DigitalSnow Final meeting

Digital Level Layers for Curve Decomposition and Vectorization

july 9th 2015, Autrans

Yan Gerard (ISIT) *yan.gerard@udamail.fr*

Plan

Plan

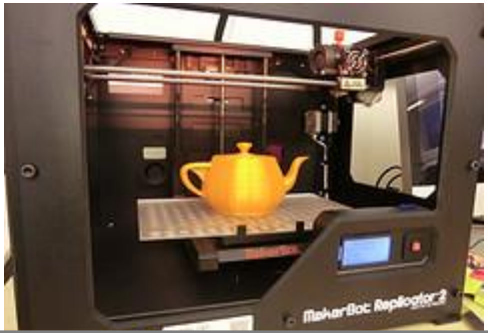
I Introduction

Geometric design
(3D artists, designers…)

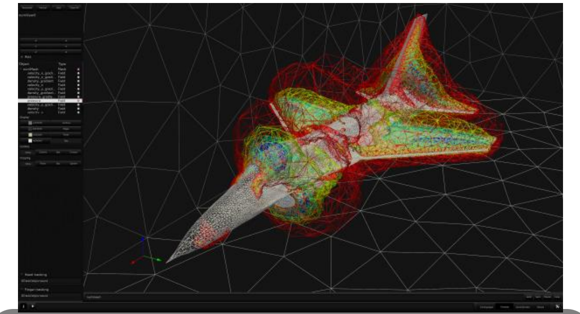
Martin Newell's
Utah Teapot

Real world data acquisition
(3D-scanners, computer vision, motion capture, medical imaging…)

3D models
(geometry)

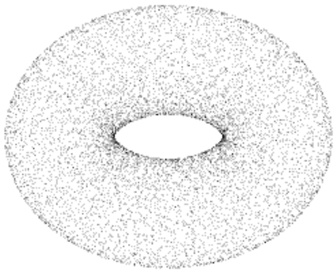


Building
(3D printer, factory…)



Computer simulation
(Finite Elements Methods…)

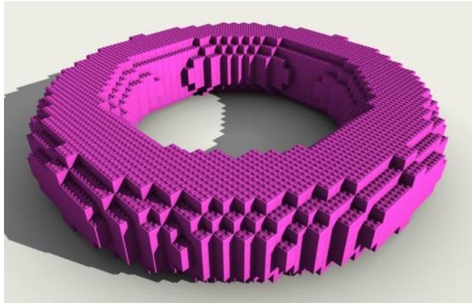
Images
(video games, movies, FX, Augmented Reality…)



Points cloud

3D models
(geometry)

Points cloud

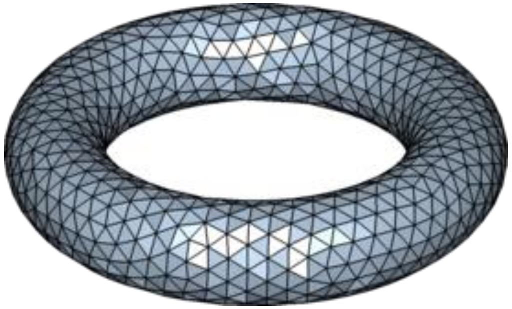


Sets of voxels

3D models
(geometry)

Points cloud

Sets of voxels



Mesh

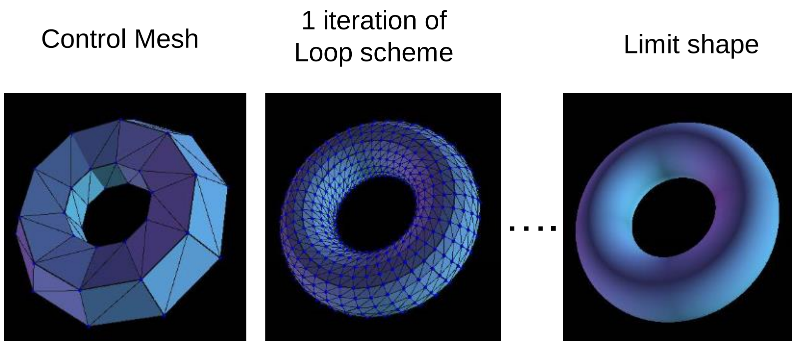
3D models
(geometry)

Points cloud

Sets of voxels

Mesh

3D models
(geometry)



Subdivision surfaces

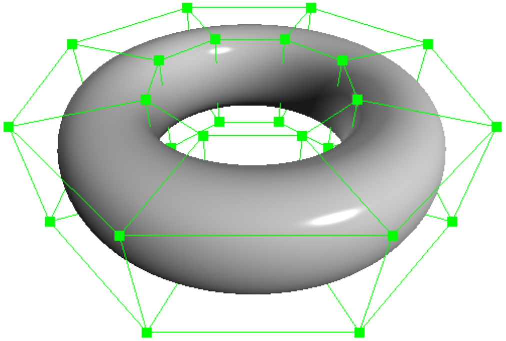
Many possible models

Points cloud

Sets of voxels

Mesh

3D models
(geometry)



Parametric shapes
(Bézier, B-splines, NURBS…)

Control Mesh

1 iteration of
Loop scheme

Limit shape

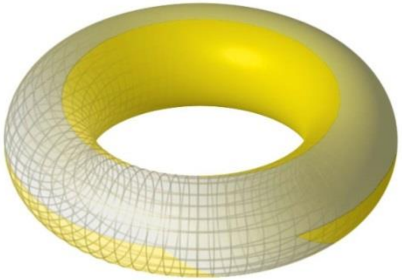
Subdivision surfaces

Points cloud

Sets of voxels

Mesh

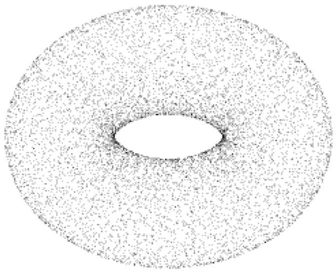
3D models
(geometry)



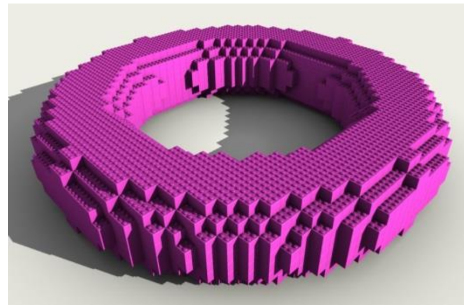
Level sets
(equation)

Parametric shapes
(Bézier, B-splines, NURBS…)

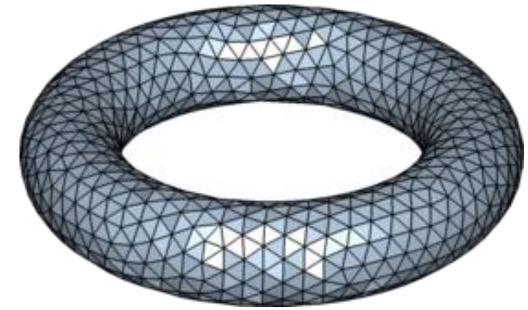
Subdivision surfaces



Points cloud

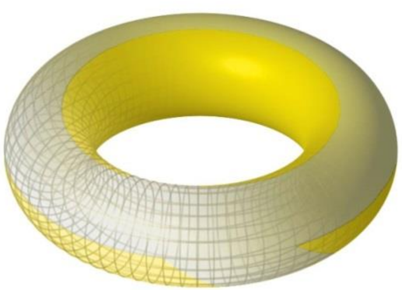


Sets of voxels

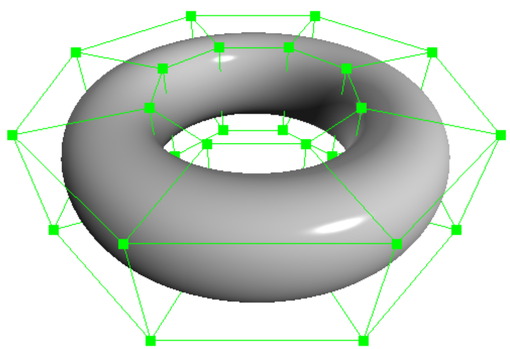


Mesh

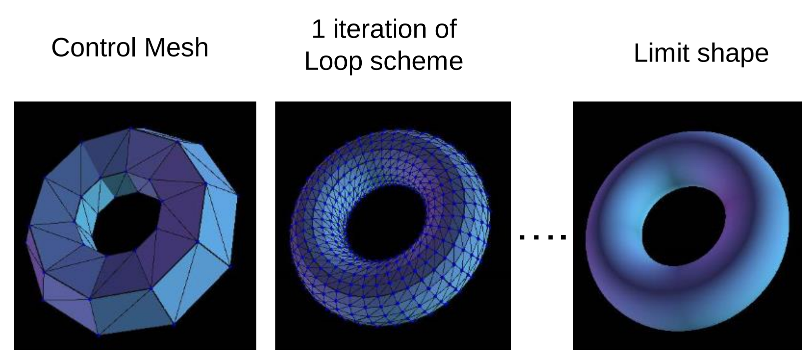
3D models
(geometry)



Level sets
(equation)

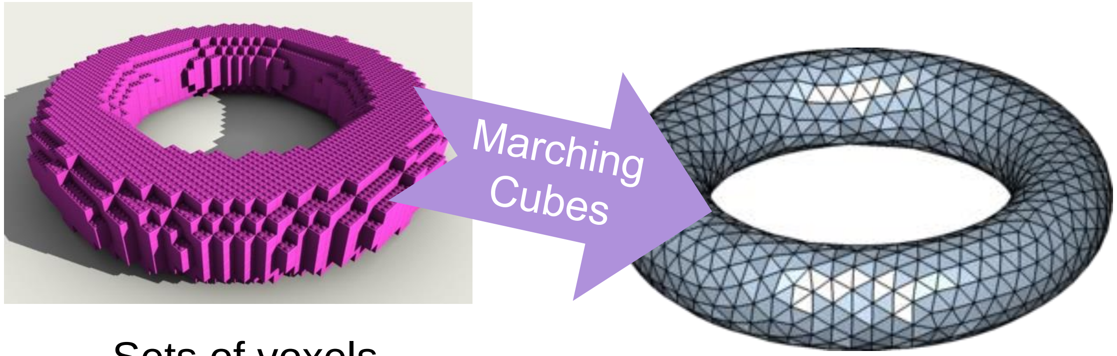


Parametric shapes
(Bézier, B-splines, NURBS…)



Subdivision surfaces

Points cloud



Sets of voxels

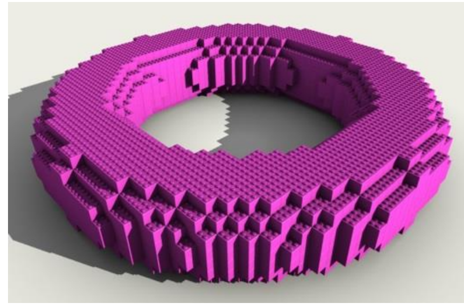
Mesh

3D models
(geometry)

Level sets
(equation)

Parametric shapes
(Bézier, B-splines, NURBS…)

Subdivision surfaces



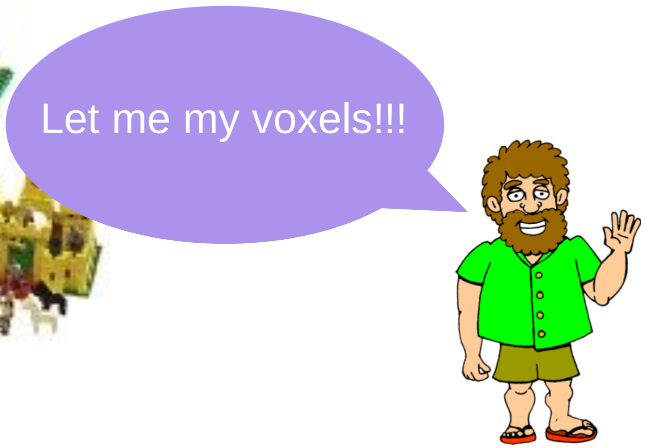
Sets of voxels

Just export the digital model in a mesh with marching cubes and simplification.

That’s the option followed by most people (it’s a good option).

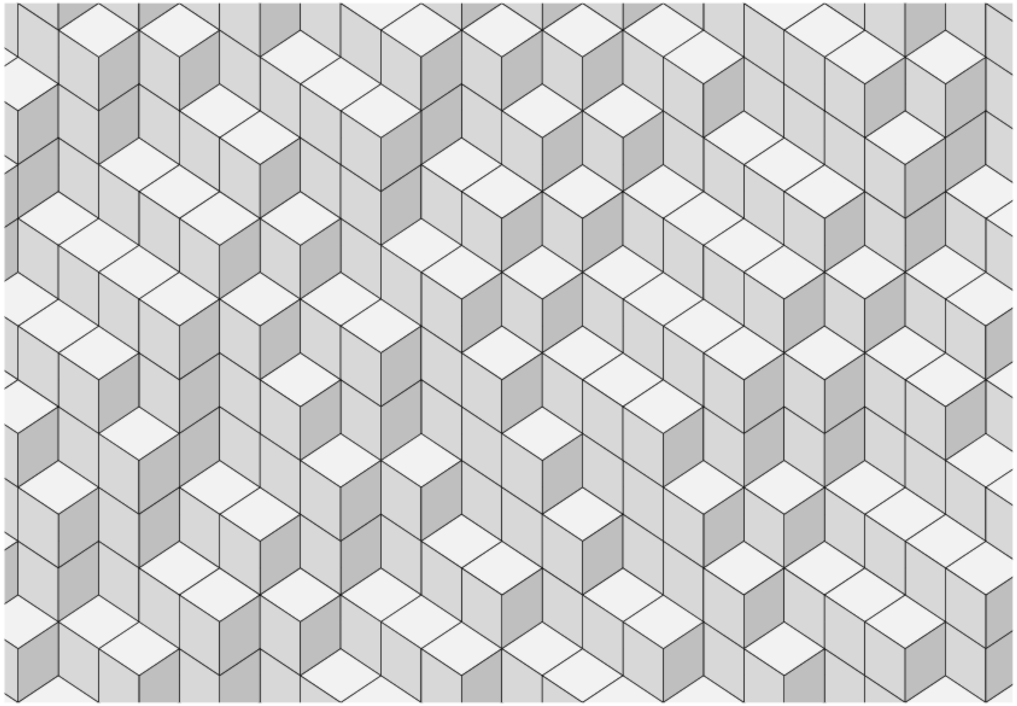
But some people are stubborn.

Are there some better reasons to do Digital Geometry ?



But some people are stubborn.
May be, they played to much legos during their childhood.

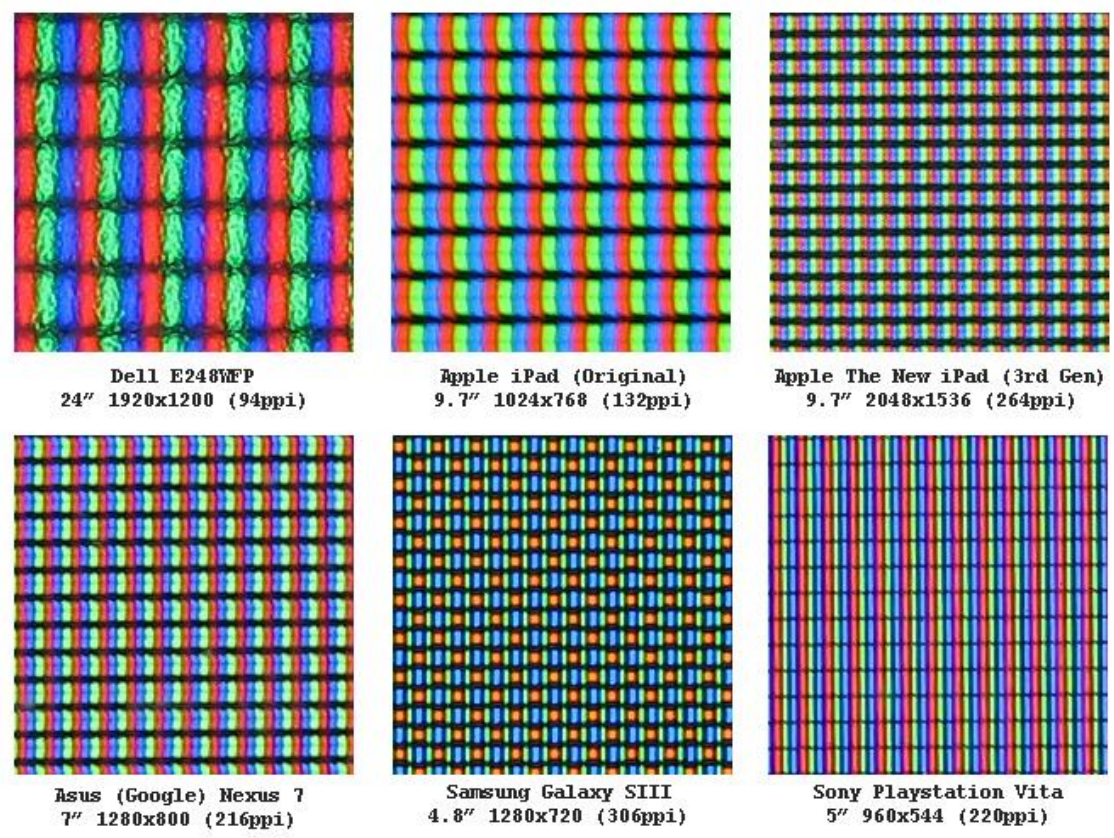
Are there some better reasons to do Digital Geometry ?



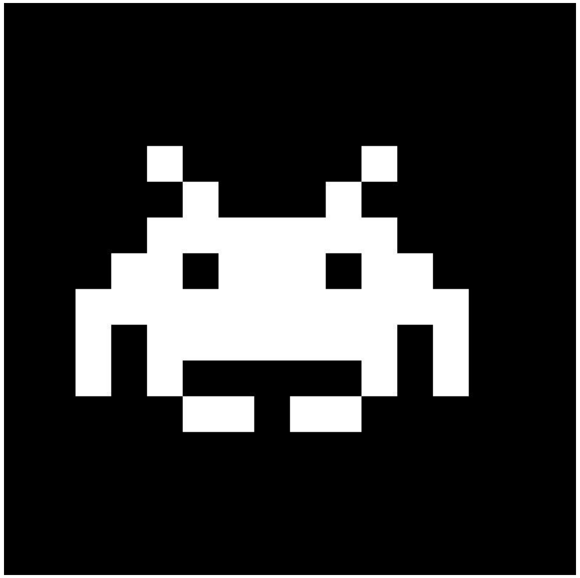
A stepped surface @ Thomas Fernique

For the beauty of theory ☺

Are there some better reasons to do Digital Geometry ?



Screens are lattices of pixels.



Binary image

Images are tabs of pixel values.

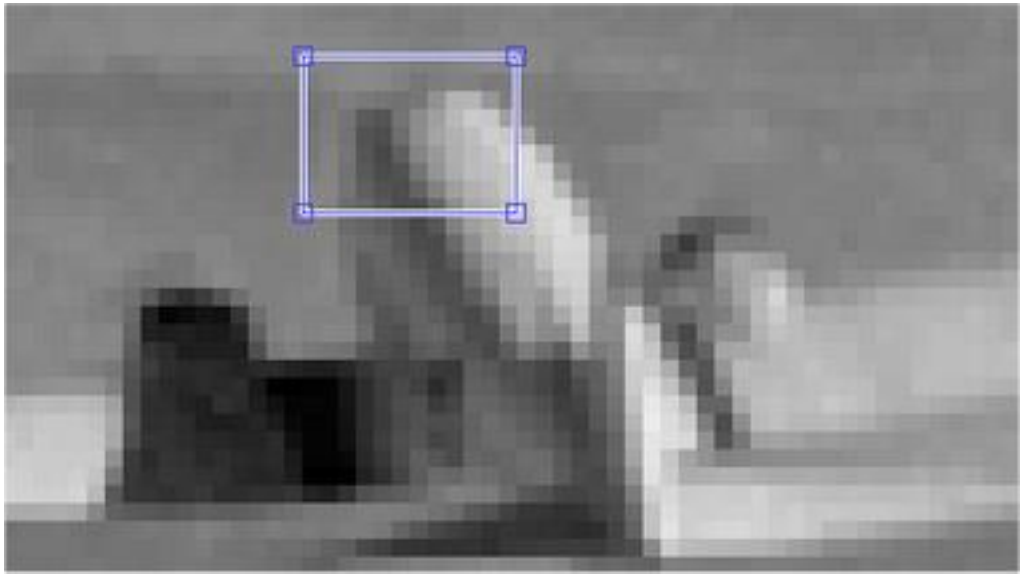
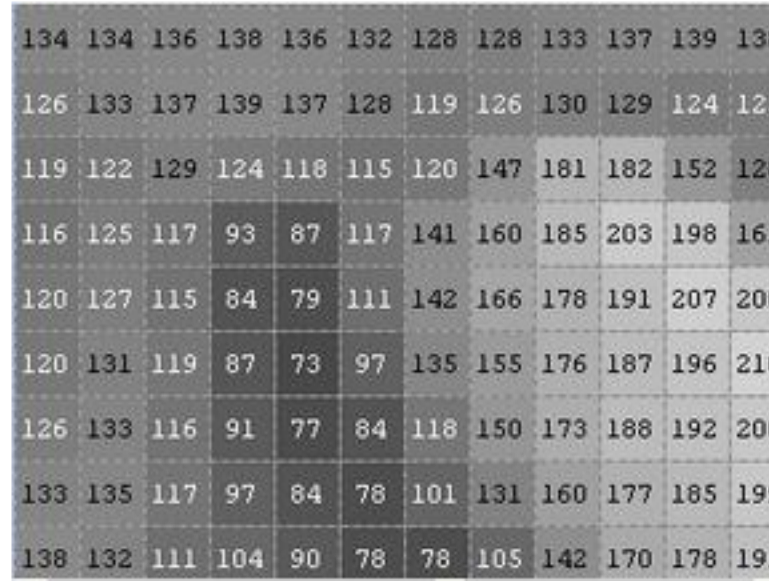


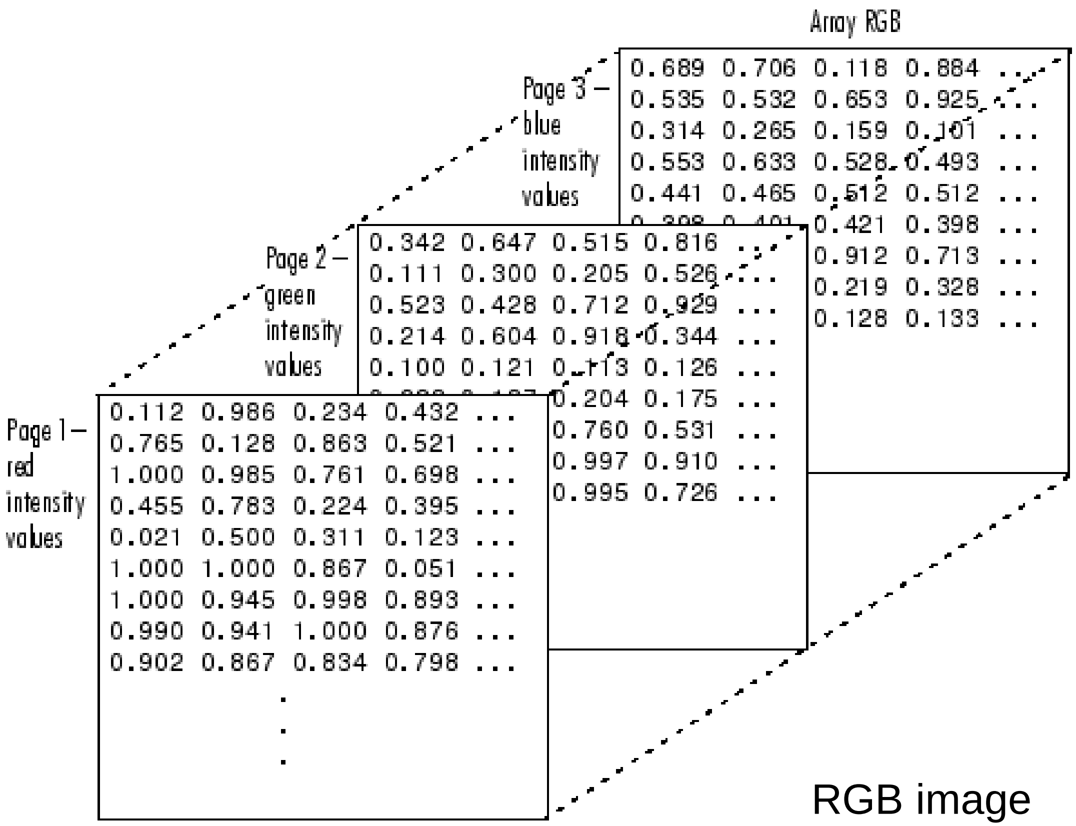
Image with grey-levels

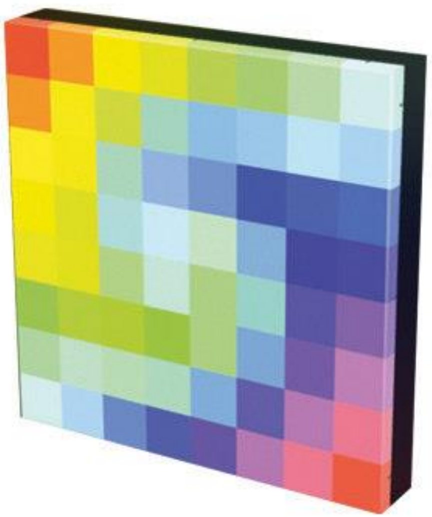


134	134	136	138	136	132	128	128	133	137	139	138
126	133	137	139	137	128	119	126	130	129	124	125
119	122	129	124	118	115	120	147	181	182	152	128
116	125	117	93	87	117	141	160	185	203	198	161
120	127	115	84	79	111	142	166	178	191	207	200
120	131	119	87	73	97	135	155	176	187	196	210
126	133	116	91	77	84	118	150	173	188	192	200
133	135	117	97	84	78	101	131	160	177	185	199
138	132	111	104	90	78	78	105	142	170	178	191

The values of the tab

Images are tabs of pixel values.





