
Distance transform with path-based distances

Second idea

Decompose the mask into several sub-masks + raster scan for each sub-mask $\Rightarrow$ propagate the min values

Example with 3×3 chamfer distance [Rosenfeld 66, Montanari 68]

$$DT(i,j) = \min_{(k,l) \in mask} \left(DT(i+k, j+l) + weight(k,l) \right)$$



Initialisation :

$$DT(i,j) = 0 \quad si \quad (i,j) \notin S$$
$$DT(i,j) = +\infty \quad si \quad (i,j) \in S$$

Complexity : $\mathcal{O}(m.N^n)$

Note : separable algorithm in $\mathcal{O}(\log^2 m.N^2)$ in 2D [Coeurjolly 2014]

Distance transform with Squared Euclidean distance

The problem

Let $p = (i, j) \in S \subset \mathbb{Z}^2$. We have :

$$DT(p) = \min_{q \notin S} \{d_2^2(p, q)\}$$

$$\Leftrightarrow$$

$$DT(i, j) = \min_{q(k,l) \notin S} \{(k - i)^2 + (l - j)^2\}$$

Distance transform with Squared Euclidean distance

The problem

Let $p = (i, j) \in S \subset \mathbb{Z}^2$. We have :

$$DT(p) = \min_{q \notin S} \{d_2^2(p, q)\}$$

$$\Leftrightarrow$$

$$DT(i, j) = \min_{q(k,l) \notin S} \{(k - i)^2 + (l - j)^2\}$$

Let's decompose [Saito et al. 94]

We can rewrite :

$$opt_x(i, {\color{red}j}) = \min_{q(k,{\color{red}j}) \notin S} \{(k - i)^2\}$$

and then

$$DT(i, j) = \min_{q(k,l) \notin S} \{(l - j)^2 + opt_x(i, l)\}$$

Distance transform with Squared Euclidean distance

The problem

Let $p = (i, j) \in S \subset \mathbb{Z}^2$. We have :

$$DT(p) = \min_{q \notin S}\{d_2^2(p, q)\}$$

$$\Leftrightarrow$$

$$DT(i, j) = \min_{q(k,l) \notin S}\{(k - i)^2 + (l - j)^2\}$$

Let's decompose [Saito et al. 94]

We can rewrite :

$$opt_x(i, j) = \min_{q(k,j) \notin S}\{(k - i)^2\}$$

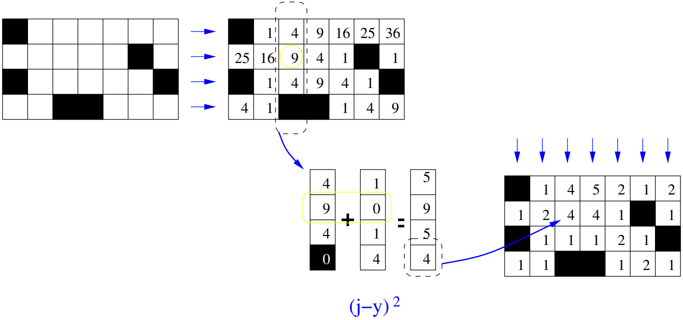
and then

$$DT(i, j) = \min_{q(k,l) \notin S}\{(l - j)^2 + opt_x(i, l)\}$$

$\Rightarrow$ *paradigm = separable algorithm*

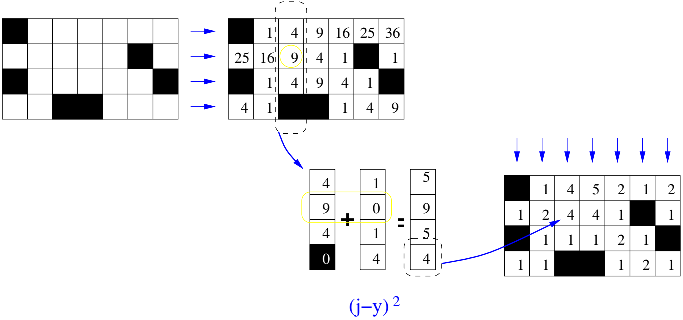
Distance transform with d_2^2 - example

Digital set S in white.



Distance transform with d_2^2 - example

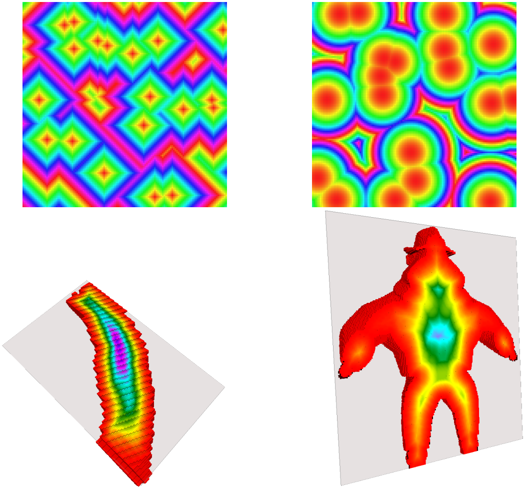
Digital set S in white.



Complexity : $O(N^2)$ for a 2D $N \times N$ domain $\Rightarrow O(N^d)$ for a nD N^n domain.

Note : works also for any distance deriving from a L_p-norm (in particular d_1, d_∞).

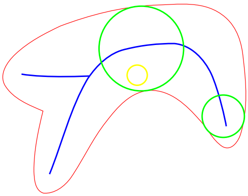
Images...



