DigitalSnow Final meeting
Digital Level Layers for Curve Decomposition and Vectorization

july 9th 2015, Autrans
Yan Gerard (ISIT) yan.gerard@udamail.fr
Plan

I

Introduction
Introduction

3D Workflows

Geometric design
(3D artists, designers…)

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…)

Martin Newell’s Utah Teapot

Building (3D printer, factory…)

Computer simulation (Finite Elements Methods…)

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

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

3D models
(geometry)

Geometric design
(3D artists, designers…)

Introduction
Introduction

Many possible models

3D models

(geometry)

Points cloud
Introduction

Points cloud

Sets of voxels

3D models

(geometry)
Introduction

Points cloud

Sets of voxels

Mesh

3D models
(geometry)
Introduction

Points cloud

Sets of voxels

Mesh

3D models

(geometr)

Many possible models

Control Mesh

1 iteration of
Loop scheme

Limit shape

Subdivision surfaces
Introduction

Many possible models

Points cloud

Sets of voxels

Mesh

3D models

gometry

Control Mesh

1 iteration of Loop scheme

Limit shape

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

Subdivision surfaces
Introduction

Many possible models

Points cloud

Sets of voxels

Mesh

3D models
(geometry)

Level sets
(equation)

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

Control Mesh

1 iteration of Loop scheme

Limit shape

Subdivision surfaces
Introduction

Points cloud

Sets of voxels

Mesh

3D models
(geometry)

Level sets
(equation)

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

Control Mesh

1 iteration of
Loop scheme

Limit shape

Subdivision surfaces
Introduction

Points cloud

Sets of voxels

Mesh

3D models (geometry)

Level sets (equation)

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

Control Mesh

1 iteration of Loop scheme

Limit shape

Subdivision surfaces

Marching Cubes
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.
But some people are stubborn. Maybe, they played too much Legos during their childhood.

Are there some better reasons to do Digital Geometry?
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.
Introduction

A better reason to do Digital Geometry?

Binary image

Images are tabs of pixel values.
A better reason to do Digital Geometry?

Image with grey-levels

The values of the tab

Images are tabs of pixel values.
Images are tabs of pixel values.
Input/output are digital

Cameras
3D scan
Kinect
MRI
US
...
Introduction

**Input/output are digital**

- Cameras
- 3D scan
- Kinect
- MRI
- US
- ...

**The input is digital**

- Integer arithmetic

**Which arithmetic for the computation?**

- **Floating Point Arithmetic**

  - IEEE 754 single precision floating-point storage
  - sign: 1 bit
  - exponent: 8 bits
  - mantissa: 23 bits

**The output is digital**

- 0111001010010
**Introduction**

Input/output are digital

---

**Integer arithmetic**

Suitable for computers (exact computations)

Requires digital mathematics…

---

**Floating Point Arithmetic**

IEEE 754 single precision floating-point storage

<table>
<thead>
<tr>
<th>sign</th>
<th>biased exponent</th>
<th>mantissa (significant)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8 bits</td>
<td>23 bits</td>
</tr>
</tbody>
</table>

32 bits total

Suitable for mathematics (continuous objects)

Problems of inaccuracy…
Input/output are digital

Floating Point Arithmetic

IEEE 754 single precision floating-point storage
- sign: 1 bit
- exponent: 8 bits
- mantissa: 23 bits

32 bits total

Problems of inaccuracy...

Suitable for mathematics (continuous objects)

\[ y = \ln\left(1 - x\right)/x \]

with IEEE 754 standard

Plot of \( \log\left(1-x\right)/x \) near \( x=0 \), computed straightforwardly, using IEEE
**Introduction**

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

Use integer arithmetic with suitable digital mathematical theories

Output = **integers**

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

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

**Output** = integers

Most popular option.

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

The challenge of digital mathematics

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

Use integer arithmetic with suitable digital mathematical theories

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

The development of digital mathematics is a huge challenge

Output = integers
Introduction

What is the main constraint?

