David
Coeurjolly

Table of
contents

Discrete Circle Recognition

David Coeurjolly

Laboratoire LIRIS
Université Claude Bernard Lyon 1
43 Bd du 11 Novembre 1918
69622 Villeurbanne CEDEX
France
david.coeurjolly@liris.cnrs.fr

January - 2005

Plan

David Coeurjolly

Plan

Plan

Table of contents

Introduction

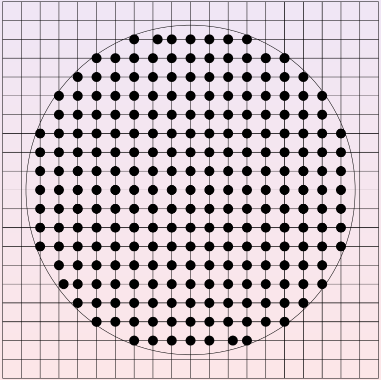
Why the discrete circle recognition ?

- Important elementary object in the discrete geometry paradigm
- Order 2 object $\rightarrow$ curvature estimation

Discrete disk

Definition of a discrete disk with center $(0,0)$

$$D : x^2 + y^2 < \left(R + \frac{1}{2}\right)^2, \qquad x, y \in \mathbb{Z} \text{ and } R \in \mathbb{R}$$

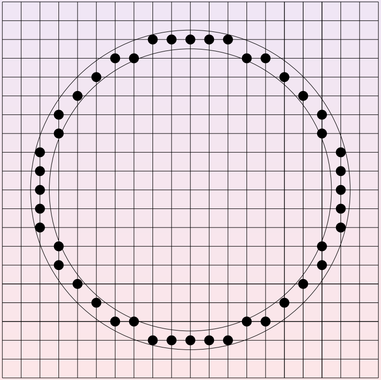


David Coeurjolly

Discrete circle

Definition of an arithmetical discrete circle with center (x_0, x_0) and radius R

$$\left(R - \frac{1}{2}\right)^2 \leq (x - x_0)^2 + (y - y_0)^2 < \left(R + \frac{1}{2}\right)^2, \qquad x, y \in \mathbb{Z}$$



Recognition based on circular separability

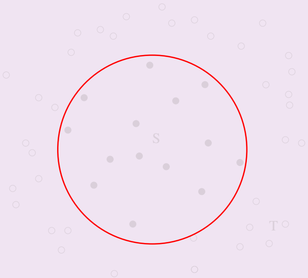
Property

A set D of pixels is a digital disk if there exists an Euclidean circle that encloses the pixels of D but excludes its complement

Circular separability in Computational Geometry

- $\Rightarrow$ Linear programming in dimension 3
- $\Rightarrow$ Linear programming in dimension 2

Notations



Recognition based on circular separability

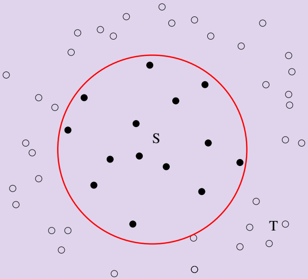
Property

A set D of pixels is a digital disk if there exists an Euclidean circle that encloses the pixels of D but excludes its complement

Circular separability in Computational Geometry

- $\Rightarrow$ Linear programming in dimension 3
- $\Rightarrow$ Linear programming in dimension 2

Notations



Preimage of discrete circle and *arc center domain*

Definition

Given a set of pixels S, its preimage must contain the set of discrete disks enclosing S and excluding its complement

- 3 parameters: (x_0, y_0) and R
- The problem is not linear
- Can be reduced to a 2-D domain representing the centers (x_0, y_0) (*arc center domain*) from which we can deduce a set of radii R according to S

acd is empty $\Leftrightarrow$ the preimage is empty $\Leftrightarrow$ S is not a discrete disk

Preimage of discrete circle and *arc center domain*

Definition

Given a set of pixels S, its preimage must contain the set of discrete disks enclosing S and excluding its complement

- 3 parameters: (x_0, y_0) and R
- The problem is not linear
- Can be reduced to a 2-D domain representing the centers (x_0, y_0) (*arc center domain*) from which we can deduce a set of radii R according to S

acd is empty $\Leftrightarrow$ the preimage is empty $\Leftrightarrow$ S is not a discrete disk