[Coeurjolly et al. 07] [DGtal Library]

Plan

Several definitions



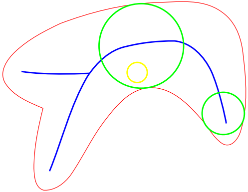
Seminal definition [Blum 67]

Meeting points of a grassfire initialized on the shape boundary

Modern definitions

- set of centers of maximal balls
- set of points having at least two closest points on the boundary

Several definitions



Seminal definition [Blum 67]

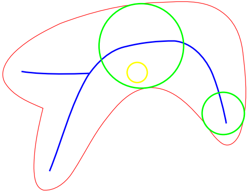
Meeting points of a grassfire initialized on the shape boundary

Modern definitions

- set of centers of maximal balls
- set of points having at least two closest points on the boundary

$\Rightarrow$ For a *shape* $F \subset \mathbb{R}^2$, these definitions lead to a *1-dimensionnal topological centered equivalent of* F

Several definitions



Seminal definition [Blum 67]

Meeting points of a grassfire initialized on the shape boundary

Modern definitions

- set of centers of maximal balls
- set of points having at least two closest points on the boundary

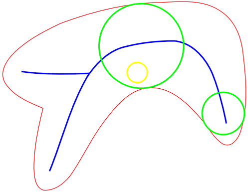
$\Rightarrow$ For a *shape* $F \subset \mathbb{R}^2$, these definitions lead to a *1-dimensionnal topological centered equivalent of F*

For a digital set/object $S \subset \mathbb{Z}^2$

Medial Axis : set of centers of maximal balls inside S

Skeleton : digital object, topologically equivalent to S and *minimal*

Several definitions



Seminal definition [Blum 67]

Meeting points of a grassfire initialized on the shape boundary

Modern definitions

- set of centers of maximal balls
- set of points having at least two closest points on the boundary

⇒ For a *shape* $F \subset \mathbb{R}^2$, these definitions lead to a *1-dimensionnal topological centered equivalent of F*

For a digital set/object $S \subset \mathbb{Z}^2$

Medial Axis : set of centers of maximal balls inside S

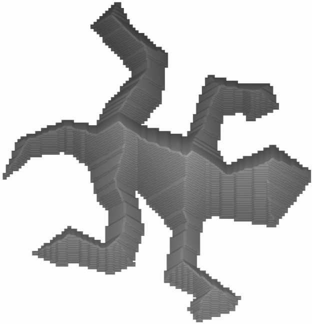
Skeleton : digital object, topologically equivalent to S and *minimal*

Medial axis and Distance Transform

Link

For a digital set S, and $p \in S$, $DT(p) =$ radius of the largest ball centered on p and included in S.

$\Rightarrow$ Medial Axis of S = local maxima of the DT



[K. Palágyi, http ://www.inf.u-szeged.hu/ palagyi/skel/skel.html]

Algorithm depends on the distance...

Path-based distance

Idea : to check whether a point $p \in S$ belongs to the Medial Axis, it is enough to compare with *a few neighbours*
$\Rightarrow$ Use a hash table

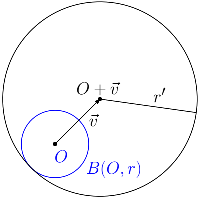
Algorithm depends on the distance...

Path-based distance

Idea : to check whether a point $p \in S$ belongs to the Medial Axis, it is enough to compare with *a few neighbours*

$\Rightarrow$ Use a hash table

Example with the Chamfer distance



$$LUT(\vec{v}, r) = \min \{r' | B(O, r) \subseteq B(O + \vec{v}, r')\}$$

$$p \in MA \iff DT(p+\vec{v}) < LUT(\vec{v}, DT(p)) , \quad \forall \vec{v} \in \mathcal{V}.$$

Neighbourhood $\mathcal{V}$? : for d_∞, $\mathcal{V} = \{(\pm 1, \pm 1), (0, \pm 1), (\pm 1, 0)\}$.

Algorithm depends on the distance...

Squared Euclidean distance

Cannot pre-define a (limited) set of neighbours and radii
$\Rightarrow$ New tool : power diagram

Algorithm depends on the distance...

Squared Euclidean distance

Cannot pre-define a (limited) set of neighbours and radii
$\Rightarrow$ New tool : power diagram

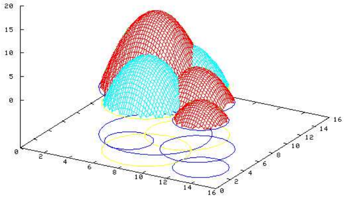
Rewriting S

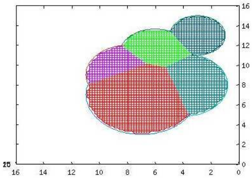
$$
\begin{aligned}
\textit{Lapalissade}: \quad S &= \bigcup_{p \in S} B(p, DT(p)) \\
&= \{(i,j) \mid \exists p,\ (i-x_p)^2 + (j-y_p)^2 < DT(p)\} \\
&= \{(i,j) \mid \exists p,\ \underbrace{DT(p) - (i-x_p)^2 - (j-y_p)^2 > 0}_{\text{paraboloid}}\}
\end{aligned}
$$

Algorithm depends on the distance...

Rewriting S

$$\textit{Lapalissade}: \; S = \{(i,j) \mid \exists p, \underbrace{DT(p) - (i - x_p)^2 - (j - y_p)^2 > 0}_{\text{paraboloid}}\}$$





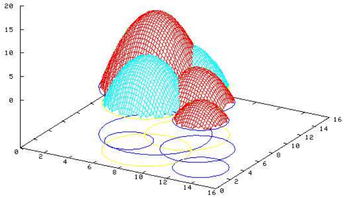
[Coeurjolly et al. 07]

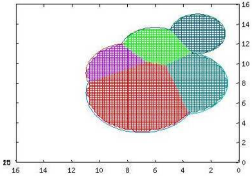
$\Rightarrow$ *Highest paraboloïds* are enough to define S : others correspond to non-maximal balls

Algorithm depends on the distance...

Rewriting S

$$\textit{Lapalissade}: \; S = \{(i,j) \mid \exists p, \underbrace{DT(p) - (i - x_p)^2 - (j - y_p)^2 > 0}_{\text{paraboloid}}\}$$





[Coeurjolly et al. 07]

⇒ *Highest paraboloïds* are enough to define S : others correspond to non-maximal balls

⇒ algorithmic tool from computational geometry = *Power Diagram*

Medial Axis and Digital Set reconstruction

Property

S is exactly defined by the set of centers of the medial axis + radii

Medial Axis and Digital Set reconstruction

Property

S is exactly defined by the set of centers of the medial axis + radii

Question

Is this set of minimum cardinality ?

