Geometry, Vision, and Graphics:

A Number-theoretic Introduction

Partha Bhowmick

CSE, IIT Kharagpur

RESEARCH PROMOTION WORKSHOP
INTRODUCTION TO GRAPH AND GEOMETRIC ALGORITHMS
14-16 MARCH 2013 (BESU SHIBPUR)

Leap year — An exception...

Is 1900 a leap year?

No!

An exception to exception: $1900 \mod 100 = 0$

Observation!?

Years ending with "00" are not leap years.

Is 2000 a leap year?

Yes!

An exception to exception to exception: $2000 \mod 400 = 0$

Non-non-leap years: $2000, 2400, 2800, \ldots$

Leap year — An exception...

Is 1900 a leap year?

No!

An exception to exception: $1900 \mod 100 = 0$

Observation!?

Years ending with "00" are not leap years.

Is 2000 a leap year?

Yes!

An exception to exception to exception: $2000 \mod 400 = 0$

Non-non-leap years: $2000, 2400, 2800, \ldots$

Leap year — An exception...

Is 1900 a leap year?

No!

An exception to exception: $1900 \mod 100 = 0$

Observation!?

Years ending with "00" are not leap years.

Is 2000 a leap year?

Yes!

An exception to exception to exception: $2000 \mod 400 = 0$

Non-non-leap years: $2000, 2400, 2800, \ldots$

Leap year — An exception...

Is 1900 a leap year?

No!
An exception to exception: $1900 \mod 100 = 0$

Observation!?

Years ending with “00” are not leap years.

Is 2000 a leap year?

Yes!
An exception to exception to exception: $2000 \mod 400 = 0$
Non-non-leap years: $2000, 2400, 2800, \ldots$

Leap year — An exception...

Is 1900 a leap year?

No!

An exception to exception: $1900 \mod 100 = 0$

Observation!?

Years ending with "00" are not leap years.

Is 2000 a leap year?

Yes!

An exception to exception to exception: $2000 \mod 400 = 0$

Non-non-leap years: $2000, 2400, 2800, \ldots$

Leap year — An exception...

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Is 1900 a leap year?

No!

An exception to exception: $1900 \mod 100 = 0$

Observation!?

Years ending with "00" are not leap years.

Is 2000 a leap year?

Yes!

An exception to exception to exception: $2000 \mod 400 = 0$

Non-non-leap years: $2000, 2400, 2800, \ldots$

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness

Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Leap year — An exception...

Is 1900 a leap year?

No!

An exception to exception: $1900 \mod 100 = 0$

Observation!?

Years ending with “00” are not leap years.

Is 2000 a leap year?

Yes!

An exception to exception to exception: $2000 \mod 400 = 0$

Non-non-leap years: $2000, 2400, 2800, \ldots$

Leap year — An exception...

Is 1900 a leap year?

No!

An exception to exception: $1900 \mod 100 = 0$

Observation!?

Years ending with “00” are not leap years.

Is 2000 a leap year?

Yes!

An exception to exception to exception: $2000 \mod 400 = 0$

Non-non-leap years: 2000, 2400, 2800, . . .

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Leap year — An exception...

Is 1900 a leap year?

No!

An exception to exception: $1900 \mod 100 = 0$

Observation!?

Years ending with "00" are not leap years.

Is 2000 a leap year?

Yes!

An exception to exception to exception: $2000 \mod 400 = 0$

Non-non-leap years: $2000, 2400, 2800, \ldots$

Leap year — An exception...

Is 1900 a leap year?

No!

An exception to exception: $1900 \mod 100 = 0$

Observation!?

Years ending with “00” are not leap years.

Is 2000 a leap year?

Yes!

An exception to exception to exception: $2000 \mod 400 = 0$

Non-non-leap years: $2000, 2400, 2800, \ldots$

Leap year — An exception...

Is 1900 a leap year?

No!
An exception to exception: $1900 \mod 100 = 0$

Observation!?

Years ending with “00” are not leap years.

Is 2000 a leap year?

Yes!
An exception to exception to exception: $2000 \mod 400 = 0$
Non-non-leap years: $2000, 2400, 2800, \ldots$

Leap year — An exception...

Is 1900 a leap year?

No!

An exception to exception: $1900 \mod 100 = 0$

Observation!?

Years ending with "00" are not leap years.

Is 2000 a leap year?

Yes!

An exception to exception to exception: $2000 \mod 400 = 0$

Non-non-leap years: $2000, 2400, 2800, \ldots$

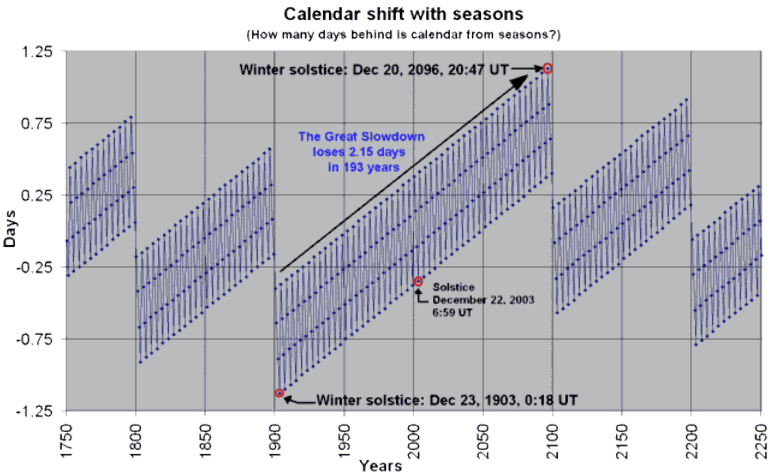
Discretization of Gregorian calendar I

Algorithm to determine leap years[1]

```
if (year % 400 == 0)
    then leap
else if (year % 100 == 0)
    then no leap
else if (year % 4 == 0)
    then leap
else no leap
```

[1] includes leap years before the official inception in 1582; one average year = 365d 5h 49m 12s.

Discretization of Gregorian calendar I



Leap years

Regular year	$\rightarrow$	365 days — 2010
Leap year	$\rightarrow$	366 days — E[2] — 2012
Non-leap year	$\rightarrow$	365 days — E to E — 2100
Non-non-leap year	$\rightarrow$	366 days — E to E to E — 2000

Decade	**Century**	**Millennium**	**Billennium**
Leap	Non-leap	Non-non-leap	Non-non-non-leap
366	365	366	365

?
exception to
exception to
exception to
exception to
⋮

[2] E = Exception

Leap years

Regular year	$\rightarrow$	365 days — 2010
Leap year	$\rightarrow$	366 days — E[2] — 2012
Non-leap year	$\rightarrow$	365 days — E to E — 2100
Non-non-leap year	$\rightarrow$	366 days — E to E to E — 2000

Decade	Century	Millennium	Billennium
Leap	Non-leap	Non-non-leap	Non-non-non-leap
366	365	366	365

?
exception to
exception to
exception to
exception to
⋮

[2] E = Exception

Leap years

Regular year	$\rightarrow$	365 days — 2010
Leap year	$\rightarrow$	366 days — E[2] — 2012
Non-leap year	$\rightarrow$	365 days — E to E — 2100
Non-non-leap year	$\rightarrow$	366 days — E to E to E — 2000

Decade	Century	Millennium	Billennium
Leap	Non-leap	Non-non-leap	Non-non-non-leap
366	365	366	365

?
exception to
exception to
exception to
exception to
⋮

[2] E = Exception

Leap years

Regular year	→	365 days — 2010
Leap year	→	366 days — E[2] — 2012
Non-leap year	→	365 days — E to E — 2100
Non-non-leap year	→	366 days — E to E to E — 2000

Decade	**Century**	**Millennium**	**Billennium**
Leap	Non-leap	Non-non-leap	Non-non-non-leap
366	365	366	365

?
exception to
exception to
exception to
exception to
⋮

[2] E = Exception

Leap years

Regular year	$\rightarrow$	365 days — 2010
Leap year	$\rightarrow$	366 days — E[2] — 2012
Non-leap year	$\rightarrow$	365 days — E to E — 2100
Non-non-leap year	$\rightarrow$	366 days — E to E to E — 2000

Decade	**Century**	**Millennium**	**Billennium**
Leap	Non-leap	Non-non-leap	Non-non-non-leap
366	365	366	365

?
exception to
exception to
exception to
exception to
⋮

[2] E = Exception

Leap years

Regular year	$\rightarrow$	365 days — 2010
Leap year	$\rightarrow$	366 days — E[2] — 2012
Non-leap year	$\rightarrow$	365 days — E to E — 2100
Non-non-leap year	$\rightarrow$	366 days — E to E to E — 2000

Decade	**Century**	**Millennium**	**Billennium**
Leap	Non-leap	Non-non-leap	Non-non-non-leap
366	365	366	365

?
exception to
exception to
exception to
exception to
⋮

[2] E = Exception

Leap years

Regular year	$\rightarrow$	365 days — 2010
Leap year	$\rightarrow$	366 days — E[2] — 2012
Non-leap year	$\rightarrow$	365 days — E to E — 2100
Non-non-leap year	$\rightarrow$	366 days — E to E to E — 2000

Decade	**Century**	**Millennium**	**Billennium**
Leap	Non-leap	Non-non-leap	Non-non-non-leap
366	365	366	365

?
exception to
exception to
exception to
exception to
⋮

[2] E = Exception

Leap years

Regular year	→	365 days — 2010
Leap year	→	366 days — E[2] — 2012
Non-leap year	→	365 days — E to E — 2100
Non-non-leap year	→	366 days — E to E to E — 2000

Decade	**Century**	**Millennium**	**Billennium**
Leap	Non-leap	Non-non-leap	Non-non-non-leap
366	365	366	365

?
exception to
exception to
exception to
exception to
⋮

[2] E = Exception

Leap years

Regular year	→	365 days — 2010
Leap year	→	366 days — E[2] — 2012
Non-leap year	→	365 days — E to E — 2100
Non-non-leap year	→	366 days — E to E to E — 2000

Decade	**Century**	**Millennium**	**Billennium**
Leap	Non-leap	Non-non-leap	Non-non-non-leap
366	365	366	365

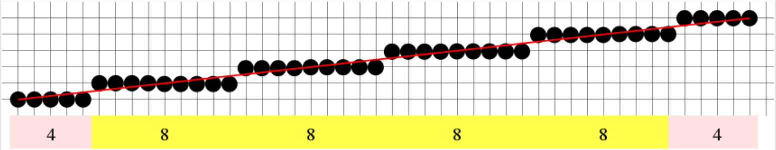
?

exception to
exception to
exception to
exception to
⋮

[2] E = Exception

Where and why lies the exception

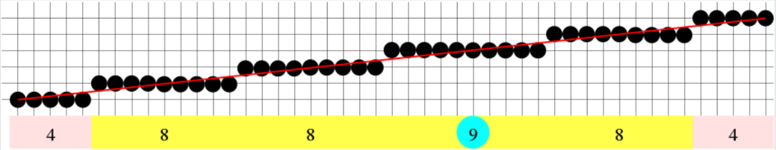
Example



slope $= 5/45$: $8\,8\,8\,8 \Rightarrow$ no exception!

Where and why lies the exception

Example



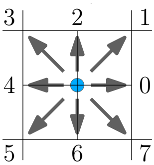
slope $= 5/46$: $8\,8\,9\,8 \Rightarrow 9 =$ exception.

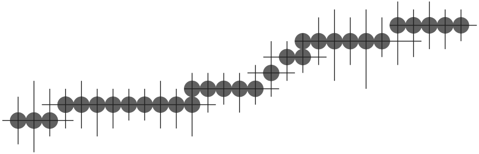
Definitions

Definition

Digital curve A sequence C of points in which each point is an 8-neighbor of its predecessor in C.

C is irreducible iff it does not remain 8-connected after removing a point that is not its end point.



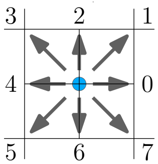


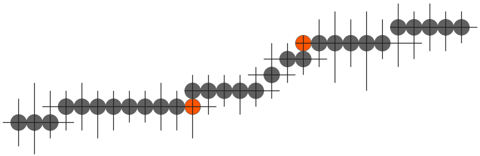
Definitions

Definition

Digital curve A sequence C of points in which each point is an 8-neighbor of its predecessor in C.

C is irreducible iff it does not remain 8-connected after removing a point that is not its end point.





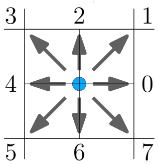
Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

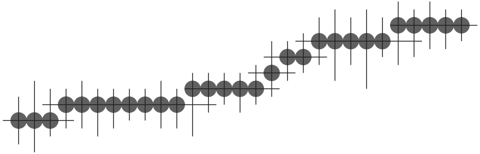
Definitions

Definition

Digital curve A sequence C of points in which each point is an 8-neighbor of its predecessor in C.

C is irreducible iff it does not remain 8-connected after removing a point that is not its end point.





Definitions

Definition

Ray $\gamma_{\alpha,\beta} = \{(x, \alpha x + \beta) \in \mathbb{R}^2 : 0 \leq x < \infty\}$.

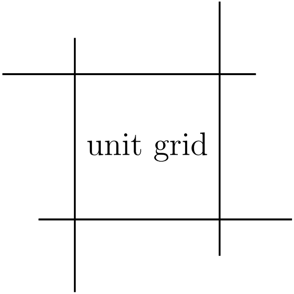
Digital Ray $I_{\alpha,\beta} = \{(n, I_n) \in \mathbb{Z}^2 : n \geqslant 0 \ \wedge I_n = \lfloor \alpha n + \beta + 0.5 \rfloor\}$, considering $0 \leqslant \alpha \leqslant 1$, w.l.o.g.

Definitions

Definition

Ray $\gamma_{\alpha,\beta} = \{(x, \alpha x + \beta) \in \mathbb{R}^2 : 0 \leq x < \infty\}$.

Digital Ray $I_{\alpha,\beta} = \{(n, I_n) \in \mathbb{Z}^2 : n \geqslant 0 \ \wedge I_n = \lfloor \alpha n + \beta + 0.5 \rfloor\}$, considering $0 \leqslant \alpha \leqslant 1$, w.l.o.g.

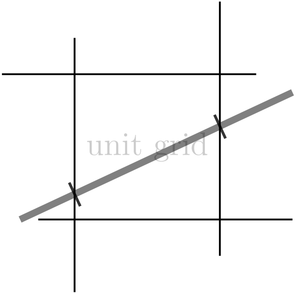


Definitions

Definition

Ray $\gamma_{\alpha,\beta} = \{(x, \alpha x + \beta) \in \mathbb{R}^2 : 0 \leq x < \infty\}$.

Digital Ray $I_{\alpha,\beta} = \{(n, I_n) \in \mathbb{Z}^2 : n \geqslant 0 \ \wedge I_n = \lfloor \alpha n + \beta + 0.5 \rfloor\}$, considering $0 \leqslant \alpha \leqslant 1$, w.l.o.g.



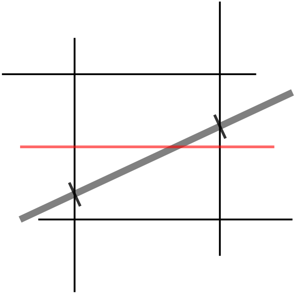
Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Definitions

Definition

Ray $\gamma_{\alpha,\beta} = \{(x, \alpha x + \beta) \in \mathbb{R}^2 : 0 \leq x < \infty\}$.

Digital Ray $l_{\alpha,\beta} = \{(n, l_n) \in \mathbb{Z}^2 : n \geqslant 0 \ \wedge l_n = \lfloor \alpha n + \beta + 0.5 \rfloor\}$, considering $0 \leqslant \alpha \leqslant 1$, w.l.o.g.

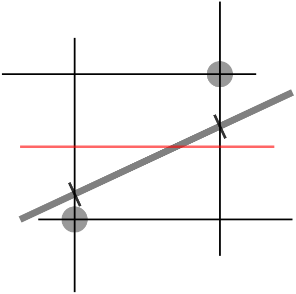


Definitions

Definition

Ray $\gamma_{\alpha,\beta} = \{(x, \alpha x + \beta) \in \mathbb{R}^2 : 0 \leq x < \infty\}$.

Digital Ray $I_{\alpha,\beta} = \{(n, I_n) \in \mathbb{Z}^2 : n \geqslant 0 \ \wedge I_n = \lfloor \alpha n + \beta + 0.5 \rfloor\}$, considering $0 \leqslant \alpha \leqslant 1$, w.l.o.g.



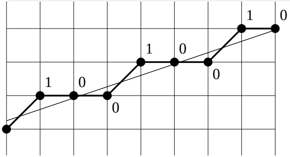
Definitions

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Definition

Ray $\gamma_{\alpha,\beta} = \{(x, \alpha x + \beta) \in \mathbb{R}^2 : 0 \leq x < \infty\}$.

Digital Ray $I_{\alpha,\beta} = \{(n, I_n) \in \mathbb{Z}^2 : n \geqslant 0 \ \wedge I_n = \lfloor \alpha n + \beta + 0.5 \rfloor\}$, considering $0 \leqslant \alpha \leqslant 1$, w.l.o.g.