Preimage of discrete circle and *arc center domain*

Definition

Given a set of pixels S, its preimage must contain the set of discrete disks enclosing S and excluding its complement

- 3 parameters: (x_0, y_0) and R
- The problem is not linear
- Can be reduced to a 2-D domain representing the centers (x_0, y_0) (*arc center domain*) from which we can deduce a set of radii R according to S

acd is empty ⇔ the preimage is empty ⇔ S is not a discrete disk

Preimage of discrete circle and *arc center domain*

Definition

Given a set of pixels S, its preimage must contain the set of discrete disks enclosing S and excluding its complement

- 3 parameters: (x_0, y_0) and R
- The problem is not linear
- Can be reduced to a 2-D domain representing the centers (x_0, y_0) (*arc center domain*) from which we can deduce a set of radii R according to S

acd is empty $\Leftrightarrow$ the preimage is empty $\Leftrightarrow$ S is not a discrete disk

David Coeurjolly

Preimage of discrete circle and *arc center domain*

Definition

Given a set of pixels S, its preimage must contain the set of discrete disks enclosing S and excluding its complement

- 3 parameters: (x_0, y_0) and R
- The problem is not linear
- Can be reduced to a 2-D domain representing the centers (x_0, y_0) (*arc center domain*) from which we can deduce a set of radii R according to S

acd is empty $\Leftrightarrow$ the preimage is empty $\Leftrightarrow$ S is not a discrete disk

Table of contents

Circular separability as a LP in dimension 3

[OKM86]

Paraboloid projection

$$(x, y) \rightarrow (x, y, x^2 + y^2)$$

i.e. projection onto the paraboloid $z = x^2 + y^2$

In this transformed space, a plane $ax + by + (x^2 + y^2) = c$ can be rewritten as:

$$(x + \frac{1}{2}a)^2 + (y - \frac{1}{2}b)^2 = c + \frac{1}{4}(a^2 + b^2)$$

Hence, assuming $c + \frac{1}{4}(a^2 + b^2) > 0$, a plane in the transformed space is a circle in $\mathbb{R}^2$. Conversely, a circle $(x - A)^2 + (y - B)^2 = R^2$ is transformed into:

$$-2Ax - 2By + (x^2 + y^2) = R^2 - (A^2 + B^2)$$

Circular separability as a LP in dimension 3

[OKM86]

Paraboloid projection

$$(x, y) \rightarrow (x, y, x^2 + y^2)$$

i.e. projection onto the paraboloid $z = x^2 + y^2$

In this transformed space, a plane $ax + by + (x^2 + y^2) = c$ can be rewritten as:

$$(x + \frac{1}{2}a)^2 + (y - \frac{1}{2}b)^2 = c + \frac{1}{4}(a^2 + b^2)$$

Hence, assuming $c + \frac{1}{4}(a^2 + b^2) > 0$, a plane in the transformed space is a circle in $\mathbb{R}^2$.
Conversely, a circle $(x - A)^2 + (y - B)^2 = R^2$ is transformed into:

$$-2Ax - 2By + (x^2 + y^2) = R^2 - (A^2 + B^2)$$

Circular separability as a LP in dimension 3

[OKM86]

Paraboloid projection

$$(x, y) \rightarrow (x, y, x^2 + y^2)$$

i.e. projection onto the paraboloid $z = x^2 + y^2$

In this transformed space, a plane $ax + by + (x^2 + y^2) = c$ can be rewritten as:

$$(x + \frac{1}{2}a)^2 + (y - \frac{1}{2}b)^2 = c + \frac{1}{4}(a^2 + b^2)$$

Hence, assuming $c + \frac{1}{4}(a^2 + b^2) > 0$, a plane in the transformed space is a circle in $\mathbb{R}^2$.
Conversely, a circle $(x - A)^2 + (y - B)^2 = R^2$ is transformed into:

$$-2Ax - 2By + (x^2 + y^2) = R^2 - (A^2 + B^2)$$

Circular separability as a LP in dimension 3

LP system

S can be separated from T by a circle if the feasible region of the system:

$$\begin{cases} ax + by + (x^2 + y^2) \leq c & \text{if } (x, y) \in S \\ ax + by + (x^2 + y^2) \geq c + d & \text{if } (x, y) \in T \end{cases}$$

is not empty.