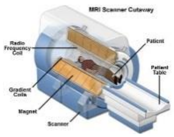
RGB image

Images are tabs of pixel values.


Cameras


3D scan


Kinect


MRI


US

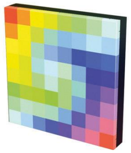
…

The input is digital

Computation

The output is digital

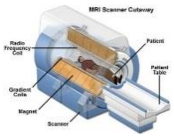
0111001010010




Cameras

3D scan

Kinect



MRI

US

…

The input is digital

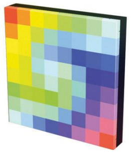
Which arithmetic for the computation ?

Integer arithmetic

Floating Point Arithmetic

The output is digital

0111001010010



Integer arithmetic

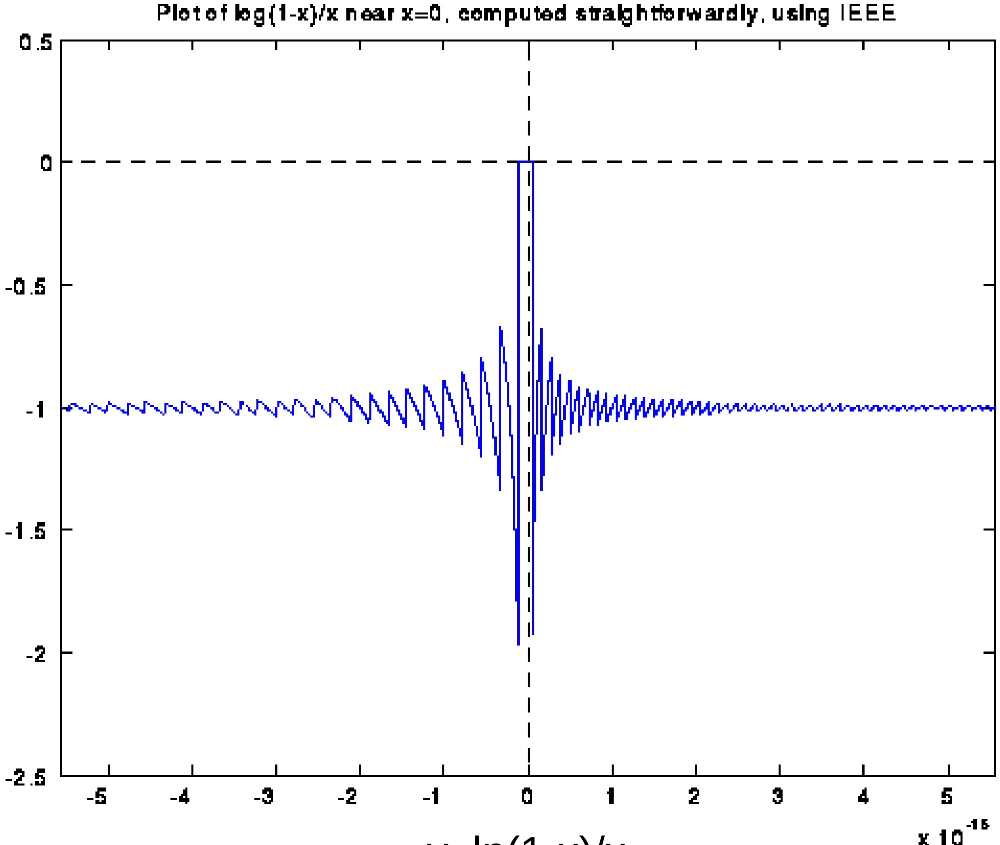
Suitable for computers
(exact computations)

Requires digital mathematics…

Floating Point Arithmetic

Suitable for mathematics
(continuous objects)

Problems of inaccuracy…

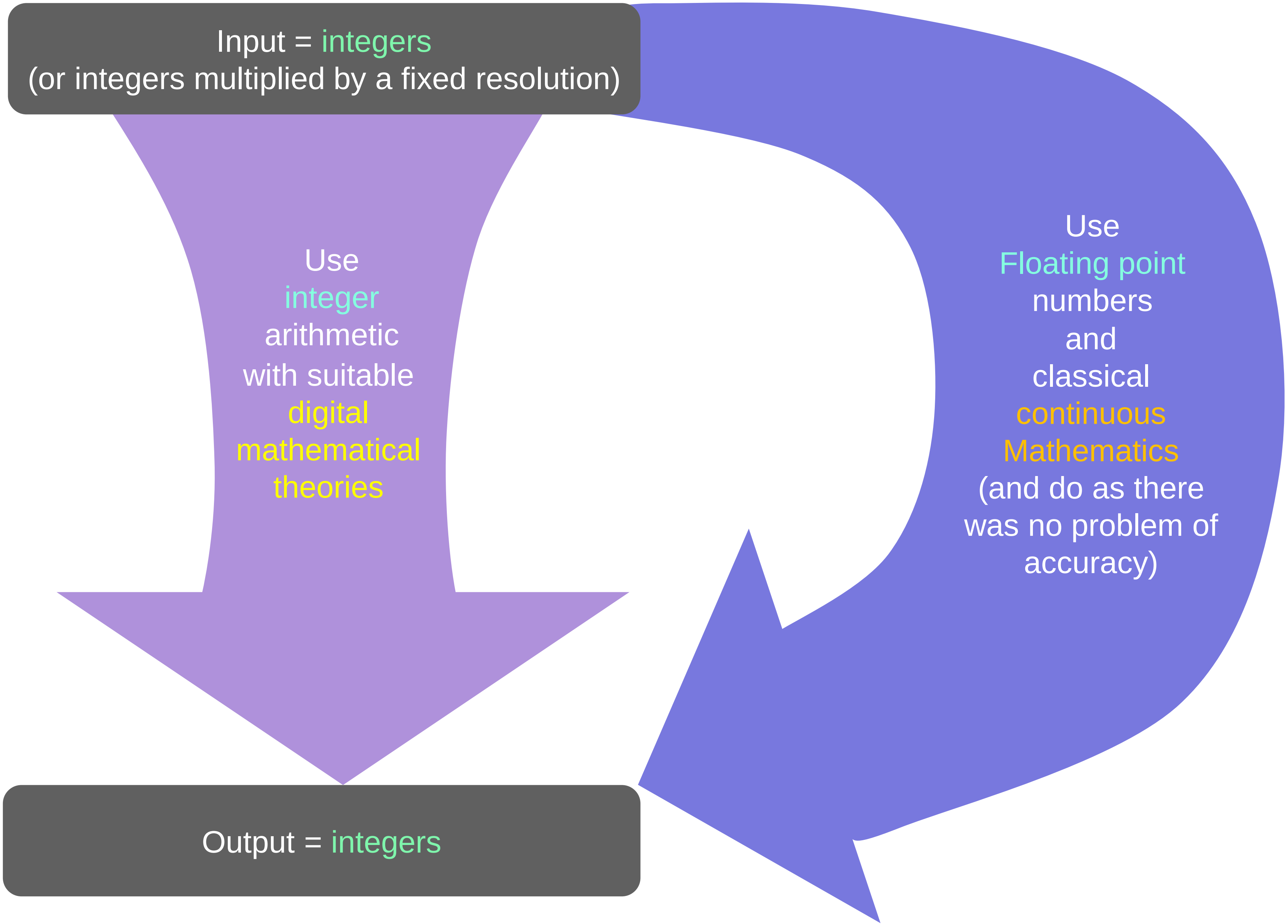


y=ln(1-x)/x
with IEEE 754 standard

Floating Point Arithmetic

Suitable for mathematics (continuous objects)

Problems of inaccuracy…


Input = integers
(or integers multiplied by a fixed resolution)
Use
integer
arithmetic
with suitable
digital
mathematical
theories
Use
Floating point
numbers
and
classical
continuous
Mathematics
(and do as there
was no problem of
accuracy)
Output = integers

Input = integers
(or integers multiplied by a fixed resolution)

Most popular option.

Use
Floating point
numbers
and
classical
continuous
Mathematics
(and do as there
was no problem of
accuracy)

Output = integers

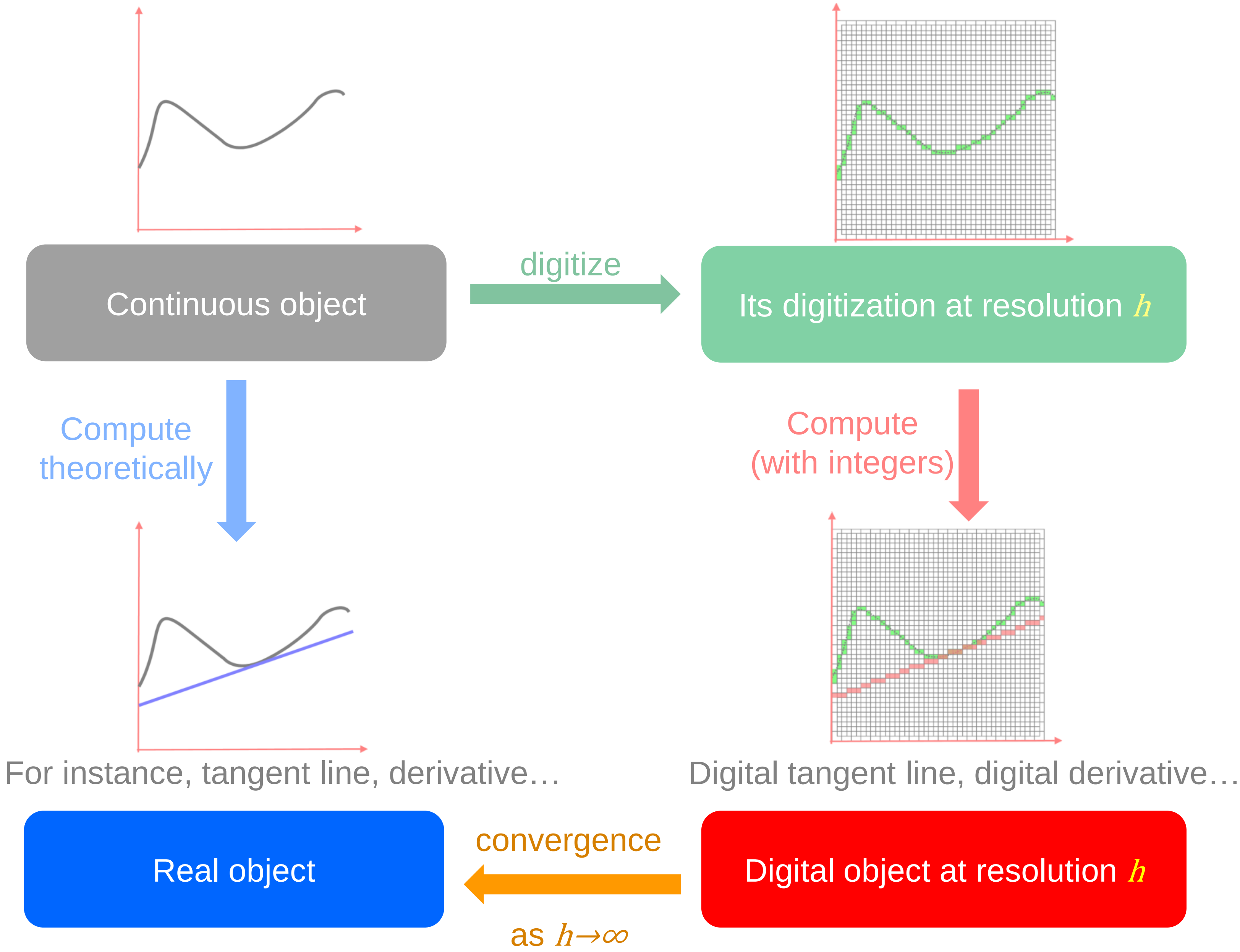
Input = integers
(or integers multiplied by a fixed resolution)

Use integer arithmetic with suitable digital mathematical theories

Output = integers

We can not do whatever we want. There are some constraints…

The developpment of digital mathematics is a **huge** challenge


Continuous object
digitize
Its digitization at resolution h
Compute theoretically
Compute (with integers)
For instance, tangent line, derivative…
Digital tangent line, digital derivative…
Real object
convergence
as $h \to \infty$
Digital object at resolution h

Plan

I — Introduction

II — About tangent estimators

III — Digital Primitives

IV — Measurements

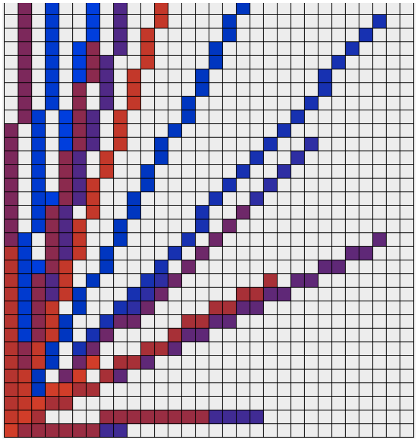
V — Transformations and Combinatorics

Plan

II

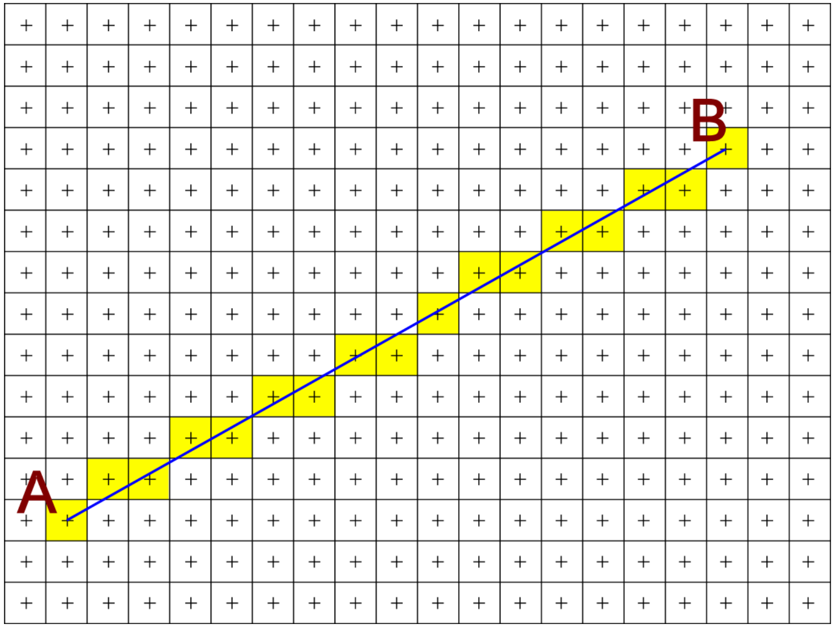
About tangent estimators

In the early 60's, the beginning of computer graphics required the first algorithms to display figures on the screen.



First requirement:
display straight lines
and other elementary figures.

Working for IBM, *Jack Elton Bresenham* developped an « optimized » algorithm to draw a line (1962).



Bresenham straight line from A to B.

The concept of digital line can be easily defined…

Digital lines definition:

Digital lines of Z^2 are subsets of Z^2 characterized by a double inequality:

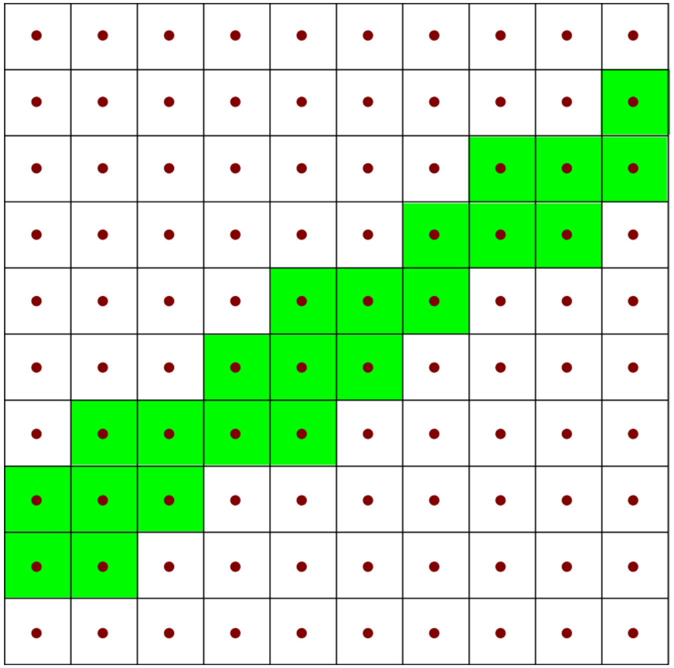
$$h \leq ax+by < h+\Delta$$

It's exactly the same for affine sub-spaces of codimension 1 (digital hyperplanes) of Z^d.

Digital lines definition:

Digital lines of Z^2 are subsets of Z^2 characterized by a double inequality:

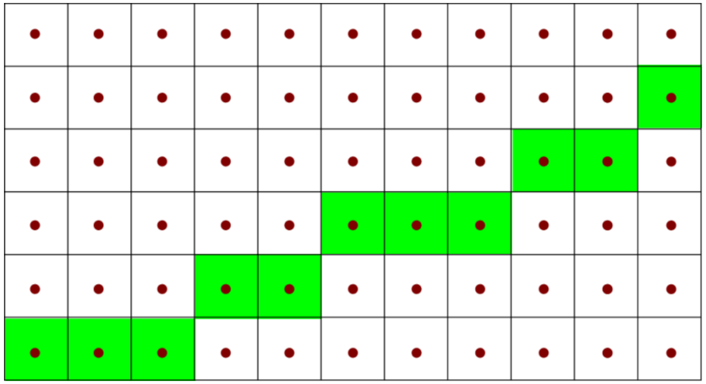
$$h \leq ax+by < h+\Delta$$



Digital lines definition:

Digital lines of Z^2 are subsets of Z^2 characterized by a double inequality:

$$h \leq ax+by < h+\Delta$$



A digital line is *naïve* if $\Delta = max\{|a|,|b|\}$.

It's 8-connected.

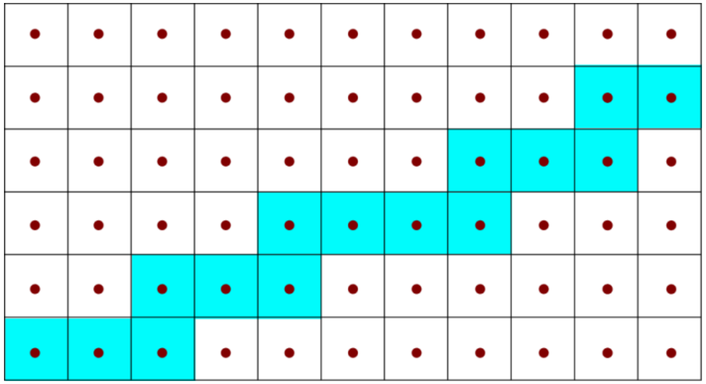
The complementary has two 4-connected components.

There is no simple point.

Digital lines definition:

Digital lines of Z^2 are subsets of Z^2 characterized by a double inequality:

$$h \leq ax+by < h+\Delta$$

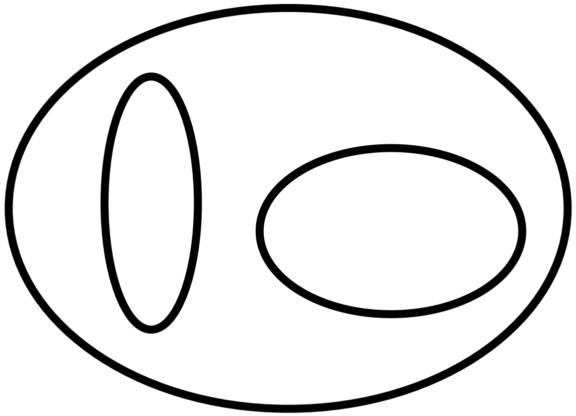


A digital line is *standard* if $\Delta = |a|+|b|$.

It's 4-connected.

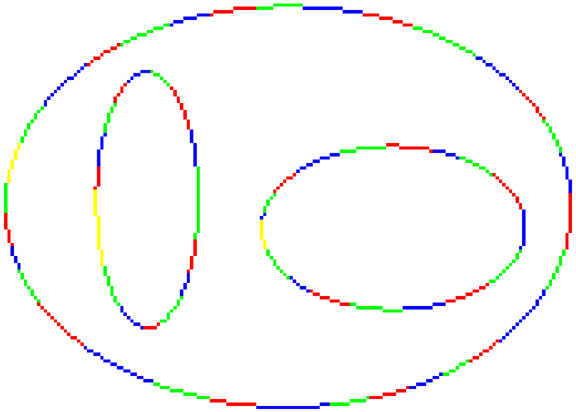
The complementary has two 8-connected components.

There is no simple point.



Input: A digital curve

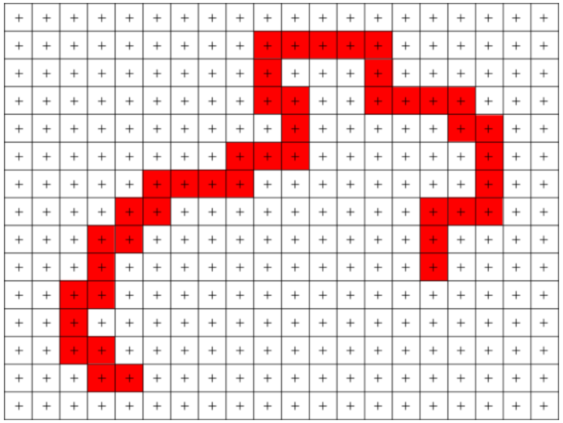
Segmentation in pieces of digital straight lines (72 pieces)



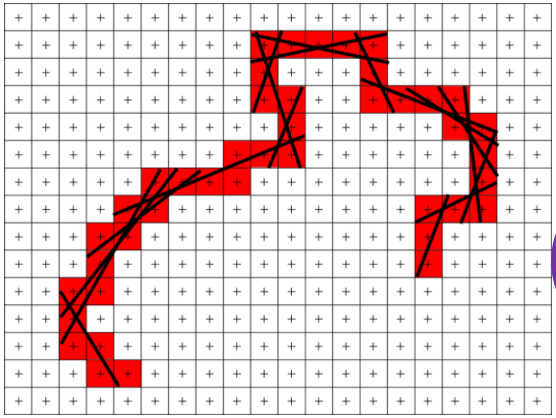
Worst case complexity: linear time

I. Debled-Rennesson, J-P Reveilles, 2th DGCI, 1992.

Output: Its decomposition in Digital Straigh Segments

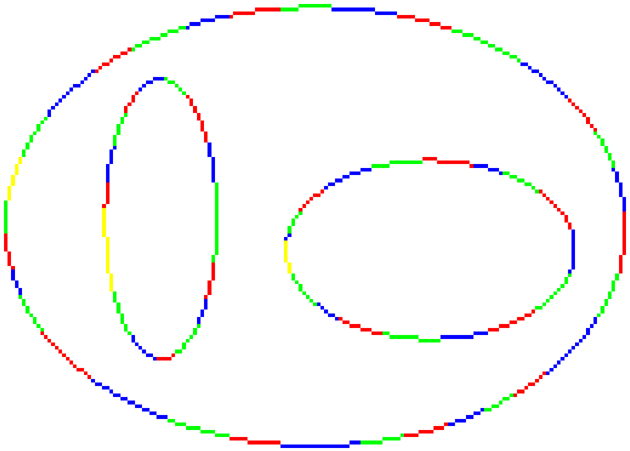


Input: A digital curve



Worst Case Complexity:
Linear Time
J-O Lachaud, A. Vialard, F. De Vieilleville, DGCI 2005.

Output: Its Tangential Cover

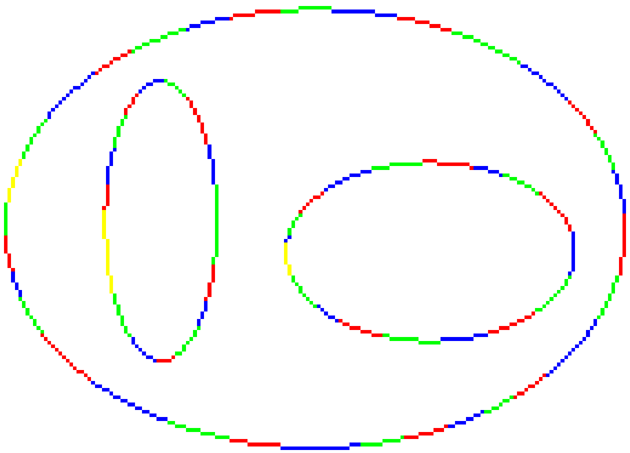


Interest: Maximal Digital Straight Segments around a point x provide
gent direction at x.

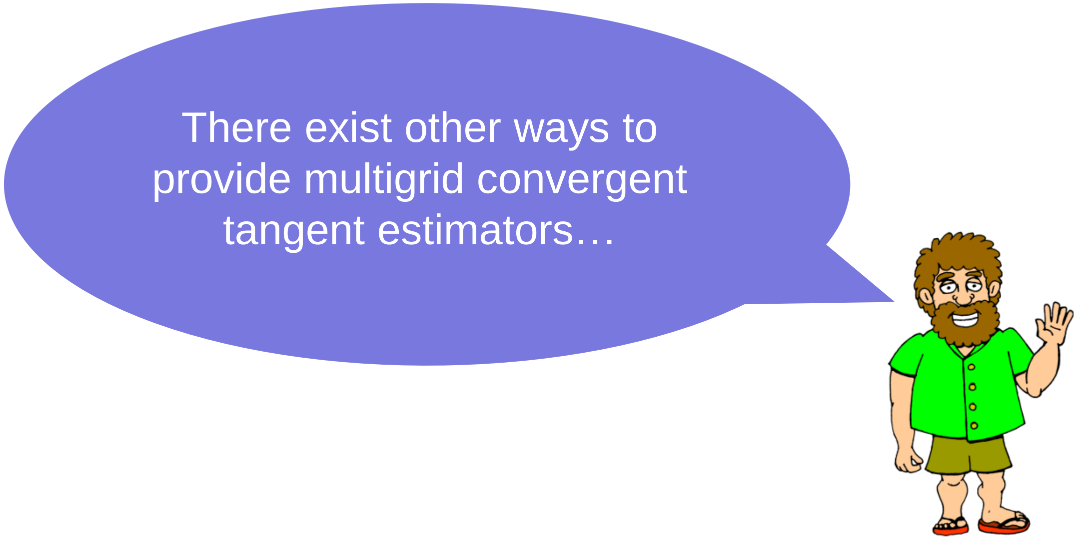
Average convergence rate
$O(h^{2/3})$
In *J-O Lachaud, 2006*

Good news:
it's multigrid convergent
(under some assumptions)
O Lachaud, A. Vialard, F.
De Vieilleville, DGCI 2005.