Medial Axis and Digital Set reconstruction

Property

S is exactly defined by the set of centers of the medial axis + radii

Question

Is this set of minimum cardinality ?

Answer

No ! Maximal balls property only ensures that no ball is included in another, but a ball can be included in the *union* of others.
Bad news : the problem is NP-complete [Coeurjolly et al. 08]

Medial Axis and Digital Set reconstruction

Property

S is exactly defined by the set of centers of the medial axis + radii

Question

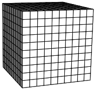
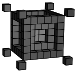
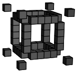
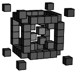
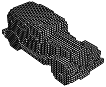
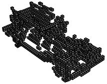
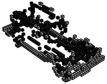
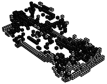
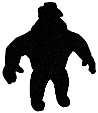
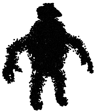
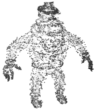
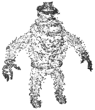
Is this set of minimum cardinality ?

Answer

No ! Maximal balls property only ensures that no ball is included in another, but a ball can be included in the *union* of others.
Bad news : the problem is NP-complete [Coeurjolly et al. 08]

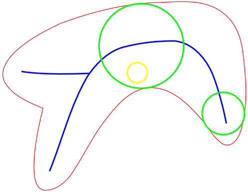
$\Rightarrow$ simplification algorithms [Ragnemalm et al. 91, Coeurjolly et al. 08]

Medial Axis and Digital Set reconstruction

Objet	$\mathcal{F} = AM(\mathcal{S})$	$\hat{\mathcal{F}}$ Ragnemalm et al.	$\hat{\mathcal{F}}$ Greedy
	 104	 56 (-46%) [<0.01s]	 66 (-36%) [< 0.01s]
	 1292	 795 (-38%) [0.1s]	 820 (-36%) [0.19s]
	 17238	 6177 (-64%) [48.53s]	 6553 (-62%) [57.79s]

Plan

Several definitions



Seminal definition [Blum 67]

Meeting points of a grassfire initialized on the shape boundary

Modern definitions

- set of centers of maximal balls
- set of points having at least two closest points on the boundary

$\Rightarrow$ For a *shape* $F \subset \mathbb{R}^2$, these definitions lead to a *1-dimensionnal topological centered equivalent of* F

For a digital set/object $S \subset \mathbb{Z}^2$

Medial Axis : set of centers of maximal balls inside S

Skeleton : digital object, topologically equivalent to S and *minimal*

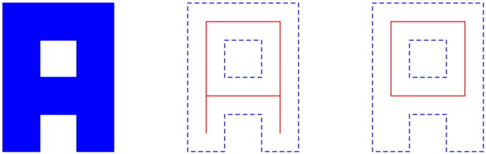
Skeleton

Let O be *a digital object* = digital set + topology (adjacency relations).

Principle

Withdraw some points of O (one by one) without "modifying the topology" :

- homotopy equivalence
- or homeomorphism ?



[Courtesy of C. Ronse]

Homotopy equivalence - Simple points

Simple point

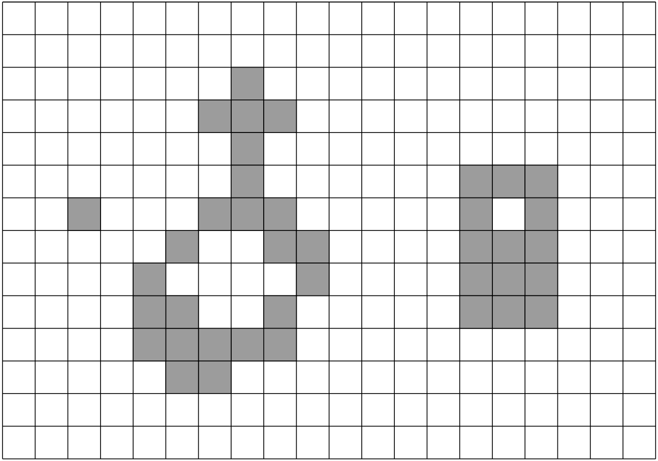
A point $p \in O$ is *simple* iff $O - \{p\}$ is homotopy equivalent to O.

In practice - Constraints

- a connected component of O cannot be deleted
- a connected component of O cannot be disconnected
- no connected component of O^c can be created
- no connected components of O^c can be merged

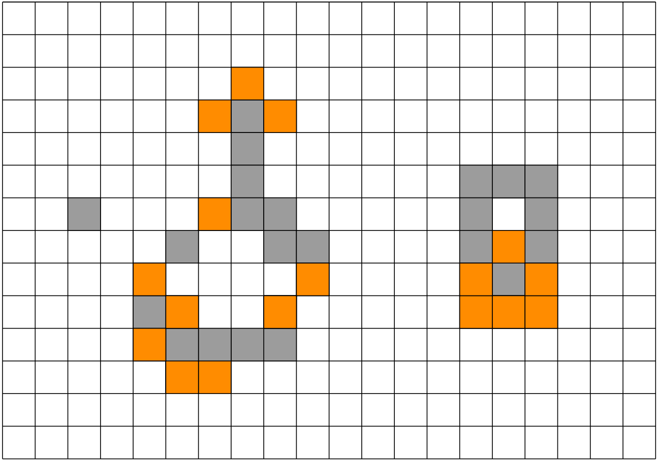
[Rosenfeld 70, Ronse 86, ...]

Example of simple points



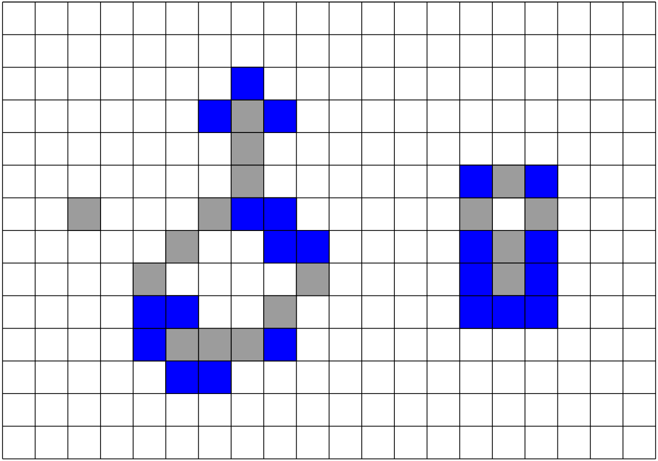
Example of simple points

Digital Object O is 4-connected.