Rem.

The parameter d can be defined by a, b and c, hence we obtain a linear system in dimension 3

Simple Discrete Circle recognition in LP_3

Algorithm

- For each pixel $s \in S$ and $t \in T$
  - Construction of the two linear constraints $s \rightarrow ax + by + (x^2 + y^2) \leq c$ and $t \rightarrow ax + by + (x^2 + y^2) \geq c + d$
- Solve the global linear inequality system
- If the feasible region is empty, S is not a discrete disk
- Otherwise, return the preimage

Computational cost

- $|S| + |T|$ linear constraints
- To detect if the feasible is empty or not : $O(|S| + |T|)$ [Meg83, Meg84]
- To construct the preimage : $O((|S| + |T|) \cdot \log(|S| + |T|))$ [PS85] (not on-line)

Simple Discrete Circle recognition in LP_3

Algorithm

- For each pixel $s \in S$ and $t \in T$
  - Construction of the two linear constraints $s \rightarrow ax + by + (x^2 + y^2) \leq c$ and $t \rightarrow ax + by + (x^2 + y^2) \geq c + d$
- Solve the global linear inequality system
- If the feasible region is empty, S is not a discrete disk
- Otherwise, return the preimage

Computational cost

- $|S| + |T|$ linear constraints
- To detect if the feasible is empty or not : $O(|S| + |T|)$ [Meg83, Meg84]
- To construct the preimage : $O((|S| + |T|) \cdot \log(|S| + |T|))$ [PS85] (not on-line)

Table of contents

Solutions in LP_2

Lemma

If a circle $C(\omega, R)$ encloses S but excludes T, its center ω necessarily satisfies the following inequality:

$$\forall s \in S,\ \forall t \in T,\ dist(\omega, s) < dist(\omega, t)$$

Definition

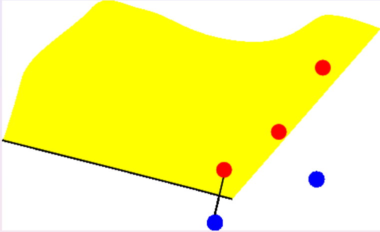
Let $s \in S$ and $t \in T$, $H(s,t)$ denotes the half-plane bounded by the perpendicular bisector of s and t, and containing s

$C(\omega, R)$ is valid for $(s,t) \Leftrightarrow \omega \in H(s,t)$ and $dist(\omega, s) \leq R < dist(\omega, t)$

Arc Center Domain

acd

$$acd(S,T) = \bigcap_{s \in S, t \in T} H(s,t)$$

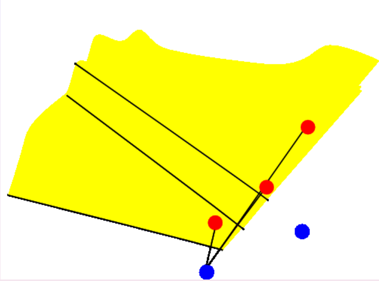


- $|S| \cdot |T|$ linear constraints
- *acd* is a 2-D convex region (maybe unbounded)
- it corresponds to the *Generalized Voronoi cell* in Computational Geometry

Arc Center Domain

acd

$$acd(S,T) = \bigcap_{s\in S, t\in T} H(s,t)$$

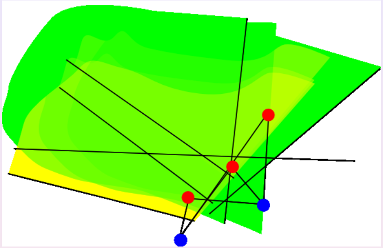


- $|S| \cdot |T|$ linear constraints
- *acd* is a 2-D convex region (maybe unbounded)
- it corresponds to the *Generalized Voronoi cell* in Computational Geometry