Locally convex shapes

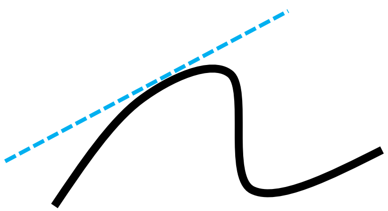


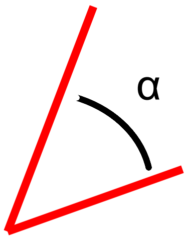
Interest: Maximal Digital Straight Segments around a point x provide the *tangent direction* at x.

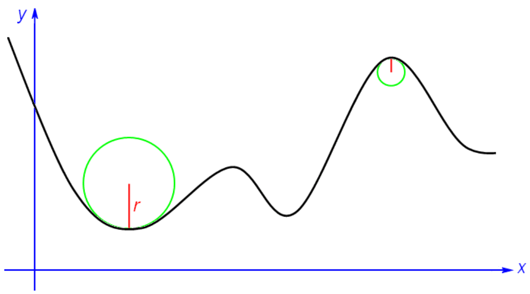


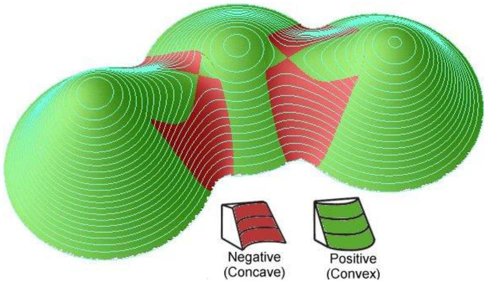
Why this interest for computing the tangent or normal direction ?

To provide measurements…









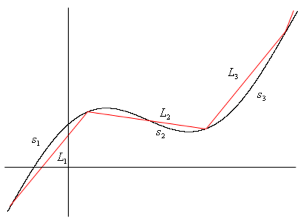
Tangent and normal directions

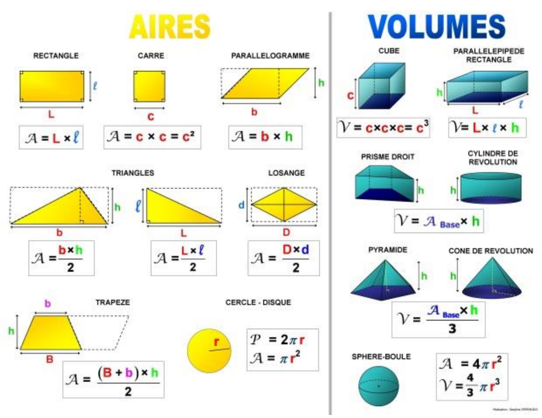
Angles

Curvatures of curves and surfaces

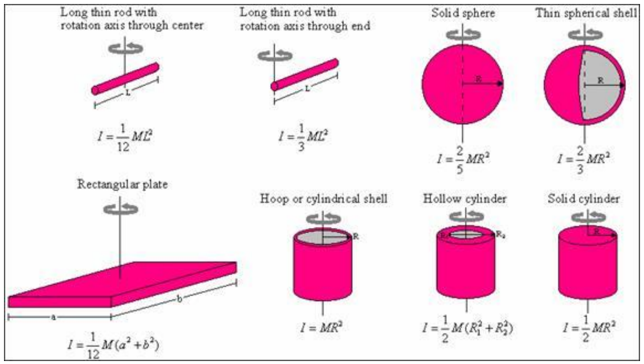
Why this interest for computing the tangent or normal direction ?

To provide measurements…





Length, areas, volumes



Sums
(barycenter coordinates, moment…)

$Length=\int 1\, ds$

$Area=\iint 1\, dS$

$Volume=\iiint 1\, dV$

$Sum=\int f(x)\, ds$

$Sum=\iint f(x)\, dS$

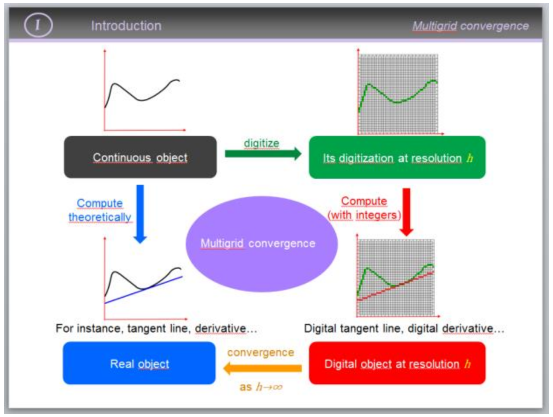
$Sum=\iiint f(x)\, dV$

Why this interest for computing the tangent or normal direction ?

To provide measurements…

Preserve the relations between measurements (*turning Number Theorem, Gauss-Bonnet…*)

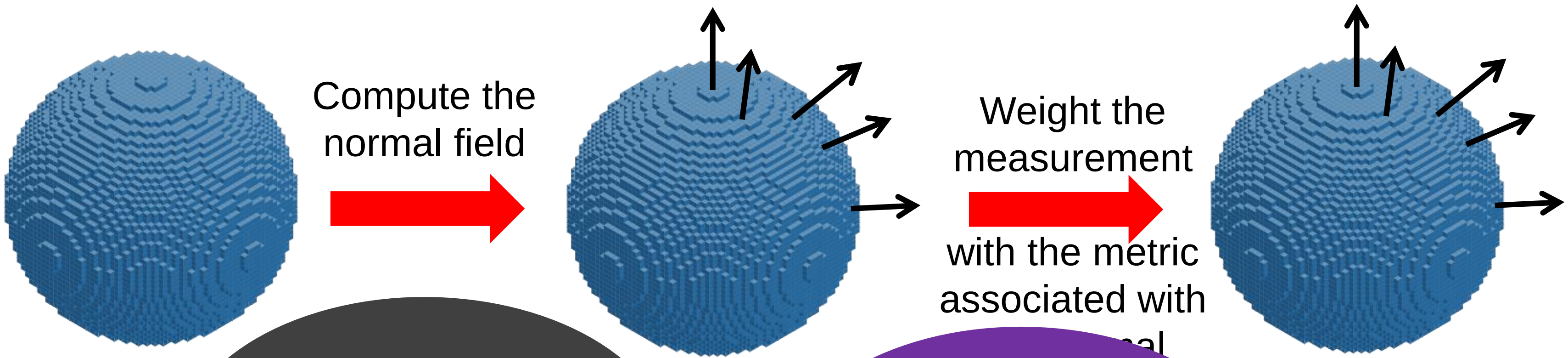
Don't forget *Multigrid convergence…*



Review for 2D in Book chapter *« Multigrid convergent Discrete estimators »* from *D. Coeurjolly*, *J-O Lachaud* and *T. Roussillon*.

Why this interest for computing the tangent or normal direction ?

To provide measurements…



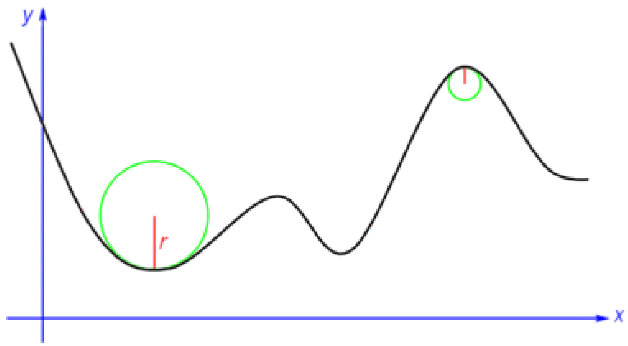
Use a multigrid convergent computation of normals…

It can guarantee the Multigrid convergence of the measurement.

Everything is cool, but…
Can we do better than using digital straight segments ?

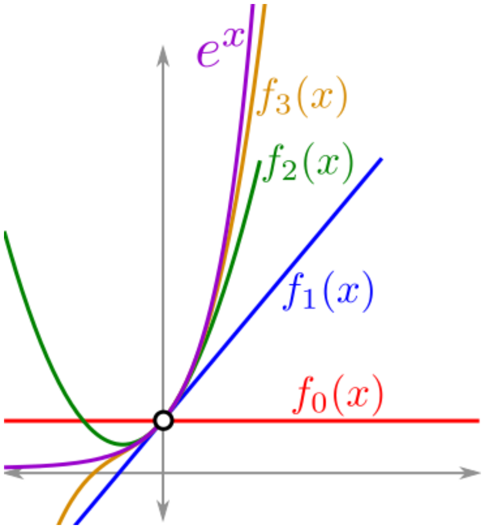
Not only for tangent estimation, but also for conversion from raster to vector graphics.

Use digital primitives of higher degree .



Curvature is defined with *osculating circles*

Use digital circles



An analytical function is approximated by its *Taylor Polynomial* of degree n.

Use a more generic approach

Use digital primitives of higher degree .

Use a more generic approach

Plan

I — Introduction

II — About tangent estimators

III — Digital Level Layers

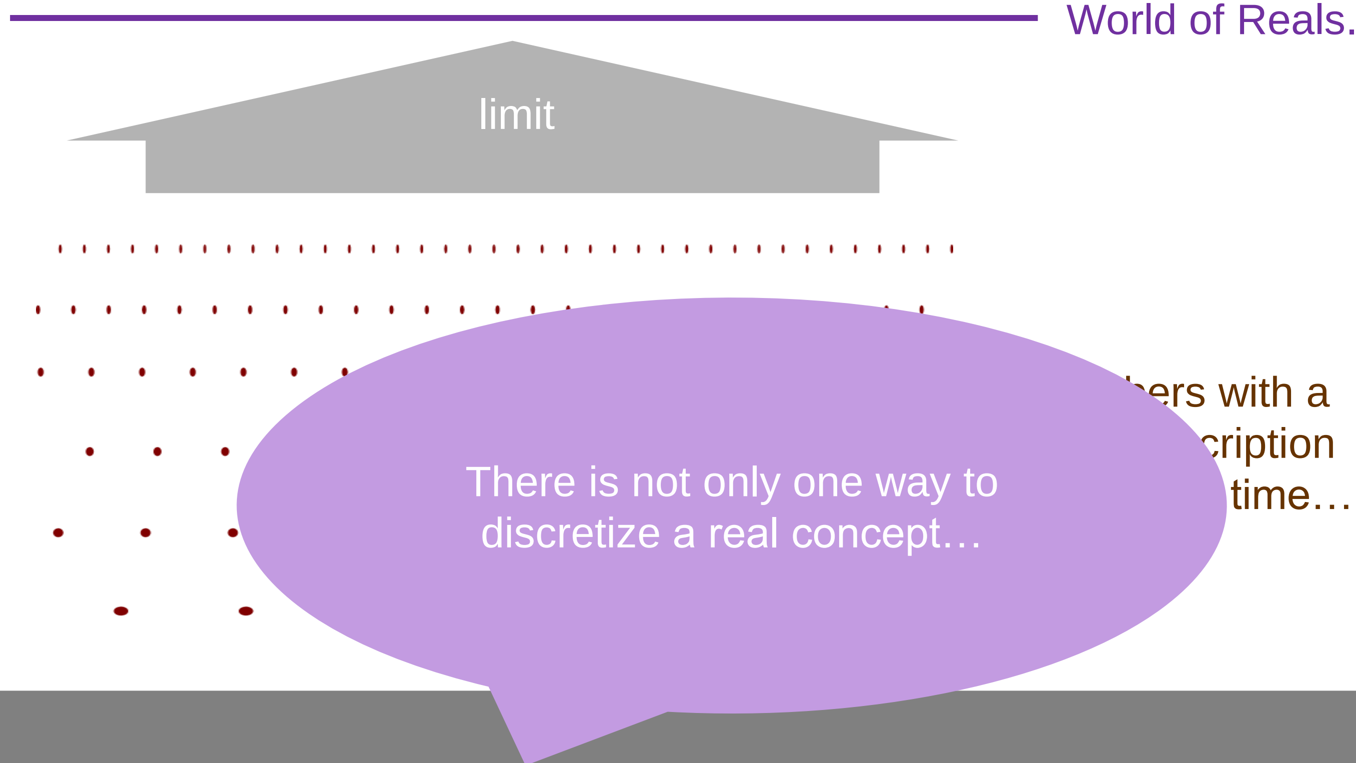
IV — DLL decomposition

V — Algorithm

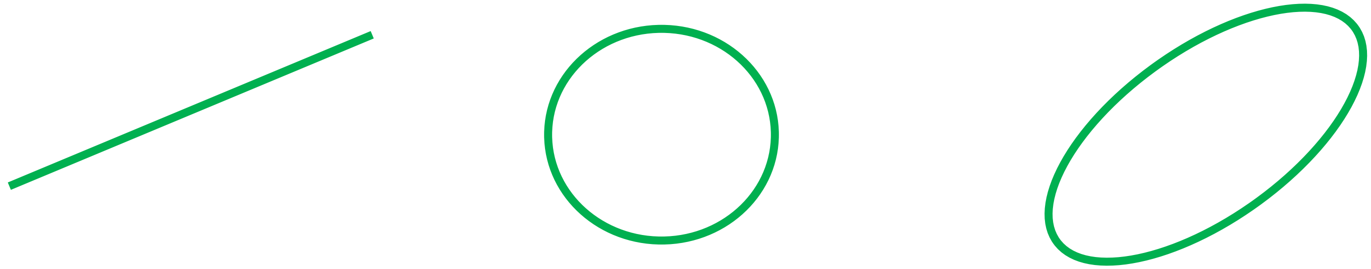
Plan

III Digital Level Layers

Usual geometry is based on real numbers, which by paradox are ’’unreal’’.

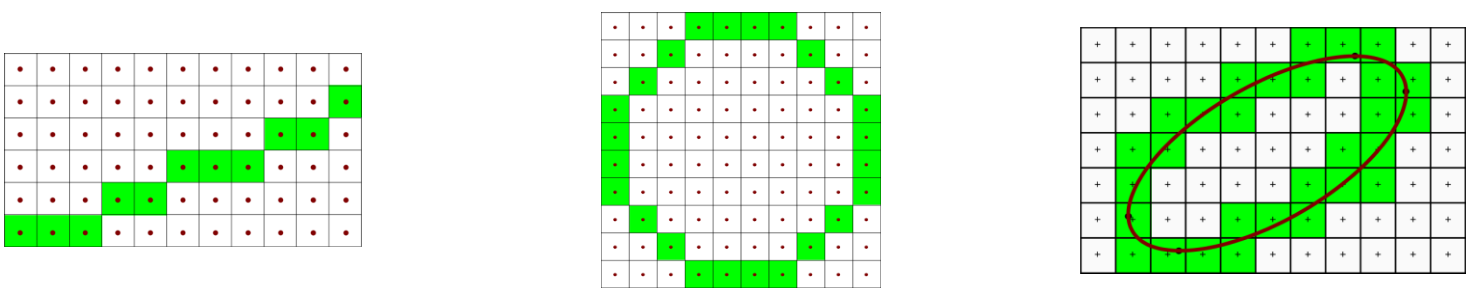


Different discrete objects or concepts have the same limit …

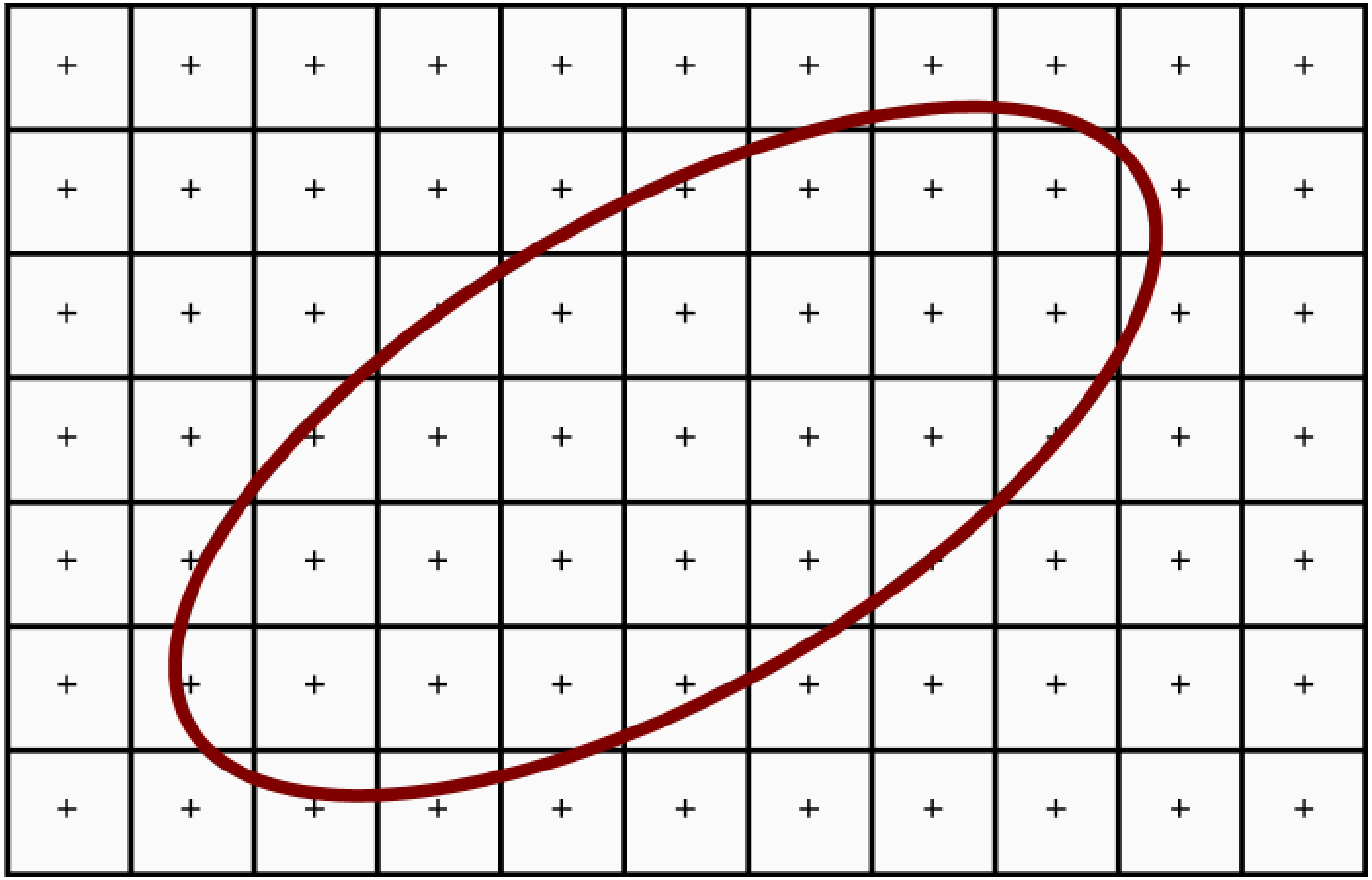


Continuous figures.

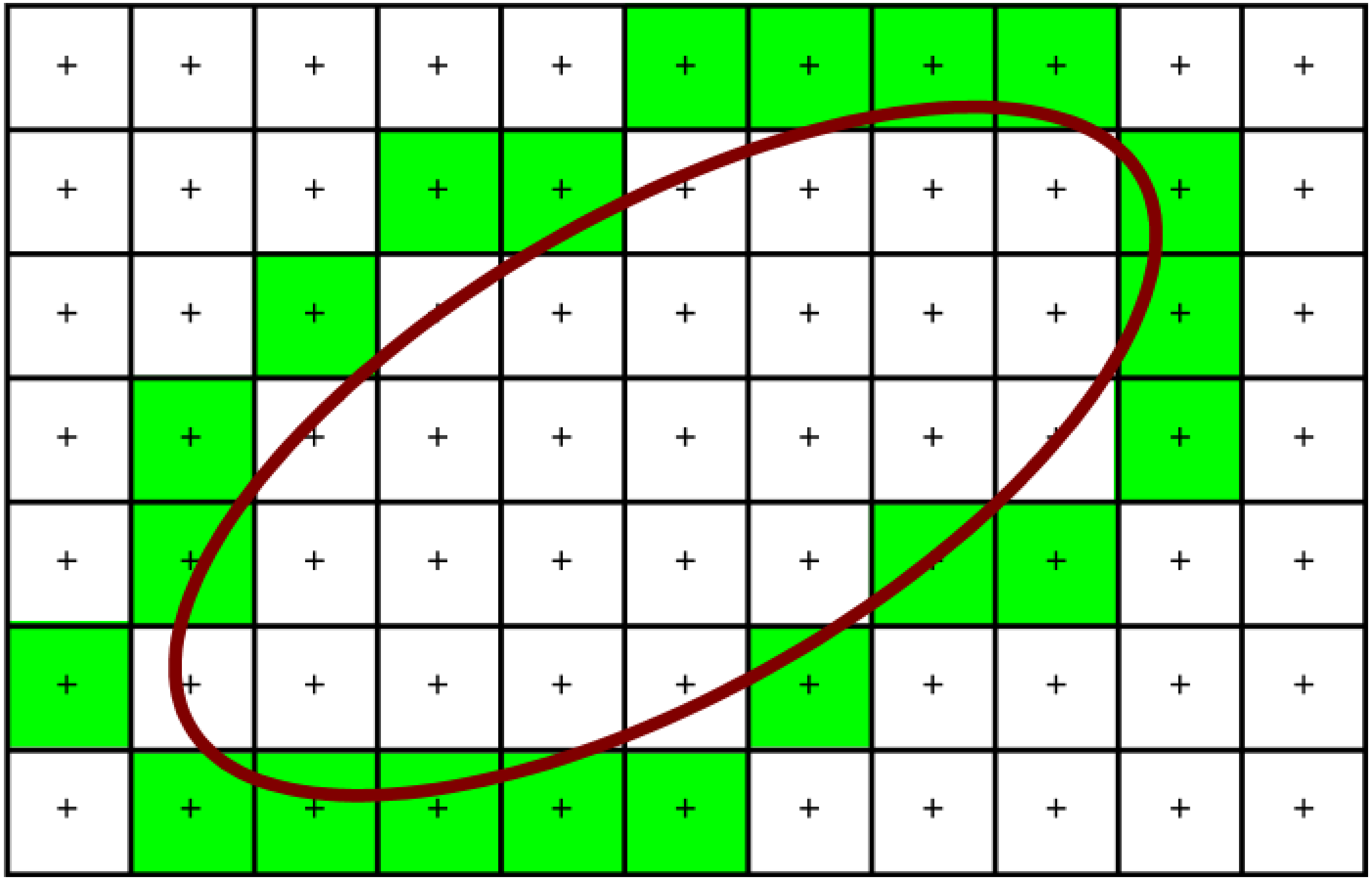
Three approaches can be used to define digital primitives:
- topological
- morphological
- analytical



Digital figures.



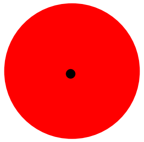
Task: define a digital primitive for S.



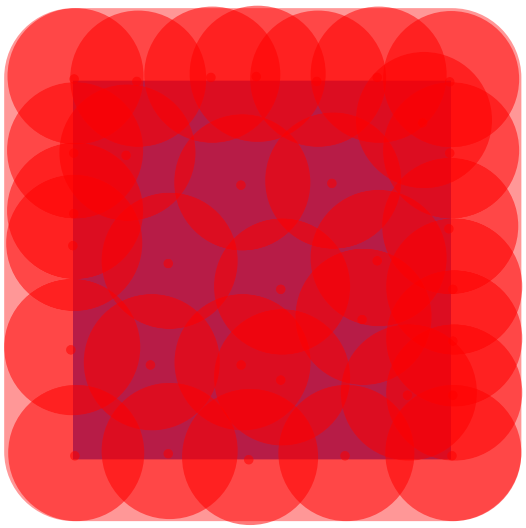
Task: define a digital primitive for S.

The *Minkowski's sum* S+B is the set of points covered by the structuring elements as it moves all along the shape.

A shape S

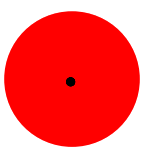


A structuring element B

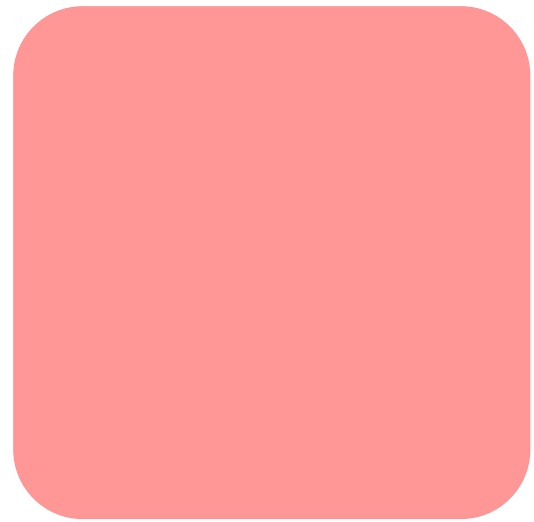


The *Minkowski’s sum* S+B is the set of points covered by the structuring elements as it moves all along the shape.