Example of simple points

Digital Object O is 8-connected.

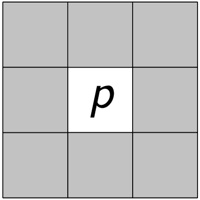


Characterization of simple points

From the definition, the characterization is *global*
$\Rightarrow$ at least $\mathcal{O}(n)$ to test *one* point in a domain of n points.

(Very) Local criterion

Let's count the connected components in the neighbourhood of p.



Neighbourhood of p

$\Rightarrow$ characterization in $\mathcal{O}(1)$!

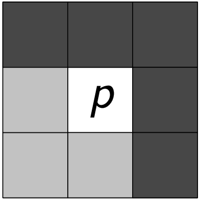
[2D : Rosenfeld 79] [3D : Morgenthaler 81, Bertrand 94, Saha et al. 94,...] [4D : Kong 97, Couprie & Bertrand 09]

Characterization of simple points

From the definition, the characterization is *global*
$\Rightarrow$ at least $\mathcal{O}(n)$ to test *one* point in a domain of n points.

(Very) Local criterion

Let's count the connected components in the neighbourhood of p.



1 connected component for O
1 for O^c
$\Rightarrow$ p is simple

$\Rightarrow$ characterization in $\mathcal{O}(1)$!

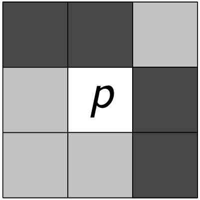
[2D : Rosenfeld 79] [3D : Morgenthaler 81, Bertrand 94, Saha et al. 94,...] [4D : Kong 97, Couprie & Bertrand 09]

Characterization of simple points

From the definition, the characterization is *global*
$\Rightarrow$ at least $\mathcal{O}(n)$ to test *one* point in a domain of n points.

(Very) Local criterion

Let's count the connected components in the neighbourhood of p.



O is 4-connected

2 connected component of O 4-connected to p
2 connected component of O^c 8-connected to p
$\Rightarrow$ p is not simple

$\Rightarrow$ characterization in $\mathcal{O}(1)$!

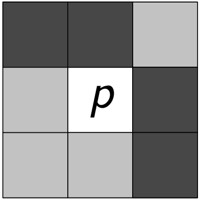
[2D : Rosenfeld 79] [3D : Morgenthaler 81, Bertrand 94, Saha et al. 94,...] [4D : Kong 97, Couprie & Bertrand 09]

Characterization of simple points

From the definition, the characterization is *global*
$\Rightarrow$ at least $\mathcal{O}(n)$ to test *one* point in a domain of n points.

(Very) Local criterion

Let's count the connected components in the neighbourhood of p.



O is 8-connected

1 connected component of O 8-connected to p
1 connected component of O^c 4-connected to p
$\Rightarrow$ p is simple

$\Rightarrow$ characterization in $\mathcal{O}(1)$!

[2D : Rosenfeld 79] [3D : Morgenthaler 81, Bertrand 94, Saha et al. 94,...] [4D : Kong 97, Couprie & Bertrand 09]

Thinning algorithm

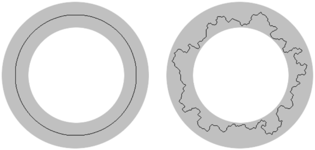
Simple sequential removal

While there exist p simple in O

- $O \leftarrow O \backslash p$

Order matters

How to choose p ?



[Chaussard 10]

Breadth-first thinning algorithm

Data: Digital Object O
Result: Digital Object $Sk(O)$
$Sk(O) \leftarrow O$
$Queue$: $SP \leftarrow \{p \in O \mid p \text{ is simple for } O\}$

Breadth-first thinning algorithm

Data: Digital Object O
Result: Digital Object $Sk(O)$
$Sk(O) \leftarrow O$
$Queue$: $SP \leftarrow \{p \in O \mid p \text{ is simple for } O\}$
while $SP \neq \emptyset$ **do**
    $Next \leftarrow \emptyset$
    forall the $p \in SP$ **do**

Breadth-first thinning algorithm

Data: Digital Object O
Result: Digital Object $Sk(O)$
$Sk(O) \leftarrow O$
$Queue$: $SP \leftarrow \{p \in O \mid p \text{ is simple for } O\}$
while $SP \neq \emptyset$ **do**
  $Next \leftarrow \emptyset$
  forall the $p \in SP$ **do**
    if p *is simple for* $Sk(O)$ **then**
      $Sk(O) \leftarrow Sk(O) \setminus \{p\}$

Breadth-first thinning algorithm

Data: Digital Object O
Result: Digital Object $Sk(O)$
$Sk(O) \leftarrow O$
$Queue$: $SP \leftarrow \{p \in O \mid p \text{ is simple for } O\}$
while $SP \neq \emptyset$ **do**
  $Next \leftarrow \emptyset$
  forall the $p \in SP$ **do**
    if p *is simple for* $Sk(O)$ **then**
      $Sk(O) \leftarrow Sk(O) \setminus \{p\}$
    forall the $q \in Sk(O)$, q *neighbour of* p **do**
      $Next \leftarrow Next \cup \{q\}$
    end
  end

Breadth-first thinning algorithm

Data: Digital Object O
Result: Digital Object $Sk(O)$

$Sk(O) \leftarrow O$
$Queue : SP \leftarrow \{p \in O \mid p \text{ is simple for } O\}$
while $SP \neq \emptyset$ **do**
    $Next \leftarrow \emptyset$
    forall the $p \in SP$ **do**
        if *p is simple for $Sk(O)$* **then**
            $Sk(O) \leftarrow Sk(O) \setminus \{p\}$
        forall the $q \in Sk(O)$, *q neighbour of p* **do**
            $Next \leftarrow Next \cup \{q\}$
        end
    end
    $SP \leftarrow \emptyset$
    forall the $p \in Next$ **do**
        if *p is simple for $Sk(O)$* **then**
            $SP \leftarrow SP \cup \{p\}$
    end
end