Arc Center Domain

acd

$$acd(S,T) = \bigcap_{s \in S, t \in T} H(s,t)$$

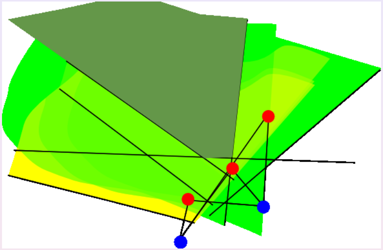


- $|S| \cdot |T|$ linear constraints
- *acd* is a 2-D convex region (maybe unbounded)
- it corresponds to the *Generalized Voronoi cell* in Computational Geometry

Arc Center Domain

acd

$$acd(S,T) = \bigcap_{s \in S, t \in T} H(s,t)$$



- $|S| \cdot |T|$ linear constraints
- *acd* is a 2-D convex region (maybe unbounded)
- it corresponds to the *Generalized Voronoi cell* in Computational Geometry

Computational cost

Intersection of $|S| \cdot |T|$ linear constraints in 2-D

- Detect if the *acd* is empty or not : $O(|S| \cdot |T|)$ [Meg83, Meg84]
- Construct the *acd* : $O((|S| \cdot |T|) \cdot \log(|S| \cdot |T|))$ [PS85] (on-line algorithms)

Summary

LP_2

- $O(|S| \cdot |T|)$ or $O((|S| \cdot |T|) \cdot \log(|S| \cdot |T|))$
- Elementary 2-D on-line algorithms

LP_3

- $O(|S| + |T|)$ or $O((|S| + |T|) \cdot \log(|S| + |T|))$
- LP solvers exist but optimal algorithms are quite complex

Discrete case: S = set of pixels and T = background pixels

$\Rightarrow$ Inefficient algorithms

Summary

LP_2

- $O(|S| \cdot |T|)$ or $O((|S| \cdot |T|) \cdot \log(|S| \cdot |T|))$
- Elementary 2-D on-line algorithms

LP_3

- $O(|S| + |T|)$ or $O((|S| + |T|) \cdot \log(|S| + |T|))$
- LP solvers exist but optimal algorithms are quite complex

Discrete case: S = set of pixels and T = background pixels

$\Rightarrow$ Inefficient algorithms

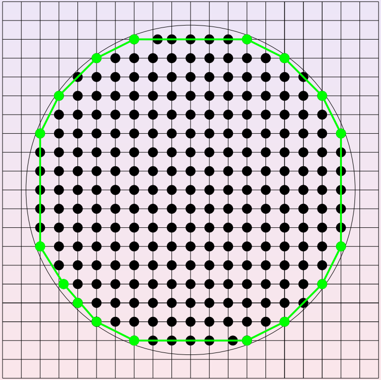
Table of contents

Table of contents

Filtering of S

Prop 1: [CGRT04]

S can be reduced to its convex hull



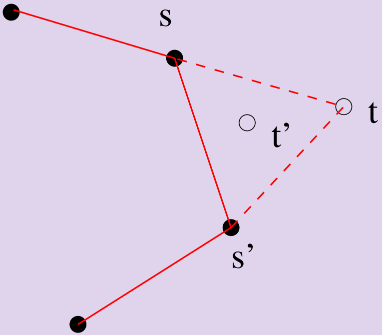
Computational cost

Discrete Convex hull of the border $\mathcal{B}$ of the object : $O(|\mathcal{B}|)$

Filtering of T

Prop 2: Triangular inclusion

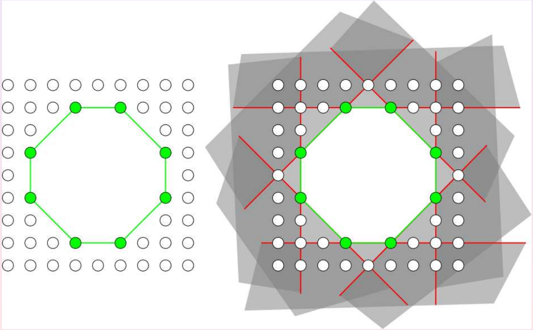
If a point $t' \in T$ belongs to the triangle (s, s', t) with $s, s' \in S$ and $t \in T$, then the point t can be removed from T without changing the *acd*