Continuous object

Compute theoretically

For instance, tangent line, derivative…

Digital object at resolution $h$

Real object

Convergence as $h \to \infty$

Its digitization at resolution $h$

Compute (with integers)

Digital tangent line, digital derivative…
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 developed 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 $\mathbb{Z}^2$ are subsets of $\mathbb{Z}^2$ 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 $\mathbb{Z}^d$. 
Digital lines definition:
Digital lines of $\mathbb{Z}^2$ are subsets of $\mathbb{Z}^2$ characterized by a double inequality:

$$h \leq ax + by < h + \Delta$$
**Digital lines** definition:

Digital lines of \( \mathbb{Z}^2 \) are subsets of \( \mathbb{Z}^2 \) 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 $\mathbb{Z}^2$ are subsets of $\mathbb{Z}^2$ 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

Output: Its decomposition in Digital Straight Segments

Segmentation in pieces of digital straight lines (72 pieces)

Worst case complexity: linear time

About tangent estimators

DSS decomposition
Input: A digital curve

Output: Its Tangential Cover

Worst Case Complexity: Linear Time
Tangent estimators

Interest: Maximal Digital Straight Segments around a point \( x \) provide tangent direction at \( x \).

Good news: it’s multigrid convergent (under some assumptions)

Average convergence rate \( O(h^{2/3}) \)

Locally convex shapes

In J-O Lachaud, 2006

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

There exist other ways to provide multigrid convergent tangent estimators...
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

\[ \text{Length} = \int 1 \, ds \]
\[ \text{Area} = \int \int 1 \, dS \]
\[ \text{Volume} = \int \int \int 1 \, dV \]

Sums (barycenter coordinates, moment…)

\[ \text{Sum} = \int f(x) \, ds \]
\[ \text{Sum} = \int \int f(x) \, dS \]
\[ \text{Sum} = \int \int \int f(x) \, dV \]
Preserve the relations between measurements (turning Number Theorem, Gauss-Bonnet...)

Don’t forget Multigrid convergence...

Why this interest for computing the tangent or normal direction?

To provide measurements...

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…

Compute the normal field

Weight the measurement with the metric associated with the normal

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.
About tangent estimators

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
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...
Three approaches can be used to define digital primitives:
- topological
- morphological
- analytical
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.
The *Minkowski’s sum* \( S + B \) is the set of points covered by the structuring elements as it moves all along the shape.
Structuring element

Morphological ellipse

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

Analytical ellipse

Digital Level Layers
The 3 definitions collapse for lines in $\mathbb{Z}^2$, planes in $\mathbb{Z}^3$ … hyperplanes in $\mathbb{Z}^d$
(affine sub-spaces of codimension 1)

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

<table>
<thead>
<tr>
<th>Topology</th>
<th>Morphology</th>
<th>Algebra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighborhood</td>
<td>Structuring element</td>
<td>value $\Delta$</td>
</tr>
<tr>
<td>$\text{Ball } N_\infty$</td>
<td>$\text{Ball } N_1$</td>
<td>$\Delta = N_\infty$ (a)</td>
</tr>
<tr>
<td>$\text{Ball } N_1$</td>
<td>$\text{Ball } N_\infty$</td>
<td>$\Delta = N_1$ (a)</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Topology</th>
<th>Morphology</th>
<th>Algebra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighborhood</td>
<td>Structuring element</td>
<td>value $\Delta$</td>
</tr>
</tbody>
</table>

$\begin{align*}
\text{Ball } N_{\infty} & \quad \text{Ball } N_1 \\
\text{Ball } N_1 & \quad \text{Ball } N_{\infty} \\
\end{align*}$

$\Delta = N_{\infty} (a)$

$\Delta = N_1 (a)$
The 3 definitions collapse for lines in $\mathbb{Z}^2$, planes in $\mathbb{Z}^3$ ... hyperplanes in $\mathbb{Z}^d$ (affine sub-spaces of codimension 1)

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

<table>
<thead>
<tr>
<th>Topology</th>
<th>Morphology</th>
<th>Algebra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighborhood</td>
<td>Structuring element</td>
<td>value $\Delta$</td>
</tr>
<tr>
<td>$Ball\ N_1$</td>
<td>$Ball\ N_\infty$</td>
<td>$\Delta = N_\infty (a)$</td>
</tr>
</tbody>
</table>