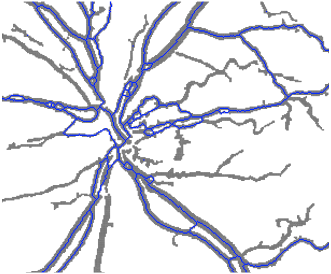
Breadth-first thinning algorithm

Data: Digital Object O
Result: Digital Object $Sk(O)$
$Sk(O) \leftarrow O$
$Queue$: $SP \leftarrow \{p \in O \mid p \text{ is simple for } O\}$
while $SP \neq \emptyset$ **do**
    $Next \leftarrow \emptyset$
    forall the $p \in SP$ **do**
        if p *is simple for* $Sk(O)$ **then**
            $Sk(O) \leftarrow Sk(O) \setminus \{p\}$
        forall the $q \in Sk(O)$, q *neighbour of* p **do**
            $Next \leftarrow Next \cup \{q\}$
        end
    end
    $SP \leftarrow \emptyset$
    forall the $p \in Next$ **do**
        if p *is simple for* $Sk(O)$ **then**
            $SP \leftarrow SP \cup \{p\}$
    end
end

Note : A priority function can also be used (for instance the Distance Transform).

Preserving the shape

Homotopy equivalence $\Rightarrow$ any digital object O with a single connected component reduces to a single point !



Preserving the shape

Homotopy equivalence $\Rightarrow$ any digital object O with a single connected component reduces to a single point !

Anchor points

Predicate defining un-removable points, for instance : points with one neighbour, points of the medial axis

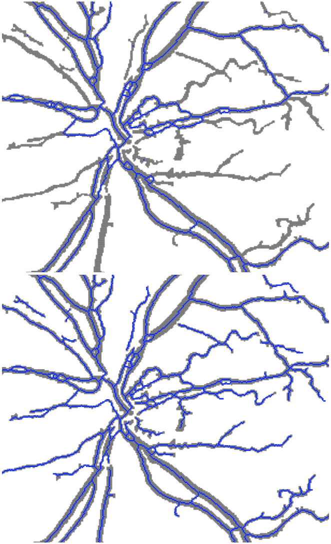
...

if *p is simple for* $Sk(O)$ *and* *not* $Anchor(p)$
then

$$Sk(O) \leftarrow Sk(O) \setminus \{p\}$$

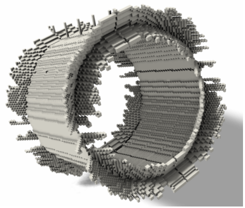
...

+ huge literature on the subject (see PhD thesis of [Chaussard 10] for instance)



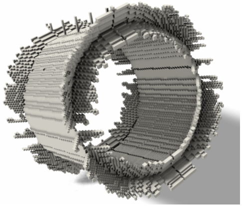
Cellular grid space framework

In higher dimension, the skeleton does not always have nice properties (remaining 3d parts for instance).

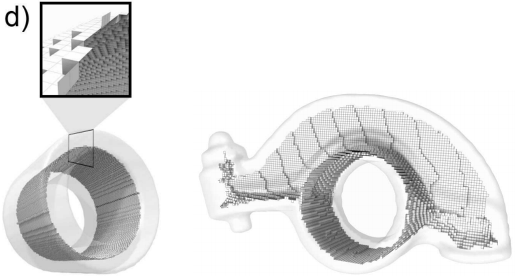


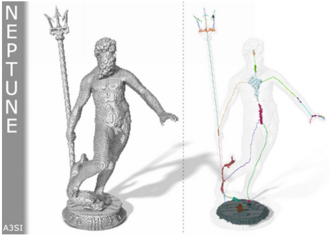
Cellular grid space framework

In higher dimension, the skeleton does not always have nice properties (remaining 3d parts for instance).



⇒ Use the cellular grid space : need to define simple cells, etc.





[Image from Chaussard 10] [see also Mazo 11]

Outline

Transformations

Measurements

- Multigrid convergence
- Area estimation
- Tangent, normal, length estimation
- Curvature and higher derivatives estimation

Goal : compute geometric quantities

Consider a family $\mathcal{F}$ of shapes in $\mathbb{R}^n$ (fulfilling given properties).

Global geometric quantities

For a given shape $S \in \mathcal{F}$, compute :

- its area (volume)
- its perimeter (area of its boundary)

Local geometric quantities

For any point $x \in \partial S$, $S \in \mathcal{F}$, compute :

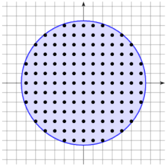
- its tangent, normal vector
- its curvature

Plan

Is the computation accurate, truthful ?

Consider :

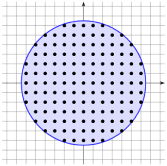
- a grid $\mathbb{G}$ of resolution $1/h$, $h > 0$ (size of the pixels).
- a digitization process Dig_h such that $\mathrm{Dig}_h(S) = S \cap (h\mathbb{Z}^2)$ is the digitized version of $S \in \mathcal{F}$



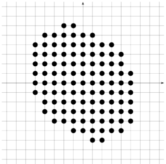
Is the computation accurate, truthful ?

Consider :

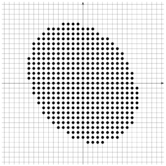
- a grid $\mathbb{G}$ of resolution $1/h$, $h > 0$ (size of the pixels).
- a digitization process Dig_h such that $\mathrm{Dig}_h(S) = S \cap (h\mathbb{Z}^2)$ is the digitized version of $S \in \mathcal{F}$



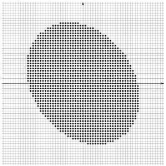
What happens when $h \to 0$?