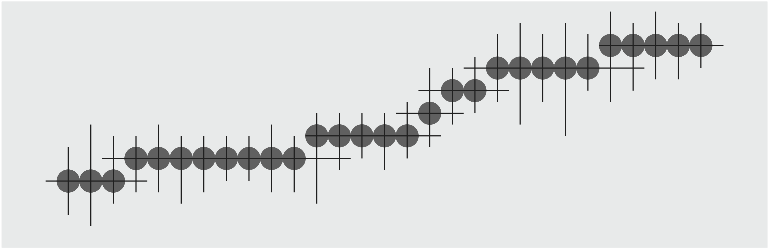
chain code $= \ldots 10010010 \ldots$

Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

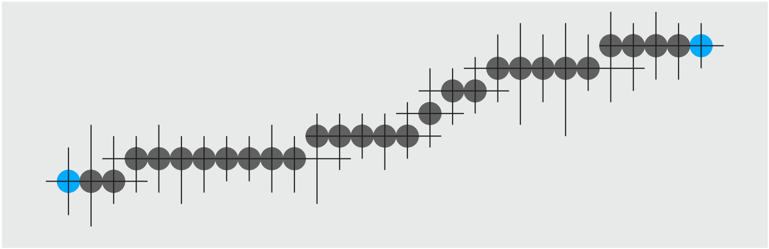


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

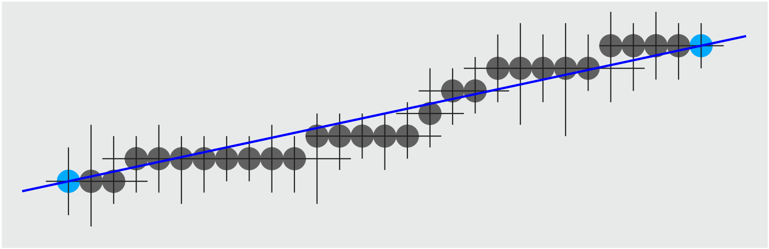


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

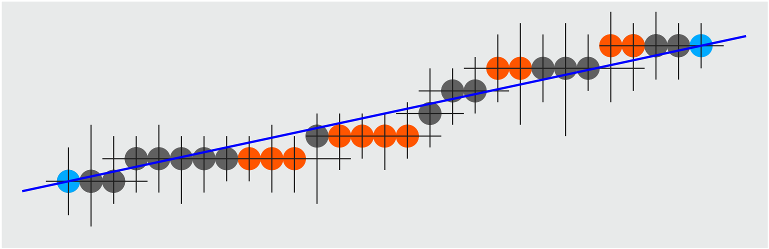


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

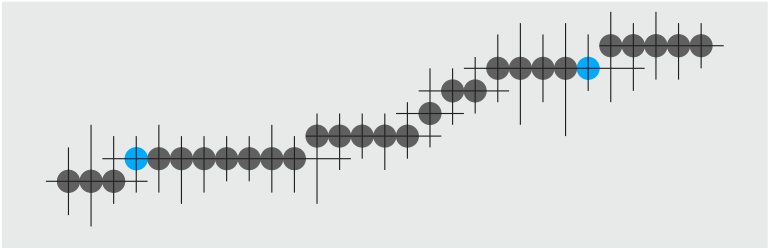


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

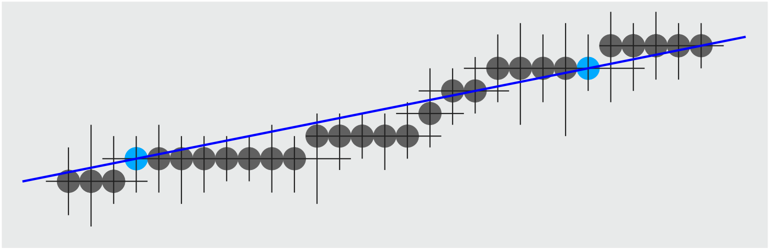


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

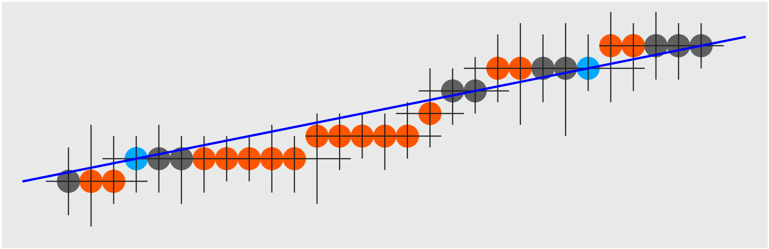


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

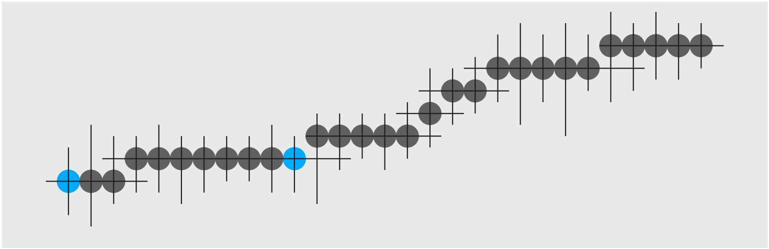


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

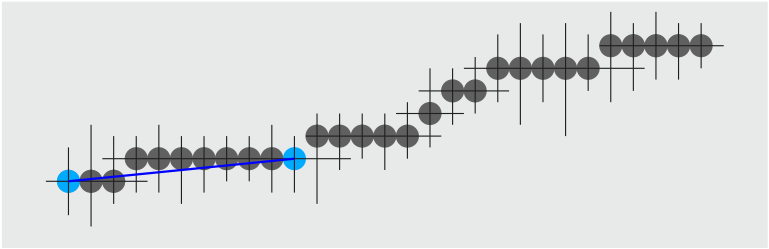


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

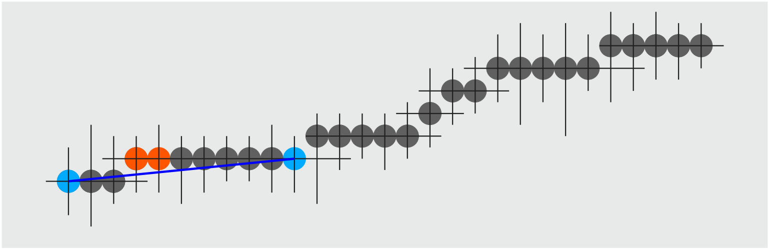


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

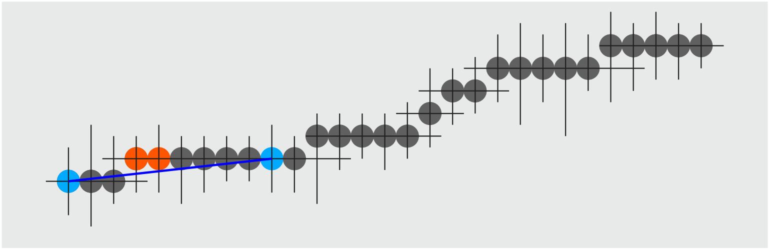


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

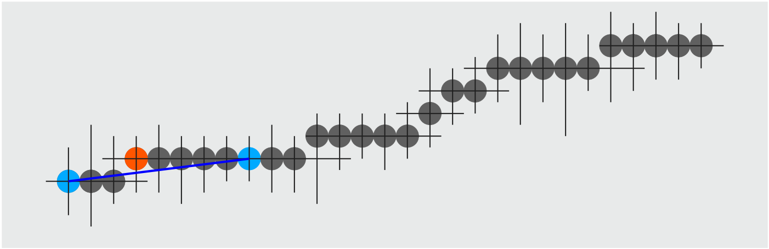


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

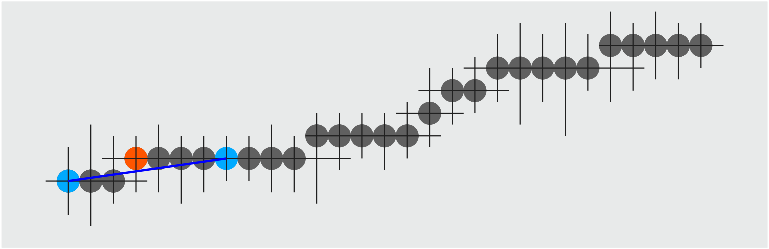


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

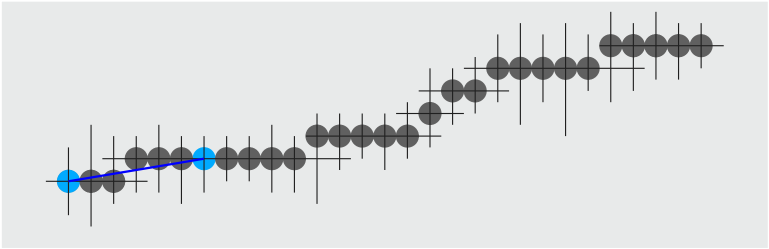


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

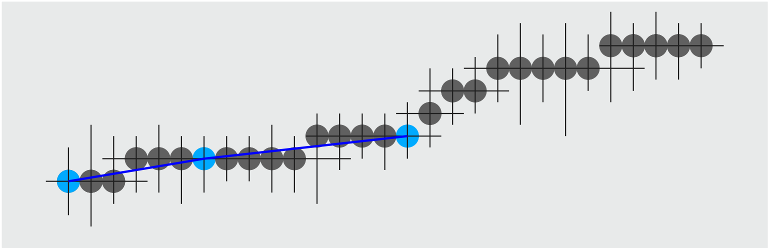


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example



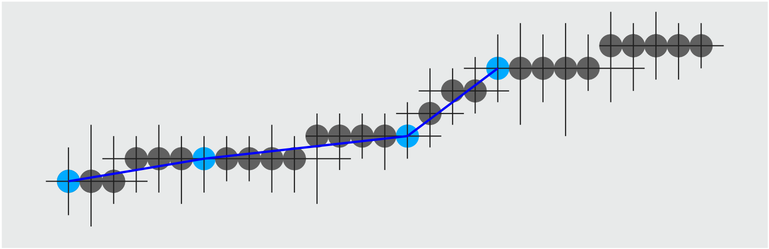
Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

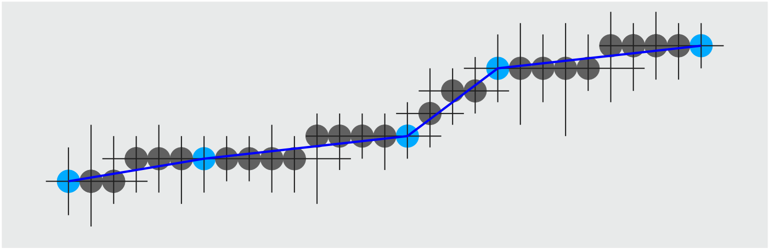


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

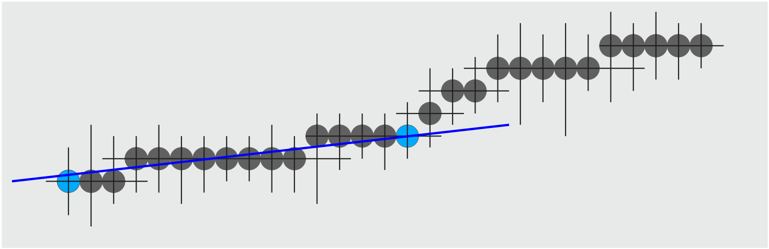


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

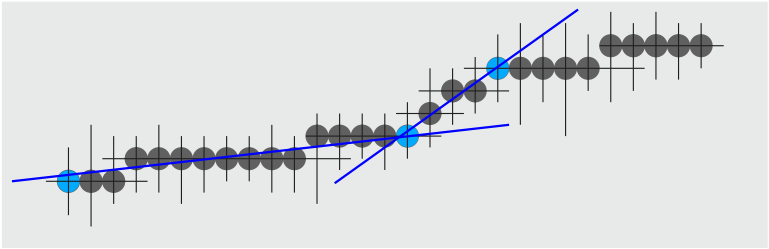


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example

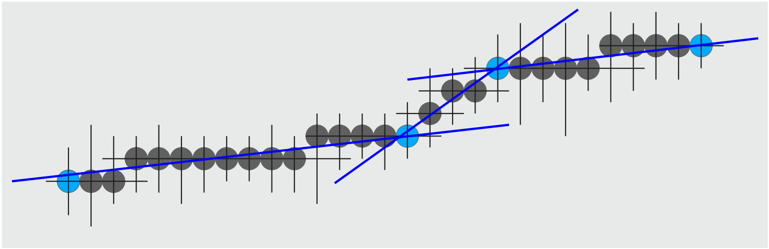


Determining the Digital Straightness

Problem statement

Given a sequence S of digital points, decide whether *there exists a real (Euclidean) line* whose digitization produces S.

Example



Definitions

Word: A (finite) word $u = a_1 a_2 \dots a_n$ over the alphabet A is a finite sequence of elements of A. $n = |u|$ is called the length of u.

Example: $A = \{0, 1\}, u = 010110, |u| = 6$.

Factor: A word v is a factor of u iff there exist words v_1, v_2 such that $u = v_1 v v_2$.

Example: $\underline{01}0110, 0\underline{10}110, \dots$

Period: $\min\{k : a_i = a_{i+k} \text{ for } i = 1, \dots, n-k\}$ is the period of a word $u = a_1 a_2 \dots a_n$.

Example: 2 is the period of 01010101.

Periodic word: An infinite word w is periodic if $w = v^\omega$, for some $v \in A^* \setminus \{\epsilon\}$.

Example: $\underline{0001}00010001\dots$

Definitions

Word: A (finite) word $u = a_1 a_2 \ldots a_n$ over the alphabet A is a finite sequence of elements of A. $n = |u|$ is called the length of u.

Example: $A = \{0, 1\}, u = 010110, |u| = 6.$

Factor: A word v is a factor of u iff there exist words v_1, v_2 such that $u = v_1 v v_2$.

Example: $\underline{01}0110, 0\underline{10}110, \ldots$

Period: $\min\{k : a_i = a_{i+k} \text{ for } i = 1, \ldots, n-k\}$ is the period of a word $u = a_1 a_2 \ldots a_n$.

Example: 2 is the period of 01010101.

Periodic word: An infinite word w is periodic if $w = v^\omega$, for some $v \in A^* \setminus \{\epsilon\}$.

Example: $\underline{0001}00010001 \ldots$

Definitions

Word: A (finite) word $u = a_1 a_2 \ldots a_n$ over the alphabet A is a finite sequence of elements of A.
$n = |u|$ is called the length of u.

Example: $A = \{0, 1\}, u = 010110, |u| = 6$.

Factor: A word v is a factor of u iff there exist words v_1, v_2 such that $u = v_1 v v_2$.

Example: $\underline{01}0110, 0\underline{10}110, \ldots$

Period: $\min\{k : a_i = a_{i+k} \text{ for } i = 1, \ldots, n-k\}$ is the period of a word $u = a_1 a_2 \ldots a_n$.

Example: 2 is the period of 01010101.

Periodic word: An infinite word w is periodic if $w = v^\omega$, for some $v \in A^* \setminus \{\epsilon\}$.

Example: $\underline{0001}00010001 \ldots$

Definitions

Word: A (finite) word $u = a_1 a_2 \ldots a_n$ over the alphabet A is a finite sequence of elements of A.
$n = |u|$ is called the length of u.

Example: $A = \{0, 1\}, u = 010110, |u| = 6$.

Factor: A word v is a factor of u iff there exist words v_1, v_2 such that $u = v_1 v v_2$.

Example: $\underline{01}0110, 0\underline{10}110, \ldots$

Period: $\min\{k : a_i = a_{i+k} \text{ for } i = 1, \ldots, n-k\}$ is the period of a word $u = a_1 a_2 \ldots a_n$.

Example: 2 is the period of 01010101.

Periodic word: An infinite word w is periodic if $w = v^\omega$, for some $v \in A^* \setminus \{\epsilon\}$.

Example: $\underline{0001}00010001 \ldots$

Definitions

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Eventually periodic: w is eventually periodic if $\exists\ u \in A^*$ and $v \in A^* - \{\epsilon\}$ s.t. $w = uv^\omega$.

Example: 01<u>0001</u>00010001 . . .

Aperiodic: w is aperiodic if it is not eventually periodic.

Example: 0101100100010111001 . . .

Definitions

Eventually periodic: w is eventually periodic if $\exists\ u \in A^*$ and $v \in A^* - \{\epsilon\}$ s.t. $w = uv^\omega$.

Example: $01\underline{0001}00010001\ldots$

Aperiodic: w is aperiodic if it is not eventually periodic.

Example: $0101100100010111 1001\ldots$

Rational vs. irrational slopes I

Theorem ([R. Brons, 1974])

Rational digital rays are periodic and irrational digital rays are aperiodic.

Example ([J.-P. Reveillès, 1991])

DSS with slope $\frac{2}{5}$: Period can be expressed as 01010, 00101, 10010, 01001, or 10100.
Which of these periods is chosen is not important, because the bounds of the period can be placed anywhere.

Rational vs. irrational slopes II

Theorem ([J.-P. Reveillès, 1991])

A word $u \in \{0,1\}^*$ is a DSS iff the corresponding digital points lie on or between two parallel real lines having a y-distance less than 1.

Theorem (Bruckstein 1991)

For irrational α, $I_{\alpha,\beta}$ uniquely determines both α and β. For rational α, $I_{\alpha,\beta}$ uniquely determines α, and β is determined up to an interval.[a]

[a] A. M. Bruckstein. **Self-similarity Properties of Digitized Straight Lines.** *Contemp. Math.*, 119:1–20, 1991.

Chain code properties

Theorem ([H. Freeman, 1970])

A chain code sequence should possess the following properties if it is a DSS:

(F1) *at most two types of elements can be present, and these can differ only by unity, modulo eight;*

(F2) *one of the two element values always occurs singly;*

(F3) *successive occurrences of the element occurring singly are as uniformly spaced as possible.*

Example

	0112112101	0110010010	0100010100	0010010010
F1	×	0110010010	0100010100	0010010010
F2	×	×	0100010100	0010010010
F3	×	×	×	0010010010

Chain code properties

Theorem ([H. Freeman, 1970])

A chain code sequence should possess the following properties if it is a DSS:

(F1) *at most two types of elements can be present, and these can differ only by unity, modulo eight;*

(F2) *one of the two element values always occurs singly;*

(F3) *successive occurrences of the element occurring singly are as uniformly spaced as possible.*

Example

	0112112101	0110010010	0100010100	0010010010
F1	×	0110010010	0100010100	0010010010
F2	✓	×	0100010100	0010010010
F3	×	×	×	0010010010

Chain code properties

Theorem ([H. Freeman, 1970])

A chain code sequence should possess the following properties if it is a DSS:

(F1) *at most two types of elements can be present, and these can differ only by unity, modulo eight;*

(F2) *one of the two element values always occurs singly;*

(F3) *successive occurrences of the element occurring singly are as uniformly spaced as possible.*

Example

	0112112101	0110010010	0100010100	0010010010
F1	×	0110010010	0100010100	0010010010
F2			0100010100	0010010010
F3	×	✓	✓	0010010010

Chain code properties

Theorem ([H. Freeman, 1970])

A chain code sequence should possess the following properties if it is a DSS:

(F1) *at most two types of elements can be present, and these can differ only by unity, modulo eight;*

(F2) *one of the two element values always occurs singly;*

(F3) *successive occurrences of the element occurring singly are as **uniformly spaced** as possible.*

Example

	0112112101	0110010010	0100010100	0010010010
F1	×	0110010010	0100010100	0010010010
F2	×	×	0100010100	0010010010
F3	×	×	×	0010010010

Chain code properties

Theorem ([H. Freeman, 1970])

A chain code sequence should possess the following properties if it is a DSS:

(F1) *at most two types of elements can be present, and these can differ only by unity, modulo eight;*

(F2) *one of the two element values always occurs singly;*

(F3) *successive occurrences of the element occurring singly are as uniformly spaced as possible.*

Example

	0112112101	0110010010	0100010100	0010010010
F1	×	0110010010	0100010100	0010010010
F2	×	×	0100010100	0010010010
F3	×	×	×	0010010010

Chain code properties

Theorem ([H. Freeman, 1970])

A chain code sequence should possess the following properties if it is a DSS:

(F1) *at most two types of elements can be present, and these can differ only by unity, modulo eight;*

(F2) *one of the two element values always occurs singly;*

(F3) *successive occurrences of the element occurring singly are as uniformly spaced as possible.*

Example

	0112112101	0110010010	0100010100	0010010010
F1	×	0110010010	0100010100	0010010010
F2	×	×	0100010100	0010010010
F3	×	×	×	0010010010

Chain code properties

Theorem ([H. Freeman, 1970])

A chain code sequence should possess the following properties if it is a DSS:

(F1) *at most two types of elements can be present, and these can differ only by unity, modulo eight;*

(F2) *one of the two element values always occurs singly;*

(F3) *successive occurrences of the element occurring singly are as uniformly spaced as possible.*

Example

	0112112101	0110010010	0100010100	0010010010
F1	×	0110010010	0100010100	0010010010
F2	×	×	0100010100	0010010010
F3	×	×	×	0010010010

Chain code properties

Theorem ([H. Freeman, 1970])

A chain code sequence should possess the following properties if it is a DSS:

(F1) *at most two types of elements can be present, and these can differ only by unity, modulo eight;*

(F2) *one of the two element values always occurs singly;*

(F3) *successive occurrences of the element occurring singly are as uniformly spaced as possible.*

Example

	0112112101	0110010010	0100010100	0010010010
F1	×	0110010010	0100010100	0010010010
F2	×	×	0100010100	0010010010
F3	×	×	×	0010010010

Chain code properties

Theorem ([H. Freeman, 1970])

A chain code sequence should possess the following properties if it is a DSS:

(F1) *at most two types of elements can be present, and these can differ only by unity, modulo eight;*

(F2) *one of the two element values always occurs singly;*

(F3) *successive occurrences of the element occurring singly are as uniformly spaced as possible.*

Example

	0112112101	0110010010	0100010100	0010010010
F1	×	0110010010	0100010100	0010010010
F2	×	×	0100010100	0010010010
F3	×	×	×	0010010010

Chain code properties

Algorithm [R. Brons, 1974]

Brons proposed grammars for chain code generation of rational digital rays based on criteria F1, F2, and F3.

Improvement [A. Rosenfeld, 1974]

- F3 is not suitable for a formal proof.
- Rosenfeld provided a formal characterization of DSS which also allowed a further specification of F3.