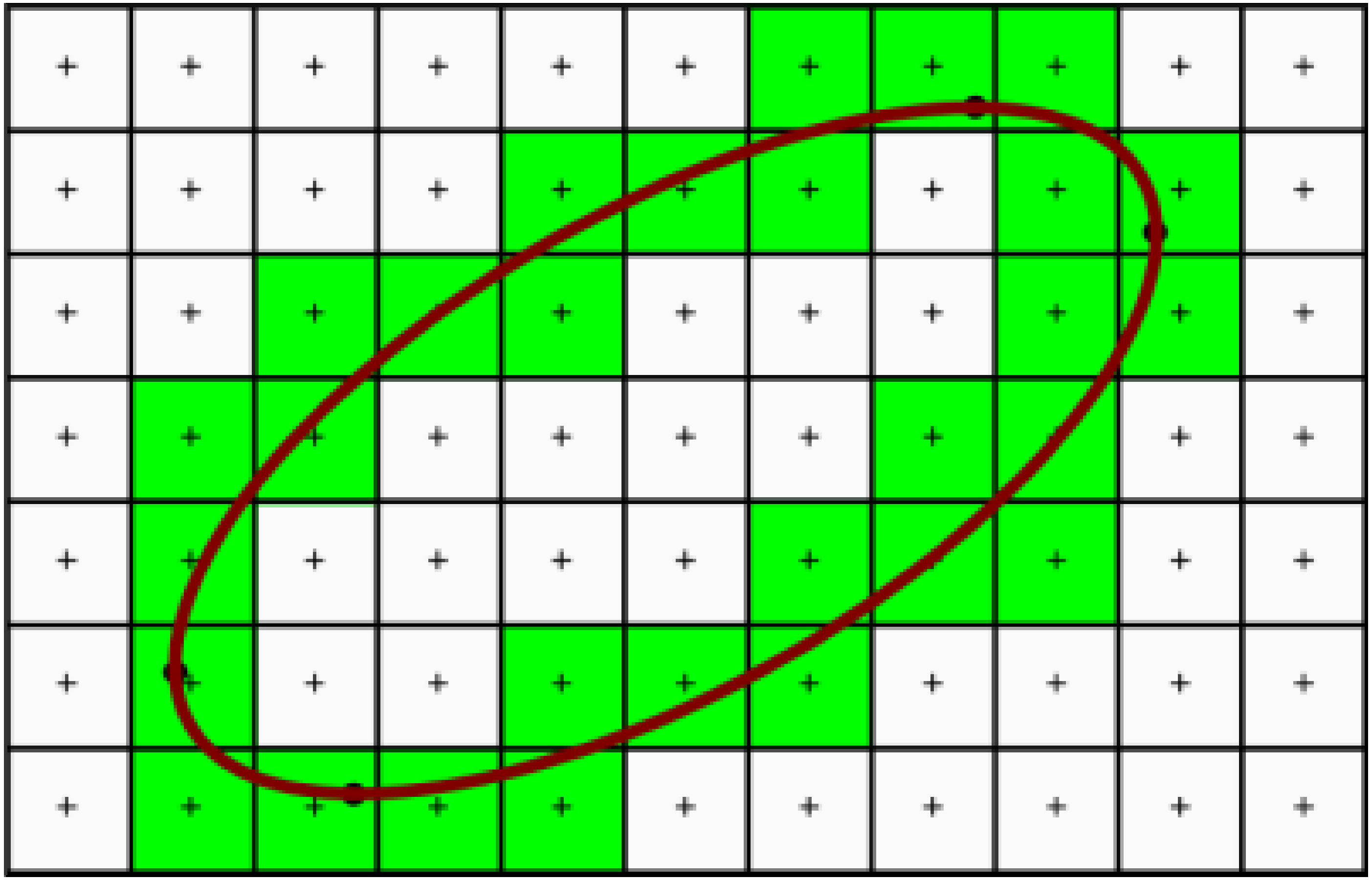
A shape S



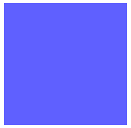
A structuring element B

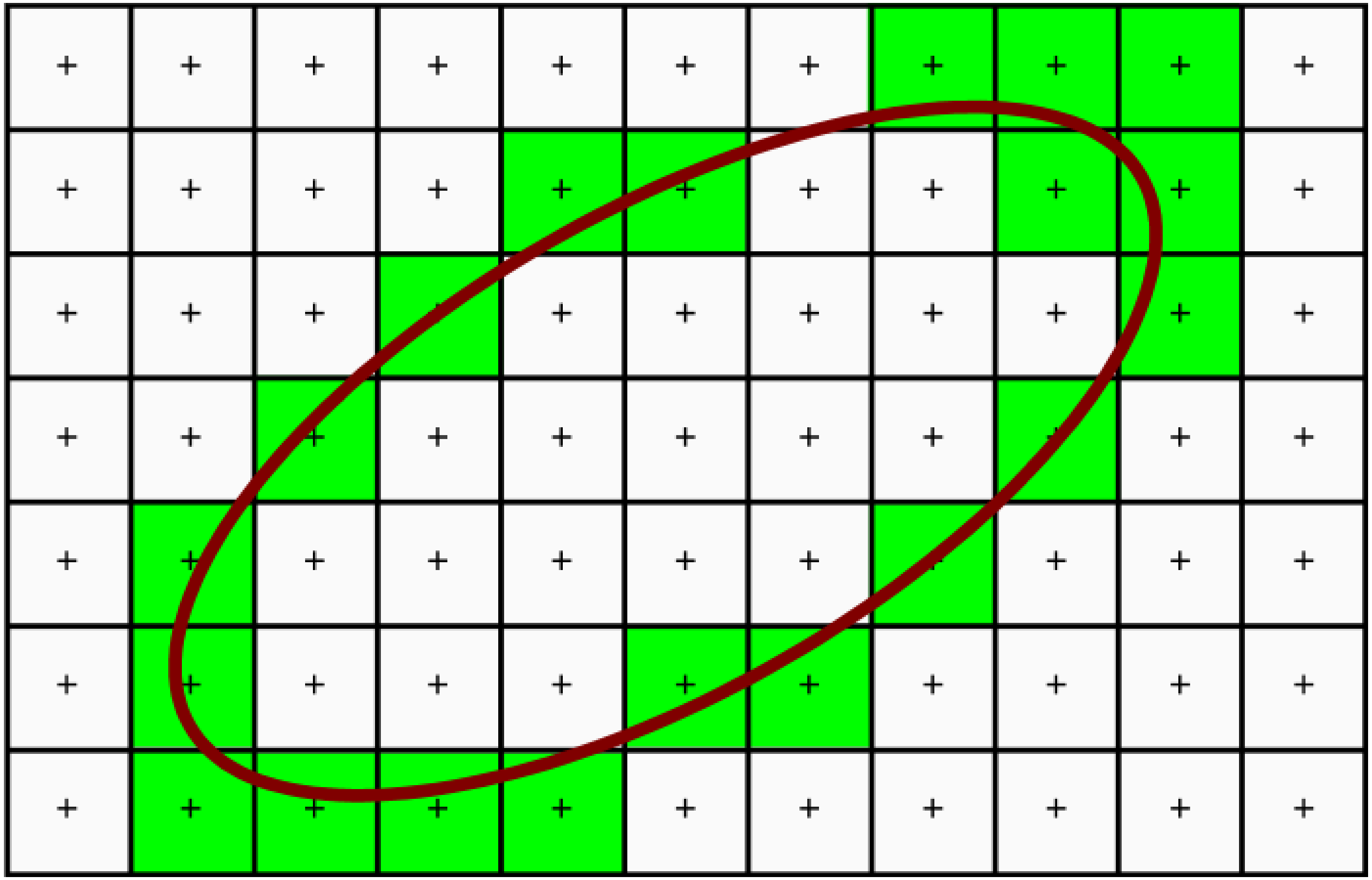


The dilation of S by B

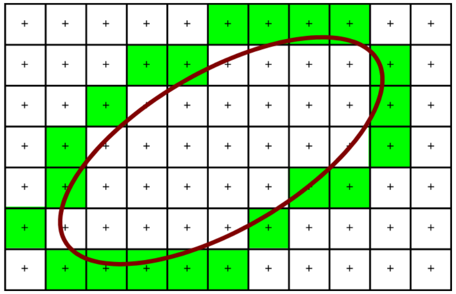


Structuring element

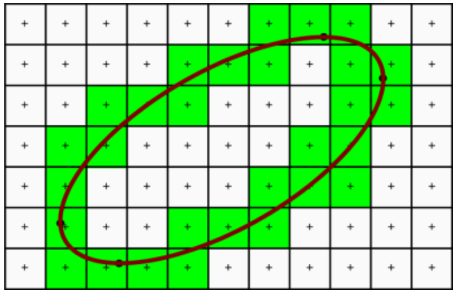




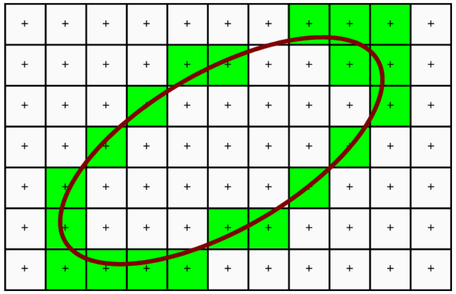
We relax the equality $f(x)=h$ in a double inequality $h-\Delta/2 \leq f(x)<h+\Delta/2$.



Topological approach.



Morphological approach.

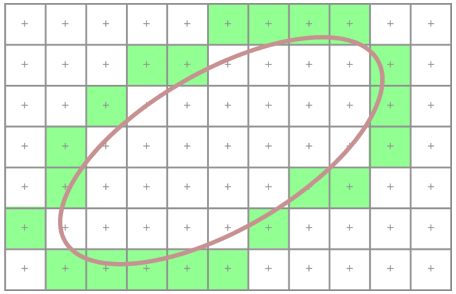


Analytical approach.

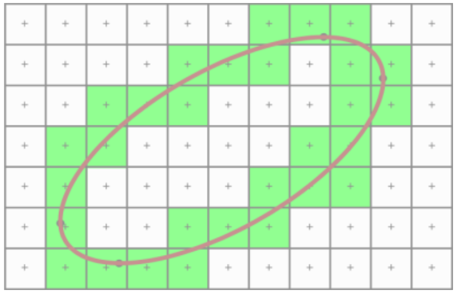
The 3 definitions collapse for lines in Z^2, planes in Z^3… hyperplanes in Z^d
(affine sub-spaces of codimension 1)

Each approach has its own parameters but there is a correspondance.

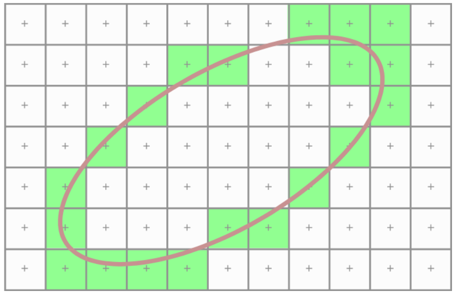
Topology	Morphology	Algebra
Neighborhood	Structuring element	value Δ
$Ball\ N_\infty$	$Ball\ N_1$	$\Delta = N_\infty(a)$
$Ball\ N_1$	$Ball\ N_\infty$	$\Delta = N_1(a)$



Topological approach.



Morphological approach.



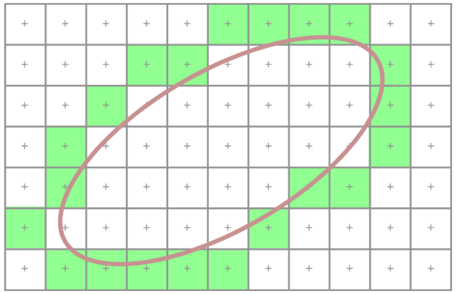
Analytical approach.

The 3 definitions collapse for lines in Z^2, planes in Z^3… hyperplanes in Z^d (affine sub-spaces of codimension 1)

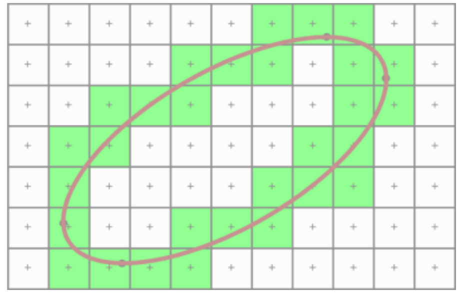
Each approach has its own parameters but there is a correspondance.

Topology	Morphology	Algebra
Neighborhood	Structuring element	value Δ
Ball N_∞	Ball N_1	$\Delta = N_\infty(a)$
Ball N_1	Ball N_∞	$\Delta = N_1(a)$

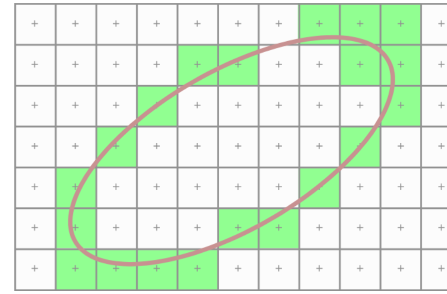
Naïve class.



Topological approach.



Morphological approach.



Analytical approach.

The 3 definitions collapse for lines in Z^2, planes in Z^3… hyperplanes in Z^d
(affine sub-spaces of codimension 1)

Each approach has its own parameters but there is a correspondance.

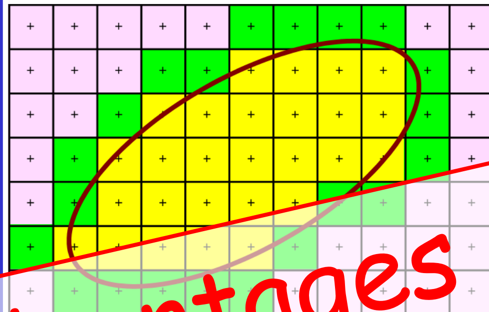
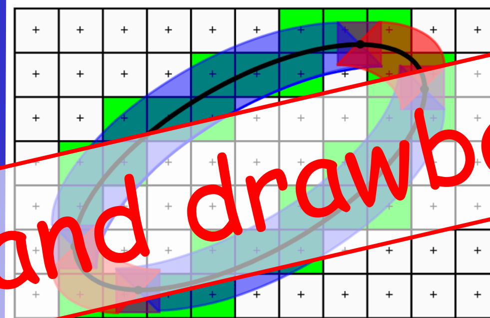
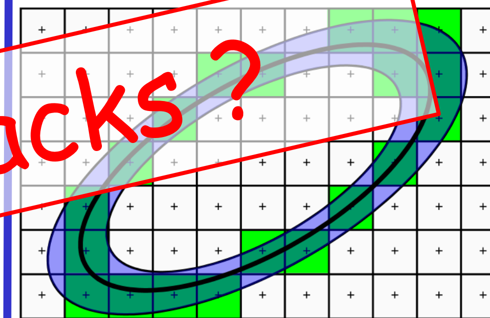
Standard class.

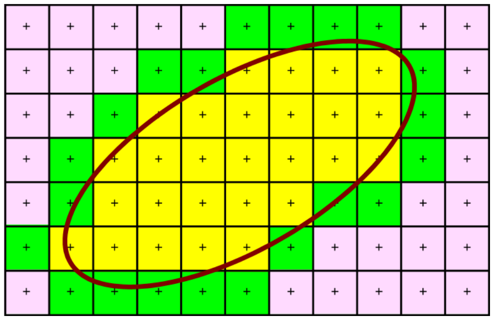
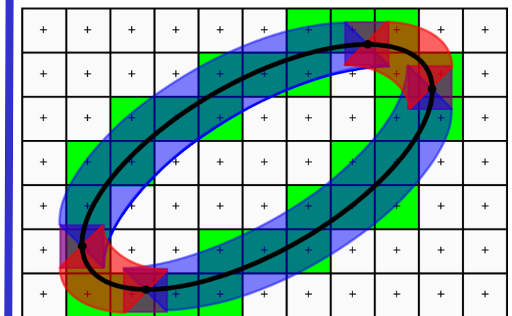
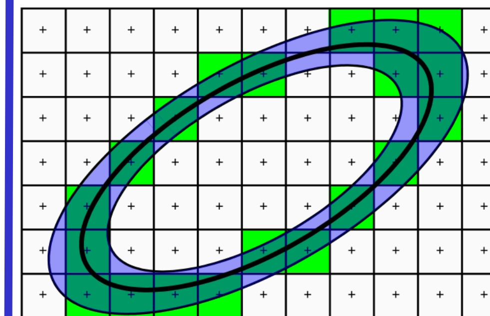
Topology	Morphology	Algebra
Neighborhood	Structuring element	value Δ
Ball N_∞	Ball N_1	$\Delta = N_\infty(a)$
Ball N_1	Ball N_∞	$\Delta = N_1(a)$

The 3 definitions collapse for lines in Z^2, planes in Z^3… hyperplanes in Z^d
(affine sub-spaces of codimension *1*)

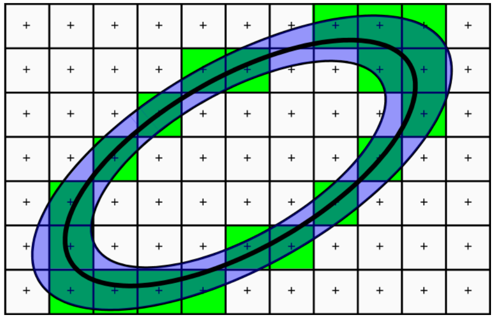
They don't collapse for arbitrary shapes.

Advantages and drawbacks ?

	Topology	Morphology	Analysis
Properties			
Topology			
Morphology			
Algebraic characterization			
Recognition algorithm			

	Topology	Morphology	Analysis
Properties			
Topology	✓	✓	✗
Morphology	✓	✓	✗
Algebraic characterization	✗	✗	✓
Recognition algorithm	✓ **SVM**	✗	✓

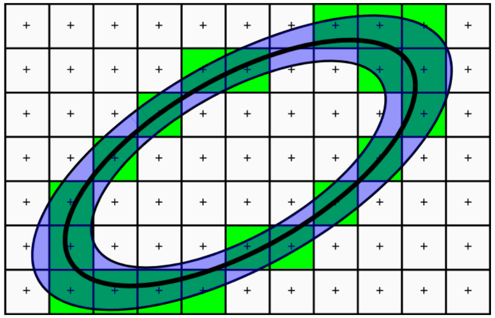
Analysis



Topology

Morphology

Analysis

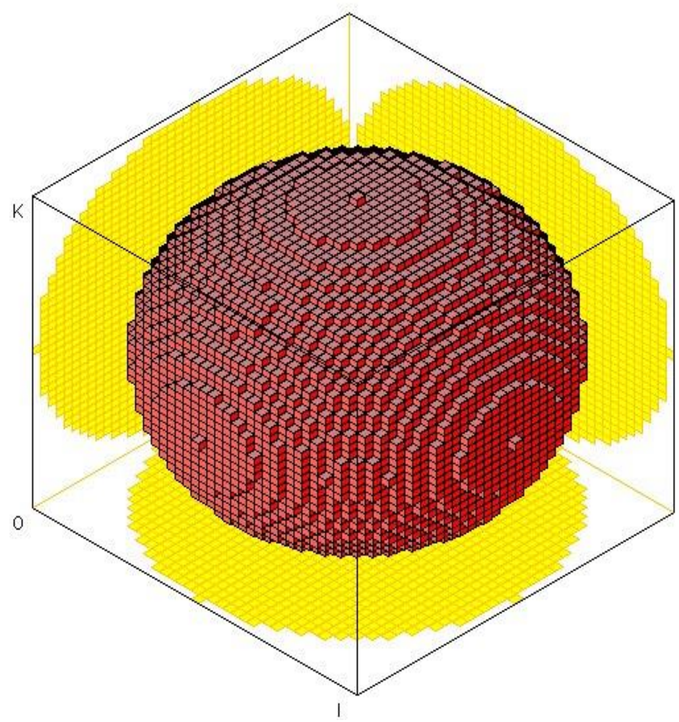


Topology 

Morphology 

Digital Level Layer definition:
A *Digital Level Layer* (name coming from *Level sets*) is a subset of Z^d characterized by a double inequality:

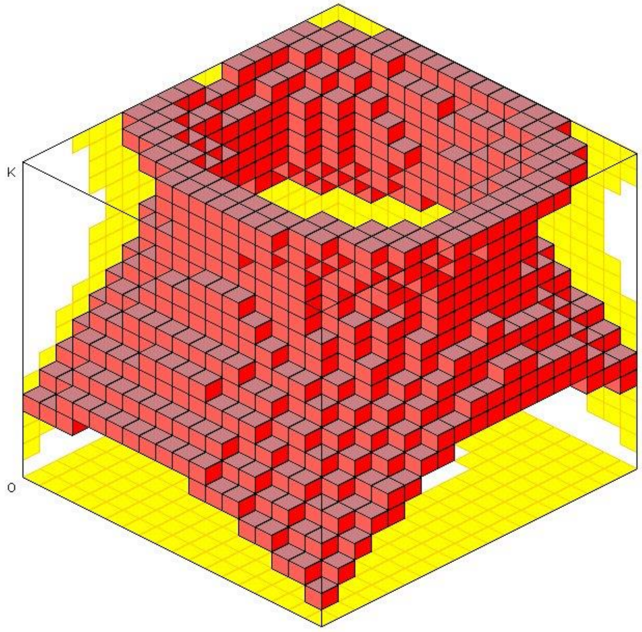
$$h \leq f(x) < h'$$



Digital Level Layer (DLL for short)

Digital Level Layer definition:
A *Digital Level Layer* (name coming from *Level sets*) is a subset of Z^d characterized by a double inequality:

$$h \leq f(x) < h'$$

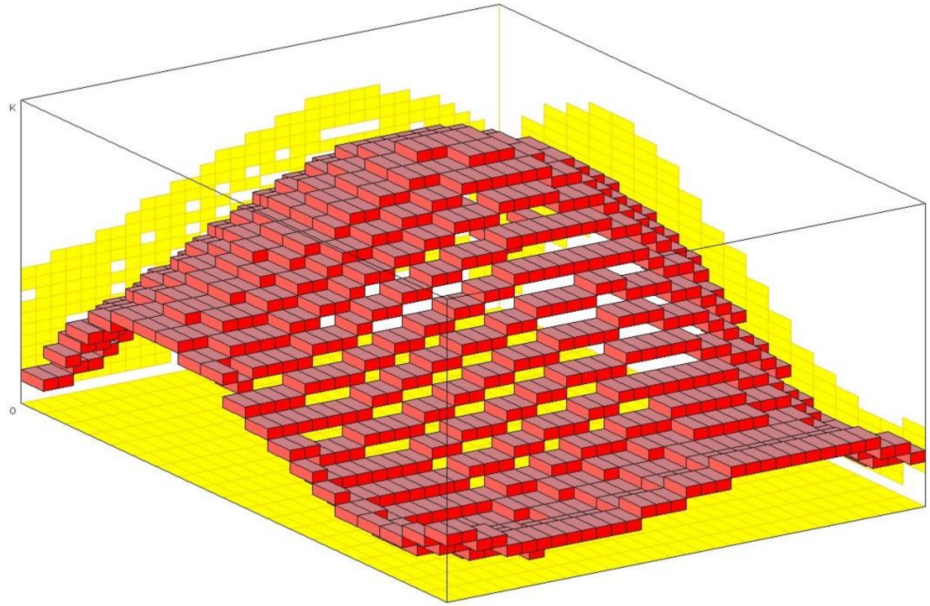


Digital Level Layer (DLL for short)

Digital Level Layer definition:
A *Digital Level Layer* (name coming from *Level sets*) is a subset of Z^d characterized by a double inequality:

$$h \leq f(x) < h'$$

The advantage of DLL is that they are described by double-inequalities:
They can be used in *Vector Graphics* (for zooming or any transformation).

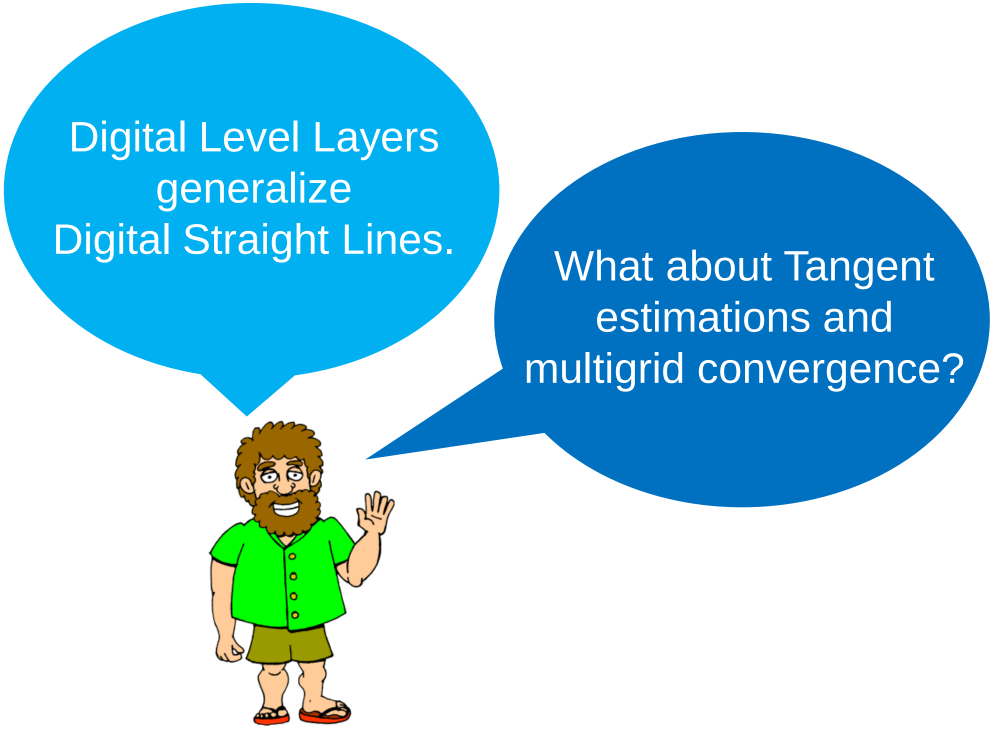


Digital Level Layer (DLL for short)

Digital Level Layer definition:

A *Digital Level Layer* (name coming from *Level sets*) is a subset of Z^d characterized by a double inequality:

$$h \leq f(x) < h'$$



Digital Level Layer definition:

A *Digital Level Layer* (name coming from *Level sets*) is a subset of Z^d characterized by a double inequality:

$$h \leq f(x) < h'$$

Review of multigrid convergent estimators developped in Digital Geometry.