Standard class.
The 3 definitions collapse for lines in $\mathbb{Z}^2$, planes in $\mathbb{Z}^3$ ... hyperplanes in $\mathbb{Z}^d$
(affine sub-spaces of codimension 1)

They don’t collapse for arbitrary shapes.
<table>
<thead>
<tr>
<th>Properties</th>
<th>Topology</th>
<th>Morphology</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Morphology</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Algebraic characterization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recognition algorithm</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Advantages and drawbacks?
<table>
<thead>
<tr>
<th>Properties</th>
<th>Topology</th>
<th>Morphology</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
</tr>
<tr>
<td>Morphology</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
</tr>
<tr>
<td>Algebraic characterization</td>
<td>❌</td>
<td>❌</td>
<td>✔️</td>
</tr>
<tr>
<td>Recognition algorithm</td>
<td>✔️  SVM</td>
<td>✗</td>
<td>✔️</td>
</tr>
</tbody>
</table>
**Digital Level Layer definition:**
A *Digital Level Layer* (name coming from *Level sets*) is a subset of $\mathbb{Z}^d$ 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 $\mathbb{Z}^d$ 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 $\mathbb{Z}^d$ 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* definition:

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

$$ h \leq f(x) < h' $$
Digital Level Layers definition:
A Digital Level Layer (name coming from Level sets) is a subset of $\mathbb{Z}^d$ characterized by a double inequality:
\[ h \leq f(x) < h' \]
### Review of multigrid convergent estimators developed in Digital Geometry.

<table>
<thead>
<tr>
<th>Method</th>
<th>Authors</th>
<th>Assumption on the continuous curve</th>
<th>Order of derivative</th>
<th>Worst case Error bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximal DSS with thickness=1</td>
<td>A. Vialard, J-O Lachaud, F De Vieilleville</td>
<td>Locally convex, $C^3$</td>
<td>$k=1$</td>
<td>$O(h^{1/3})$</td>
</tr>
<tr>
<td>Convolutions</td>
<td>S. Fourey, F. Brunet, A. Esbelin, B. R. Malgouyres</td>
<td>$C^3$ or $C^2$</td>
<td>Any $k$</td>
<td>$O(h^{2/3})^k$</td>
</tr>
<tr>
<td>Maximal DLL with thickness&gt;1</td>
<td>L. Provot, Y. Gerard</td>
<td>$C^{k+1}$</td>
<td>Any $k$</td>
<td>$O(h^{1/(k+1)})^k$</td>
</tr>
</tbody>
</table>
Review of multigrid convergent estimators developed 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 developed in Digital Geometry.

<table>
<thead>
<tr>
<th>Maximal DSS with thickness=1</th>
<th>A. Vialard, J-O Lachaud, F De Vieillelville</th>
<th>Locally convex, $C^3$</th>
<th>$k=1$</th>
<th>$O(h^{1/3})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolutions</td>
<td>S. Fourey, F. Brunet, A. Esbelin, B. R. Malgouyres</td>
<td>$C^3$ or $C^2$</td>
<td>Any $k$</td>
<td>$O(h^{(2/3)^k})$</td>
</tr>
<tr>
<td>Maximal DLL with thickness&gt;1</td>
<td>L. Provot, Y. Gerard</td>
<td>$C^{k+1}$</td>
<td>Any $k$</td>
<td>$O(h^{1/(k+1)})$</td>
</tr>
</tbody>
</table>
Review of multigrid convergent estimators developed in Digital Geometry.

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

<table>
<thead>
<tr>
<th>Maximal DLL with thickness&gt;1</th>
<th>L. Provot, Y. Gerard</th>
<th>$C^{k+1}$</th>
<th>Any $k$</th>
<th>$O(h^{1/(k+1))}$</th>
</tr>
</thead>
</table>

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

Applied on a digital function $f: \mathbb{Z} \to \mathbb{Z}$
Review of multigrid convergent estimators developed in Digital Geometry.