Chain code properties

Algorithm [R. Brons, 1974]

Brons proposed grammars for chain code generation of rational digital rays based on criteria F1, F2, and F3.

Improvement [A. Rosenfeld, 1974]

- F3 is not suitable for a formal proof.
- Rosenfeld provided a formal characterization of DSS which also allowed a further specification of F3.

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Chain code properties

Algorithm [R. Brons, 1974]

Brons proposed grammars for chain code generation of rational digital rays based on criteria F1, F2, and F3.

Improvement [A. Rosenfeld, 1974]

- F3 is not suitable for a formal proof.
- Rosenfeld provided a formal characterization of DSS which also allowed a further specification of F3.

Chain code properties

Algorithm [R. Brons, 1974]

Brons proposed grammars for chain code generation of rational digital rays based on criteria F1, F2, and F3.

Improvement [A. Rosenfeld, 1974]

- F3 is not suitable for a formal proof.
- Rosenfeld provided a formal characterization of DSS which also allowed a further specification of F3.

Chain code properties

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness

Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Theorem ([A. Rosenfeld, 1974])

Necessary conditions for (the chain code sequences of) digital straight segments [A run is a maximum-length factor a^n, for $a \in A$.]

(R1) *The runs have at most two directions, differing by 45^0, and for one of these directions, the run length must be 1.*

(R2) *The runs can have only two lengths, which are consecutive integers.*

(R3) *One of the runs can occur only once at a time.*

(R4) *... for the run length that occurs in runs, these runs can themselves have only two lengths, which are consecutive integers; and so on.*

Chain code properties

Theorem ([A. Rosenfeld, 1974])

Necessary conditions for (the chain code sequences of) digital straight segments [A run is a maximum-length factor a^n, for $a \in A$.]

(R1) *The runs have at most two directions, differing by 45^0, and for one of these directions, the run length must be 1.*

(R2) *The runs can have only two lengths, which are consecutive integers.*

(R3) *One of the runs can occur only once at a time.*

(R4) *... for the run length that occurs in runs, these runs can themselves have only two lengths, which are consecutive integers; and so on.*

Chain code properties

Theorem ([A. Rosenfeld, 1974])

Necessary conditions for (the chain code sequences of) digital straight segments [A run is a maximum-length factor a^n, for $a \in A$.]

(R1) *The runs have at most two directions, differing by 45^0, and for one of these directions, the run length must be 1.*

(R2) *The runs can have only two lengths, which are consecutive integers.*

(R3) *One of the runs can occur only once at a time.*

(R4) *... for the run length that occurs in runs, these runs can themselves have only two lengths, which are consecutive integers; and so on.*

Chain code properties

Theorem ([A. Rosenfeld, 1974])

Necessary conditions for (the chain code sequences of) digital straight segments [A run is a maximum-length factor a^n, for $a \in A$.]

(R1) *The runs have at most two directions, differing by 45^0, and for one of these directions, the run length must be 1.*

(R2) *The runs can have only two lengths, which are consecutive integers.*

(R3) *One of the runs can occur only once at a time.*

(R4) *... for the run length that occurs in runs, these runs can themselves have only two lengths, which are consecutive integers; and so on.*

Chain code properties

Theorem ([A. Rosenfeld, 1974])

Necessary conditions for (the chain code sequences of) digital straight segments [A run is a maximum-length factor a^n, for $a \in A$.]

(R1) *The runs have at most two directions, differing by 45^0, and for one of these directions, the run length must be 1.*

(R2) *The runs can have only two lengths, which are consecutive integers.*

(R3) *One of the runs can occur only once at a time.*

(R4) *... for the run length that occurs in runs, these runs can themselves have only two lengths, which are consecutive integers; and so on.*

Number-theoretic properties

Continued Fraction

Let slope of a DSS $= a_1/a_0$ $(a_0 > a_1 > 1; a_0, a_1 \in \mathbb{Z})$.

$$\frac{a_1}{a_0} = \frac{1}{q_1 + \cdots}$$

$$= [q_1, q_2, \ldots, q_n] \text{ (Euclidean algorithm)}$$

Example

$$46/87 = \frac{1}{1 + \frac{1}{1+\frac{1}{8+\frac{1}{5}}}} = [1, 1, 8, 5].$$

Number-theoretic properties

Continued Fraction

Let slope of a DSS $= a_1/a_0$ ($a_0 > a_1 > 1; a_0, a_1 \in \mathbb{Z}$).

$$\frac{a_1}{a_0}$$

$$= [q_1, q_2, \ldots, q_n] \text{ (Euclidean algorithm)}$$

Example

$$46/87 = \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{8 + \cfrac{1}{5}}}} = [1, 1, 8, 5].$$

Number-theoretic properties

Continued Fraction

Let slope of a DSS $= a_1/a_0$ ($a_0 > a_1 > 1; a_0, a_1 \in \mathbb{Z}$).

$$\frac{a_1}{a_0} = \frac{1}{q_1 + \frac{1}{\cdots}}$$

$$= [q_1, q_2, \ldots, q_n] \text{ (Euclidean algorithm)}$$

Example

$$46/87 = \frac{1}{1 + \frac{1}{1 + \frac{1}{8 + \frac{1}{5}}}} = [1, 1, 8, 5].$$

Number-theoretic properties

Continued Fraction

Let slope of a DSS $= a_1/a_0$ ($a_0 > a_1 > 1; a_0, a_1 \in \mathbb{Z}$).

$$\frac{a_1}{a_0} = \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cdots}}$$

$$= [q_1, q_2, \ldots, q_n] \text{ (Euclidean algorithm)}$$

Example

$$46/87 = \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{8 + \frac{1}{5}}}} = [1, 1, 8, 5].$$

Number-theoretic properties

Continued Fraction

Let slope of a DSS $= a_1/a_0$ ($a_0 > a_1 > 1; a_0, a_1 \in \mathbb{Z}$).

$$\frac{a_1}{a_0} = \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{\ddots \cfrac{1}{q_{n-1} + \cfrac{1}{q_n}}}}}$$

$$= [q_1, q_2, \ldots, q_n] \text{ (Euclidean algorithm)}$$

Example

$$46/87 = \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{8 + \cfrac{1}{5}}}} = [1, 1, 8, 5].$$

Number-theoretic properties

Continued Fraction

Let slope of a DSS $= a_1/a_0$ $(a_0 > a_1 > 1; a_0, a_1 \in \mathbb{Z})$.

$$\frac{a_1}{a_0} = \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{\ddots \, \cfrac{1}{q_{n-1} + \cfrac{1}{q_n}}}}}$$

$$= [q_1, q_2, \ldots, q_n] \text{ (Euclidean algorithm)}$$

Example

$$46/87 = \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{8 + \cfrac{1}{5}}}} = [1, 1, 8, 5].$$

Number-theoretic
P. Bhowmick
Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Continued Fraction

Let slope of a DSS $= a_1/a_0$ ($a_0 > a_1 > 1; a_0, a_1 \in \mathbb{Z}$).

$$\frac{a_1}{a_0} = \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{\ddots \cfrac{1}{q_{n-1} + \cfrac{1}{q_n}}}}}$$

$= [q_1, q_2, \ldots, q_n]$ (Euclidean algorithm)

Example

$$46/87 = \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{8 + \cfrac{1}{5}}}} = [1, 1, 8, 5].$$

Number-theoretic properties

Continued Fraction

Let slope of a DSS $= a_1/a_0$ $(a_0 > a_1 > 1; a_0, a_1 \in \mathbb{Z})$.

$$\frac{a_1}{a_0} = \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{\ddots \cfrac{1}{q_{n-1} + \frac{1}{q_n}}}}}$$

$= [q_1, q_2, \ldots, q_n]$ (Euclidean algorithm)

Example

$$46/87 = \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{8 + \frac{1}{5}}}} = [1, 1, 8, 5].$$

Number-theoretic properties

Continued Fraction

Let slope of a DSS $= a_1/a_0$ ($a_0 > a_1 > 1$; $a_0, a_1 \in \mathbb{Z}$).

$$\frac{a_1}{a_0} = \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{\ddots \; \cfrac{1}{q_{n-1} + \cfrac{1}{q_n}}}}}$$

$$= [q_1, q_2, \ldots, q_n] \text{ (Euclidean algorithm)}$$

Example

$$46/87 = \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{8 + \cfrac{1}{5}}}} = [1, 1, 8, 5].$$

Number-theoretic properties

Continued Fraction

Let slope of a DSS $= a_1/a_0$ $(a_0 > a_1 > 1; a_0, a_1 \in \mathbb{Z})$.

$$\frac{a_1}{a_0} = \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{\ddots \cfrac{1}{q_{n-1} + \cfrac{1}{q_n}}}}}$$

$$= [q_1, q_2, \ldots, q_n] \text{ (Euclidean algorithm)}$$

Example

$$46/87 = \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{8 + \cfrac{1}{5}}}} = [1, 1, 8, 5].$$

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Splitting a Continued Fraction

$$\frac{a1}{a0} = [q_1, q_2, \ldots, q_n]$$

$$= \frac{\alpha_n q_n + \beta_n}{\gamma_n q_n + \delta_n}$$

where α_ns are defined by $q_1, q_2, \ldots, q_n$

$$= \frac{(\alpha_{n-1} q_{n-1} + \beta_{n-1}) q_n + \alpha_{n-1}}{(\gamma_{n-1} q_{n-1} + \delta_{n-1}) q_n + \gamma_{n-1}}$$

$$= \frac{(\alpha_{n-1} q_{n-1} + \beta_{n-1})(q_n - 1) + \alpha_{n-1}(q_{n-1} + 1) + \beta_{n-1}}{(\gamma_{n-1} q_{n-1} + \delta_{n-1})(q_n - 1) + \gamma_{n-1}(q_{n-1} + 1) + \delta_{n-1}}.$$

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Splitting a Continued Fraction

$$\frac{a1}{a0} = [q_1, q_2, \ldots, q_n]$$

$$= \frac{\alpha_n q_n + \beta_n}{\gamma_n q_n + \delta_n}$$

where α_ns are defined by $q_1, q_2, \ldots, q_n$

$$= \frac{(\alpha_{n-1} q_{n-1} + \beta_{n-1}) q_n + \alpha_{n-1}}{(\gamma_{n-1} q_{n-1} + \delta_{n-1}) q_n + \gamma_{n-1}}$$

$$= \frac{(\alpha_{n-1} q_{n-1} + \beta_{n-1})(q_n - 1) + \alpha_{n-1}(q_{n-1} + 1) + \beta_{n-1}}{(\gamma_{n-1} q_{n-1} + \delta_{n-1})(q_n - 1) + \gamma_{n-1}(q_{n-1} + 1) + \delta_{n-1}}.$$

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Splitting a Continued Fraction

$$\frac{a1}{a0} = [q_1, q_2, \ldots, q_n]$$

$$= \frac{\alpha_n q_n + \beta_n}{\gamma_n q_n + \delta_n}$$

where α_ns are defined by $q_1, q_2, \ldots, q_n$

$$= \frac{(\alpha_{n-1} q_{n-1} + \beta_{n-1}) q_n + \alpha_{n-1}}{(\gamma_{n-1} q_{n-1} + \delta_{n-1}) q_n + \gamma_{n-1}}$$

$$= \frac{(\alpha_{n-1} q_{n-1} + \beta_{n-1})(q_n - 1) + \alpha_{n-1}(q_{n-1} + 1) + \beta_{n-1}}{(\gamma_{n-1} q_{n-1} + \delta_{n-1})(q_n - 1) + \gamma_{n-1}(q_{n-1} + 1) + \delta_{n-1}}.$$

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Number-theoretic properties

Definition

Concatenation of a_1/b_1 and a_2/b_2 is
$(a_1/b_1) \otimes (a_2/b_2) = a/b$,
where $a = (a_1 + a_2)/c$ and $b = (b_1 + b_2)/c$, for an integer c s.t. $gcd(a, b) = 1$.

Definition (Splitting formula)

$$[q_1, q_2, \ldots, q_n] = \begin{cases} [q_1, q_2, \ldots, q_{n-1} + 1] \otimes (q_n - 1)[q_1, q_2, \ldots, q_{n-1}]; & \text{if } n \text{ is even} \\ (q_n - 1)[q_1, q_2, \ldots, q_{n-1}] \otimes [q_1, q_2, \ldots, q_{n-1} + 1]. & \text{if } n \text{ is odd} \end{cases}$$

Number-theoretic properties

Definition

Concatenation of a_1/b_1 and a_2/b_2 is
$(a_1/b_1) \otimes (a_2/b_2) = a/b$,
where $a = (a_1 + a_2)/c$ and $b = (b_1 + b_2)/c$, for an integer c s.t. $gcd(a, b) = 1$.

Definition (Splitting formula)

$$[q_1, q_2, \ldots, q_n] = \begin{cases} [q_1, q_2, \ldots, q_{n-1} + 1] \otimes (q_n - 1)[q_1, q_2, \ldots, q_{n-1}]; & \text{if } n \text{ is even} \\ (q_n - 1)[q_1, q_2, \ldots, q_{n-1}] \otimes [q_1, q_2, \ldots, q_{n-1} + 1]. & \text{if } n \text{ is odd} \end{cases}$$

Example

$$\begin{aligned}\frac{46}{87} &= [1,1,8,5] \ (n \text{ is even}) \\ &= [1,1,9] \otimes 4 \cdot [1,1,8] \\ &= (8 \cdot [1,1] \otimes [1,2]) \otimes 4 \cdot (7 \cdot [1,1] \otimes [1,2]) \\ &= (8 \cdot [2] \otimes ([2] \otimes [1])) \otimes 4 \cdot (7 \cdot [2] \otimes ([2] \otimes [1])),\end{aligned}$$

which gives DSS chain codes:

(0101010101010101)(011)
(01010101010101)(011)
(01010101010101)(011)
(01010101010101)(011)
(01010101010101)(011).

Approximate straightness

- avoiding tight enforcing of the DSS constraints
  (especially for a curve representing the gross pattern of a real-life image with digital imperfections)
- enabling extraction of approximately straight pieces from a digital curve
  (straightening a part of the DC when the concerned part is not exactly "digitally straight")
- reducing the number of extracted segments
  (hence reducing the storage and CPU time)
- usage of integer operations only

Approximate straightness

- avoiding tight enforcing of the DSS constraints
  (especially for a curve representing the gross pattern of a real-life image with digital imperfections)
- enabling extraction of approximately straight pieces from a digital curve
  (straightening a part of the DC when the concerned part is not exactly "digitally straight")
- reducing the number of extracted segments
  (hence reducing the storage and CPU time)
- usage of integer operations only

Approximate straightness

- avoiding tight enforcing of the DSS constraints
  (especially for a curve representing the gross pattern of a real-life image with digital imperfections)
- enabling extraction of approximately straight pieces from a digital curve
  (straightening a part of the DC when the concerned part is not exactly “digitally straight”)
- reducing the number of extracted segments
  (hence reducing the storage and CPU time)
- usage of integer operations only

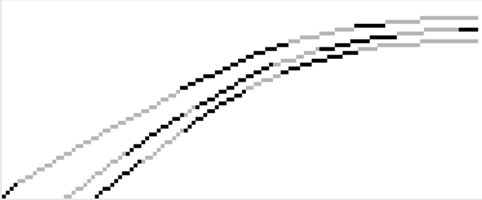
Approximate straightness

- avoiding tight enforcing of the DSS constraints
  (especially for a curve representing the gross pattern of a real-life image with digital imperfections)
- enabling extraction of approximately straight pieces from a digital curve
  (straightening a part of the DC when the concerned part is not exactly "digitally straight")
- reducing the number of extracted segments
  (hence reducing the storage and CPU time)
- usage of integer operations only

Why Approximate straightness

Example

Exactly straight pieces (48 nos.)



Approximately straight pieces (20 nos.)

How Approximate straightness

Straight line
- Time
- Gregorian
- DSL & DSS
- Problem
- Periodicity
- Properties
- Approximate straightness

Circle
- Construction
- Properties
- DCS
- DCR & DCH
- Segmentation
- Properties
- DCT
- DCG
- Surface

- **orientations parameters**
  - n (non-singular element)
  - s (singular element)
  - l (length of leftmost run of n)
  - r (length of rightmost run of n)
- **run length interval parameters**: p and q
  $[p, q]$ is the range of possible lengths (excepting l and r) of n
- **conditions**:
  - $q - p \leq d = \lfloor (p+1)/2 \rfloor$
  - $(l - p), (r - p) \leq e = \lfloor (p + 1/2) \rfloor$

How Approximate straightness

- **orientations parameters**
  - *n* (non-singular element)
  - *s* (singular element)
  - *l* (length of leftmost run of *n*)
  - *r* (length of rightmost run of *n*)
- **run length interval parameters**: *p* and *q*
  [*p*, *q*] is the range of possible lengths (excepting *l* and *r*) of *n*
- **conditions**:
  - $q - p \leqslant d = \lfloor (p+1)/2 \rfloor$
  - $(l - p), (r - p) \leqslant e = \lceil (p+1)/2 \rceil$

Straight line
- Time
- Gregorian
- DSL & DSS
- Problem
- Periodicity
- Properties
- Approximate straightness

Circle
- Construction
- Properties
- DCS
- DCR & DCH
- Segmentation
- Properties
- DCT
- DCG
- Surface

How Approximate straightness

- **orientations parameters**
  - n (non-singular element)
  - s (singular element)
  - l (length of leftmost run of n)
  - r (length of rightmost run of n)
- **run length interval parameters**: p and q
  $[p, q]$ is the range of possible lengths (excepting l and r) of n
- **conditions**:
  - $q - p \leq d = \lfloor (p+1)/2 \rfloor$
  - $(l - p), (r - p) \leq e = \lceil (p+1)/2 \rceil$

Straight line
- Time
- Gregorian
- DSL & DSS
- Problem
- Periodicity
- Properties
- Approximate straightness

Circle
- Construction
- Properties
- DCS
- DCR & DCH
- Segmentation
- Properties
- DCT
- DCG
- Surface

How Approximate straightness

- **orientations parameters**
  - *n* (non-singular element)
  - *s* (singular element)
  - *l* (length of leftmost run of *n*)
  - *r* (length of rightmost run of *n*)
- **run length interval parameters**: *p* and *q*
  $[p, q]$ is the range of possible lengths (excepting *l* and *r*) of *n*
- **conditions**:
  - $q - p \leq d = \lfloor (p+1)/2 \rfloor$
  - $(l - p), (r - p) \leq e = \lceil (p+1)/2 \rceil$

How Approximate straightness

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

- **orientations parameters**
  - *n* (non-singular element)
  - *s* (singular element)
  - *l* (length of leftmost run of *n*)
  - *r* (length of rightmost run of *n*)
- **run length interval parameters**: *p* and *q*
  [*p*, *q*] is the range of possible lengths (excepting *l* and *r*) of *n*
- **conditions**:
  - $q - p < d = \lfloor (p+1)/2 \rfloor$
  - $(l - p), (r - p) < e = \lfloor (p + 1/2) \rfloor$

How Approximate straightness

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

- **orientations parameters**
  - *n* (non-singular element)
  - *s* (singular element)
  - *l* (length of leftmost run of *n*)
  - *r* (length of rightmost run of *n*)
- **run length interval parameters**: *p* and *q*
  [*p*, *q*] is the range of possible lengths (excepting *l* and *r*) of *n*
- **conditions**:
  - $q - p \leqslant d = \lfloor (p+1)/2 \rfloor$
  - $(l - p), (r - p) \leqslant e = \lfloor (p + 1/2) \rfloor$

How Approximate straightness

- **orientations parameters**
  - n (non-singular element)
  - s (singular element)
  - l (length of leftmost run of n)
  - r (length of rightmost run of n)
- **run length interval parameters**: p and q
  $[p, q]$ is the range of possible lengths (excepting l and r) of n
- **conditions**:
  - $q - p \leqslant d = \lfloor (p+1)/2 \rfloor$
  - $(l-p), (r-p) \leqslant e = \lfloor (p+1/2) \rfloor$

How Approximate straightness

- **orientations parameters**
  - *n* (non-singular element)
  - *s* (singular element)
  - *l* (length of leftmost run of *n*)
  - *r* (length of rightmost run of *n*)
- **run length interval parameters**: *p* and *q*
  [*p*, *q*] is the range of possible lengths (excepting *l* and *r*) of *n*
- **conditions**:
  - $q - p \leqslant d = \lfloor (p+1)/2 \rfloor$
  - $(l-p), (r-p) \leqslant e = \lfloor (p+1/2) \rfloor$

How Approximate straightness

- **orientations parameters**
  - n (non-singular element)
  - s (singular element)
  - l (length of leftmost run of n)
  - r (length of rightmost run of n)
- **run length interval parameters**: p and q
  $[p, q]$ is the range of possible lengths (excepting l and r) of n
- **conditions**:
  - $q - p \leqslant d = \lfloor (p + 1)/2 \rfloor$
  - $(l - p), (r - p) \leqslant e = \lfloor (p + 1/2) \rfloor$

How approximate straightness

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Theorem ([Bhowmick and Bhattacharya, 2007])

Isothetic error of a run length p_i in an ADSS (approximate DSS) comprising of N ADSS, is given by

$$\epsilon \leqslant \left(1 - \frac{1}{N}\right)\left(1 + \frac{d}{p+1}\right) \leqslant 1 + \frac{d}{p+1}. \tag{1}$$

Remarks

- Error incurred with an ADSS can be controlled by d.
- For a given error bound, d decreases linearly with p.