$\mathrm{Dig}_h(S)$



$\mathrm{Dig}_{h/2}(S)$

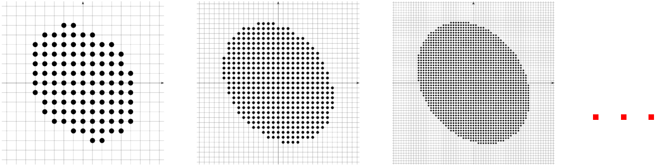


$\mathrm{Dig}_{h/4}(S)$

. . .

. . .

Global geometric estimators - multigrid convergence



$\mathrm{Dig}_h(S)$ $\mathrm{Dig}_{h/2}(S)$ $\mathrm{Dig}_{h/4}(S)$ ...

Multigrid convergence [Serra82]

Let $\mathcal{F}$ be a family of shapes. The geometric estimator $\hat{\epsilon}$ is said to be multigrid convergent for $\mathcal{F}$ toward the geometric descriptor ϵ iff $\forall S \in \mathcal{F}$

$$\lim_{h \to 0} |\hat{\epsilon}(\mathrm{Dig}_h(S)) - \epsilon(S)| \leq \tau_S(h)$$

$$\lim_{h \to 0} \tau_S(h) = 0$$

The *speed of convergence* is given by $\tau_S(h)$.

Local geometric estimators - multigrid convergence

Local multigrid convergence

Let $\mathcal{F}$ be a family of shapes. The geometric estimator $\hat{\epsilon}$ is said to be multigrid convergent for $\mathcal{F}$ toward the geometric descriptor ϵ iff $\forall S \in F, \forall x \in \partial S$,

$$\forall y \in \partial \mathrm{Dig}_h(S) \text{ with } \|y - x\|_1 \leq h,$$

$$|\hat{\epsilon}(\mathrm{Dig}_h(S), y) - \epsilon(S, x)| \leq \tau_{S,x}(h)$$

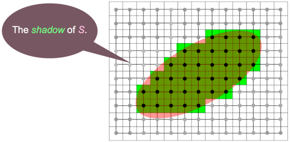
$$\lim_{h \to 0} \tau_{S,x}(h) = 0$$

The *speed of convergence* is given by $\tau_{S,x}(h)$.

[See Coeurjolly et al. 12 for a survey on digital estimators]

Plan

First example : area estimation



Consider the *shadow* digitization scheme Dig_h.

For a shape $S \in \mathcal{F}$, let's count the number of points in $\mathrm{Dig}_h(S)$, and define $\hat{\epsilon}(\mathrm{Dig}_h(S)) = h^2.|S \cap (h\mathbb{Z}^2)|$.

- For the family of convex shapes $\mathcal{F}$, $\tau_X(h) = O(h)$ [Gauss,Dirichlet]
- For the family of $\mathcal{C}^3$-convex shapes $\mathcal{F}$, $\tau_X(h) = O(h^{\frac{15}{11}+\epsilon})$ [Huxley 90]

Speed of convergence

h	1	0.1	0.01	0.001	...
$h^{\frac{15}{11}}$	1	0.04328	0.00187	0.00008	...

Plan

Transformations

Measurements

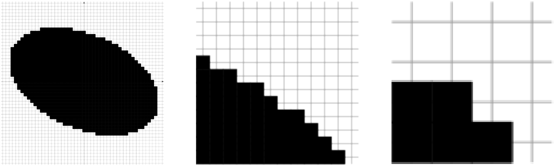
- Multigrid convergence
- Area estimation
- Tangent, normal, length estimation
- Curvature and higher derivatives estimation

Length estimation - first trial

Let C be a digital curve, defined by a sequence of elementary displacements.

Question

Can we define a *convergent length estimator* of C by counting the number of elementary displacements ?



Remark

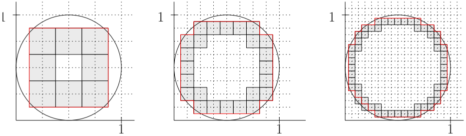
Even if the Hausdorff distance between the digital boundary and the shape boundary ∂S tends towards 0 when $h \to 0$, "stairs effect" remains.

Local weights - Stair effect

Let's try for different digitization schemes.

"Shadow" of a shape

Let S be a disk of radius $\frac{1}{2}$, and Dig_h be the "shadow".
Let C be $\partial\mathrm{Dig}_h(S)$.



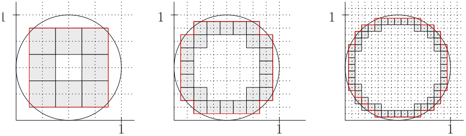
Length of C tends to 4 instead of π !

Local weights - Stair effect

Let's try for different digitization schemes.

"Shadow" of a shape

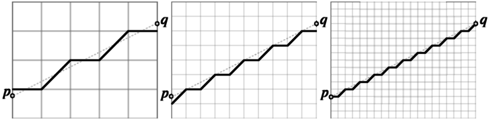
Let S be a disk of radius $\frac{1}{2}$, and Dig_h be the "shadow".
Let C be $\partial\mathrm{Dig}_h(S)$.



Length of C tends to 4 instead of π !

Digitization of a segment

Let S be a segment of slope $\alpha \in [0,1]$ and length l.
$\mathrm{Dig}_h(S)$ is the digitization that "chooses the closest point".



Length of S is equal to $\frac{5(1+\sqrt{(2)})}{2}$ $\forall n \geq 1$ instead of $\frac{5\sqrt{5}}{2}$!

[Klette, 2004]

Doomed local weights

Local weights estimator

- Consider the decomposition of C into parts of m elementary displacements such that $C = w_1 w_2 \dots w_n \lambda$
- Give a weight $p(.)$ to each w_i
- Define the estimator $\hat{\epsilon}(C) = \sum_{i=1}^{n} p(w_i)$

Doomed local weights

Local weights estimator

- Consider the decomposition of C into parts of m elementary displacements such that $C = w_1 w_2 \dots w_n \lambda$
- Give a weight $p(.)$ to each w_i
- Define the estimator $\hat{\epsilon}(C) = \sum_{i=1}^{n} p(w_i)$

Result when $C = \mathrm{Dig}_h(S)$, S a straight segment

For all m and all $p(.)$, the set of segments for which the estimator converges to the length of S is countable. Meaning that most of the time, the estimator does not converge. [Tajine, Daurat 03].