Filtering of T in the discrete case

Prop 3

We can independently process the edges of $Conv(\mathcal{B})$



Filtering of T and Bezout's points

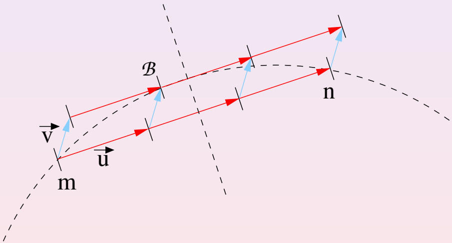
[CGRT04]

Definition: Bezout's point

A point $\mathcal{P}$ is a Bezout point of a straight segment $[mn]$ defined by an arithmetical directional vector $\vec{u}$ if and only if:

$$\overrightarrow{m\mathcal{P}} = \vec{v} + k\vec{u} \quad \textit{with} \quad k \in \mathbb{Z} \quad \textit{and} \quad det(\vec{u}, \vec{v}) = \pm 1$$

and $\mathcal{P}$ is the closest point to the middle of $[mn]$.



Uni-modular parallelogram

Let $O, P, Q, R \in \mathbb{Z}^2$ be a parallelogram. There is no discrete point inside $(OPQR)$ $\Leftrightarrow$ $det(\vec{OP}, \vec{OQ}) = \pm 1$

Filtering of T and Bezout's points

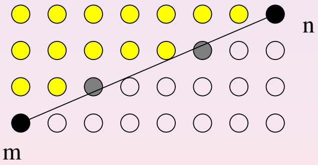
[CGRT04]

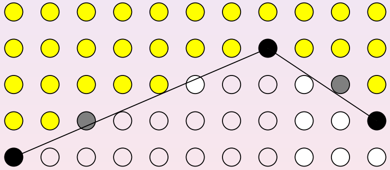
Definition: Bezout's point

A point $\mathcal{P}$ is a Bezout point of a straight segment $[mn]$ defined by an arithmetical directional vector $\overrightarrow{u}$ if and only if:

$$\overrightarrow{m\mathcal{P}} = \overrightarrow{v} + k\overrightarrow{u} \quad \textit{with} \quad k \in \mathbb{Z} \quad \textit{and} \quad det(\overrightarrow{u}, \overrightarrow{v}) = \pm 1$$

and $\mathcal{P}$ is the closest point to the middle of $[mn]$.

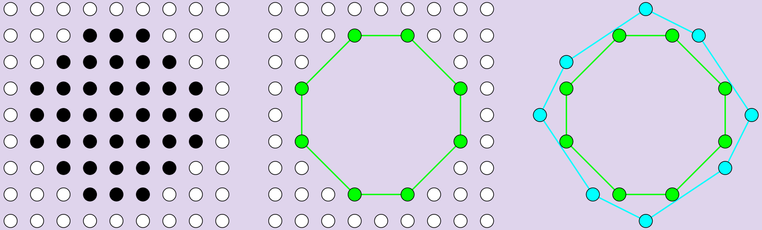




Theorem

Let us consider the oriented edge $e = [mn]$ with $m, n \in \mathbb{Z}$ and T_e the set of grid points in $\bar{\mathcal{O}}$ belonging to the half-space defined by e. Then, all triangles (m, n, t) contain another point $t' \in T$ except for Bezout points

Conclusion on the optimizations

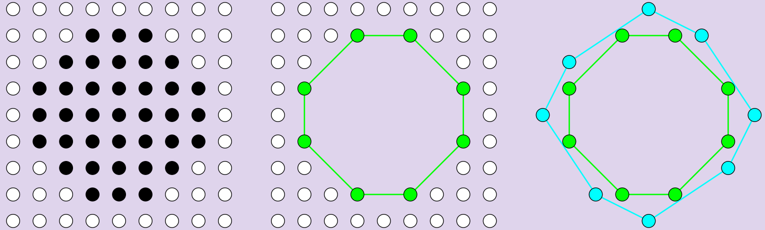


- $S = Conv(\mathcal{B})$
- T = one Bezout's point per edge of the $Conv(\mathcal{B})$

Objective

Bound on the number of edges of the discrete convex hull

Conclusion on the optimizations



- $S = Conv(\mathcal{B})$
- $T =$ one Bezout's point per edge of the $Conv(\mathcal{B})$