How approximate straightness

Theorem ([Bhowmick and Bhattacharya, 2007])

Isothetic error of a run length p_i in an ADSS (approximate DSS) comprising of N ADSS, is given by

$$\epsilon \leqslant \left(1 - \frac{1}{N}\right)\left(1 + \frac{d}{p+1}\right) \leqslant 1 + \frac{d}{p+1}. \quad (1)$$

Remarks

- Error incurred with an ADSS can be controlled by d.
- For a given error bound, d decreases linearly with p.

Cumulative error (criterion $\mathrm{C}_{\max}$)

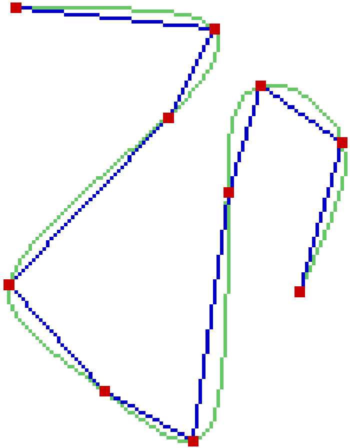
Theorem ([Bhowmick and Bhattacharya, 2007])

An ordered set of ADSS, $\langle \mathbf{L}^{(k)} \rangle_{j_1}^{j_2}$, can be replaced by a single straight line segment, $\widetilde{\mathbf{L}}$, such that isothetic deviation of no point in $\langle \mathbf{L}^{(k)} \rangle_{j_1}^{j_2}$ from $\widetilde{\mathbf{L}}$ exceeds τ, if

$$\max_{j_1 \leqslant j \leqslant j_2 - 1} \left| \triangle \left(s(\mathbf{L}_{j_1}^{(k)}), e(\mathbf{L}_{j}^{(k)}), e(\mathbf{L}_{j_2}^{(k)}) \right) \right| \leqslant \tau d_{\top} \left(s(\mathbf{L}_{j_1}^{(k)}), e(\mathbf{L}_{j_2}^{(k)}) \right)$$

$\widetilde{\mathbf{L}}$ passes through the start point $s(\mathbf{L}_j^{(k)})$ of $\mathbf{L}_{j_1}^{(k)}$ and the end point $e(\mathbf{L}_j^{(k)})$ of $\mathbf{L}_{j_2}^{(k)}$;
$|\triangle(p, q, r)| = 2\times$ area of the triangle pqr;
$d_{\top}(p, q) =$ maximum isothetic distance between p and q.

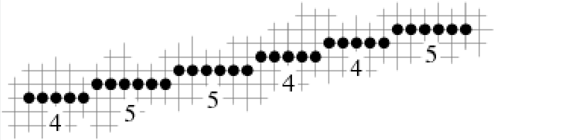
C_{max}: An example



$\tau = 8$

Approximate straightness: Example

Example



$0^4 1 0^5 1 0^5 1 0^4 1 0^4 1 0^5$

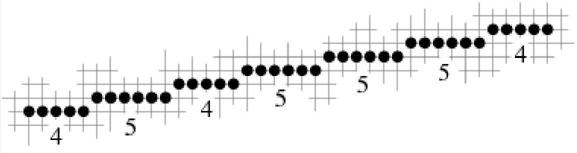
$\Rightarrow p = 4, q = 5, l = 4, r = 5$

$\Rightarrow$ R3 fails

$\Rightarrow$ not a DSS but an ADSS.

Approximate straightness: Example

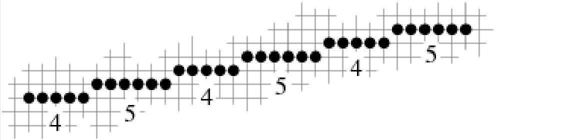
Example



$0^4 1 0^5 1 0^4 1 0^5 1 0^5 1 0^5 1 0^4$
$\Rightarrow p = 4, q = 5, l = 4, r = 4$
$\Rightarrow$ R4 fails
$\Rightarrow$ not a DSS but an ADSS.

Approximate straightness: Example

Example



$0^4 1 0^5 1 0^4 1 0^5 1 0^4 1 0^5$

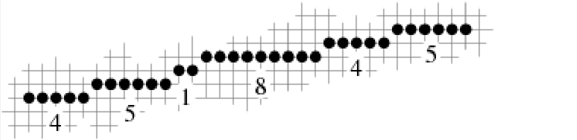
$\Rightarrow p = 4, q = 5, l = 4, r = 5$

$\Rightarrow$ R1–R4 and c1, c2

$\Rightarrow$ an ADSS as well as a DSS.

Approximate straightness: Example

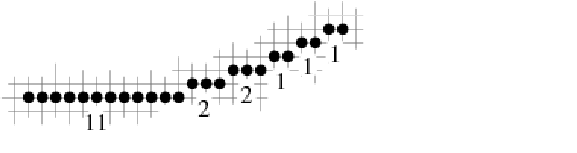
Example



$0^4 1 0^5 1 0 1 0^8 1 0^4 1 0^5$
$\Rightarrow p = 1, q = 8, l = 4, r = 5$
R2, c1, and c2 fail
$\Rightarrow$ neither a DSS nor an ADSS.

Approximate straightness: Example

Example

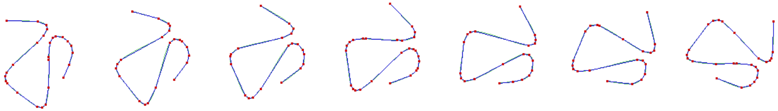


$0^{11}10^{2}10^{2}101010$
$\Rightarrow p = 1, q = 2, l = 11, r = 1$
$\Rightarrow$ R2 and c2 fail
$\Rightarrow$ not a DSS or an ADSS.

Results (test curve)

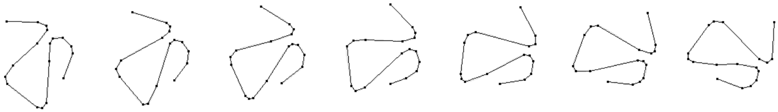
input

Results (test curve)



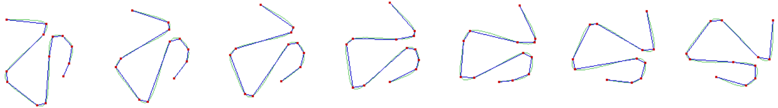
$\tau = 1$

Results (test curve)



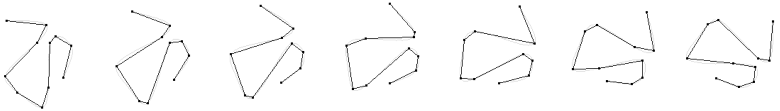
$\tau = 2$

Results (test curve)



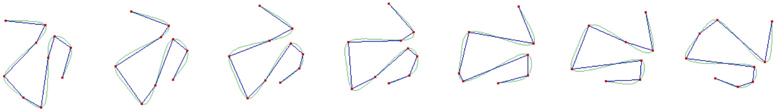
$\tau = 4$

Results (test curve)



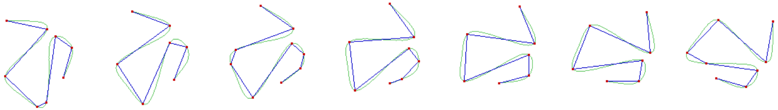
$\tau = 6$

Results (test curve)



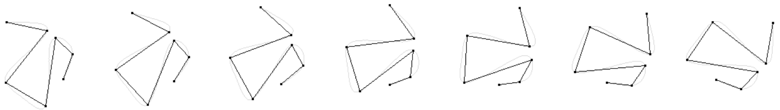
$\tau = 8$

Results (test curve)



$\tau = 11$

Results (test curve)



$\tau = 14$

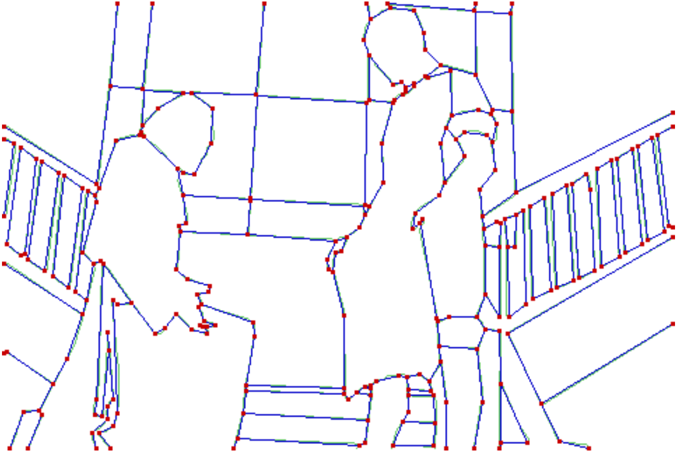
Results (statue)

a real-world image

Results (statue)

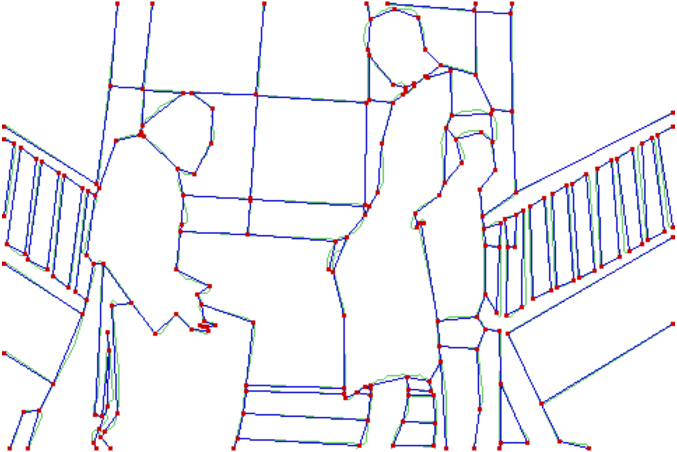
edge map

Results (statue)



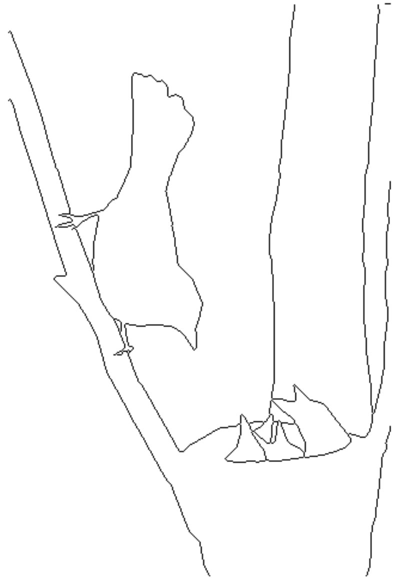
approximation for $\tau = 2$

Results (statue)



approximation for $\tau = 4$

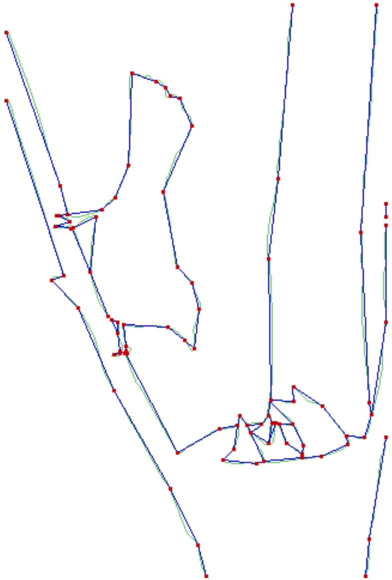
Results (nestlings)



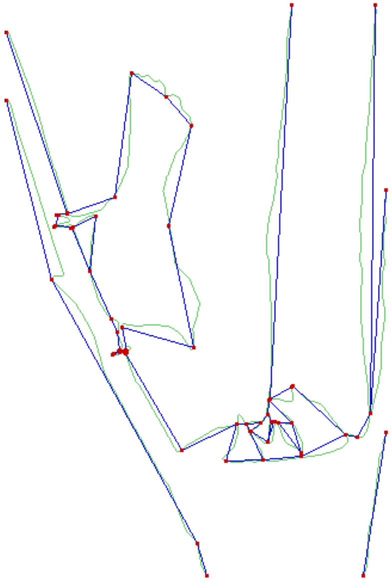
Results (nestlings)

Results (nestlings)

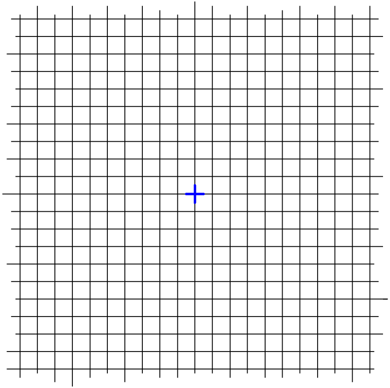
Results (nestlings)



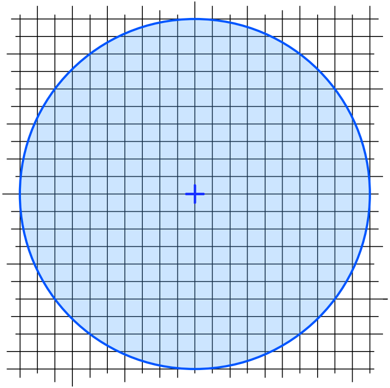
Results (nestlings)



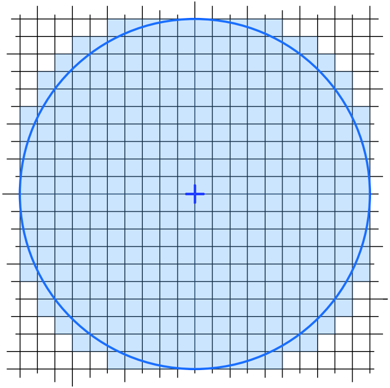
Construction by Digitization



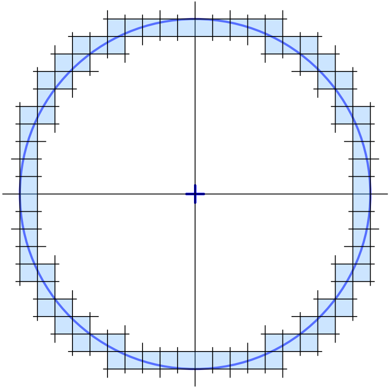
Construction by Digitization



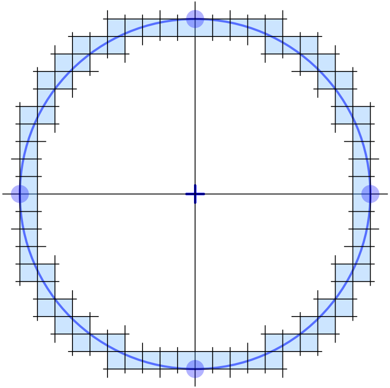
Construction by Digitization



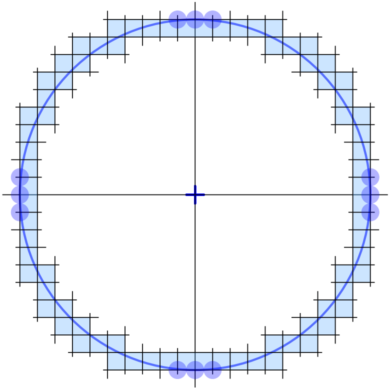
Construction by Digitization



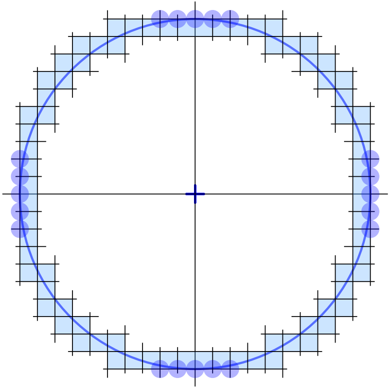
Construction by Digitization



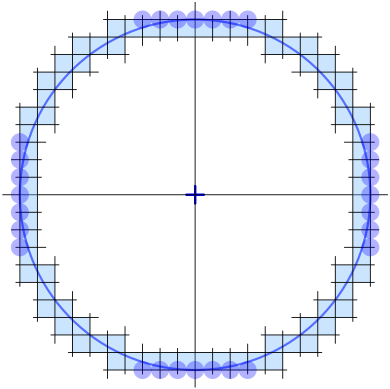
Construction by Digitization



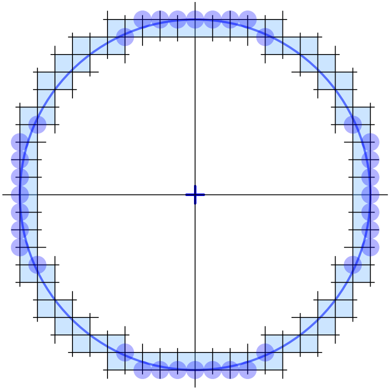
Construction by Digitization



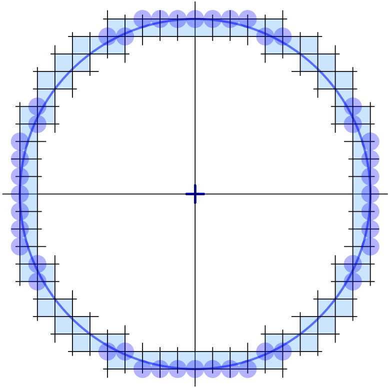
Construction by Digitization



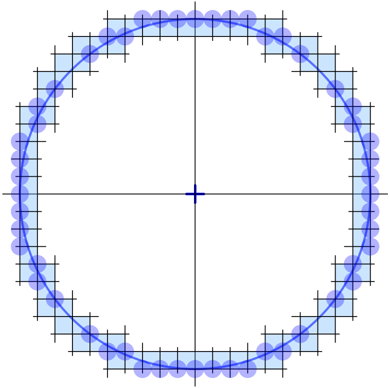
Construction by Digitization



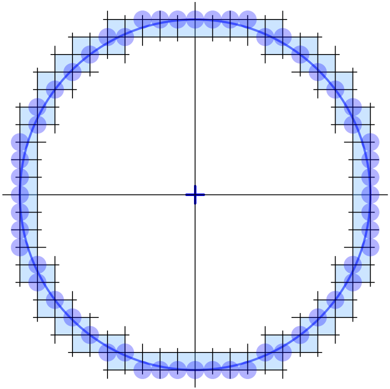
Construction by Digitization



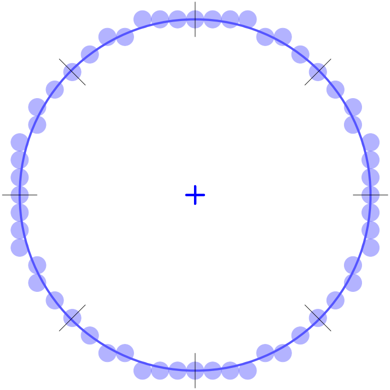
Construction by Digitization



Construction by Digitization



Construction by Digitization



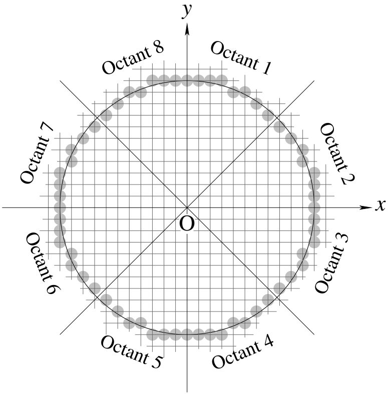
Construction by Digitization

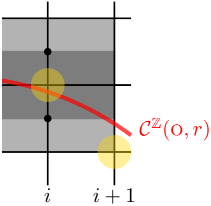
Algorithm	Inventors	Year
Incremental	Bresenham	1977
Optimized midpoint	Foley *et al.*	1993
Short run	Hsu *et al.*	1993
Hybrid run slice	Yao & Rokne	1995
Number-theoretic[a]	Bhowmick & Bhattacharya	2008

[a] P. Bhowmick and B. B. Bhattacharya,
Number-theoretic interpretation and construction of a digital circle,
Discrete Applied Mathematics, **156** : 2381–2399, **2008.**

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

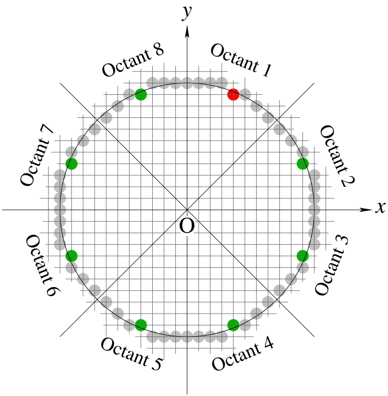
Octants





A real circle, $\mathcal{C}^{\mathbb{R}}(o, 11)$, and the eight octants of the corresponding digital circle, $\mathcal{C}^{\mathbb{Z}}(o, 11)$.