Method	Authors	Assumption on the continuous curve	Order of derivative	Worst case Error bound
Maximal DSS with *thickness=1*	A. Vialard, J-O Lachaud, F De Vieilleville	Locally convex, C^3	$k=1$	$O(h^{1/3})$
Convolutions	S. Fourey, F. Brunet, A. Esbelin, B. R. Malgouyres	C^3 or C^2	Any k	$O(h^{(2/3)^k})$
Maximal DLL with *thickness>1*	L. Provot, Y. Gerard	C^{k+1}	Any k	$O(h^{(1/(k+1))})$

Review of multigrid convergent estimators developped in Digital Geometry.

Maximal DSS with *thickness=1*	A. Vialard, J-O Lachaud, F De Vieilleville	Locally convex, C^3	$k=1$	$O(h^{1/3})$

Parameter free because the parameter i.e the class of digital straight lines has been fixed…

Review of multigrid convergent estimators developped in Digital Geometry.

All approaches are able to deal with noisy shapes (using their parameters).

Maximal DSS with *thickness=1*	A. Vialard, J-O Lachaud, F De Vieilleville	Locally convex, C^3	$k=1$	$O(h^{1/3})$
Convolutions	S. Fourey, F. Brunet, A. Esbelin, B. R. Malgouyres	C^3 or C^2	Any k	$O(h^{(2/3)^k})$
Maximal DLL with *thickness>1*	L. Provot, Y. Gerard	C^{k+1}	Any k	$O(h^{(1/(k+1))})$

Review of multigrid convergent estimators developped in Digital Geometry.

Can be applied on contours of a shape,
not only the graph of a function…

Maximal DSS with *thickness=1*	A. Vialard, J-O Lachaud, F De Vieilleville	Locally convex, C^3	$k=1$	$O(h^{1/3})$
Maximal DLL with *thickness>1*	L. Provot, Y. Gerard	C^{k+1}	Any k	$O(h^{(1/(k+1))})$

Applied on a digital function
$f: Z \rightarrow Z$

Review of multigrid convergent estimators developped in Digital Geometry.

Maximal DSS with *thickness=1*	A. Vialard, J-O Lachaud, F De Vieilleville	Locally convex, C^3	$k=1$	$O(h^{1/3})$
Increase the thickness		weaker assumption	More general	Better worst case convergence rate.
Maximal DLL with *thickness>1*	L. Provot, Y. Gerard	C^{k+1}	Any k	$O(h^{(1/(k+1))})$

Maximal DLL with *thickness=1*

There is the convergence result for $k=1$.
There should exist extensions for $k>1$ under some conditions…

Review of multigrid convergent estimators developped in Digital Geometry.

Iterative version with deleting and points insertion for computing the derivative along a curve. It remains linear.

Computation in worst case linear time for a single DLL.

Maximal DSS with *thickness=1*	A. Vialard, J-O Lachaud, F De Vieilleville	Locally convex, C^3	$k=1$	$O(h^{1/3})$

Maximal DLL with *thickness>1*	L. Provot, Y. Gerard	C^{k+1}	Any k	$O(h^{(1/(k+1))})$

Computation in $O(n^{2(k+1)})$ in theory but close to linear time in practice for a single DLL.

No iterative version with deleting and points insertion for computing the derivative along a curve. It becomes quadratic.

Review of multigrid convergent estimators developped in Digital Geometry.

More restrictive and less accurate but faster…

Maximal DSS with *thickness=1*	A. Vialard, J-O Lachaud, F De Vieilleville	Locally convex, C^3	$k=1$	$O(h^{1/3})$
Maximal DLL with *thickness>1*	L. Provot, Y. Gerard	C^{k+1}	Any k	$O(h^{(1/(k+1))})$

Review of multigrid convergent estimators developped in Digital Geometry.

Maximal DLL with *thickness>1*	L. Provot, Y. Gerard	C^{k+1}	Any k	$O(h^{(1/(k+1))})$

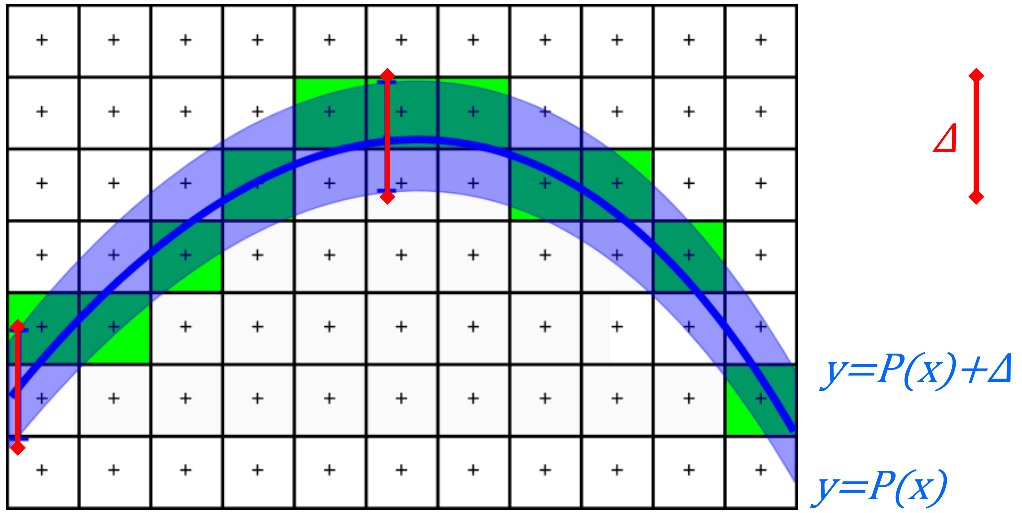
How does it work ?

We use DLL with double inequality:

$$P(x) \leq y < P(x)+\Delta$$

with a fixed $\Delta>1$ and a chosen maximal degree k for $P(x)$.

If we choose a high Δ, it allows more noise, but becomes less precise.

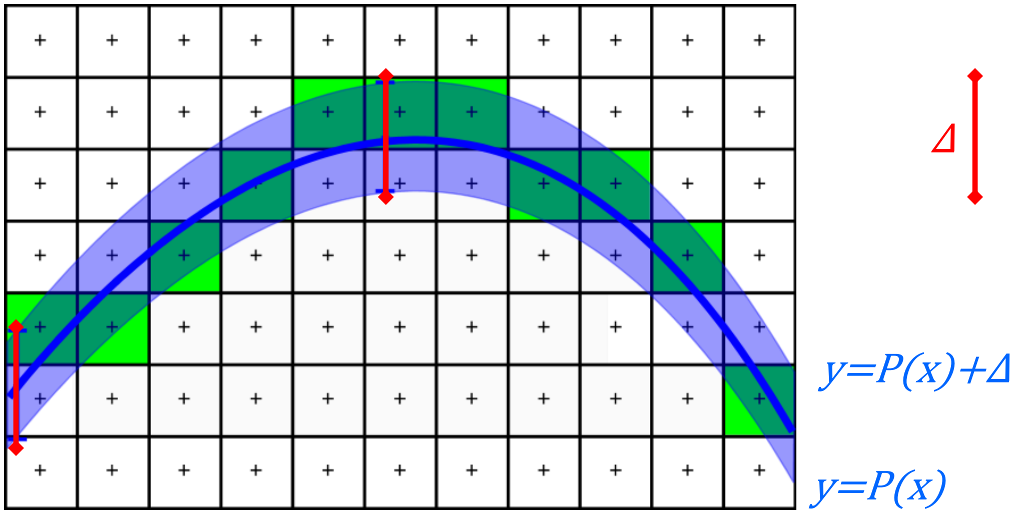


How does it work ?

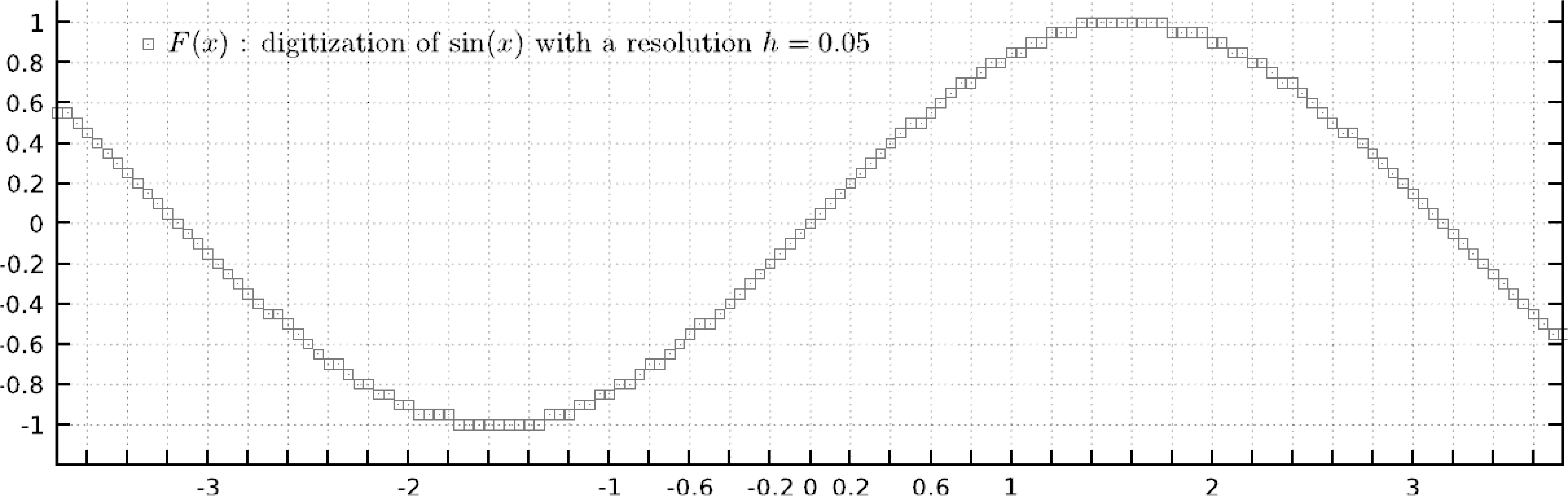
We use DLL with double inequality:

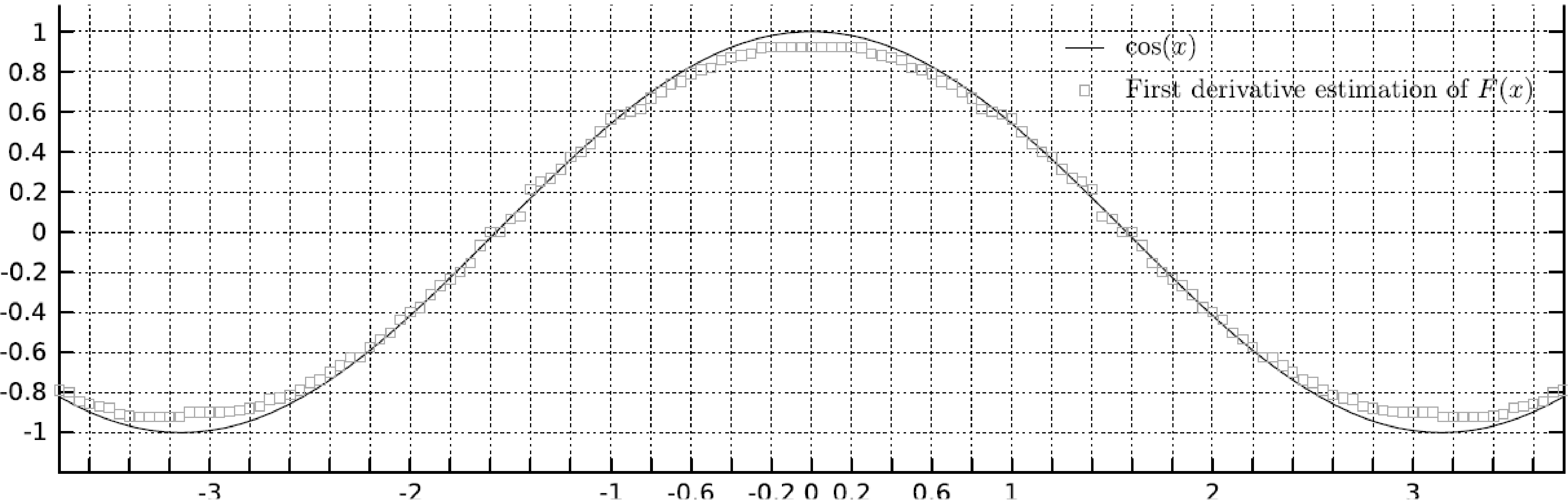
$$P(x) \leq y < P(x)+\Delta$$

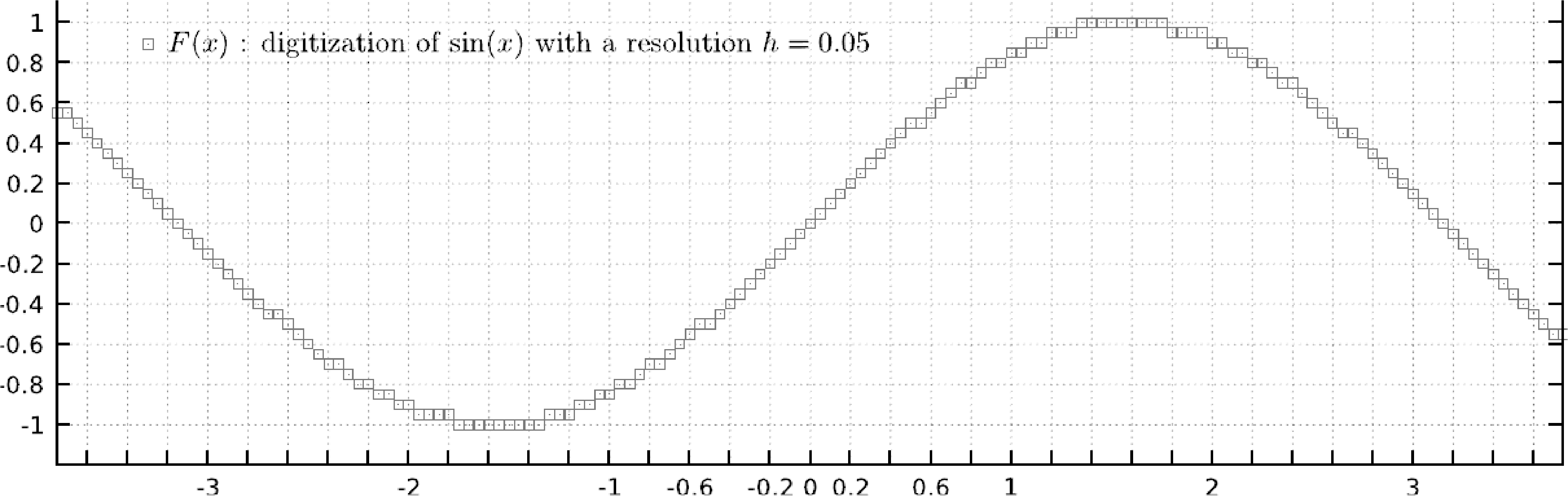
with a fixed $\Delta > 1$ and a chosen maximal degree k for $P(x)$.
If we choose a high Δ, it allows more noise, but becomes less precise.



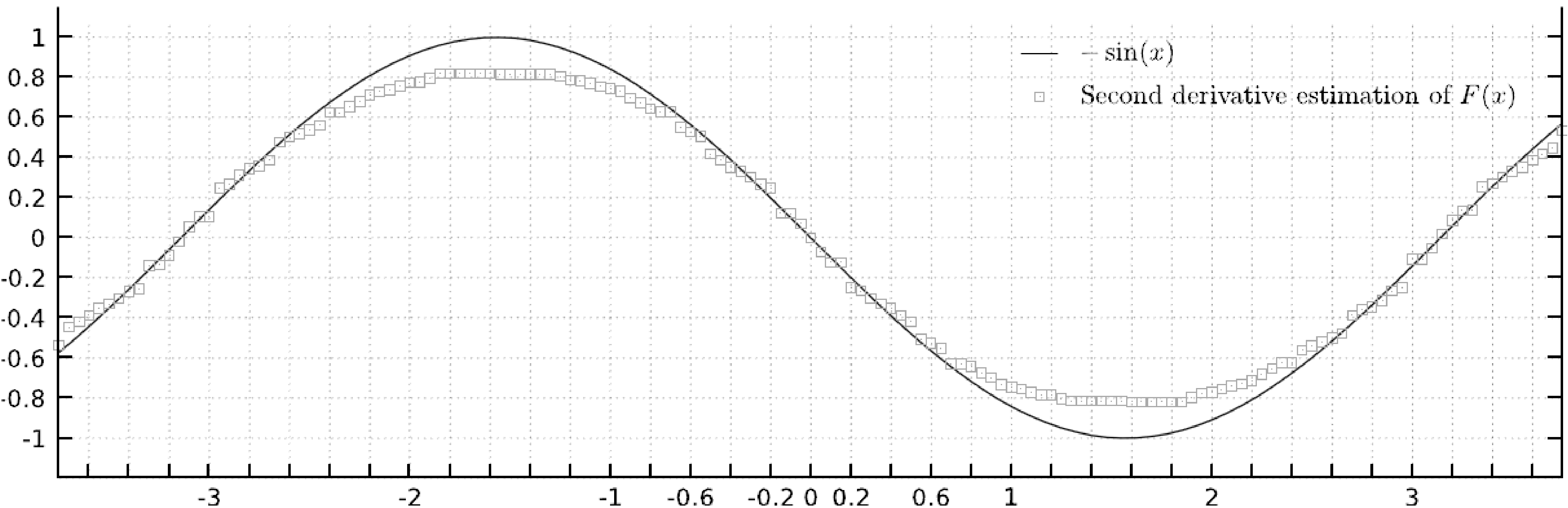
$P(x)$ provides directly the derivative of order k.


$F(x)$: digitization of $\sin(x)$ with a resolution $h = 0.05$
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-3
-2
-1
-0.6
-0.2
0
0.2
0.6
1
2
3

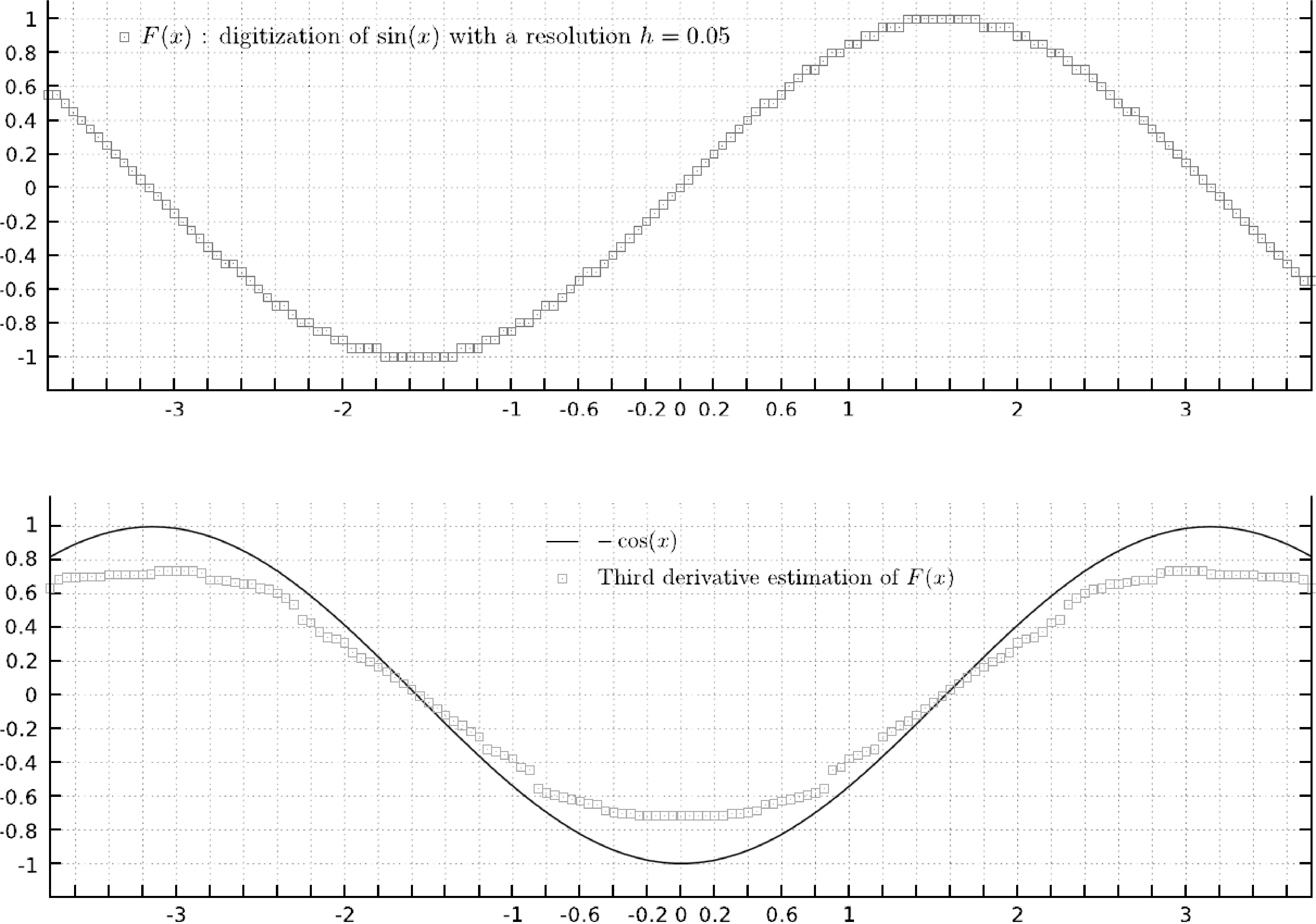

$\cos(x)$
First derivative estimation of $F(x)$
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-3
-2
-1
-0.6
-0.2
0
0.2
0.6
1
2
3



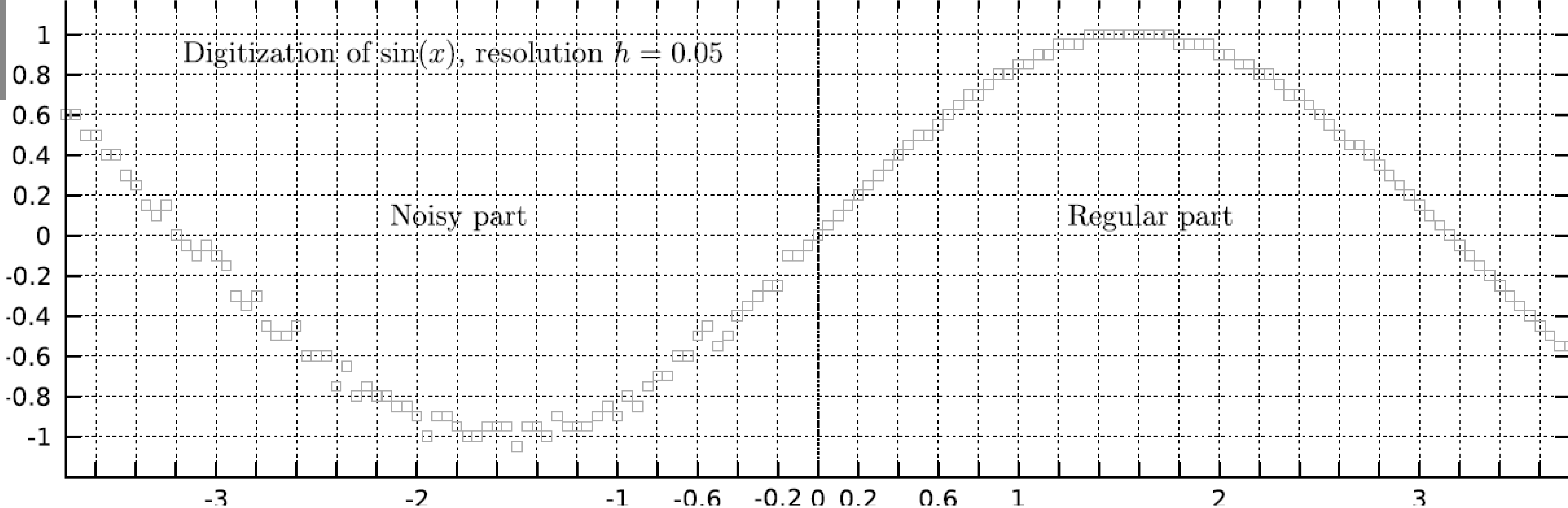
F(x) : digitization of sin(x) with a resolution h = 0.05
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-3
-2
-1
-0.6
-0.2
0
0.2
0.6
1
2
3