Doomed local weights

Local weights estimator

- Consider the decomposition of C into parts of m elementary displacements such that $C = w_1 w_2 \ldots w_n \lambda$
- Give a weight $p(.)$ to each w_i
- Define the estimator $\hat{\epsilon}(C) = \sum_{i=1}^{n} p(w_i)$

Result when $C = \mathrm{Dig}_h(S)$, S a straight segment

For all m and all $p(.)$, the set of segments for which the estimator converges to the length of S is countable. Meaning that most of the time, the estimator does not converge. [Tajine, Daurat 03].

Another solution ?

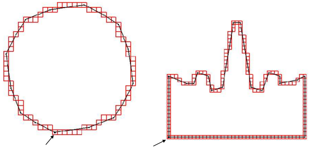
Decompose C into parts such that the "length" of the parts adapts itself to the curve. $\Rightarrow$ use digital segments

[see also semi-local estimators in Daurat et al. 11, Mazo & Baudrier 14]

Length estimation through polygonalization

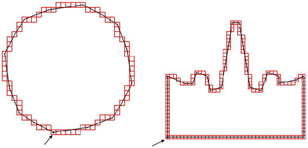
Principle

Compute a polygon from the digital curve C using digital straight segments.
$\Rightarrow$ length of C = length of the polygon.



Length estimation through polygonalization

Principle



Compute a polygon from the digital curve C using digital straight segments.
$\Rightarrow$ length of C = length of the polygon.

Several approaches

- greedy as long as possible DSS [Kovalesvsky et al. 92]
- Miminum Length Polygon : shortest curve that separates inside from outside [Sloboda 98, Provençal & Lachaud 09]
- Faithful Polygon : polygon that preserves convex and concave parts [Roussillon & Sivignon 11]

Multigrid convergence (theoretical and/or experimental) in $\mathcal{O}(h)$.

Tangent estimation

Let $C = p_i$ be a digital curve in $\mathbb{Z}^2$.
Goal : estimate the first derivative for all $p \in C$.

Tangent estimation

Let $C = p_i$ be a digital curve in $\mathbb{Z}^2$.
Goal : estimate the first derivative for all $p \in C$.

Maximal Digital Segment

A digital segment S_{ij} is *maximal* on C if there is no segment $S \subset C$ such that $S_{ij} \subset S$.

Tangent estimation

Let $C = p_i$ be a digital curve in $\mathbb{Z}^2$.
Goal : estimate the first derivative for all $p \in C$.

Maximal Digital Segment

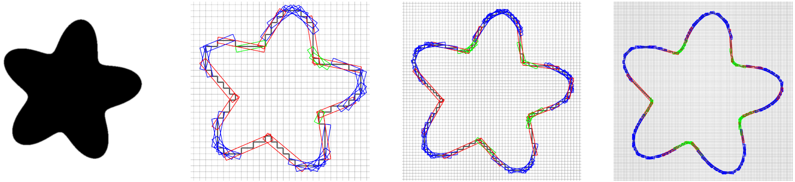
A digital segment S_{ij} is *maximal* on C if there is no segment $S \subset C$ such that $S_{ij} \subset S$.

Algorithmically

- Maximality can be tested locally, checking $S_{i-1,j}$ and $S_{i,j+1}$
- algorithms to add or remove a point at the front or at the back of a DSS in $\mathcal{O}(1)$ [Debled & Reveilles 95, Lachaud et al. 07]

$\Rightarrow$ algorithm in $\mathcal{O}(|C|)$ to compute all the maximal DSS on C.

Maximal Digital Segments



S $\mathrm{Dig}_h(S)$ $\mathrm{Dig}_{\frac{h}{2}}(S)$ $\mathrm{Dig}_{\frac{h}{4}}(S)$

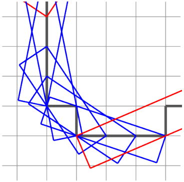
Swiss knife

- convergent tangent estimator
- convergent length estimator
- convex, concave parts, extremal points

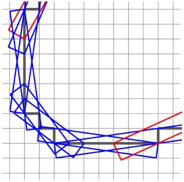
[Courtesy of T. Roussillon, DGtal]

Maximal Digital Segments

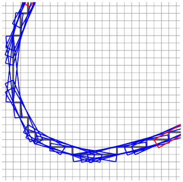
S



$\mathrm{Dig}_h(S)$



$\mathrm{Dig}_{\frac{h}{2}}(S)$



$\mathrm{Dig}_{\frac{h}{4}}(S)$

Swiss knife

- convergent tangent estimator
- convergent length estimator
- convex, concave parts, extremal points

[Courtesy of T. Roussillon, DGtal]

Maximal Digital Segments

Theorem [Lachaud et al. 07]

Let $x \in \partial S$. The direction of any maximal digital segment of $\mathrm{Dig}_h(S)$ that covers x converges to the tangent at x when $h \to 0$.

Maximal Digital Segments

Theorem [Lachaud et al. 07]

Let $x \in \partial S$. The direction of any maximal digital segment of $\mathrm{Dig}_h(S)$ that covers x converges to the tangent at x when $h \to 0$.

Convergent tangent estimator

- choose any maximal digital segment
- use a convex combination of all the maximal digital segments (λ-MST) [Lachaud et al. 07]

Maximal Digital Segments

Theorem [Lachaud et al. 07]

Let $x \in \partial S$. The direction of any maximal digital segment of $\mathrm{Dig}_h(S)$ that covers x converges to the tangent at x when $h \to 0$.