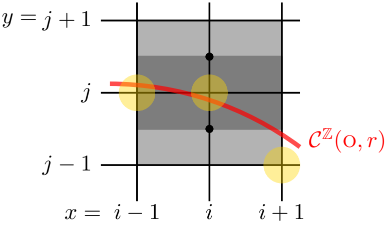
Property 1



- Each point $p(i, j) \in \mathcal{C}^{\mathbb{Z}}(o, r)$ has seven other points of reflection in $\mathcal{C}^{\mathbb{Z}}(o, r)$.
  (Properties of Octant 1 are applicable to other octants.)

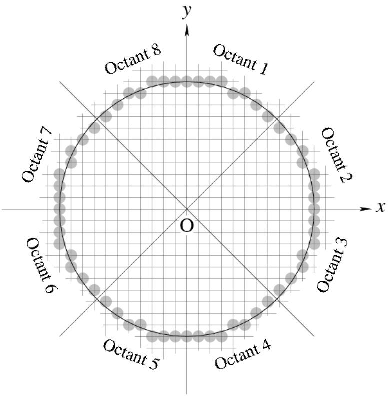
Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Property 2



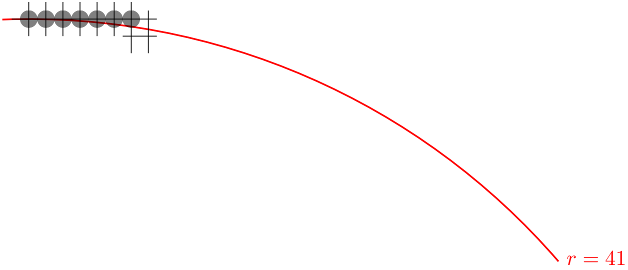
- y-distance of the grid-intersection point of $\mathcal{C}^{\mathbb{R}}(o, r)$ from the digital point of $\mathcal{C}^{\mathbb{Z}}(o, r)$ is less than $1/2$.

Property 3



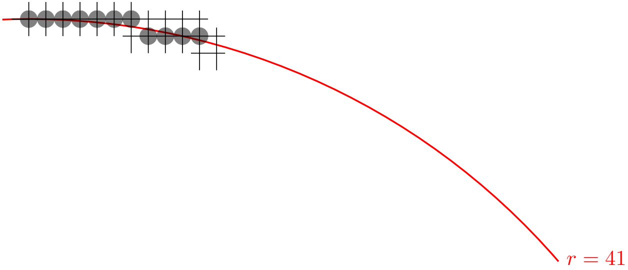
- $\mathcal{C}^{\mathbb{Z}}(o, r)$ is a closed and irreducible digital curve.

Property 4



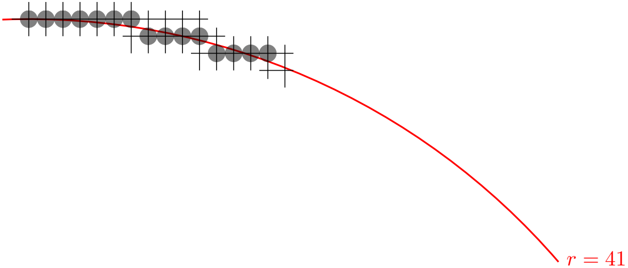
An upper run is usually longer than a lower run in Octant 1.

Property 4



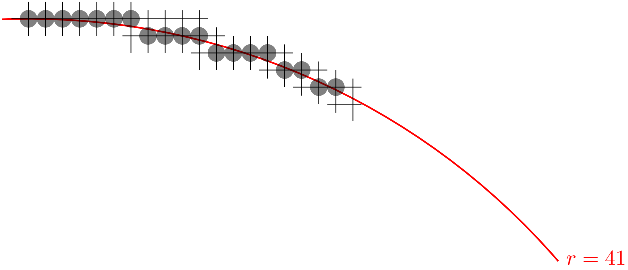
An upper run is usually longer than a lower run in Octant 1.

Property 4



An upper run is usually longer than a lower run in Octant 1.

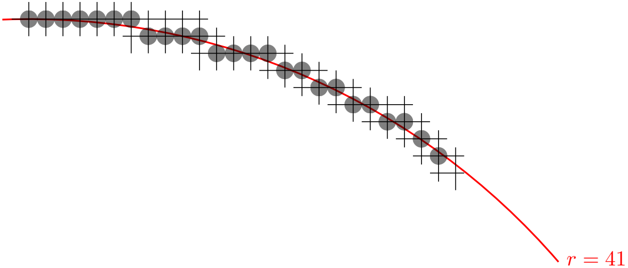
Property 4



An upper run is usually longer than a lower run in Octant 1.

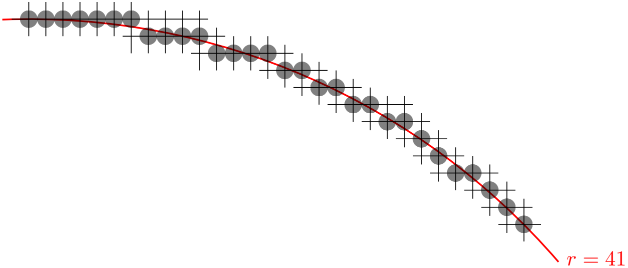
Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Property 4



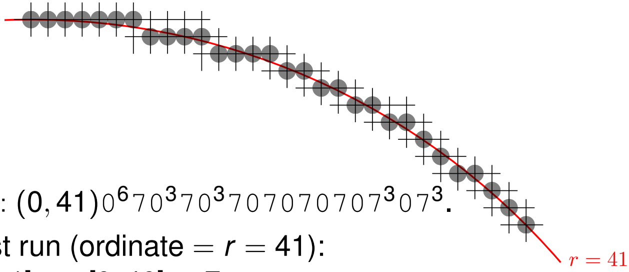
An upper run is usually longer than a lower run in Octant 1.

Property 4



An upper run is usually longer than a lower run in Octant 1.

Number-theoretic Properties I

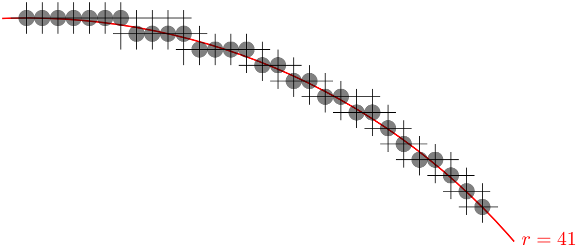


- $r = 41 : (0,41)0^6 7 0^3 7 0^3 7 0 7 0 7 0 7 0 7 0 7^3 0 7^3$.
- topmost run (ordinate $= r = 41$):
  $s[0, r-1] = s[0, 40] = 7$,
  next run ($y = r - 1 = 40$): $s[r, 3r-3] = s[41, 120] = 4$,
  next run ($y = r - 2 = 39$):
  $s[3r-2, 5r-7] = s[121, 198] = 4$,...
- *square numeric code* = $\langle 7, 4, 4, 2, 2, 2, 2, 1, 1, 2, 1, 1, 1 \rangle$
  = $\langle 7, 4^2, 2^4, 1^2, 2, 1^3 \rangle$.

Number-theoretic Properties II

Lemma

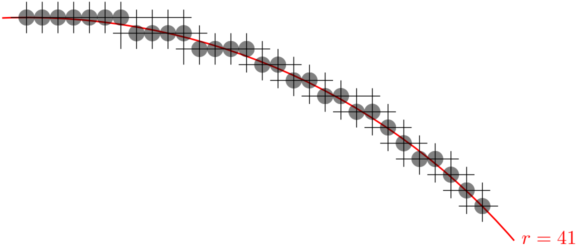
The interval $I_k = [(2k-1)r - k(k-1), (2k+1)r - k(k+1) - 1]$ *contains the squares of abscissae of the grid points of* $\mathcal{C}^{\mathbb{Z},I}(o,r)$ *whose ordinates are* $r-k$*, for* $k \geqslant 1$*.*



Number-theoretic Properties III

Lemma

The lengths of the intervals containing the squares of equi-ordinate abscissae of the grid points in $\mathcal{C}^{\mathbb{Z},I}(o,r)$ decrease constantly by 2, starting from l_1.



Number-theoretic Properties IV

Theorem

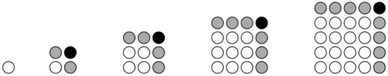
The squares of abscissae of grid points, lying on $\mathcal{C}^{\mathbb{Z},I}(o,r)$ *and having ordinate* $r-k$*, lie in the interval* $[u_k, v_k := u_k + l_k - 1]$*, where* u_k *and* l_k *are given as follows.*

$$u_k = \begin{cases} u_{k-1} + l_{k-1} & \text{if } k \geqslant 1 \\ 0 & \text{if } k = 0 \end{cases}$$

$$l_k = \begin{cases} l_{k-1} - 2 & \text{if } k \geqslant 2 \\ 2r - 2 & \text{if } k = 1 \\ r & \text{if } k = 0 \end{cases}$$

Algorithm **DCS**

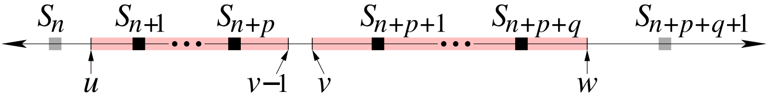
```
Algorithm DCS (int r) {
int i = 0, j = r, s = 0, w = r − 1;
int l = w << 1;
while (j ⩾ i) {
        do {  sym_8 (i, j);
              s = s + i;
              i + +;
              s = s + i;  } while (s ⩽ w);
        w = w + l;
        l = l − 2;
        j − −;  }}
```



Number-theoretic properties I

Lemma

The number of perfect squares in a closed interval $[v, w]$ is at most one more than the number of perfect squares in the preceding closed interval $[u, v-1]$ of equal length, where the intervals are taken from the non-negative integer axis.

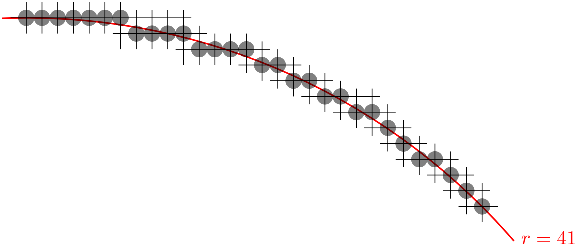


Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Number-theoretic properties II

Theorem

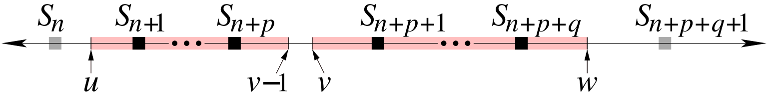
The run length of grid points of $\mathcal{C}^{\mathbb{Z},I}(o,r)$ with ordinate $j-1$ never exceeds one more than the run length of its grid points with ordinate j.



Number-theoretic properties III

Lemma

If $[u, v-1]$ be the interval $I_k, k \geqslant 1$, and $[v, w]$ be the interval of same length as $[u, v-1]$, then the number of perfect squares in $[v, w]$ is at least (floor of) half the number of perfect squares less one in $[u, v-1]$.

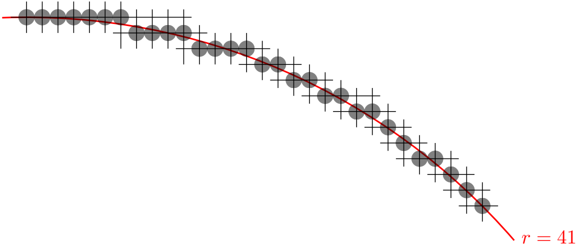


Number-theoretic properties IV

Theorem

If $\lambda(j)$ be the run length of grid points of $\mathcal{C}^{\mathbb{Z},I}(o,r)$ with ordinate j, then the run length of grid points with ordinate $j-1$ for $j \leqslant r-1$ and $r \geqslant 2$, is given by

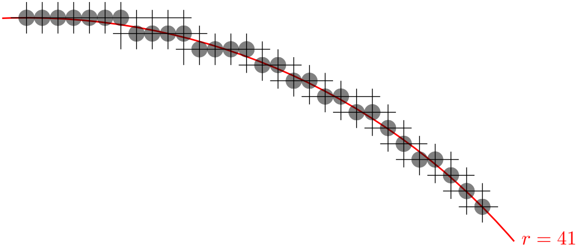
$$\lambda(j-1) \geqslant \left\lfloor \frac{\lambda(j)-1}{2} \right\rfloor - 1.$$



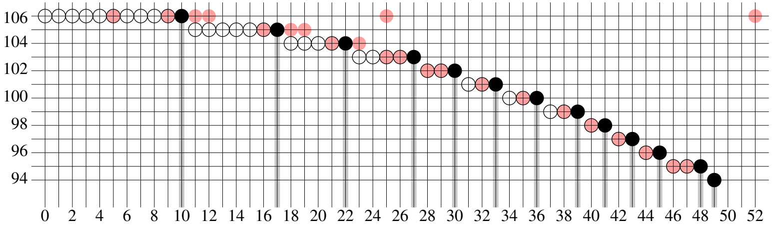
Constructive bounds

Constructive bounds

$$\left\lfloor \frac{\lambda(j) - 1}{2} \right\rfloor - 1 \leqslant \lambda(j-1) \leqslant \lambda(j) + 1$$



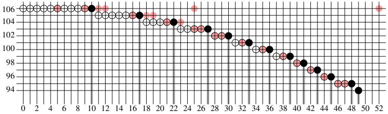
Algorithm **DCR**



Demonstration of **DCR** for $r = 106$.

Algorithm **DCR**: Square search

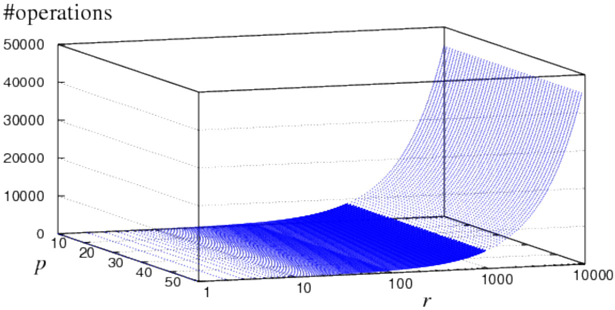
```
Algorithm DCR (int r) {
    int i = 0, j = r, w = r − 1, m;
    int s = 0, t = r, l = w << 1;
    while (j ⩾ i) {
        while (s < t) {
            m = s + t;
            m = m >> 1;
            if (w ⩽ square[m])
                t = m;
            else
                s = m + 1; }
        if (w < square[s])
            s − −;
        s + +;
        include_run (i, s − i, j);
        t = s + s − i + 1;
        i = s;
        w = w + l;
        l = l − 2;
        j − −; }}
```



Hybrid algorithm **DCH** I

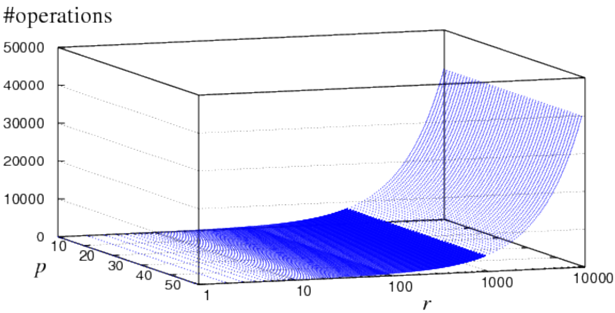

```
Algorithm DCH (int r, int p) {
int i = 0, j = r, w = r − 1, m;
int s = 0, t = r, l = w << 1;
while (j ⩾ i) {
    while (s < t) {
        m = s + t;
        m = m >> 1;
        if (w ⩽ square[m])
            t = m;
        else
            s = m + 1; }
    if (w < square[s])
        s − −;
    s + +;
    include_run (i, s − i, j);
    if (s − i < p)
        break;
    t = s + s − i + 1;
    i = s;
    w = w + l;
    l = l − 2;
    j − −; }
i = s − 1;
s = square[s];
w = w + l;
l = l − 2;
j − −;
while (j ⩾ i) {
    do {sym_8 (i, j);
        s = s + i;
        i + +;
        s = s + i; } while (s ⩽ w);
    w = w + l;
    l = l − 2;
    j − −; }}
```

Test Results...



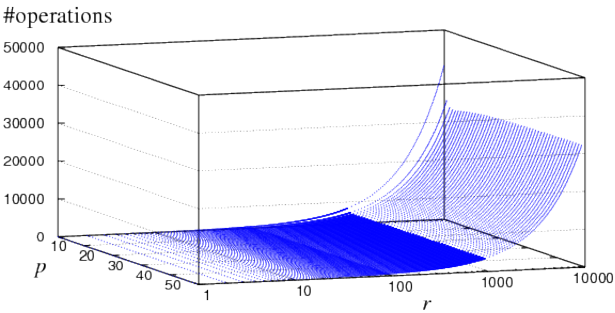
DCB

Test Results...



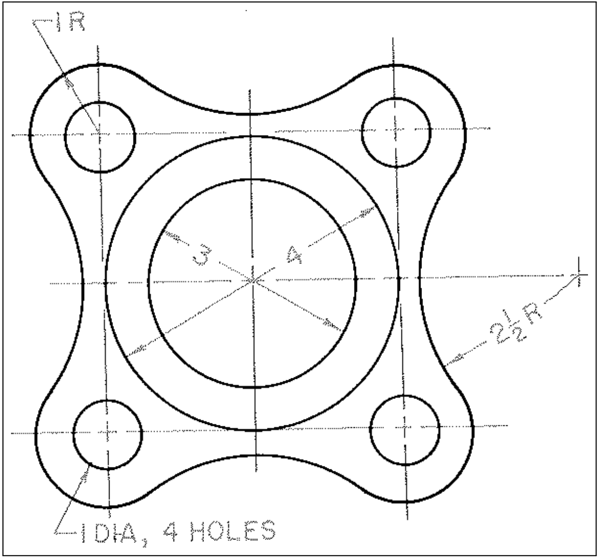
DCR

Test Results...