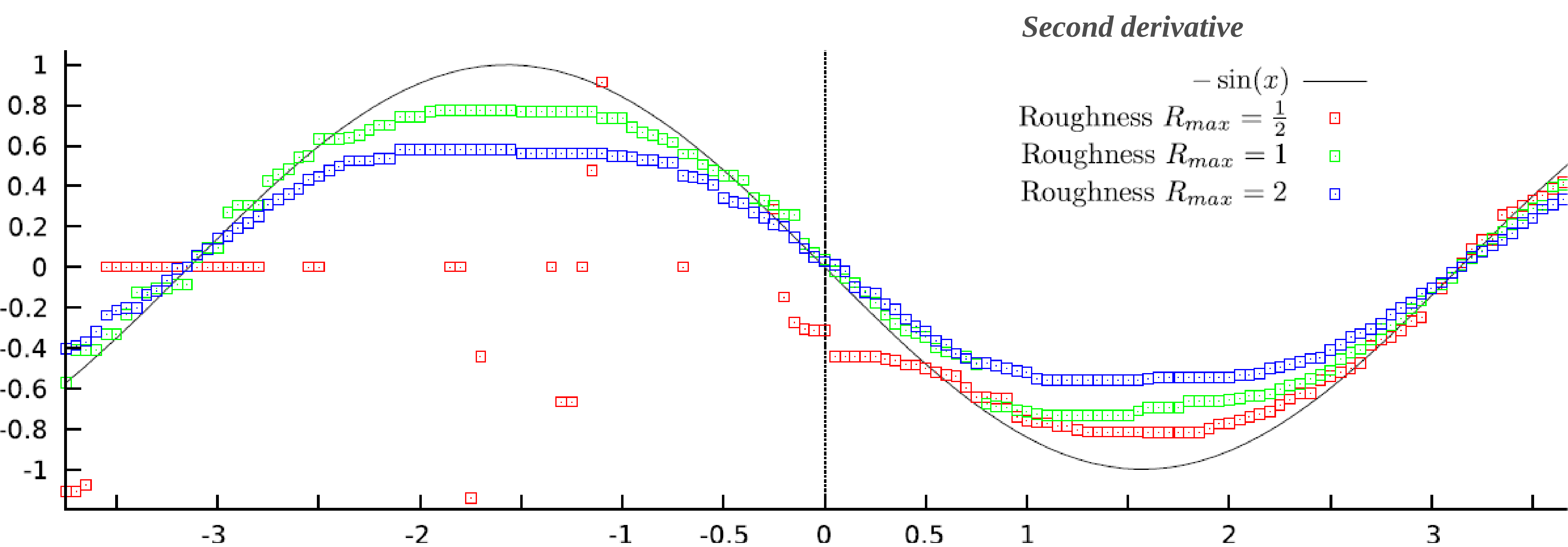


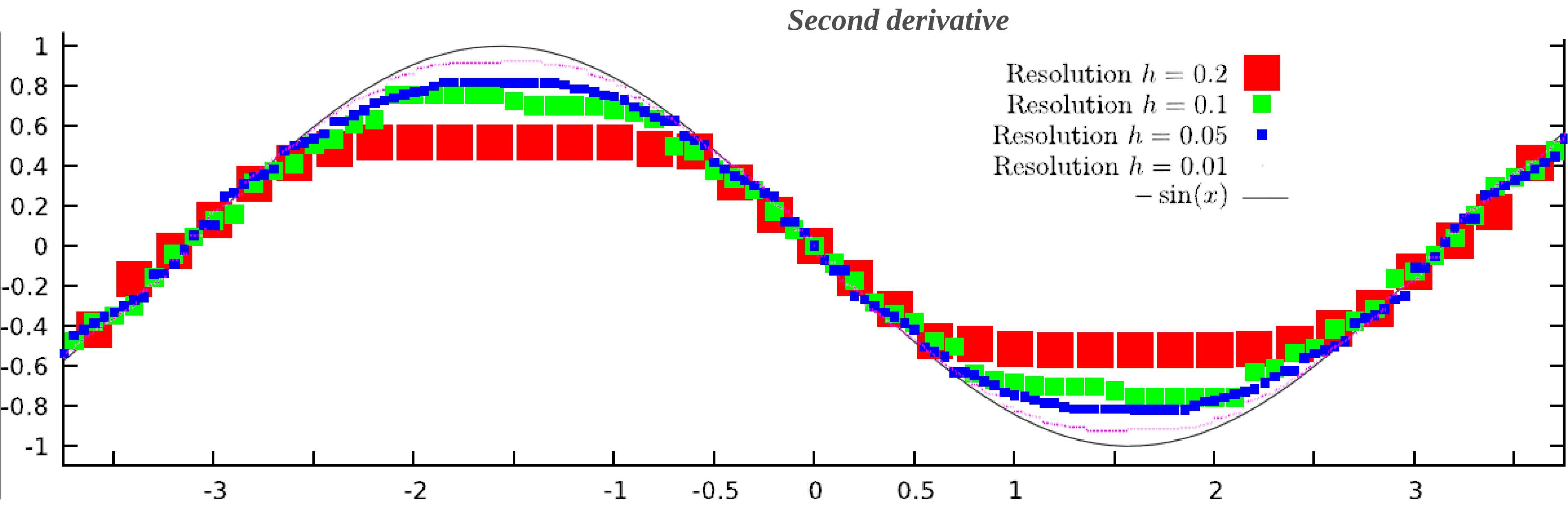
− sin(x)
Second derivative estimation of F(x)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-3
-2
-1
-0.6
-0.2
0
0.2
0.6
1
2
3



$F(x)$: digitization of $\sin(x)$ with a resolution $h = 0.05$
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-3
-2
-1
-0.6
-0.2
0
0.2
0.6
1
2
3
$-\cos(x)$
Third derivative estimation of $F(x)$


Digitization of sin(x), resolution $h = 0.05$
Noisy part
Regular part


Second derivative
$-\sin(x)$
Roughness $R_{max} = \frac{1}{2}$
Roughness $R_{max} = 1$
Roughness $R_{max} = 2$


Second derivative
Resolution $h = 0.2$
Resolution $h = 0.1$
Resolution $h = 0.05$
Resolution $h = 0.01$
$-\sin(x)$
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-3
-2
-1
-0.5
0
0.5
1
2
3

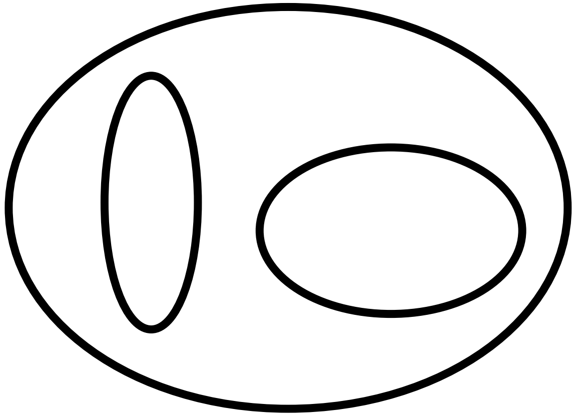
Plan

- I — Introduction
- II — About tangent estimators
- III — Digital Level Layers
- IV — DLL decomposition
- V — Algorithm

Plan

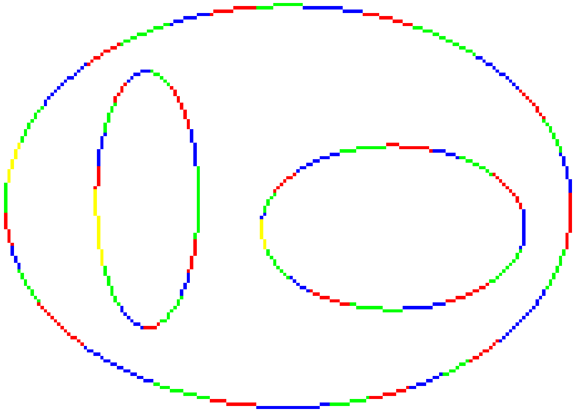
IV

DLL decomposition



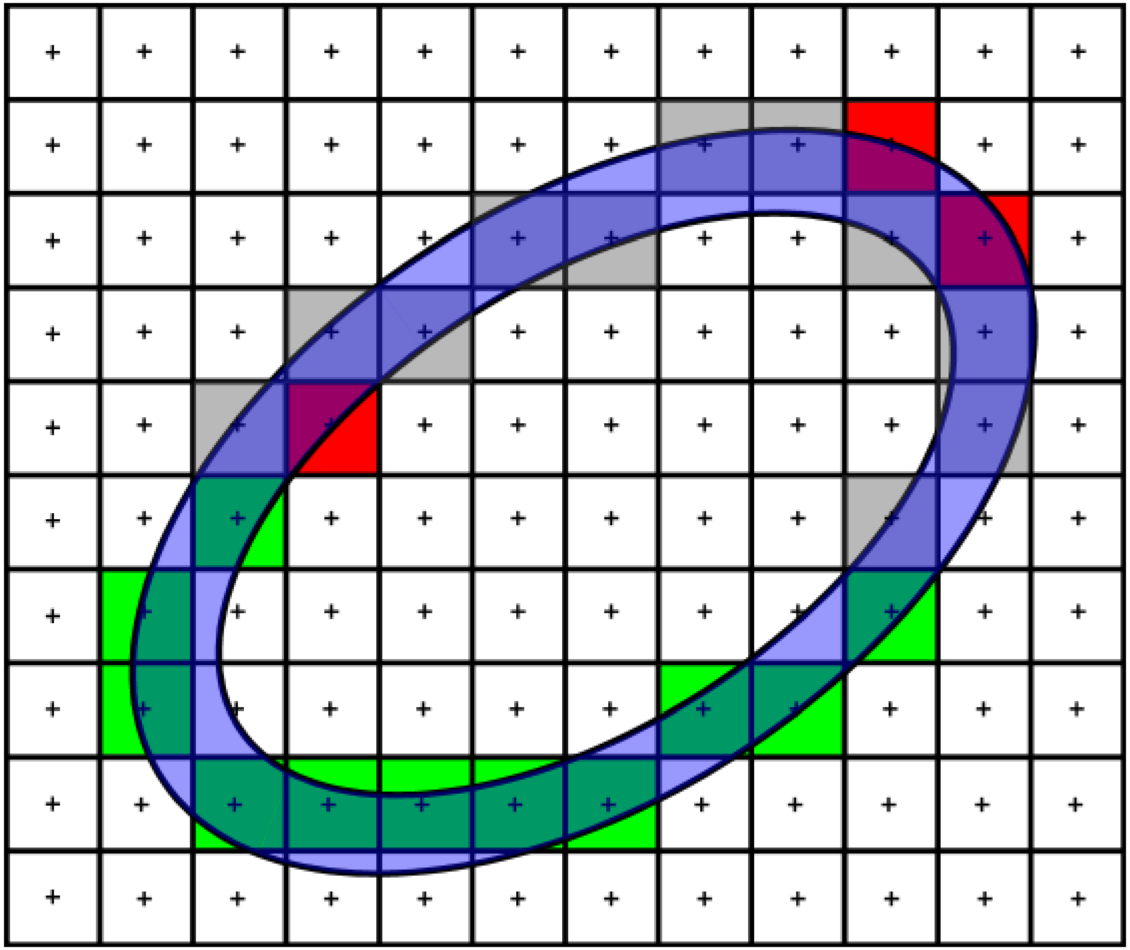
Input: A digital curve

Segmentation in pieces of
digital straight lines
(72 pieces)



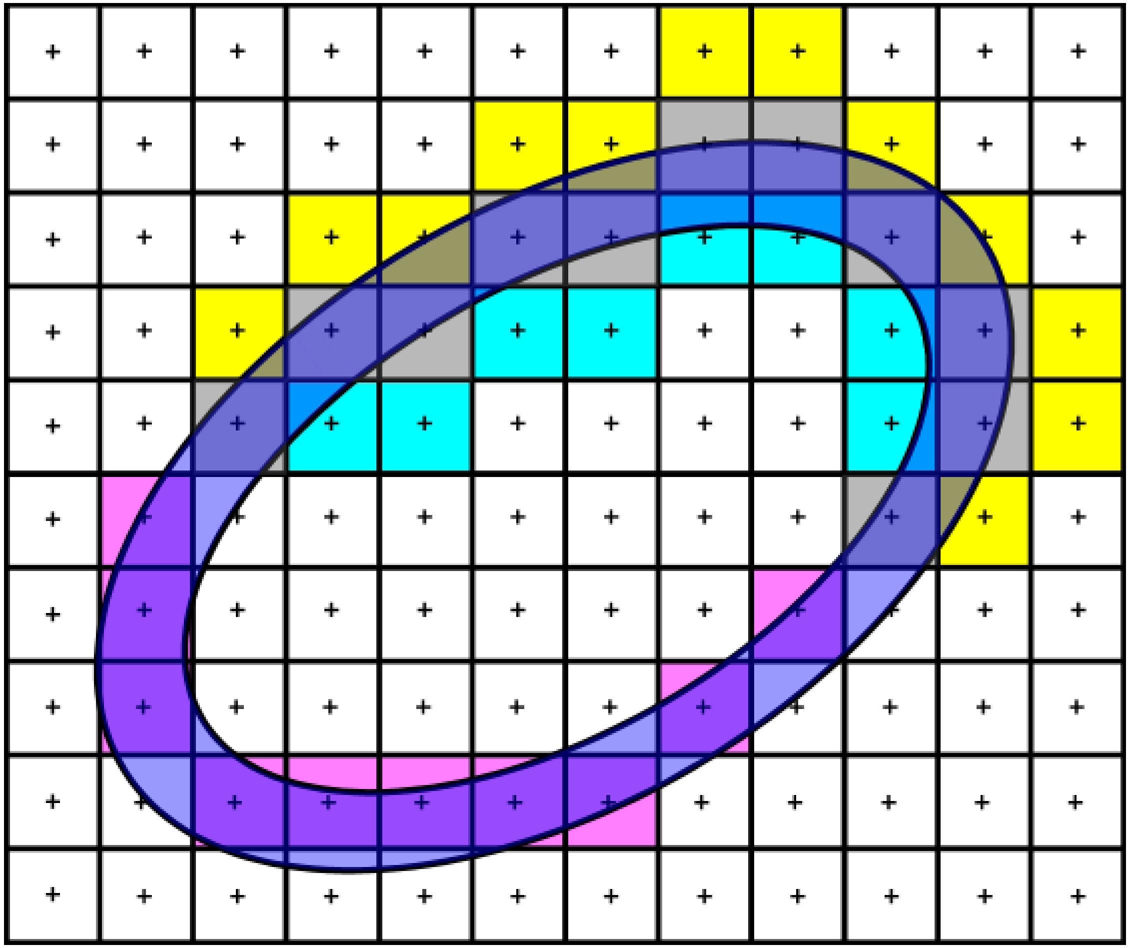
Output: Its decomposition in Digital Straigh Segments

Principle :



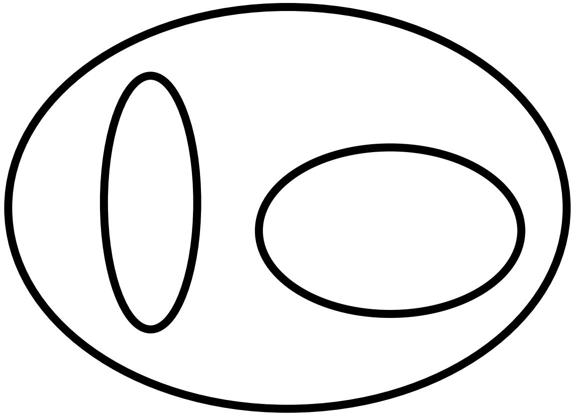
Input: Lattice set S → Recognition → DLL containing S → Digitization → Undesired neighbors

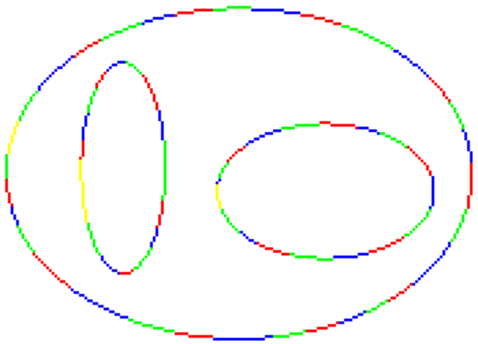
Principle :



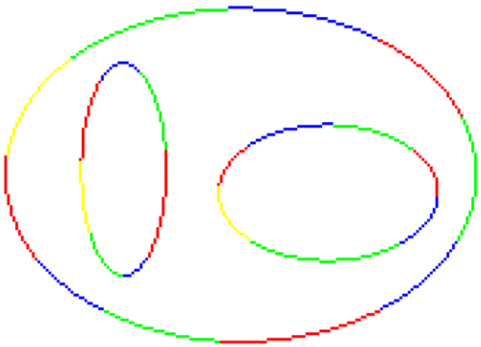
Input: Lattice set S → Recognition → DLL containing S → Digitization → Undesired neighbors

\+ Forbidden neighbors → Recognition → DLL between the inliers and outliers

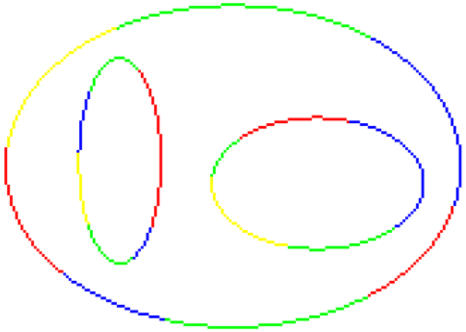




Segmentation in pieces of digital straight lines (72 pieces)

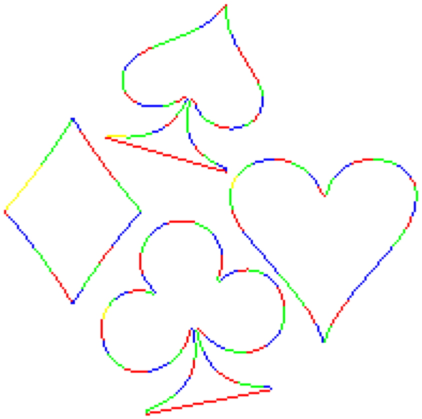


Segmentation in pieces of digital circles (DLL) (24 pieces)

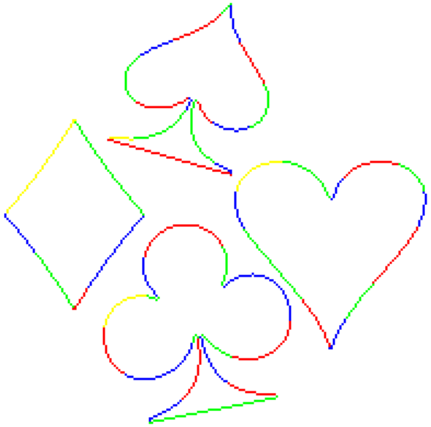


Segmentation in pieces of digital conics (DLL) (18 pieces)

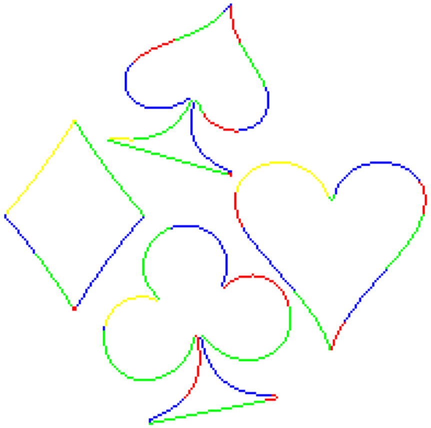
We decompose the digital curve in Digital Level Layers (DLL)



Segmentation in pieces of digital straight lines (116 pieces)



Segmentation in pieces of digital circles (DLL) (50 pieces)

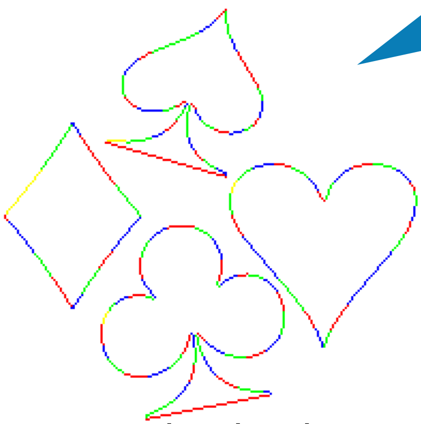


Segmentation in pieces of digital conics (DLL) (42 pieces)

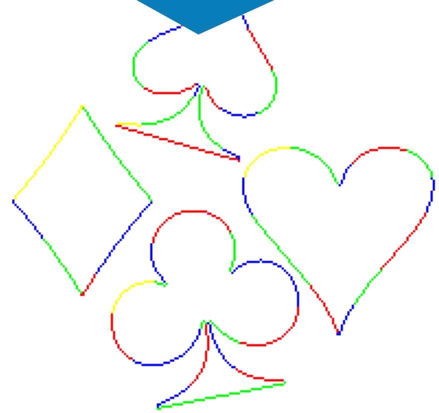
We decompose the digital curve in Digital Level Layers (DLL)

It provides a vector description of a digital curve wich is smoother than DSS.

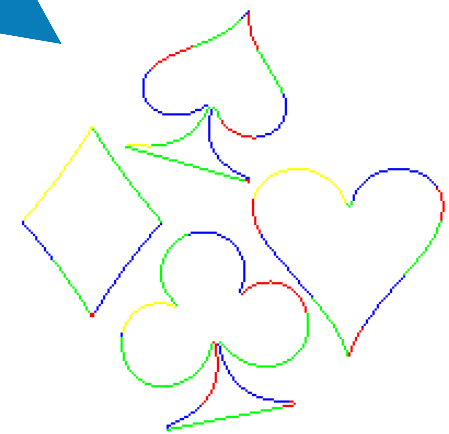
All cases computed with a UNIQUE algorithm
(with, as parameter, a chosen basis of polynomials like in SVM)



Segmentation in pieces of digital straight lines (116 pieces)

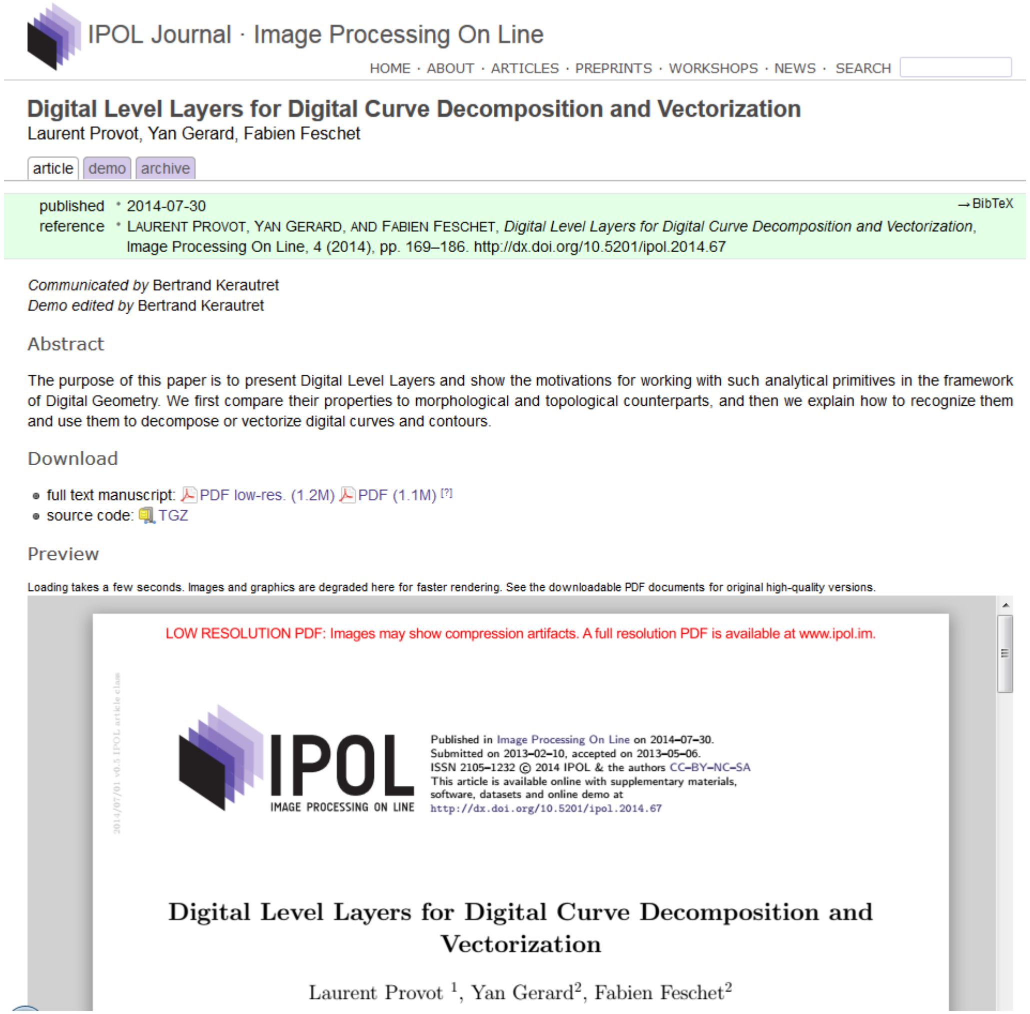


Segmentation in pieces of digital circles (DLL) (50 pieces)



Segmentation in pieces of digital conics (DLL) (42 pieces)

Paper, Demo and code are available on IPOL *(thanks to Bertrand Kerautret)*



IPOL Journal · Image Processing On Line

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

Digital Level Layers for Digital Curve Decomposition and Vectorization

Laurent Provot, Yan Gerard, Fabien Feschet

article | demo | archive

published · 2014-07-30 →BibTeX

reference · LAURENT PROVOT, YAN GERARD, AND FABIEN FESCHET, *Digital Level Layers for Digital Curve Decomposition and Vectorization*, Image Processing On Line, 4 (2014), pp. 169–186. http://dx.doi.org/10.5201/ipol.2014.67

Communicated by Bertrand Kerautret
Demo edited by Bertrand Kerautret

Abstract

The purpose of this paper is to present Digital Level Layers and show the motivations for working with such analytical primitives in the framework of Digital Geometry. We first compare their properties to morphological and topological counterparts, and then we explain how to recognize them and use them to decompose or vectorize digital curves and contours.

Download

- full text manuscript: PDF low-res. (1.2M) PDF (1.1M) [?]
- source code: TGZ

Preview

Loading takes a few seconds. Images and graphics are degraded here for faster rendering. See the downloadable PDF documents for original high-quality versions.

LOW RESOLUTION PDF: Images may show compression artifacts. A full resolution PDF is available at www.ipol.im.