Objective

Bound on the number of edges of the discrete convex hull

Table of contents

Discrete convex hull in a $m \times m$-grid

[BB91, AŽ95]

Let $e(m)$ be the number of edges of a convex polygon whose vertices belong to a $m \times m$ grid:

$$e(m) = \frac{12}{(4\pi^2)^{1/3}} m^{2/3} + O(m^{1/3} log(m))$$

Given a convex object $\mathcal{O}$ in a $m \times m$-grid

- $|S| = O(m^{2/3})$
- $|T| = O(m^{2/3})$

Discrete convex hull in a $m \times m$-grid

[BB91, AŽ95]

Let $e(m)$ be the number of edges of a convex polygon whose vertices belong to a $m \times m$ grid:

$$e(m) = \frac{12}{(4\pi^2)^{1/3}} m^{2/3} + O(m^{1/3} log(m))$$

Given a convex object $\mathcal{O}$ in a $m \times m$-grid

- $|S| = O(m^{2/3})$
- $|T| = O(m^{2/3})$

Overall algorithm

Alg

1. Compute the convex hull of $\mathcal{O}$ — $O(m)$
2. For each edge of $Conv(\mathcal{B})$ — $O(m^{2/3})$ edges
   - Compute Bezout's point and append it to T — $O(\log(m))$
3. Solve the Separability problem using either LP_2 or LP_3

LP_2

- $O(|S| \cdot |T|)$ or $O((|S| \cdot |T|) \cdot \log(|S| \cdot |T|))$
- $O(m^{4/3})$ or $O(m^{4/3} \log(m))$

LP_3

- $O(|S| + |T|)$ or $O((|S| + |T|) \cdot \log(|S| + |T|))$
- $O(m^{2/3})$ or $O(m^{2/3} \log(m))$

David Coeurjolly

Table of contents

On-line recognition and Discrete arc segmentation

Specific problems

- The LP solver must be on-line to update the *acd*
- If the input is an $8-$arc, we need to estimate the local orientation (kind of *local convex hull*)

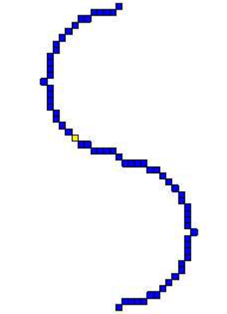
$\Rightarrow$ *greedy process*

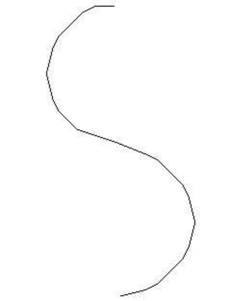
A possible solution:

1. Use an on-line DSS segmentation to extract *Strictly convex or concave part* (SCoC) from the $8-$arc
2. On-line construction of the convex hull and Bezout's points
3. On-line solver in LP_2 [PS85] ($O(m^{4/3}\log(m))$)

Some results

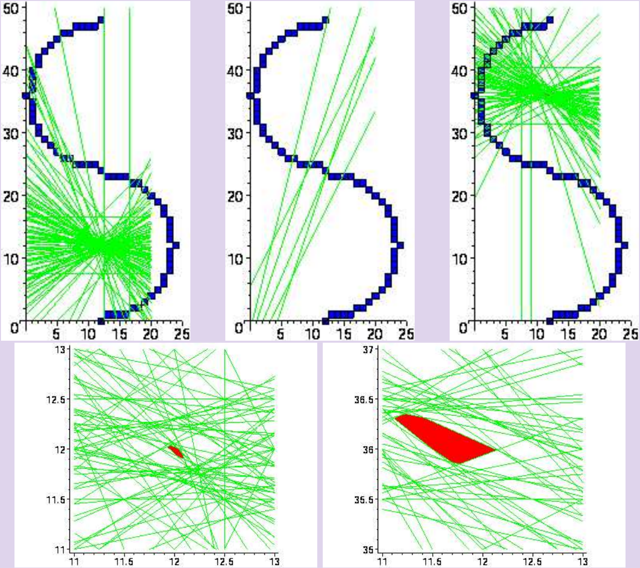
SCoC and the *local convex hull* in *S*





Some results

[CGRT04]