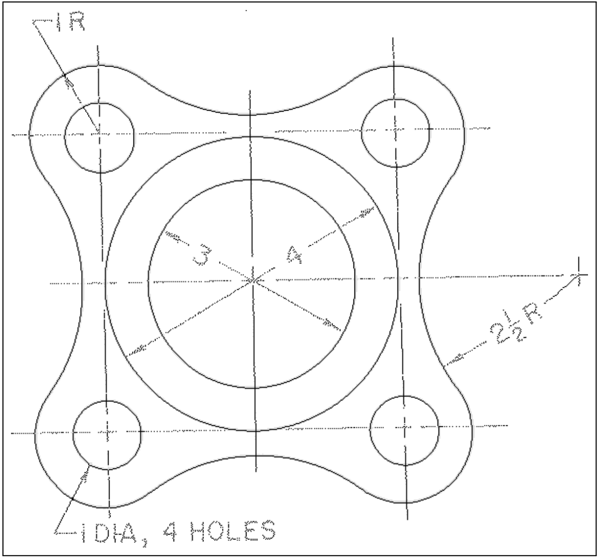


DCH

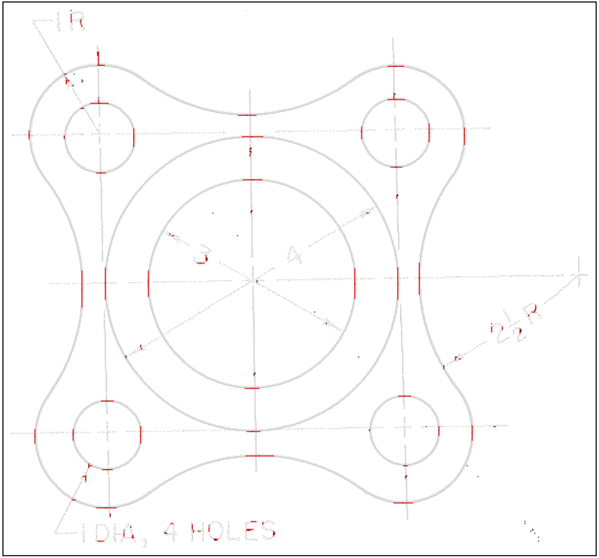
Arc Segmentation



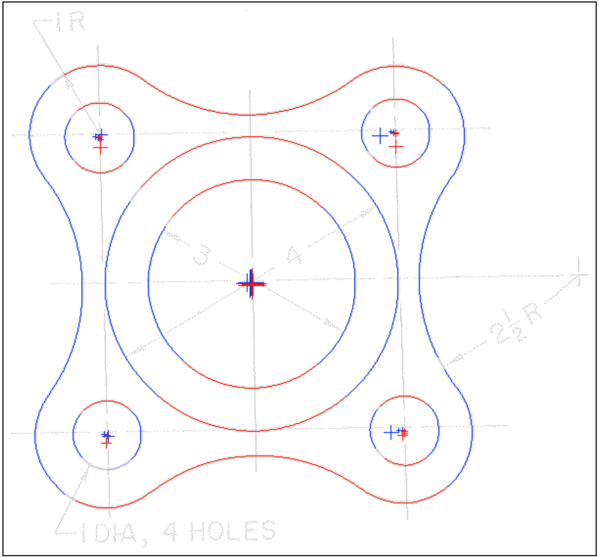
Arc Segmentation



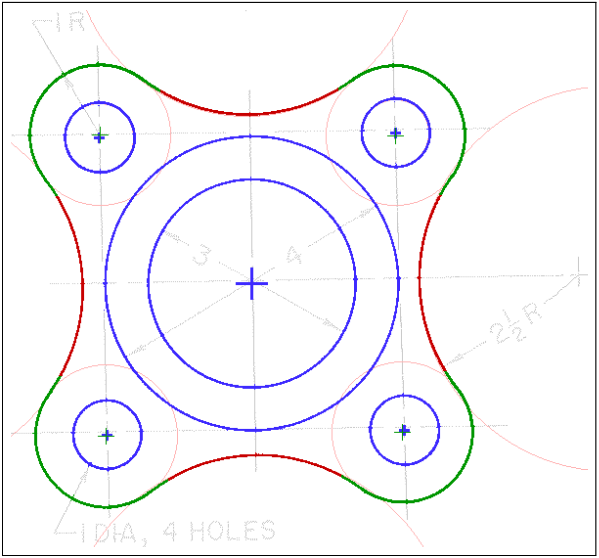
Arc Segmentation



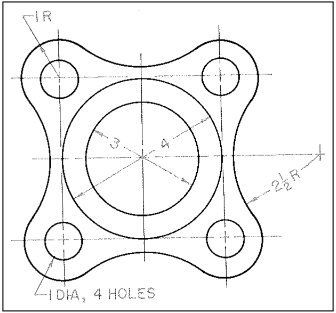
Arc Segmentation



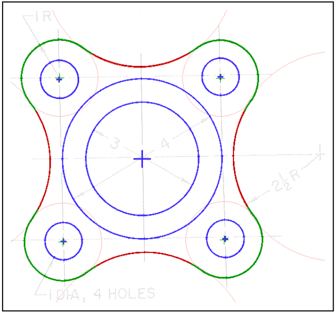
Arc Segmentation



Arc Segmentation



input



output

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Arc Segmentation

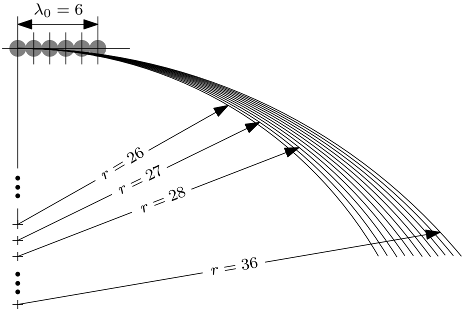
Algorithm	Inventors	Year
Hough transform Davies **1984**, Illingworth & Kittler **1988**, Yip et al. **1992**, Chen & Chung **2001**, Kim & Kim **2005**, Chiu & Liaw **2005**,...		
Voronoi diagram	Coeurjolly *et al.*	2004
Chord & Sagitta	Bera, Bhowmick & Bhattacharya	2010
Discrete Curvature[a]	Pal, Dutta & Bhowmick	in press
Number Theory[b]	Pal & Bhowmick	2012
Number Theory & Graph Theory[c]: Bhowmick & Pal		accepted

[a] S. Pal, R. Dutta & P. Bhowmick, Circular Arc Segmentation by Curvature Estimation and Geometric Validation, *Intl. Journal Image & Graphics* (in press).

[b] S. Pal & P. Bhowmick, Determining Digital Circularity Using Integer Intervals *Journal of Mathematical Imaging & Vision*, **42**(1):1-24, 2012.

[c] S. Pal & P. Bhowmick, Fast Circular Arc Segmentation Based on Approximate Circularity and Cuboid Graph, *Journal of Mathematical Imaging & Vision* (accepted).

Conflicting Radii I

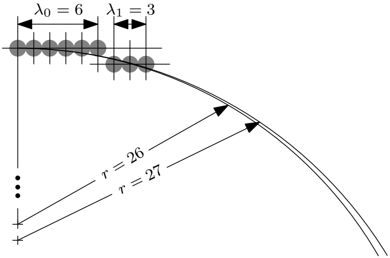


$r \in [26, 36]$

Lemma

λ_0 is the length of top run of a digital circle $\mathcal{C}^{\mathbb{Z}}(o, r)$ iff $r \in R_0 := [(\lambda_0 - 1)^2 + 1, \lambda_0^2]$.

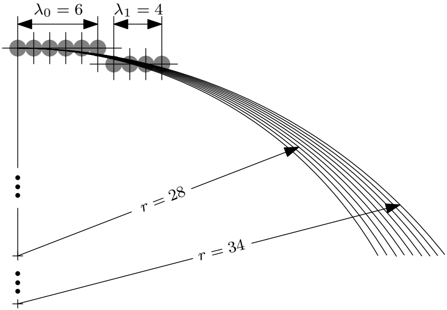
Conflicting Radii II



$r \in [26, 27]$

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

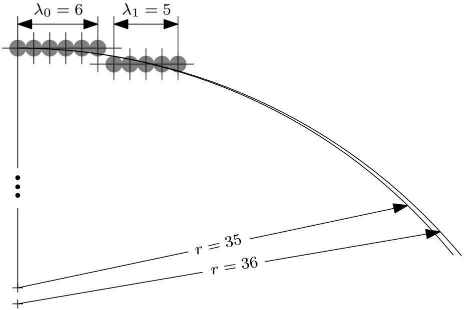
Conflicting Radii III



$r \in [28, 34]$

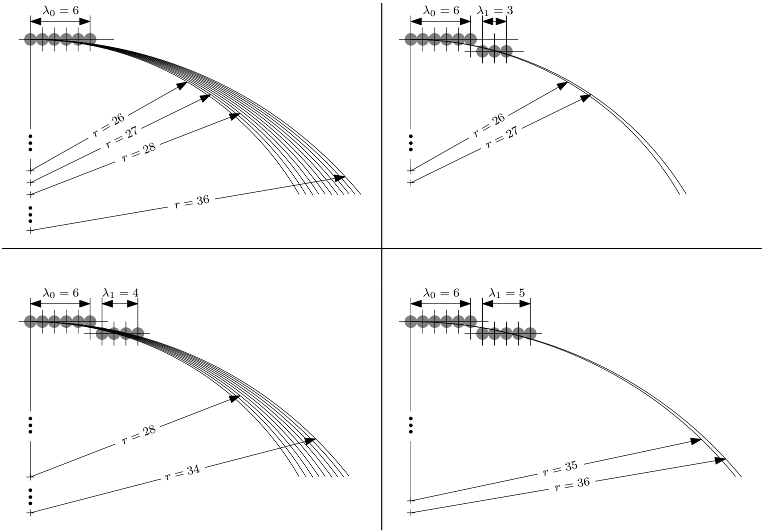
Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Conflicting Radii IV



$r \in [35, 36]$

Radii Nesting I



Radii Nesting II

Lemma

λ_0 and λ_1 are the lengths of top two runs of $\mathcal{C}^{\mathbb{Z}}(o, r)$ iff $r \in R_0 \cap R_1$, where, $R_1 = \left[\left\lceil \frac{(\Lambda_1 - 1)^2 + 3}{3} \right\rceil, \left\lfloor \frac{\Lambda_1^2 + 2}{3} \right\rfloor\right]$, $\Lambda_1 = \lambda_0 + \lambda_1$.

(If $R_0 \cap R_1 = \emptyset$, then there exists no digital circle ... λ_0 and λ_1.)

Radii Nesting III

Theorem (Radii interval)

$\langle \lambda_0, \ldots, \lambda_n \rangle$ is the sequence of top $n+1$ run-lengths of $\mathcal{C}^{\mathbb{Z}}(o, r)$ iff

$$r \in \bigcap_{k=0}^{n} R_k$$

where,

$$R_k = \left[\left\lceil \frac{1}{2k+1} \left((\Lambda_k - 1)^2 + k(k+1) + 1 \right) \right\rceil, \left\lfloor \frac{1}{2k+1} \left(\Lambda_k^2 + k(k+1) \right) \right\rfloor \right]$$

and

$$\Lambda_k = \sum_{j=0}^{k} \lambda_j.$$

(If $\bigcap_{k=0}^{n} R_k = \emptyset$, then there exists no digital circle whose top $n+1$ runs have length $\langle \lambda_0, \lambda_1, \ldots, \lambda_n \rangle$.)

Algorithm DCT

$\Lambda \leftarrow S[0]$

$[r', r''] \leftarrow [(\Lambda - 1)^2 + 1, \Lambda^2]$

for $k \leftarrow 1$ to $n - 1$

    $\Lambda \leftarrow \Lambda + S[k]$

    $s' \leftarrow \left\lceil ((\Lambda - 1)^2 + k(k+1) + 1)/(2k+1) \right\rceil$

    $s'' \leftarrow \left\lfloor (\Lambda^2 + k(k+1))/(2k+1) \right\rfloor$

    if $s'' < r'$ **or** $s' > r''$

        print "S is circular up to $(k-1)$th run for $[r', r'']$."

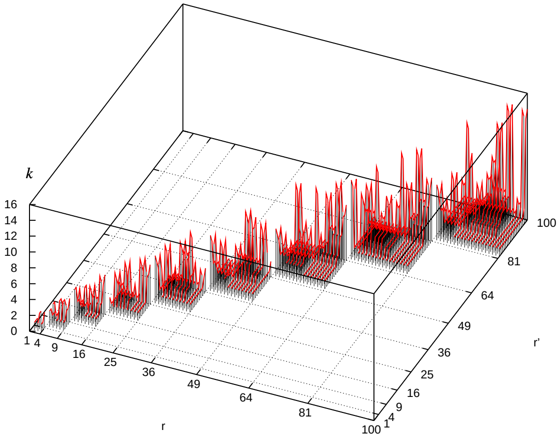
        return

    else

        $[r', r''] \leftarrow [\max(r', s'), \min(r'', s'')]$

print "S is circular in entirety for $[r', r'']$."

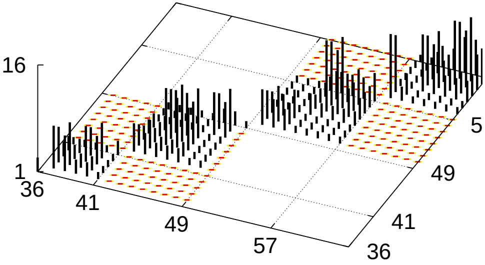
Conflicting Radii: Resolved how fast? I



Conflicting radii starting from $k = 0$

Conflicting Radii: Resolved how fast? II

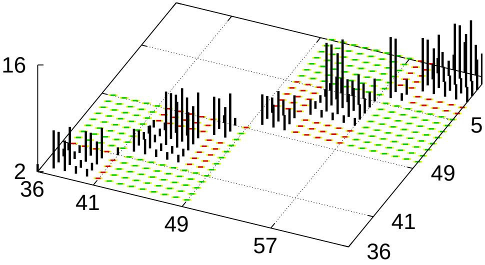
Resolving the conflicting radii r' with increasing k



$k = 1$

Conflicting Radii: Resolved how fast? III

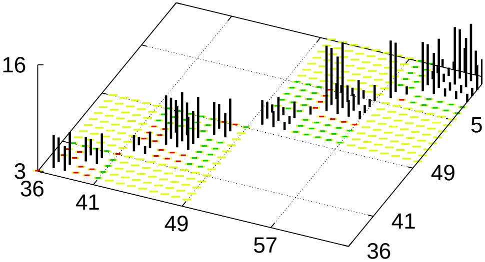
Resolving the conflicting radii r' with increasing k



$k = 2$

Conflicting Radii: Resolved how fast? IV

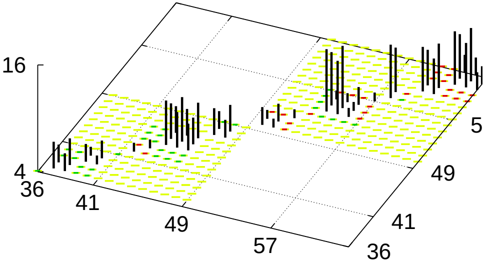
Resolving the conflicting radii r' with increasing k



$k = 3$

Conflicting Radii: Resolved how fast? V

Resolving the conflicting radii r' with increasing k



$k = 4$

General Case & DCG I

Lemma

If a digital circle of radius r contains a given run of length λ, then there exist two positive integers a and k such that $r \geqslant \left\lceil \max\left(f_{1,\lambda}(a,k), f_{2,\lambda}(a,k)\right)\right\rceil$, where

$$f_{1,\lambda}(a,k) = \frac{(a-1)^2 + k(k-1) + 1}{2k-1}$$

and

$$f_{2,\lambda}(a,k) = \frac{(a+\lambda-1)^2 + k(k+1) + 1}{2k+1}.$$

General Case & DCG II

Lemma

If a digital circle of radius r contains a given run of length λ*, then there exist two positive integers a and k such that* $r \leqslant \left\lfloor \min\left(f_{3,\lambda}(a,k), f_{4,\lambda}(a,k)\right)\right\rfloor$*, where*

$$f_{3,\lambda}(a,k) = \frac{a^2 + k(k-1)}{2k-1}$$

and

$$f_{4,\lambda}(a,k) = \frac{(a+\lambda)^2 + k(k+1)}{2k+1}.$$

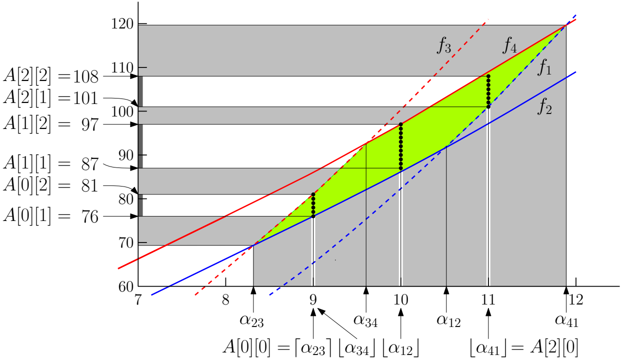
General Case & DCG III

Theorem

An arbitrary run of given length λ belongs to only those digital circles whose radii are in the range

$$\mathcal{R}_{ak} = \begin{array}{c} \left\{ r \mid r \geqslant \left\lceil \max\limits_{a,k \in \mathbb{Z}^+} \left(f_{1,\lambda}(a,k), f_{2,\lambda}(a,k) \right) \right\rceil \right\} \\ \bigcap \\ \left\{ r \mid r \leqslant \left\lfloor \min\limits_{a,k \in \mathbb{Z}^+} \left(f_{3,\lambda}(a,k), f_{4,\lambda}(a,k) \right) \right\rfloor \right\}. \end{array}$$

General Case & DCG IV



General Case & DCG V

Points of intersection (in $\mathbb{R}^2$) among the parabolas $\{f_{i,\lambda} \mid i = 1, 2, 3, 4\}$ defining $\mathcal{R}_{ak}$.

$$(\underline{k} = 2k - 1, \overline{k} = 2k + 1, \underline{\hat{k}} = k(k-1), \hat{\hat{k}} = k(k+1), \underline{\lambda} = \lambda - 1)$$

Parabolas		Point	Abscissa of the point
$f_{1,\lambda}$	$f_{2,\lambda}$	α_{12}	$\frac{1}{2}\left(\underline{k}\lambda + \sqrt{(\underline{k}\lambda + 2)^2 + 2(\underline{k\lambda}^2 + 2\underline{\hat{k}} - 3)} + 2\right)$
$f_{2,\lambda}$	$f_{3,\lambda}$	α_{23}	$\frac{1}{2}\left(\underline{k\lambda} + \sqrt{(\underline{k\lambda})^2 + 2(\underline{k\lambda}^2 + 2\hat{\hat{k}} - 1)}\right)$
$f_{3,\lambda}$	$f_{4,\lambda}$	α_{34}	$\frac{1}{2}\left(\underline{k}\lambda + \sqrt{(\underline{k}\lambda)^2 + 2(\underline{k}\lambda^2 + 2k^2)}\right)$
$f_{4,\lambda}$	$f_{1,\lambda}$	α_{41}	$\frac{1}{2}\left(\underline{k}\lambda + \overline{k} \pm \sqrt{(\underline{k}\lambda + \overline{k})^2 + 2(\underline{k}\lambda^2 + 2\underline{\hat{k}} - \overline{k} - 1)}\right)$

General Case & DCG VI

Specifications of the parabolas $\{f_{i,\lambda} \mid i = 1, 2, 3, 4\}$.