IPOL IMAGE PROCESSING ON LINE

Published in Image Processing On Line on 2014–07–30.
Submitted on 2013–02–10, accepted on 2013–05–06.
ISSN 2105–1232 © 2014 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials, software, datasets and online demo at
http://dx.doi.org/10.5201/ipol.2014.67

Digital Level Layers for Digital Curve Decomposition and Vectorization

Laurent Provot [1], Yan Gerard[2], Fabien Feschet[2]

Plan

I Introduction

II About tangent estimators

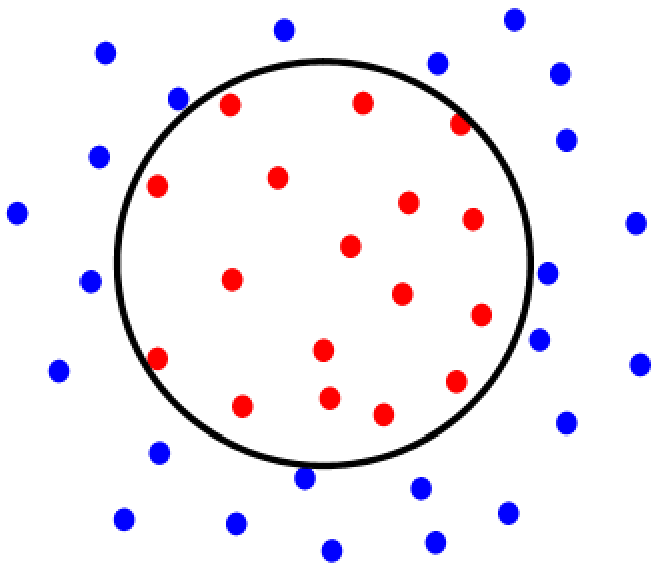
III Digital Level Layers

IV DLL decomposition

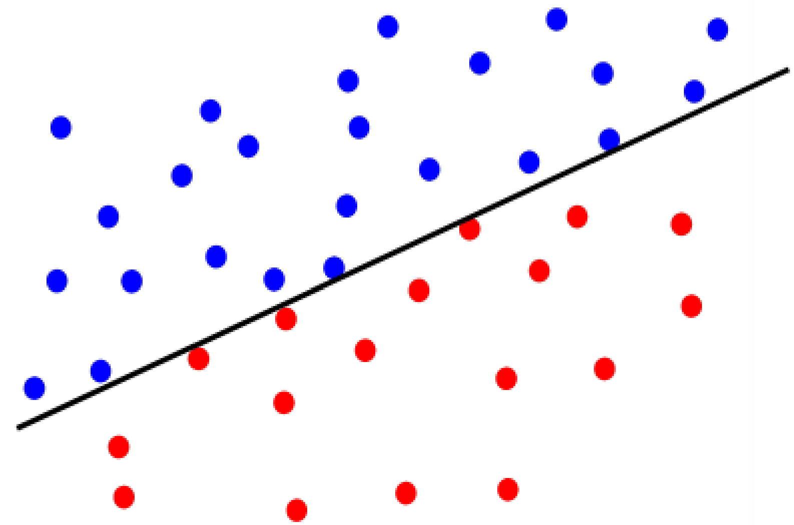
V Algorithm

Plan

V Algorithm

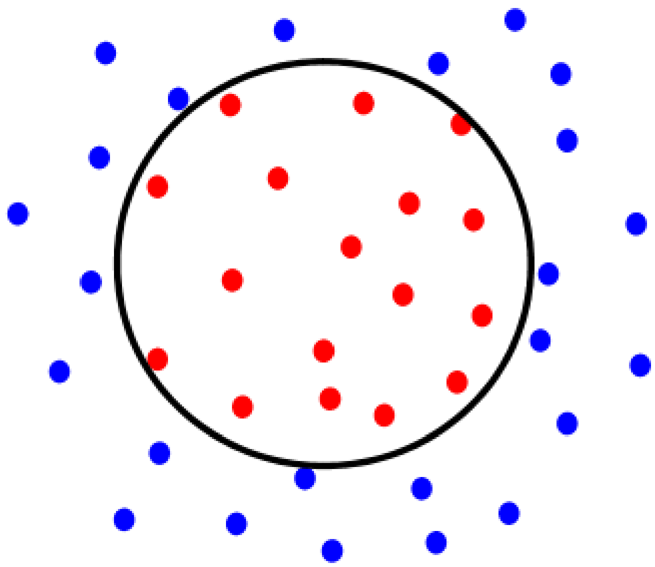


Problem of separation
by a level set *$f(x)=0$*
with *f* in a given linear space

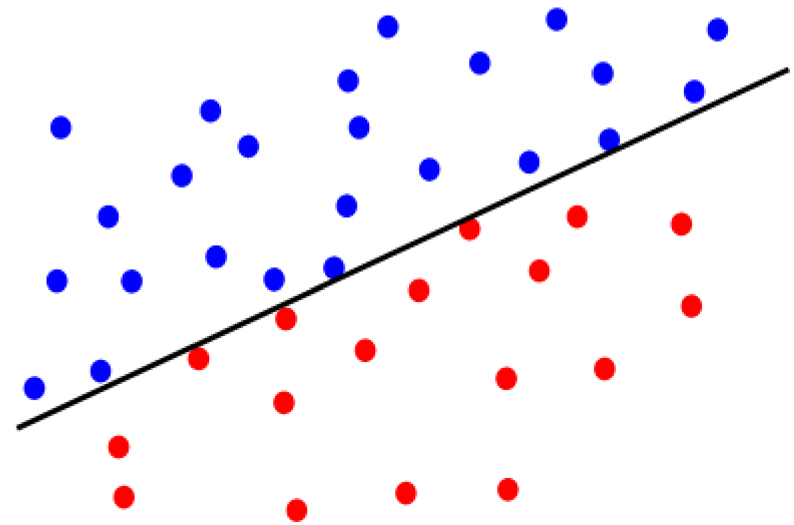


Problem of linear separability
in a descriptive space of higher dimension

Kernel trick (Aïzerman et al. 1964) is the principle of *Support Vector Machines*.

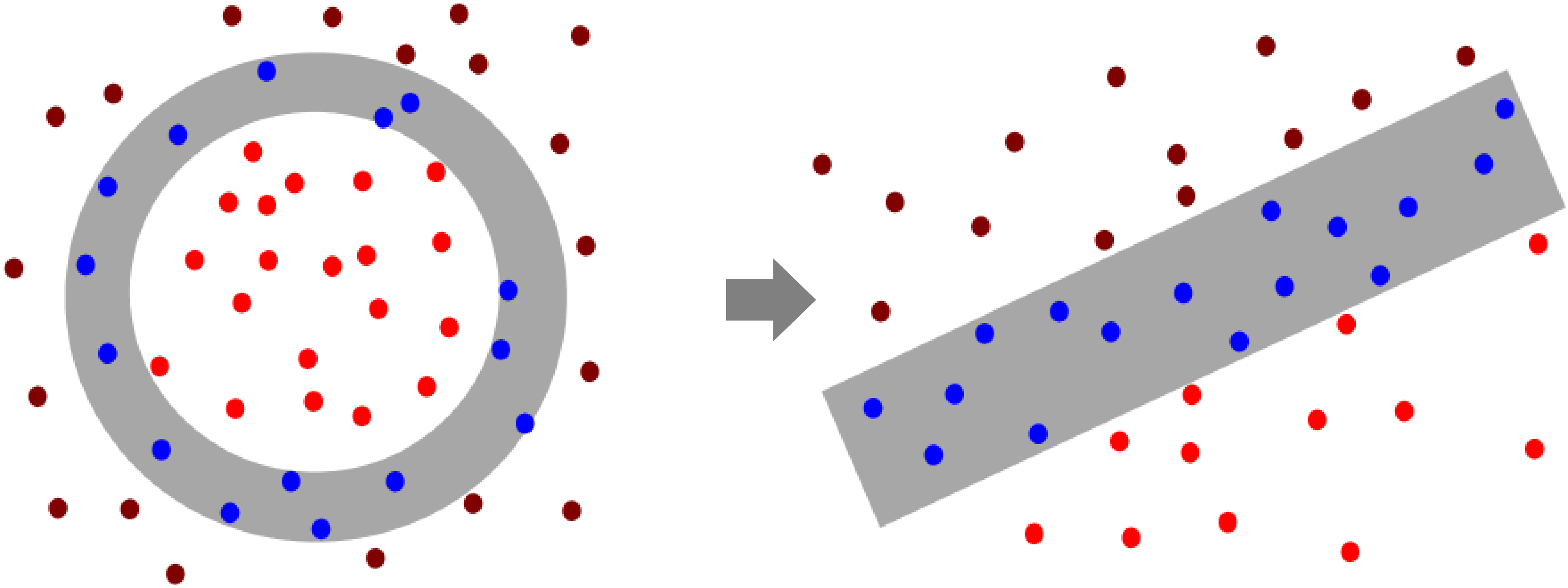


Problem of separation
by a level set $f(x)=0$
with f in a given linear space



Problem of linear separability
in a descriptive space of higher dimension

GJK (*Gilbert Johnson Keerthi*, 1988) computes
the closest pair of points from the two convex hulls.
It's widely used for collision detection.

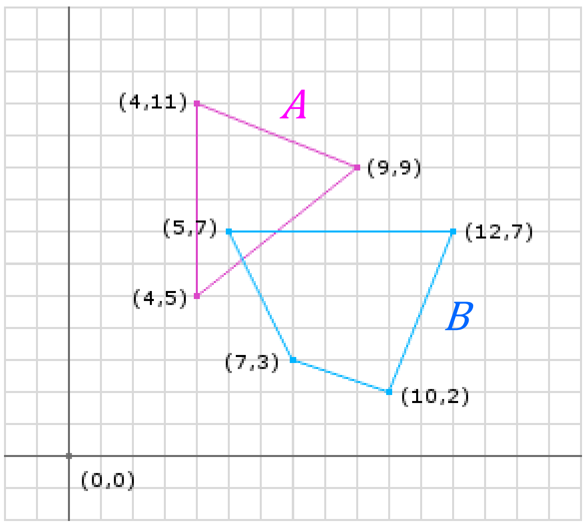


Problem of separation
by two level sets $f(x)=h$ and $f(x)=h'$
with f in a given linear space

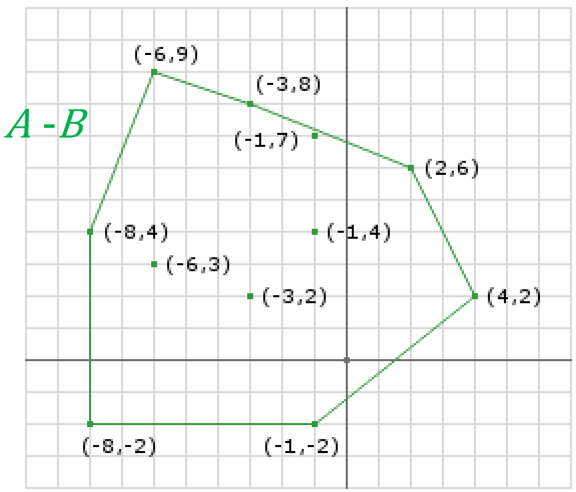
Problem of linear separability
by two parallel hyperplanes
We introduce a variant of GJK in *nD*

GJK (*Gilbert Johnson Keerthi*, 1988) computes
the closest pair of points from the two convex hulls.
It's widely used for collision detection.

Input : two polytopes $A \subset R^d$ and $B \subset R^d$ given by their vertices.
Question : do they intersect ?
More general question: compute their minimal distance.



A and B

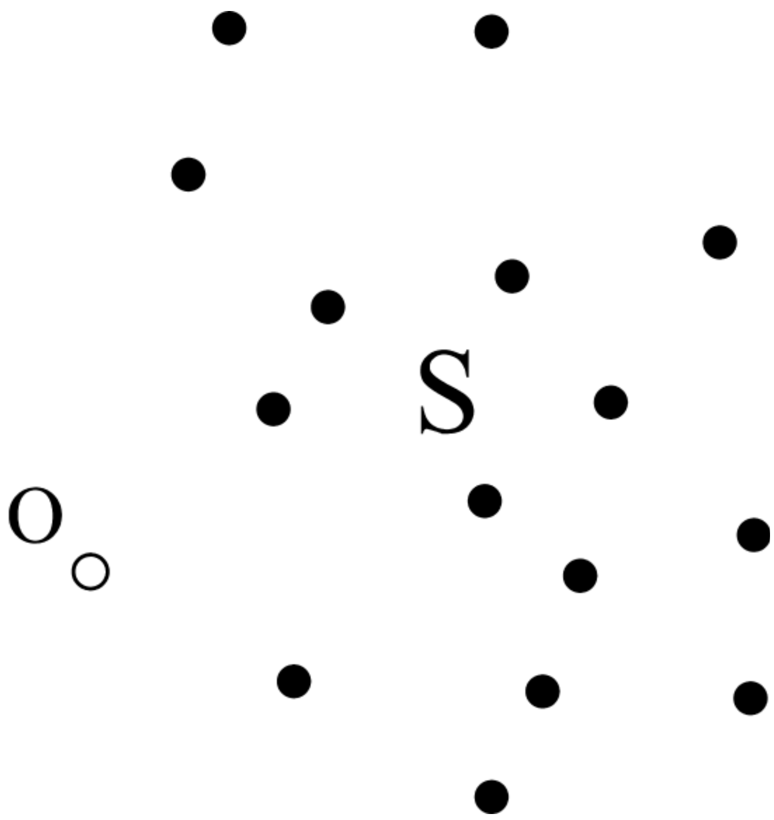


Difference $A - B$

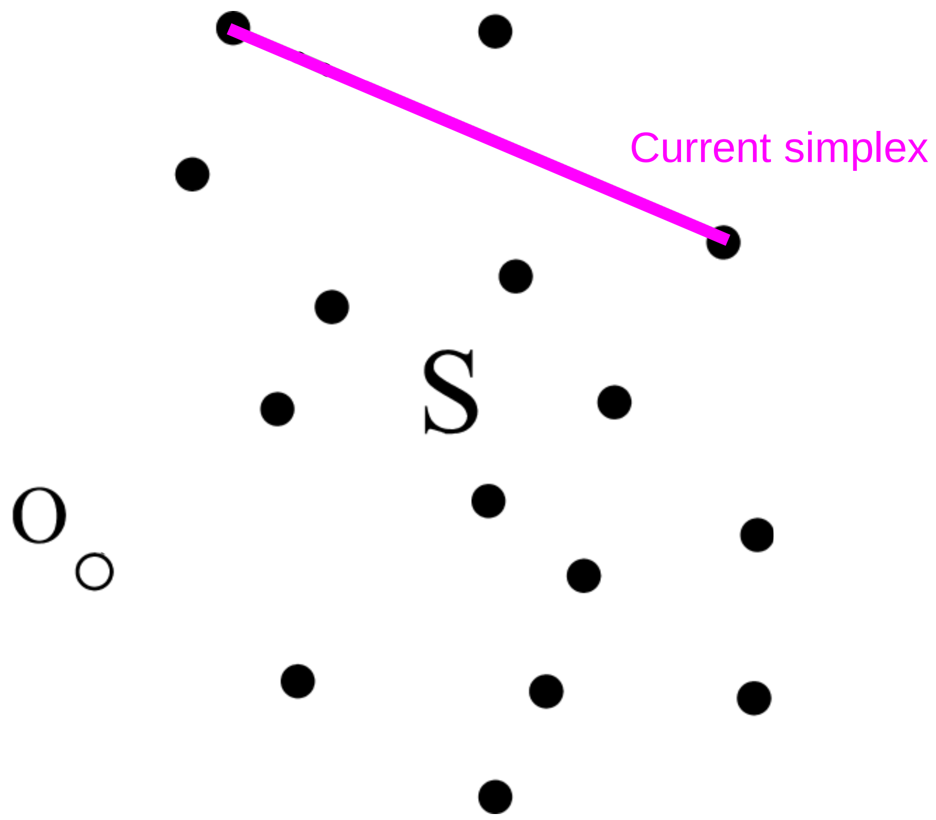
distance (A,B)=distance(0,A-B)

Principe of GJK algorithm :
compute the distance between the origin O and B-A.

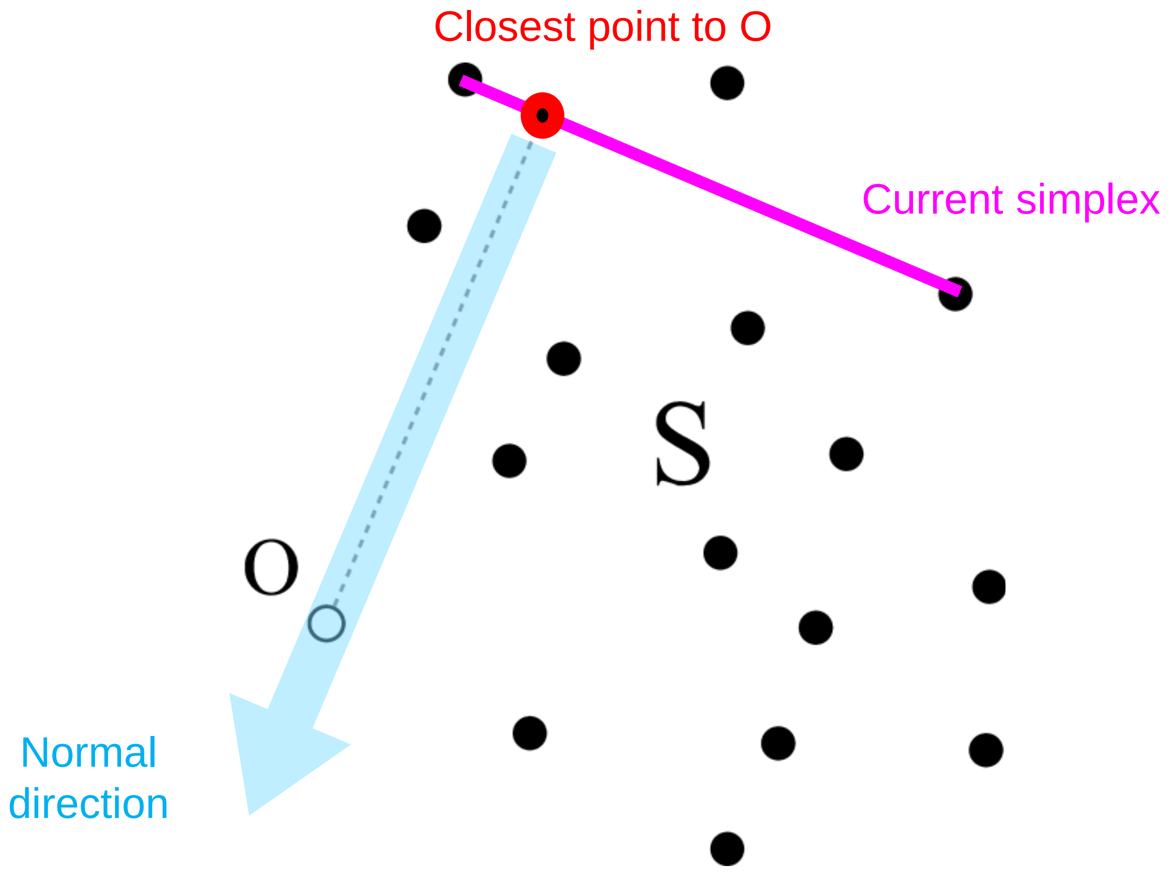
Principe of GJK algorithm :
compute the distance between the origin *O* and *B-A*.



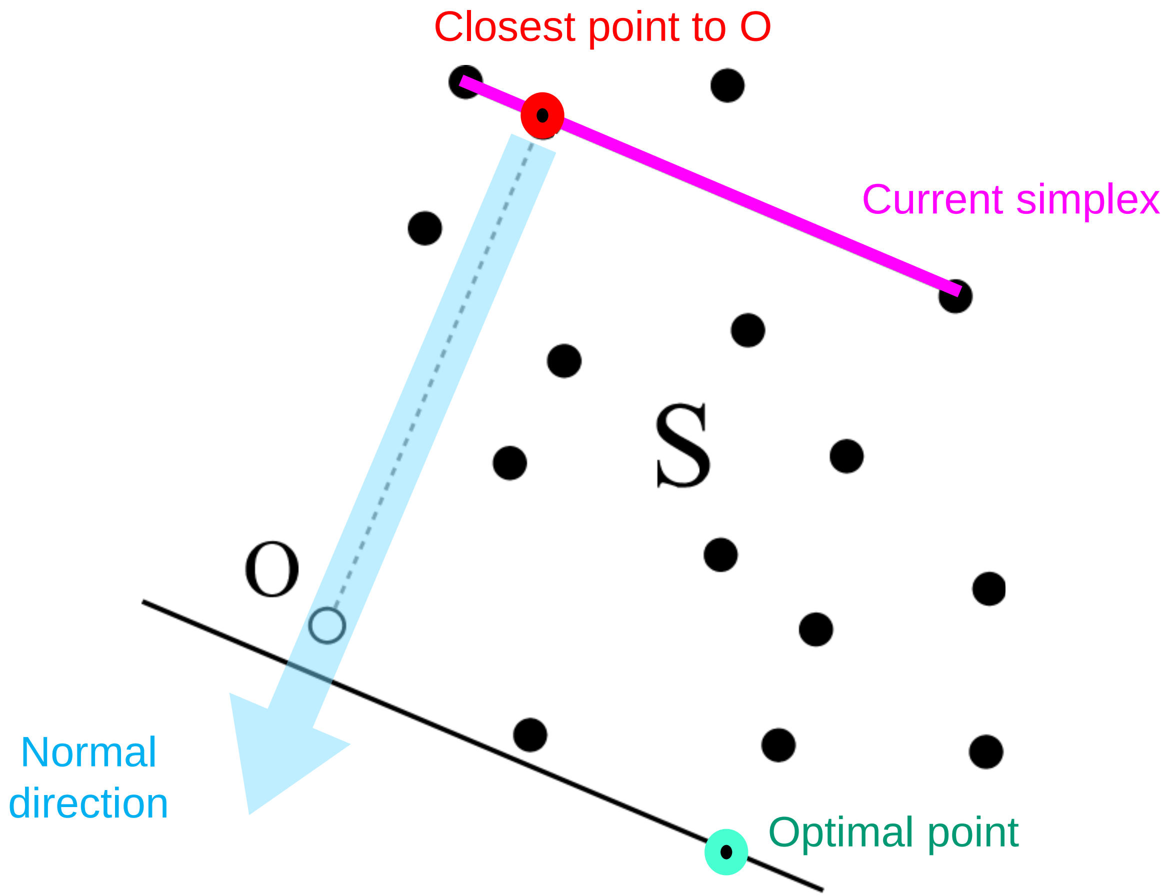
Principe of GJK algorithm :
compute the distance between the origin *O* and *B-A*.



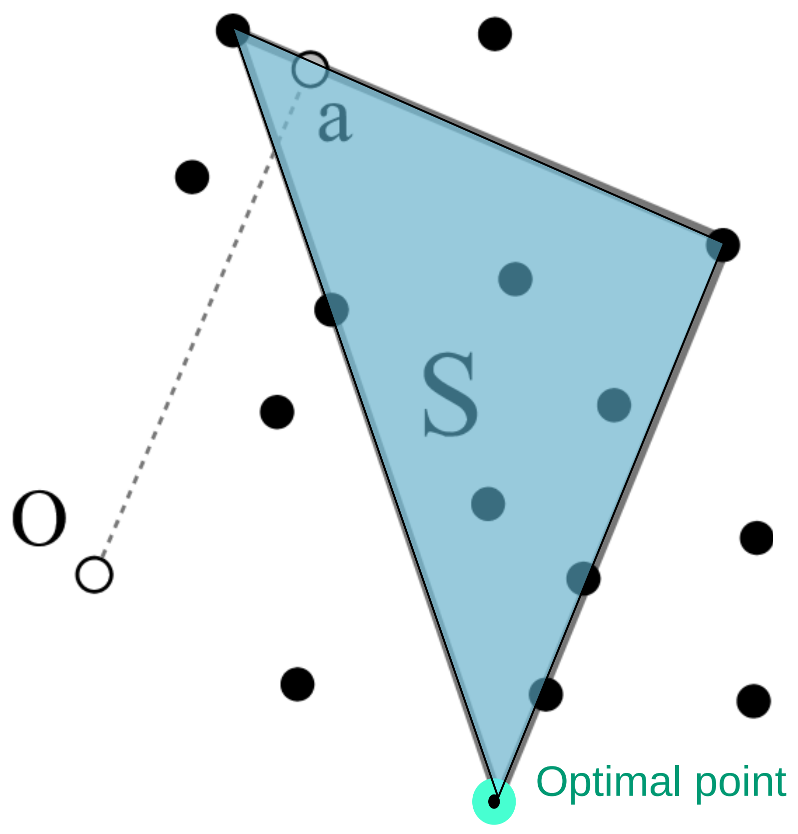
Principe of GJK algorithm :
compute the distance between the origin *O* and *B-A*.



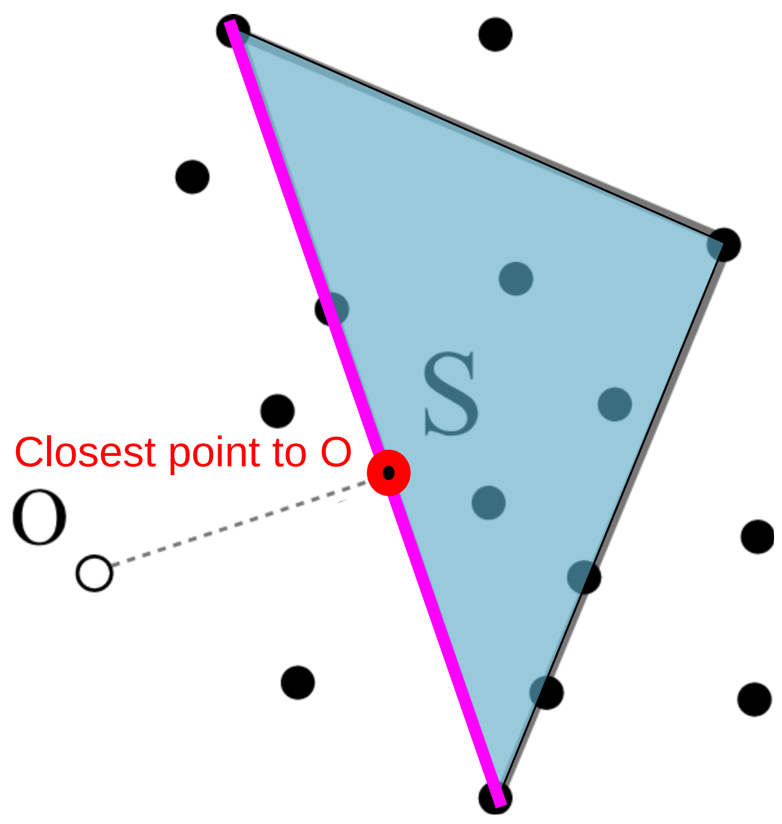
Principe of GJK algorithm :
compute the distance between the origin *O* and *B-A*.