Final result of the segmentation

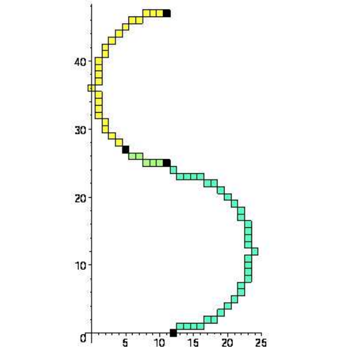
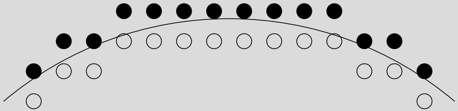


Table of contents

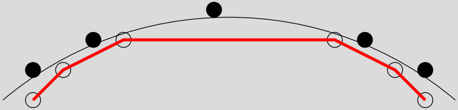
Conclusion

Importance of the arithmetic in geometrical problems

Without Bezout's points [Kov90]:



With Bezout's points:



Recognition in LP_2:

Brute-force algorithm

- $O(n^2 \log n)$
- update $O(\log n)$
- memory $O(n^2)$

With the arthimetical properties

- $O(n^{4/3} \log n)$
- update $O(\log n)$
- memory $O(n^{4/3})$

Other references...

[NA84, Fis86, Sau93, WS95, Dam95, TC89, Kim84, KA84]

D.M. Acketa and J.D. Žunić.
On the maximal number of edges of convex digital polygons included into a $m \times m$-grid.
Journal of Combinatorial Theory, Serie A(69):358–368, 1995.

A. Balog and I. Bárány.
On the convex hull of the integer points in a disc.
In ACM-SIGACT ACM-SIGGRAPH, editor, *Proceedings of the 7th Annual Symposium on Computational Geometry (SCG '91)*, pages 162–165, North Conway, NH, USA, June 1991. ACM Press.

D. Coeurjolly, Y. Gerard, J.-P. Reveillès, and L. Tougne.
An elementary algorithm for digital arc segmentation.
Discrete Applied Mathematics, 139(1-3):31–50, 2004.

P. Damaschke.
The linear time recognition of digital arcs.
Pattern Recognition Letters, 16(5):543–548, May 1995.

S. Fisk.
Separating points sets by circles, and the recognition of digital disks.
IEEE Transaction on Pattern Analysis and Machine Intelligence, 8(4):554–556, 1986.

C. E. Kim and T. A. Anderson.
Digital disks and a digital compactness measure.
In *Seventh International Conference on Pattern Recognition (Montreal, Canada, July 30-August 2, 1984)*, IEEE Publ. 84CH2046-1, pages 254–257. IEEE, 1984.

C. E. Kim.
Digital disks.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 6:372–374, 1984.

V. A. Kovalevsky.
New definition and fast recognition of digital straight segments and arcs.
Proceedings of the tenth international conference on Pattern Analysis and Machine Intelligence, pages 31–34, June 1990.

N. Megiddo.
Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems.
SIAM Journal of Computing, 12(4):759–776, November 1983.

N. Megiddo.

Linear programming in linear time when the dimension is fixed.
Journal of the ACM, 31(1):114–127, January 1984.

A. Nakamura and K. Aizawa.
Digital circles.
Computer Vision, Graphics, and Image Processing, 26(2):242–255, May 1984.

Joseph O'Rourke, S. R. Kosaraju, and N. Megiddo.
Computing circular separability.
Discrete and Computational Geometry, 1(1):105–113, 1986.

F. P. Preparata and M. I. Shamos.
Computational Geometry : An Introduction.
Springer-Verlag, 1985.

P. Sauer.
On the recognition of digital circles in linear time.
Computational Geometry: Theory and Application 2, pages 287–302, 1993.

S. M. Thomas and Y. T. Chan.
A simple approach for the estimation of circular arc center and its radius.
Computer Vision, Graphics and Image Processing, 45:362–370, 1989.

M. Worring and W.M. Smeulders.
Digitized circular arcs: characterization and parameter estimation.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(6):587–598, jun 1995.