Parabola	Axis	Directrix	Length of Latus Rectum	Vertex	Focus
$f_{1,\lambda}$	$x = 1$	$\underline{k}\, y = 3/4$	$\underline{k}$	$\left(1, (\hat{\underline{k}} + 1)/\underline{k}\right)$	$\left(1, (8\hat{\overline{k}} + 5)/(4\underline{k})\right)$
$f_{2,\lambda}$	$x = -\underline{\lambda}$	$\overline{k}\, y = 3/4$	$\overline{k}$	$\left(-\underline{\lambda}, (\hat{\overline{k}} + 1)/\overline{k}\right)$	$\left(-\underline{\lambda}, (8\hat{\underline{k}} + 5)/(4\overline{k})\right)$
$f_{3,\lambda}$	$x = 0$	$\underline{k}\, y = -1/4$	$\underline{k}$	$\left(0, (\hat{\underline{k}})/\underline{k}\right)$	$\left(0, (8\hat{\overline{k}} + 1)/(4\underline{k})\right)$
$f_{4,\lambda}$	$x = -\lambda$	$\overline{k}\, y = -1/4$	$\overline{k}$	$\left(-\lambda, \hat{\overline{k}}/\overline{k}\right)$	$\left(-\lambda, (8\hat{\underline{k}} + 1)/(4\overline{k})\right)$

Specifications of the parabolas $\{f_{i,\lambda} \mid i = 1, 2, 3, 4\}$.

POINTS OF INTERSECTION (IN $\mathbb{R}^2$) AMONG THE PARABOLAS $\{f_{i,\lambda} : i = 1, 2, 3, 4\}$ DEFINING $\mathcal{R}_{ak}$.

To obtain the value of $\{\alpha_{ij} \mid j = (i \bmod 4) + 1, i = 1, 2, 3, 4\}$, we have solved the following quadratic equations in a. Out of the two values of a obtained, say $a = C \pm \sqrt{D}$, we define α as $C + \sqrt{D}$.

α_{23}: $\frac{(a+\lambda-1)^2+k(k+1)+1}{2k+1} = \frac{a^2+k(k-1)}{2k-1}$

or, $(2k-1)(a^2 + 2(\lambda-1)a + (\lambda-1)^2 + k(k+1) + 1) = (2k+1)(a^2 + k(k-1))$

or, $2a^2 - 2(2k-1)(\lambda-1)a - (2k-1)(\lambda-1)^2 - 2k^2 - 2k + 1 = 0$

or, $a = \frac{1}{2}\left((2k-1)(\lambda-1) \pm \sqrt{(2k-1)^2(\lambda-1)^2 + 2((2k-1)(\lambda-1)^2 + 2k^2 + 2k - 1)}\right)$

or,

$$\alpha_{23} = \frac{1}{2}\left((2k-1)(\lambda-1) + \sqrt{(2k-1)^2(\lambda-1)^2 + 2((2k-1)(\lambda-1)^2 + 2k^2 + 2k - 1)}\right).$$

α_{12}: $\frac{(a-1)^2+k(k-1)+1}{2k-1} = \frac{(a+\lambda-1)^2+k(k+1)+1}{2k+1}$

or, $(2k+1)((a-1)^2 + k(k-1) + 1) = (2k-1)((a+\lambda-1)^2 + k(k+1) + 1)$

or, $2a^2 - 2((2k-1)\lambda + 2)a - (2k-1)(\lambda-1)^2 - 2k^2 + 2k + 3 = 0$

or, $a = \frac{1}{2}\left((2k-1)\lambda + 2 \pm \sqrt{((2k-1)\lambda + 2)^2 + 2((2k-1)(\lambda-1)^2 + 2k^2 - 2k - 3)}\right)$

or, $\alpha_{12} = \frac{1}{2}\left((2k-1)\lambda + 2 + \sqrt{((2k-1)\lambda + 2)^2 + 2((2k-1)(\lambda-1)^2 + 2k^2 - 2k - 3)}\right).$

General Case & DCG VIII

α_{41}: $\frac{(a+\lambda)^2+k(k+1)}{2k+1} = \frac{(a-1)^2+k(k-1)+1}{2k-1}$

or, $(2k-1)((a+\lambda)^2+k(k+1)) = (2k+1)((a-1)^2+k(k-1)+1)$

or, $2a^2 - 2(2k(1+\lambda) - \lambda + 1)a - (2k-1)\lambda^2 - 2k^2 + 4k + 2 = 0$

or,

$$a = \frac{1}{2}\left((2k-1)\lambda + 2k + 1 \pm \sqrt{((2k-1)\lambda + 2k + 1)^2 + 2((2k-1)\lambda^2 + 2k^2 - 4k - 2)}\right)$$

or, $\alpha_{41} =$

$$\frac{1}{2}\left((2k-1)\lambda + 2k + 1 + \sqrt{((2k-1)\lambda + 2k + 1)^2 + 2((2k-1)\lambda^2 + 2k^2 - 4k - 2)}\right).$$

α_{34}: $\frac{a^2+k(k-1)}{2k-1} = \frac{(a+\lambda)^2+k(k+1)}{2k+1}$

or, $(2k+1)(a^2+k(k-1)) = (2k-1)((a+\lambda)^2+k(k+1))$

or, $2a^2 - 2(2k-1)\lambda - (2k-1)\lambda^2 - 2k^2 = 0$

or, $a = \frac{1}{2}\left((2k-1)\lambda \pm \sqrt{(2k-1)^2\lambda^2 + 2((2k-1)\lambda^2 + 2k^2)}\right)$

or, $\alpha_{34} = \frac{1}{2}\left((2k-1)\lambda + \sqrt{(2k-1)^2\lambda^2 + 2((2k-1)\lambda^2 + 2k^2)}\right)$.

Algorithm DCG I

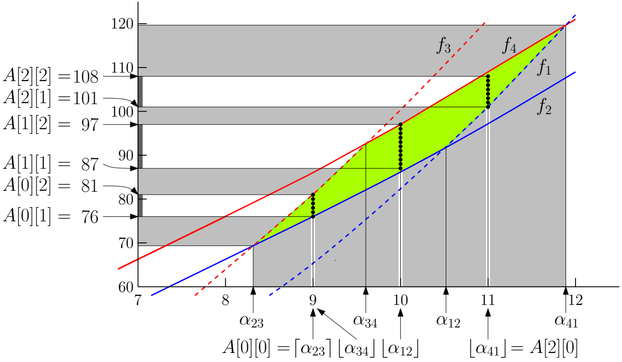
$n_{\max} \leftarrow 0$
for $k' \leftarrow k_{\min}$ to $k_{\max}$
  $\Lambda \leftarrow S[0]$, $i \leftarrow 0$
  FIND-PARAMS(A, Λ, k')
  while $i < m$ **and** $n_{\max} < n$ ▷ for all a's of first run
    $[s', s''] \leftarrow [r', r''] \leftarrow [A[i][1], A[i][2]]$
    $\Lambda \leftarrow A[i][0] + S[0]$, $j \leftarrow 1$
    while $j < n$ **and** $s'' \geqslant r'$ **and** $s' \leqslant r''$ ▷ verifying other $n-1$ runs
      $\Lambda \leftarrow \Lambda + S[j]$, $k \leftarrow k' + j$
      $s' \leftarrow \left\lceil \frac{(\Lambda-1)^2+k(k+1)+1}{2k+1} \right\rceil$, $s'' \leftarrow \left\lfloor \frac{\Lambda^2+k(k+1)}{2k+1} \right\rfloor$
      if $s'' \geqslant r'$ **and** $s' \leqslant r''$
        $[r', r''] \leftarrow [\max(r', s'), \min(r'', s'')]$
    if $n_{\max} < j$
      $n_{\max} \leftarrow j$, $k_{\mathrm{off}} \leftarrow k'$, $[r_{\min}, r_{\max}] \leftarrow [r', r'']$
print "S is circular for $n_{\max}$ runs; starting run $= k_{\mathrm{off}}$; $r \in [r_{\min}, r_{\max}]$."

Algorithm DCG II

Procedure FIND-PARAMS

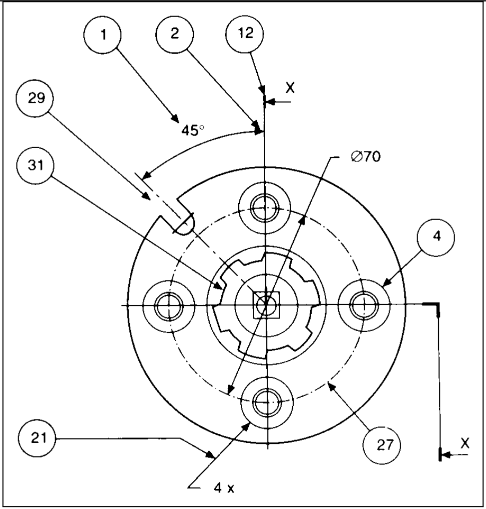
Compute $\{\alpha_{uv} \mid 1 \leqslant u \leqslant 4 \ \wedge v = (u+1) \bmod 4\}$ ▷ (from Tables)
$i \leftarrow 0$
for $a \leftarrow \lceil \alpha_{23} \rceil$ to $\lfloor \alpha_{41} \rfloor$
    $A[i][0] \leftarrow a$ ▷ computing r'
    if $a < \alpha_{12}$
        $A[i][1] \leftarrow \lceil f_{2,\lambda}(a,k) \rceil$
    else
        $A[i][1] \leftarrow \lceil f_{1,\lambda}(a,k) \rceil$ ▷ computing r''
    if $a < \alpha_{34}$
        $A[i][2] \leftarrow \lfloor f_{3,\lambda}(a,k) \rfloor$
    else
        $A[i][2] \leftarrow \lfloor f_{4,\lambda}(a,k) \rfloor$
    $i \leftarrow i + 1$
$m \leftarrow i$

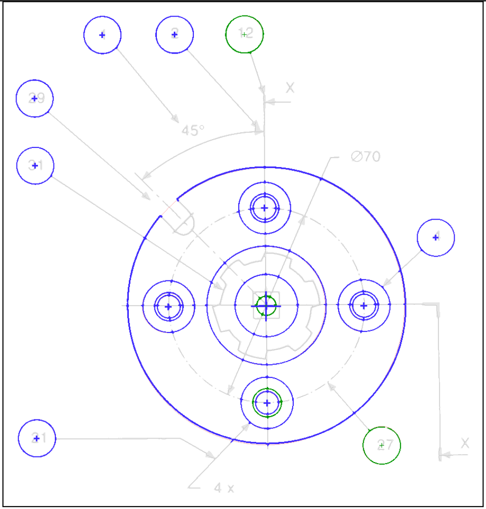
Algorithm DCG III

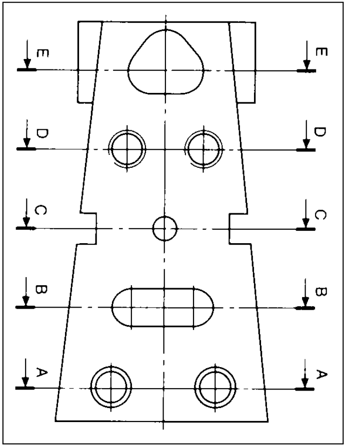


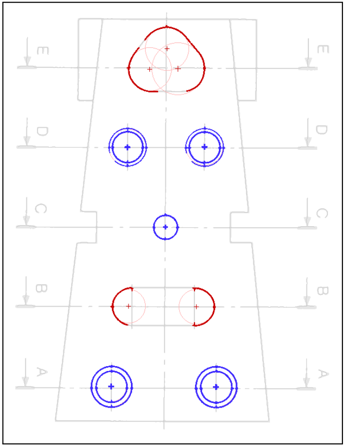
FIND-PARAMS on a run-length 7:

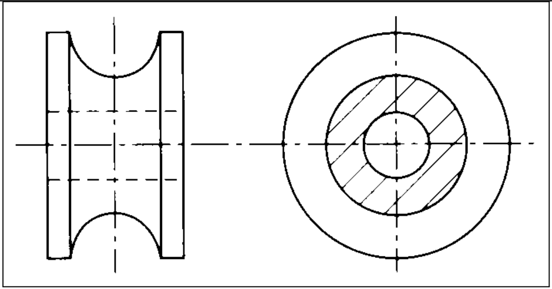
Solution space $\mathcal{R}_{ak}$ of the radius intervals $\{[r'_j, r''_j] \mid j = 0, 1, 2\}$ corresponding to $m = 3$ square numbers lying in $\left[\lceil \alpha_{23} \rceil^2, \lfloor \alpha_{41} \rfloor^2\right] = [9^2, 11^2]$.

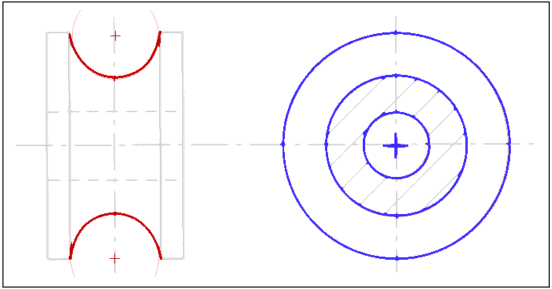

1
2
12
X
29
45°
31
∅70
4
21
27
X
4 x


X
45°
∅70
X
4 x

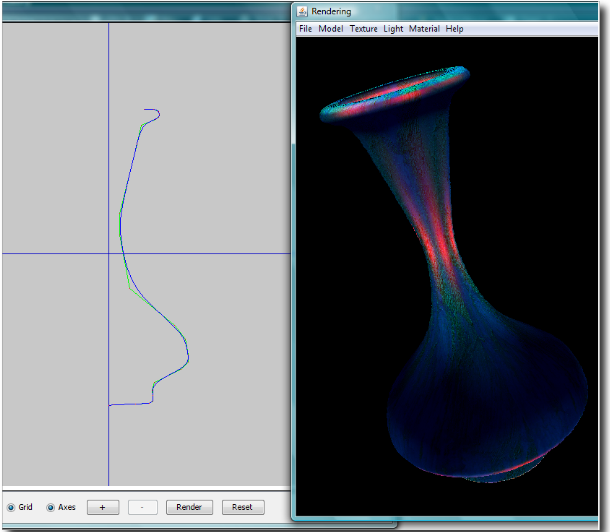






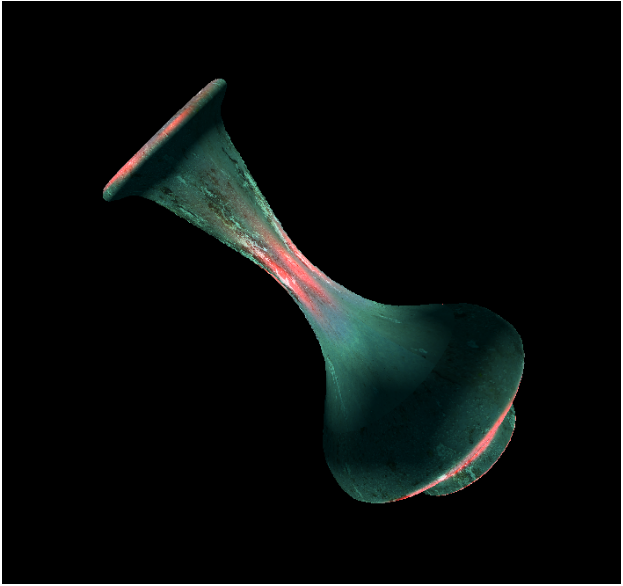


Snapshots of Our Algorithm

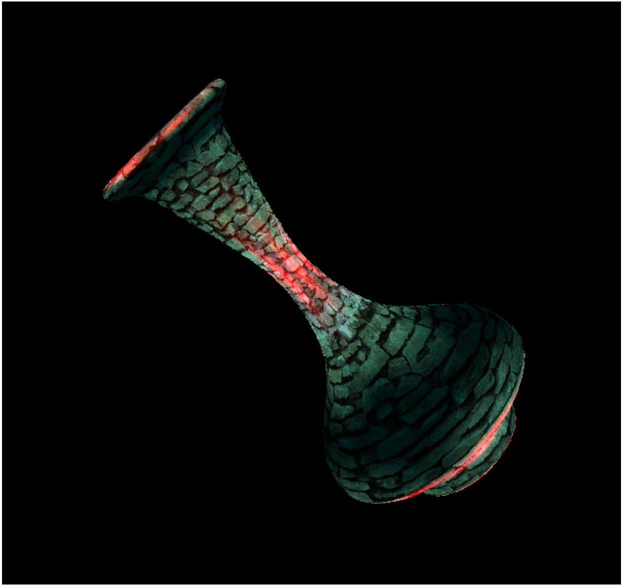


Snapshots of Our Algorithm

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface



Snapshots of Our Algorithm



Snapshots of Our Algorithm

Snapshots of Our Algorithm

Algorithm	Inventors	Year
Polyhedra Represntn.	Galyean & Hughes	1991
Finite Element	Han *et al.*	2007
Cylindrical Element	Han *et al.*	2007
Circular Sector	Lee *et al.*	2008
Number-theoretic [a]	Kumar *et al.*	2010

---

[a] G. Kumar, N.K. Sharma, and P. Bhowmick, Wheel-throwing in Digital Space Using Number-theoretic Approach, *International Journal of Arts and Technology*, 2010 (in press).

A preliminary version appeared in: *Proc. of International Conference on Arts and Technology: ArtsIT 2009*, **LNICST:** 30, Springer, pp. 181–189, 2010.