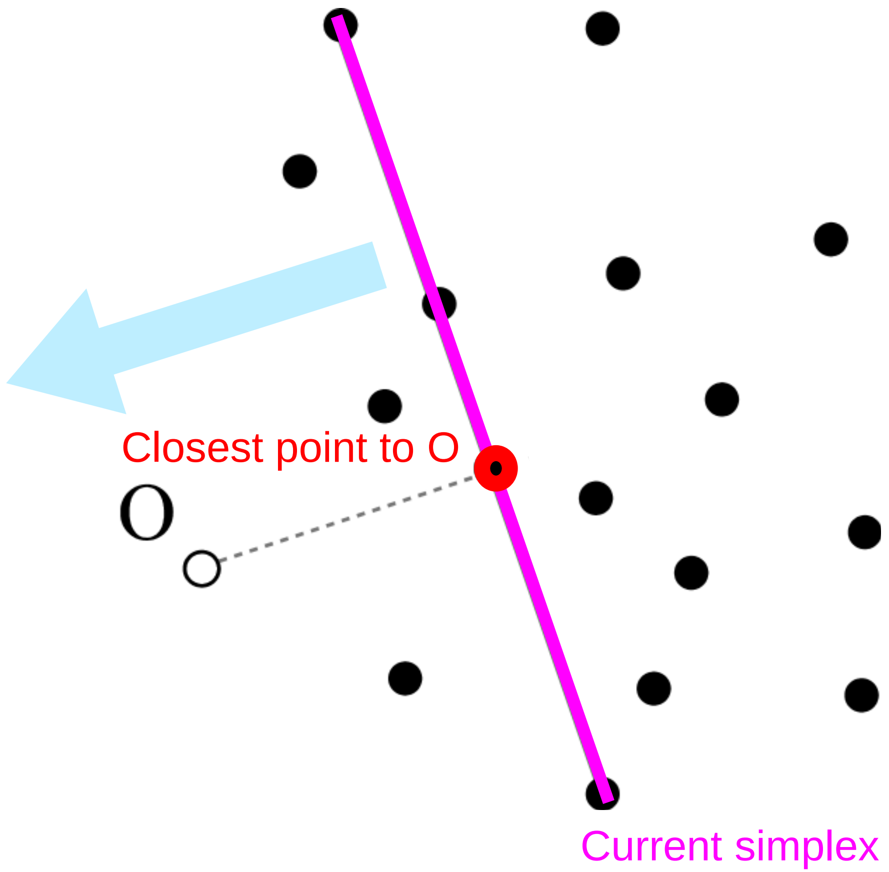
Principe of GJK algorithm :
compute the distance between the origin O and $B\text{-}A$.



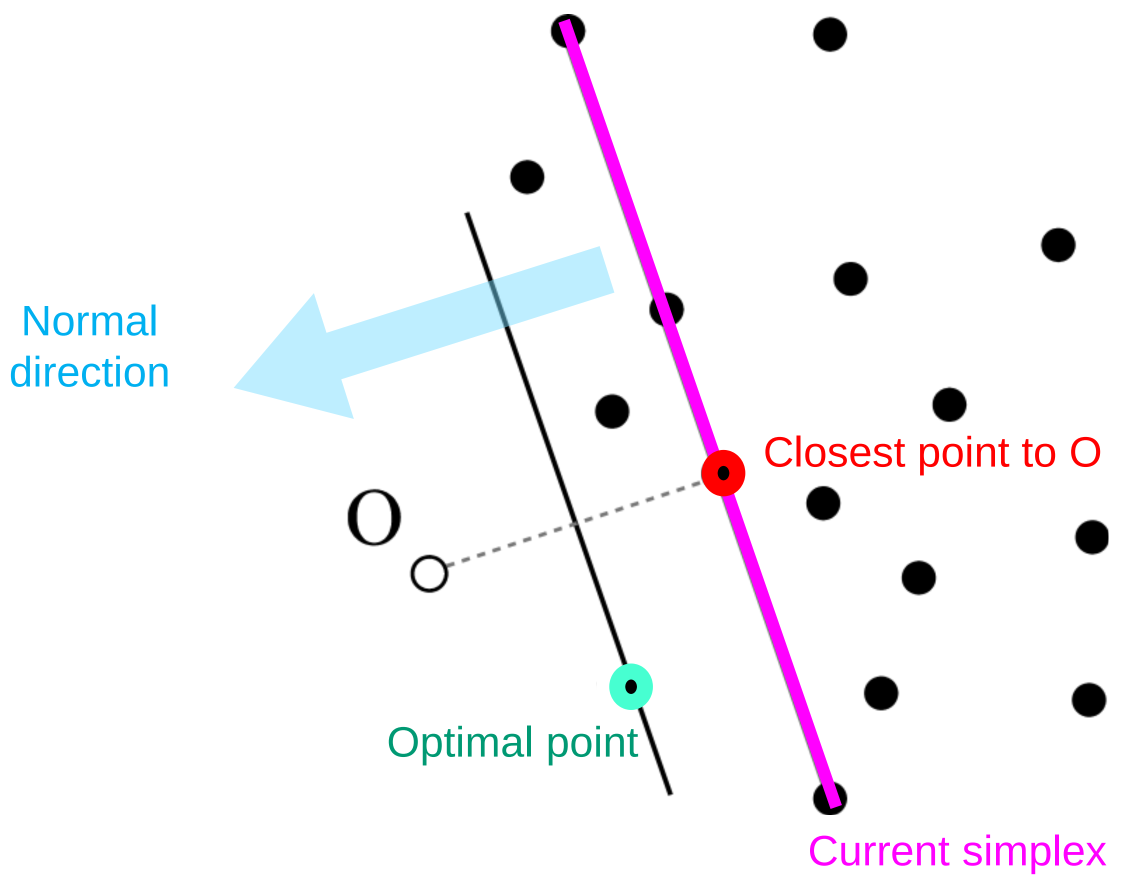
Principe of GJK algorithm :
compute the distance between the origin O and $B\text{-}A$.



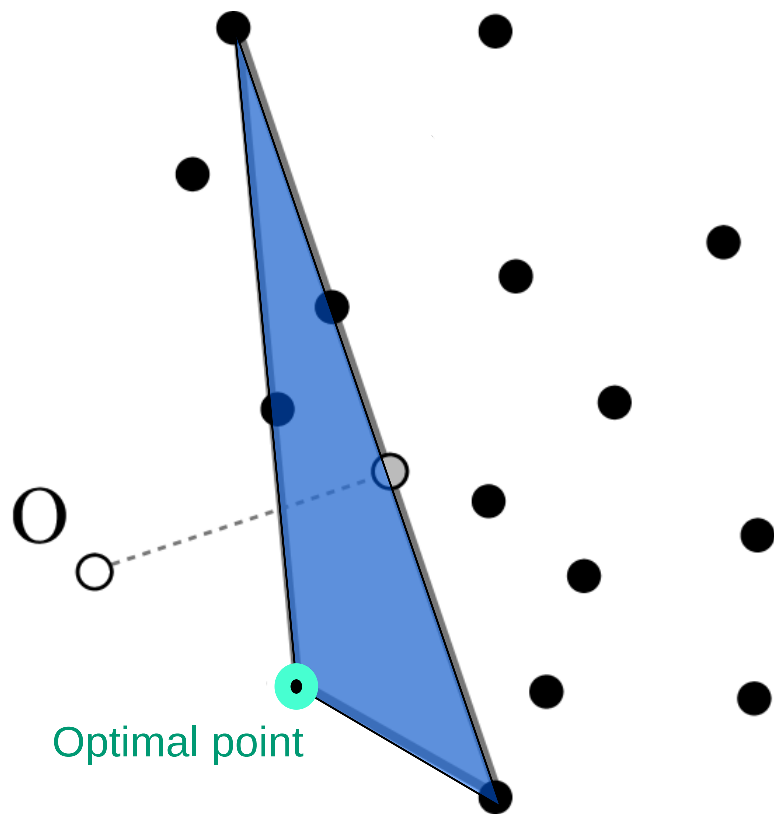
Principe of GJK algorithm :
compute the distance between the origin *O* and *B-A*.



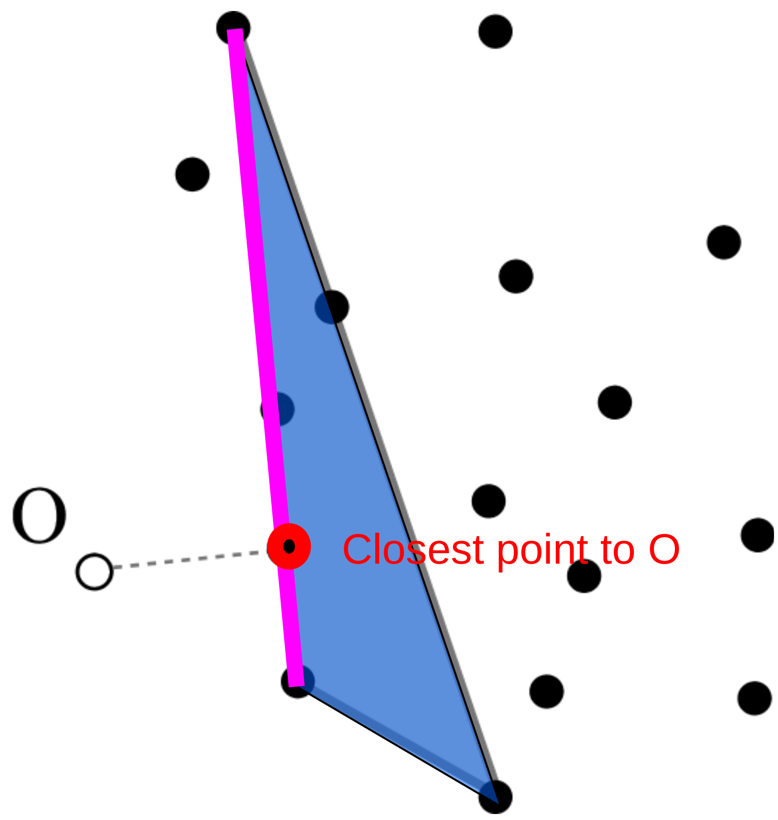
Principe of GJK algorithm :
compute the distance between the origin O and B-A.



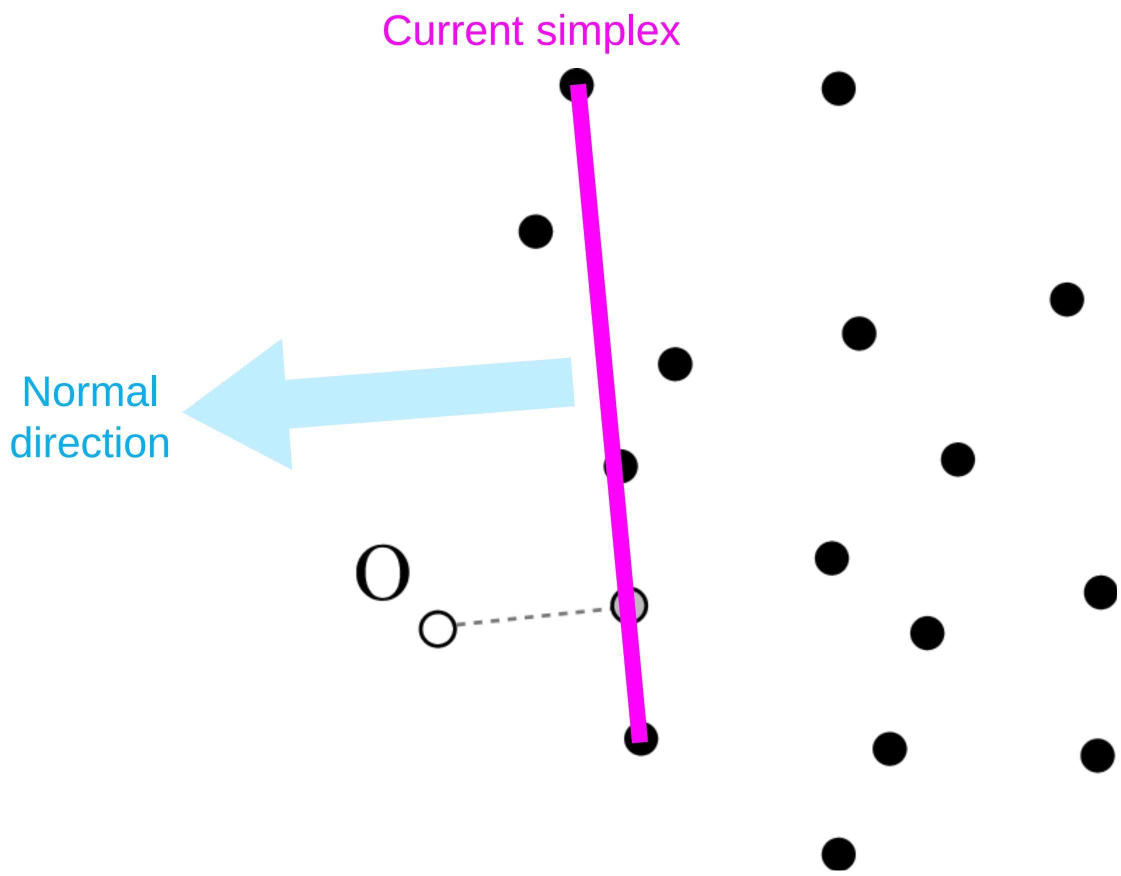
Principe of GJK algorithm :
compute the distance between the origin *O* and *B-A*.



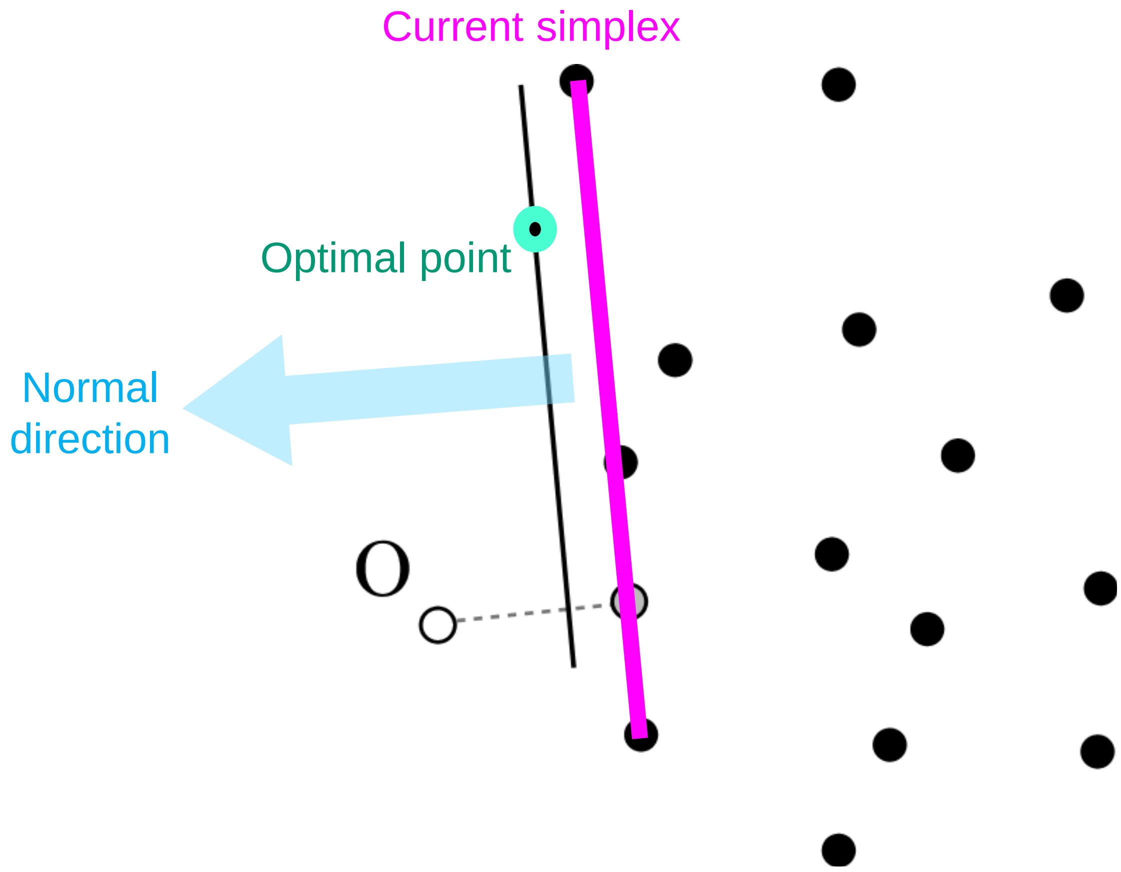
Principe of GJK algorithm :
compute the distance between the origin O and $B\text{-}A$.



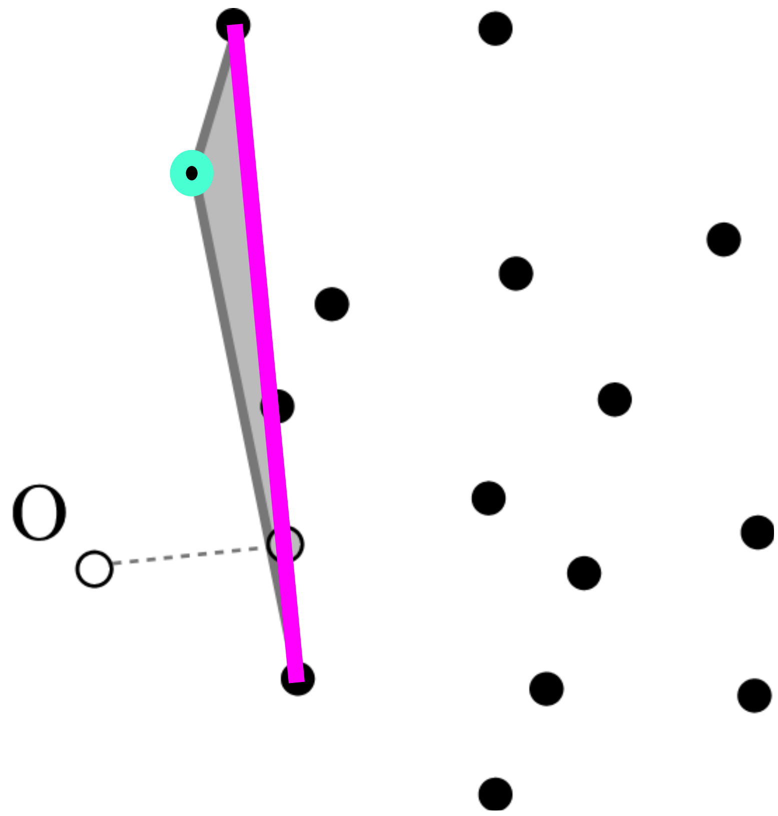
Principe of GJK algorithm :
compute the distance between the origin *O* and *B-A*.



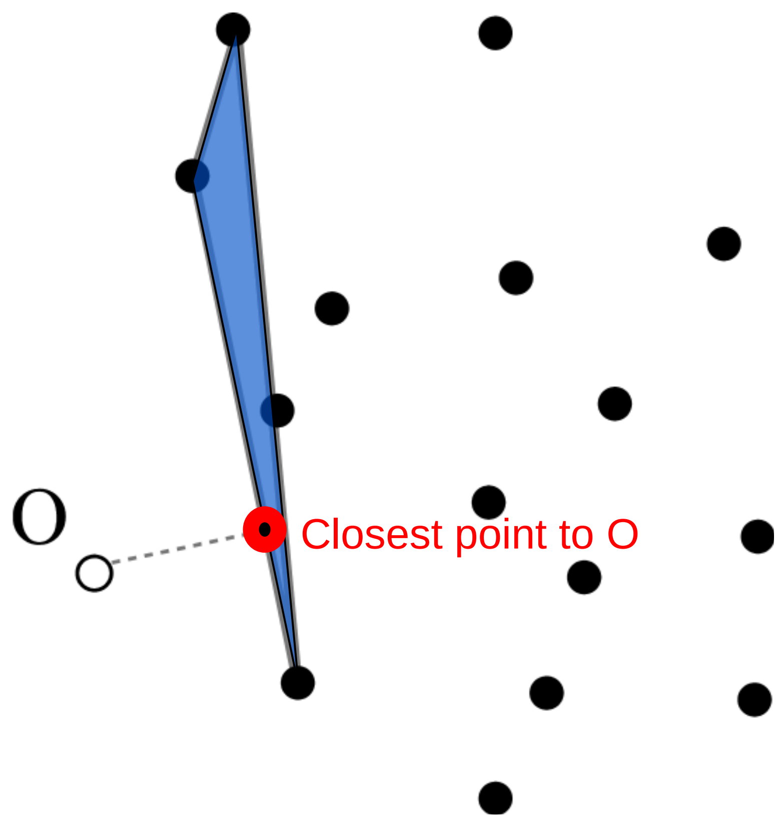
Principe of GJK algorithm :
compute the distance between the origin *O* and *B-A*.



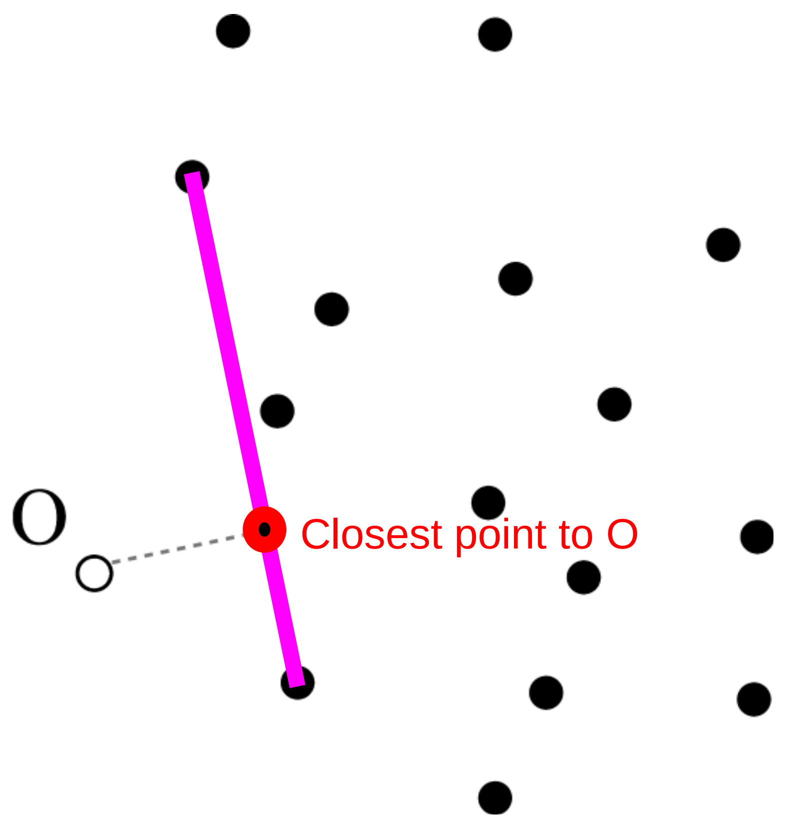
Principe of GJK algorithm :
compute the distance between the origin *O* and *B-A*.



Principe of GJK algorithm :
compute the distance between the origin *O* and *B-A*.

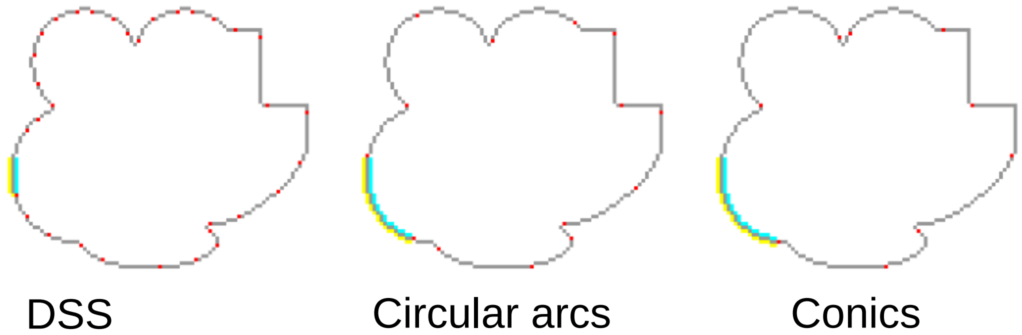


Principe of GJK algorithm :
compute the distance between the origin *O* and *B-A*.



Principe of GJK algorithm :
compute the distance between the origin *O* and *B-A*.

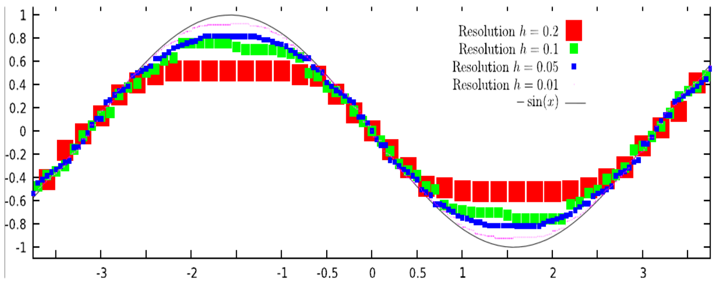
Conclusion



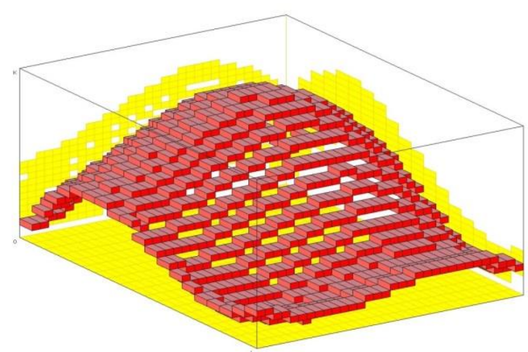
DLL provide a nice extension of the decomposition of a curve in DSS with a single algorithm and the choice of the primitive used:

- DSS (kernel functions are x and y)
- Circular arcs (kernel function are x^2+y^2, x and y)
- Conics (kernel function are x^2, y^2, xy, x and y)

Do we have Multigrid Convergence properties in this framework of digital contours and shapes, as for DSS ?



It's time consuming to compute the derivatives all along a curve :
Provide an enhanced version
with point deletion and insertion…



DLL works also in 3D and more:
Provide also multigrid convergence results…