<table>
<thead>
<tr>
<th>Maximal DSS with thickness=1</th>
<th>A. Vialard, J-O Lachaud, F De Vieilleville</th>
<th>Locally convex, $C^3$</th>
<th>$k=1$</th>
<th>$O(h^{1/3})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increase the thickness</td>
<td>More general</td>
<td>Better worst case convergence rate.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximal DLL with thickness&gt;1</td>
<td>L. Provot, Y. Gerard</td>
<td>$C^k$</td>
<td>Any $k$</td>
<td>$O(h^{1/(k+1)})$</td>
</tr>
</tbody>
</table>

- 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 developed 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.

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

<table>
<thead>
<tr>
<th>Maximal DLL with thickness&gt;1</th>
<th>L. Provot, Y. Gerard</th>
<th>$C^{k+1}$</th>
<th>Any $k$</th>
<th>$O(h^{1/(k+1))}$</th>
</tr>
</thead>
</table>

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 developed in Digital Geometry.

#### More restrictive and less accurate but faster...

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

<table>
<thead>
<tr>
<th>Maximal DLL with thickness&gt;1</th>
<th>L. Provot, Y. Gerard</th>
<th>$C^{k+1}$</th>
<th>Any $k$</th>
<th>$O(h^{(1/(k+1))})$</th>
</tr>
</thead>
</table>
Review of multigrid convergent estimators developed 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 \( \Delta \), 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 \( \Delta \), 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 $h = 0.05$
$F(x)$: digitization of $\sin(x)$ with a resolution $h = 0.05$
$F(x)$: digitization of $\sin(x)$ with a resolution $h = 0.05$
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 $R_{max} = 2$
Digital Level Layers

Multigrid convergence

Second derivative

Resolution $h = 0.2$  
Resolution $h = 0.1$  
Resolution $h = 0.05$  
Resolution $h = 0.01$  
$- \sin(x)$  

Graph showing the second derivative with different resolutions.
Plan

DLL decomposition
Input: A digital curve

Segmentation in pieces of digital straight lines (72 pieces)

Output: Its decomposition in Digital Straight 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 which 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)*
Plan

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 $A \subset \mathbb{R}^d$ and $B \subset \mathbb{R}^d$ given by their vertices.

Question: do they intersect?

More general question: compute their minimal distance.

Principle of GJK algorithm:
compute the distance between the origin $O$ and $B - A$. 
Principle of GJK algorithm:
compute the distance between the origin \(O\) and \(B-A\).
Principle of GJK algorithm:
compute the distance between the origin $O$ and $B-A$. 
Principle of GJK algorithm:
compute the distance between the origin $O$ and $B-A$. 
Principle of GJK algorithm:
compute the distance between the origin $O$ and $B-A$. 

**Current simplex**

**Closest point to $O$**

**Normal direction**
Principle of GJK algorithm:
compute the distance between the origin $O$ and $B - A$. 

Current simplex

Closest point to $O$

Normal direction

Optimal point
Principle of GJK algorithm:
compute the distance between the origin \( O \) and \( B-A \).
Principle of GJK Algorithm:
compute the distance between the origin \( O \) and \( B-A \).
Principle of GJK algorithm:
compute the distance between the origin \(O\) and \(B-A\).
Principle of GJK algorithm:
compute the distance between the origin $O$ and $B-A$. 
Principle of GJK algorithm:
compute the distance between the origin $O$ and $B-A$. 
Principle of GJK algorithm:
compute the distance between the origin \( O \) and \( B-A \).
Principle of GJK algorithm:
compute the distance between the origin $O$ and $B-A$. 
Principle of GJK algorithm:
compute the distance between the origin $O$ and $B-A$. 
Principle of GJK algorithm:
compute the distance between the origin $O$ and $B-A$. 
Principle of GJK algorithm:
compute the distance between the origin $O$ and $B-A$. 

Closest point to $O$
Principle of GJK algorithm:
compute the distance between the origin $O$ and $B-A$. 

Closest point to $O$
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?
Further works

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…