Convergent tangent estimator

- choose any maximal digital segment
- use a convex combination of all the maximal digital segments (λ-MST) [Lachaud et al. 07]

Convergent length estimator

- "Digitize" $\int_0^1 t(s)ds$: $\widehat{Length}(\mathrm{Dig}_h(S)) = \sum_{e \in \partial \mathrm{Dig}_h(S)} \hat{t}(e).t_{elem}(e)$.
- Convergence speed : $\mathcal{O}(h^{\frac{1}{3}})$ ($\mathcal{O}(h^{\frac{4}{3}})$ experimentally)

[Coeurjolly & Klette 04, Lachaud 06]

Plan

Transformations

Measurements

- Multigrid convergence
- Area estimation
- Tangent, normal, length estimation
- Curvature and higher derivatives estimation

Curvature

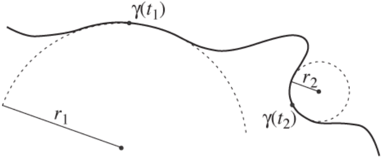
Let $\gamma(s)$ be a (at least $\mathcal{C}^2$) curve. The curvature κ along γ is given by :

Definitions

(i) norm of the second derivative $\kappa(s) = |\frac{d^2\gamma}{ds^2}|$

(ii) derivative of the tangent orientation $\kappa(s) = \frac{d\phi}{ds}$

(iii) inverse of the osculating circle radius $\kappa(s) = \frac{1}{r(s)}$



Families of digital curvature estimators (1/2)

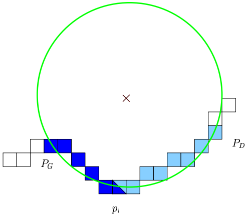
Many algorithms that *mimic* these definitions :

- in (i) (ii) convolutions (local weighted means) are used to mimic derivatives [Worring & Smeulders 93, Feschet & Tougne 99, ...]

Families of digital curvature estimators (1/2)

Many algorithms that *mimic* these definitions :

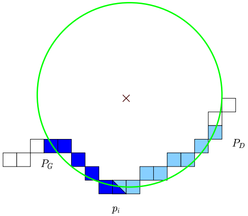
- in (i) (ii) convolutions (local weighted means) are used to mimic derivatives [Worring & Smeulders 93, Feschet & Tougne 99, ...]
- in (ii) (iii) digital segments or digital circular arcs are used to compute tangents or osculating circles [Coeurjolly et al. 01, Herman & Klette 07, Roussillon et al. 11,...]

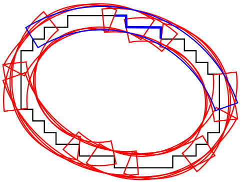


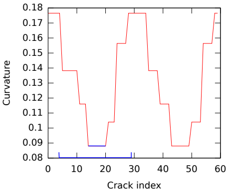
Families of digital curvature estimators (1/2)

Many algorithms that *mimic* these definitions :

- in (i) (ii) convolutions (local weighted means) are used to mimic derivatives [Worring & Smeulders 93, Feschet & Tougne 99, ...]
- in (ii) (iii) digital segments or digital circular arcs are used to compute tangents or osculating circles [Coeurjolly et al. 01, Herman & Klette 07, Roussillon et al. 11,...]







Families of digital curvature estimators (2/2)

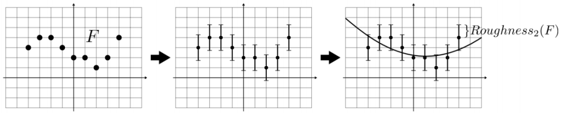
Difficulties : not user-parameter free and/or multigrid convergence difficult to obtain

Families of digital curvature estimators (2/2)

Difficulties : not user-parameter free and/or multigrid convergence difficult to obtain

Other estimators :

- fitting of higher order polynomial [Provot & Gérard 11]
  - multigrid convergent estimation of higher order derivatives in $\mathcal{O}(h^{\frac{1}{k+1}})$ for k-th derivative
  - not parameter-free

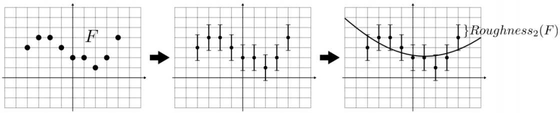


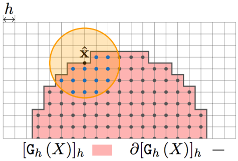
Families of digital curvature estimators (2/2)

Difficulties : not user-parameter free and/or multigrid convergence difficult to obtain

Other estimators :

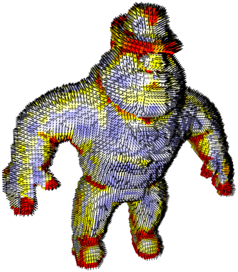
- fitting of higher order polynomial [Provot & Gérard 11]
  - multigrid convergent estimation of higher order derivatives in $\mathcal{O}(h^{\frac{1}{k+1}})$ for k-th derivative
  - not parameter-free
- integral invariant estimator [Pottman et al. 07, Coeurjolly et al. 14, Levallois 15]
  - multigrid convergent in $\mathcal{O}(h^{\frac{1}{3}})$
  - can be parameter-free

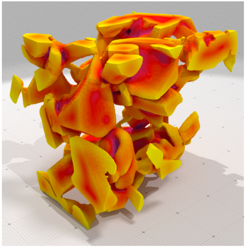




For 3D Digital Objects

- Normal vector estimation : multigrid convergent in $\mathcal{O}(h^{\frac{1}{8}})$ and stable algorithm [Cuel et al. 14, ...]
- Surface area : as for length estimation, integrate normals
- Mean and Gaussian curvature : integral invariant estimators are multigrid convergent in $\mathcal{O}(h^{\frac{1}{3}})$ [Coeurjolly et al. 14, Levallois 15]





The last words...

A short and incomplete overview

For instance, what about :

- more “complex” digital primitives : curves in 3D, circles, etc
- affine transformations : how to perform a rotation ?

The last words...

A short and incomplete overview

For instance, what about :

- more “complex” digital primitives : curves in 3D, circles, etc
- affine transformations : how to perform a rotation ?

Books to go further

- *Géométrie discrète et images numériques*, 2007. Collective book (in french).
- *Digital Geometry*, 2004. R. Klette & A. Rosenfeld (in english)

DGtal

Digital Geometry Tools and Algorithms

- aim at gathering digital geometry algorithms in a common programming framework
- open-source, collaborative library
- Let's try it during lab session this afternoon !

Thank you !