Theoretical Foundation: A Glimpse

Disconnected generatrix

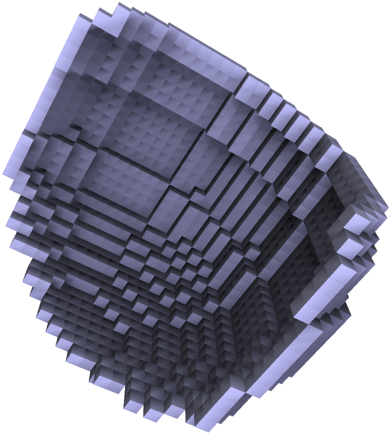
Theoretical Foundation: A Glimpse

Reducible generatrix

Theoretical Foundation: A Glimpse

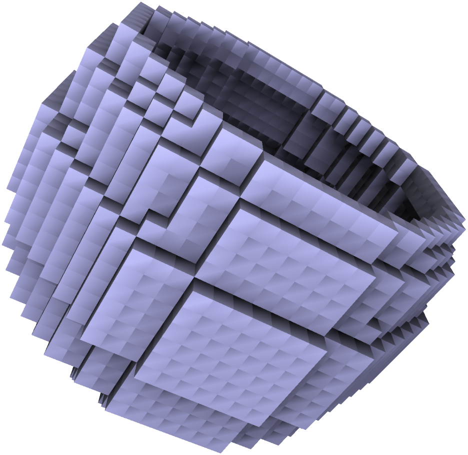
Perfect generatrix: Connected & irreducible

Theoretical Foundation: A Glimpse



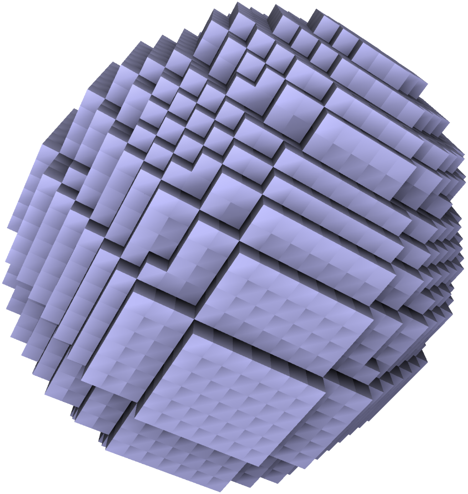
open and irreducible digital surface

Theoretical Foundation: A Glimpse



open and irreducible digital surface

Theoretical Foundation: A Glimpse



closed and irreducible digital surface

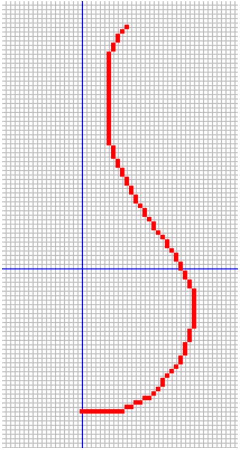
Surface of Revolution in $\mathbb{Z}^3$

Number-theoretic

P. Bhowmick

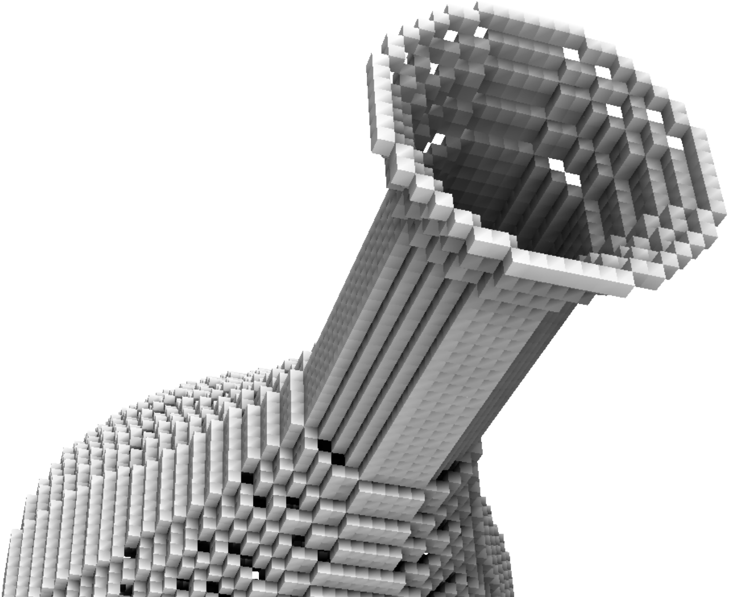
Straight line
- Time
- Gregorian
- DSL & DSS
- Problem
- Periodicity
- Properties
- Approximate straightness

Circle
- Construction
- Properties
- DCS
- DCR & DCH
- Segmentation
- Properties
- DCT
- DCG
- Surface

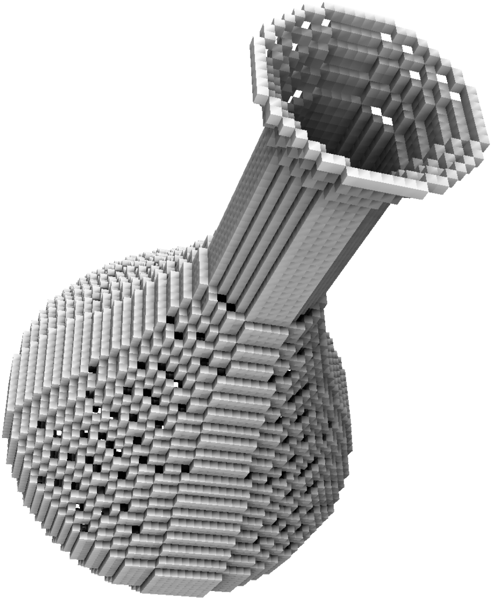


Digital generatrix

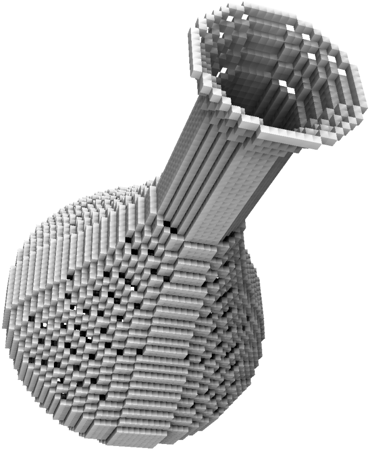
Surface of Revolution in $\mathbb{Z}^3$



Surface of Revolution in $\mathbb{Z}^3$

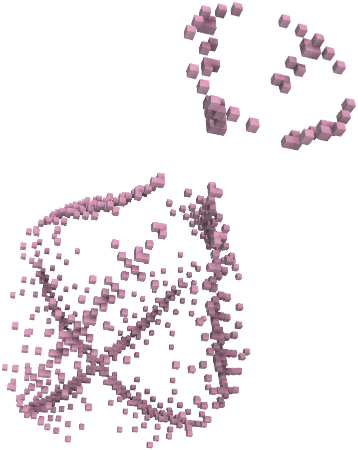


Surface of Revolution in $\mathbb{Z}^3$



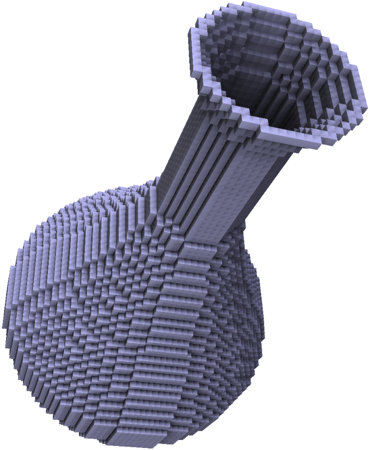
A disconnected surface of revolution created due to missing voxels

Surface of Revolution in $\mathbb{Z}^3$



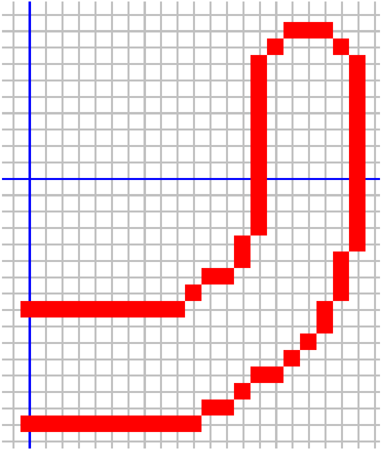
Missing voxels

Surface of Revolution in $\mathbb{Z}^3$



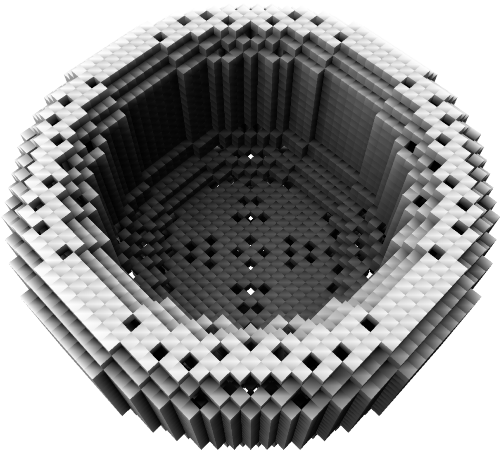
Connected and irreducible surface of revolution

Double-layered Surface of Revolution in $\mathbb{Z}^3$



2-layered digital generatrix

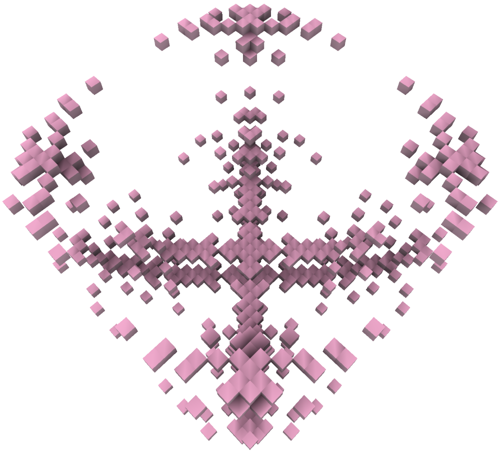
Double-layered Surface of Revolution in $\mathbb{Z}^3$



A disconnected surface of revolution created due to missing voxels

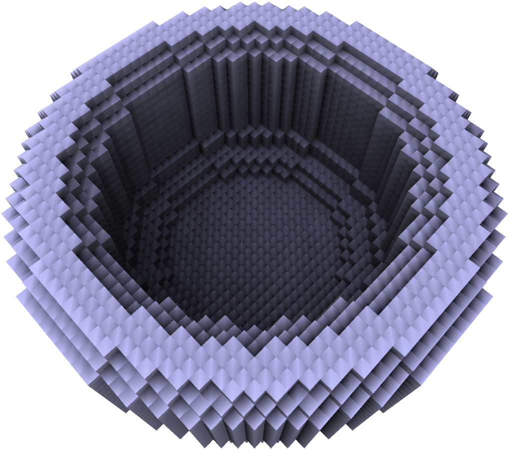
Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Double-layered Surface of Revolution in $\mathbb{Z}^3$



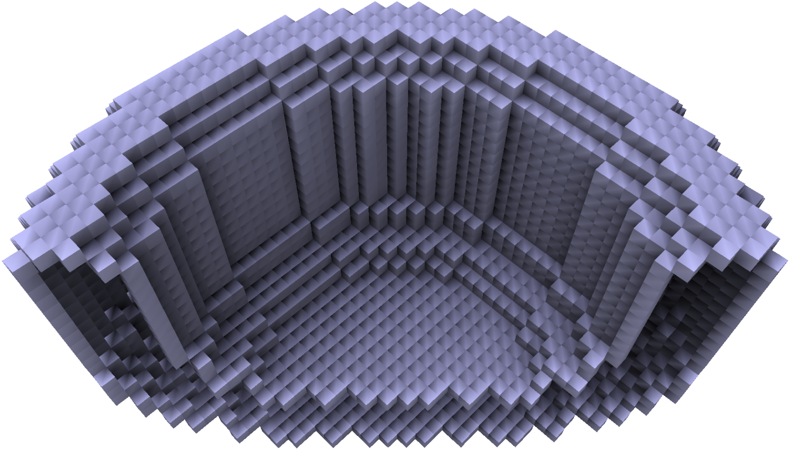
Missing voxels

Double-layered Surface of Revolution in $\mathbb{Z}^3$



Connected and irreducible 2-layered surface of revolution

Double-layered Surface of Revolution in $\mathbb{Z}^3$

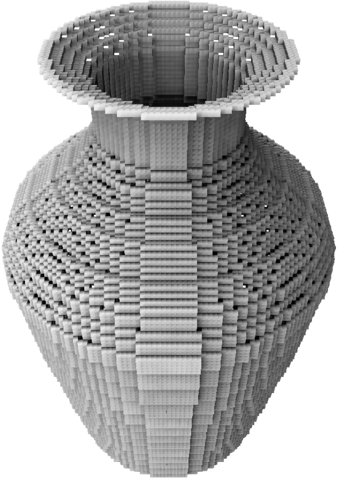


A fragmented piece

Double-layered Surface of Revolution in $\mathbb{Z}^3$

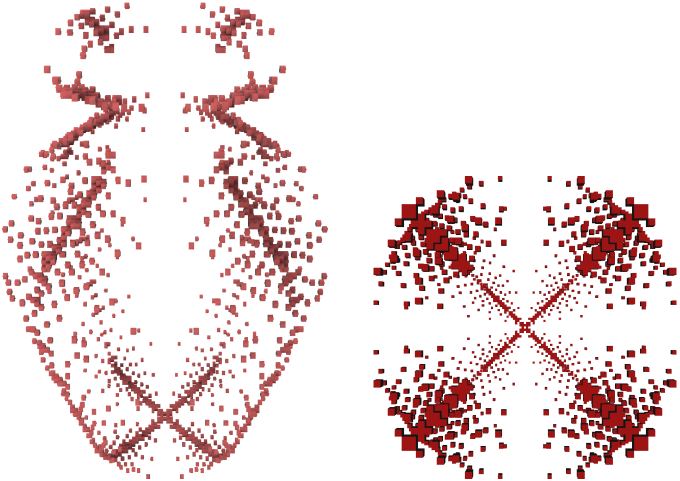
A sample set of finished potteries produced by our algorithm

Missing Voxels: Parabolic Characterization I



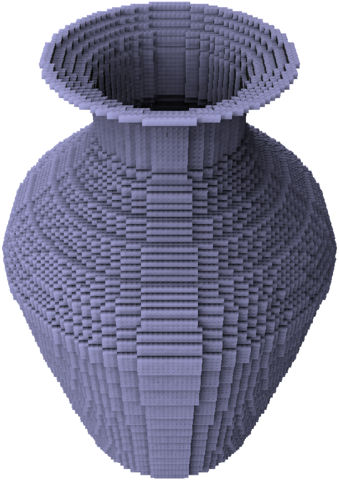
Surface with absentee voxels

Missing Voxels: Parabolic Characterization II



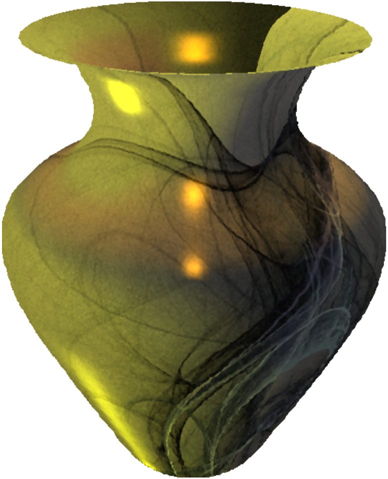
Absentee voxels (Left: front view, Right: top view)

Missing Voxels: Parabolic Characterization III



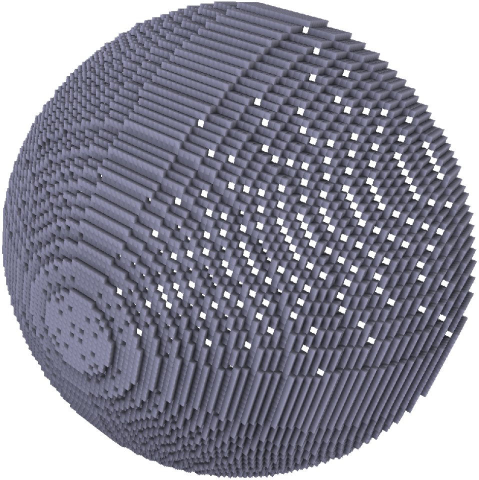
The *perfect & irreducible* digital surface of revolution

Missing Voxels: Parabolic Characterization IV



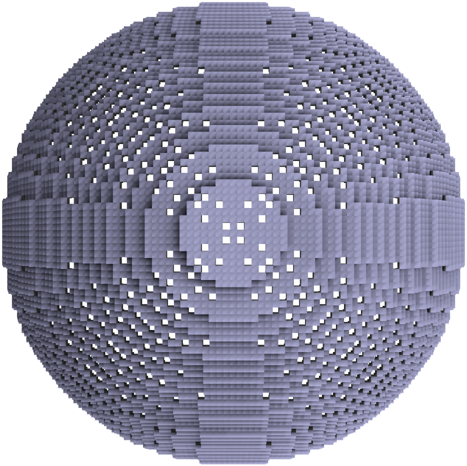
After a realistic finish.

Missing Voxels: Parabolic Characterization V



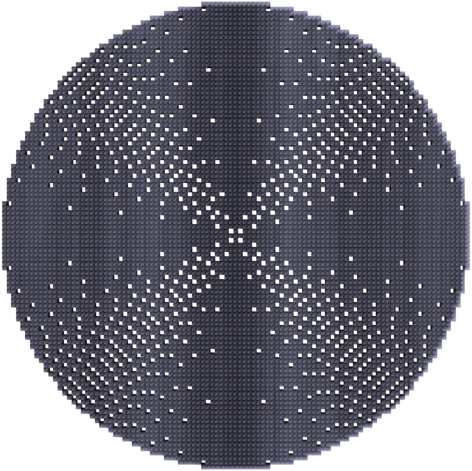
Digital hemisphere ($r = 50$): Oblique view

Missing Voxels: Parabolic Characterization VI



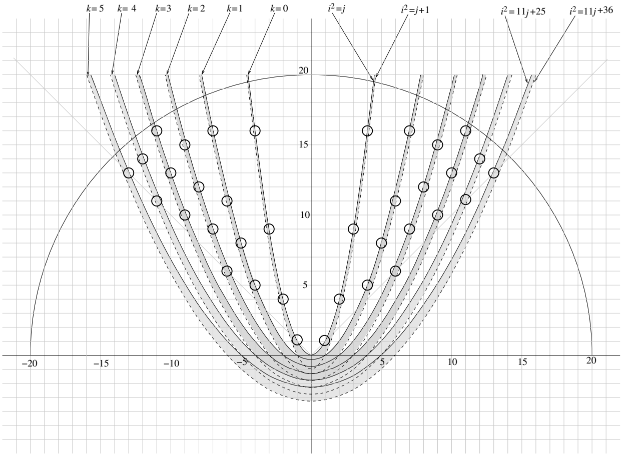
Top view

Missing Voxels: Parabolic Characterization VII



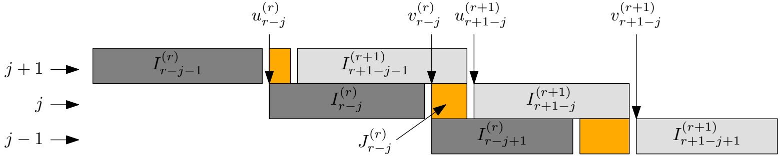
Projection

Missing Voxels: Parabolic Characterization VIII



Infimum parabolas = solid curves
supremum parabolas = dashed curves.

Missing Voxels: Parabolic Characterization IX



The interval $J_{r-j}^{(r)}$ in which an absentee lies.
Light gray $\Rightarrow r+1$, Deep gray $\Rightarrow r$.

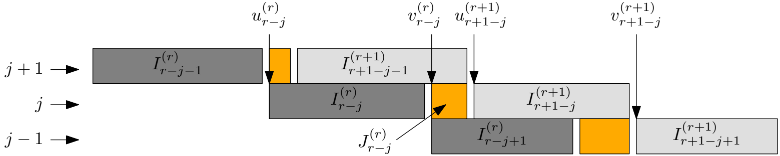
Lemma

The squares of abscissae of the pixels in $\mathcal{C}_1^{\mathbb{Z}}(o, r)$ whose ordinates are j lie in the interval $I_{r-j}^{(r)} = \left[u_{r-j}^{(r)}, v_{r-j}^{(r)}\right)$, where

$$u_{r-j}^{(r)} = r^2 - j^2 - j,$$

$$v_{r-j}^{(r)} = r^2 - j^2 + j.$$

Missing Voxels: Parabolic Characterization X

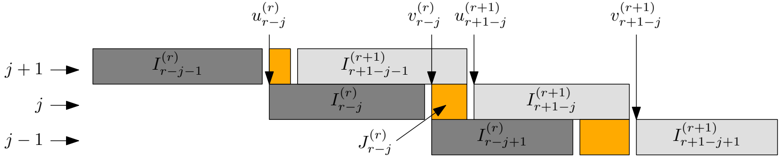


The difference between the lower limit of $I^{(r)}_{r-j}$ and the upper limit of $I^{(r+1)}_{r+1-j}$ is given by

$$u^{(r+1)}_{r+1-j} - v^{(r)}_{r-j} = ((r+1)^2 - j^2 - j) - (r^2 - j^2 + j) = 2(r-j) + 1.$$

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

Missing Voxels: Parabolic Characterization XI



Lemma

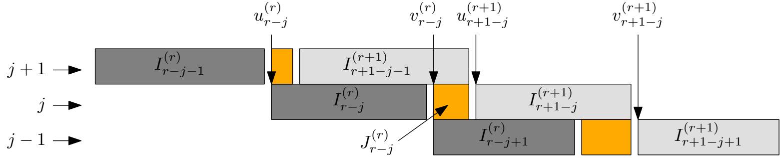
For $r > 0$, the intervals $I^{(r)}_{r-j}$ and $I^{(r+1)}_{r+1-j}$ are disjoint and $u^{(r+1)}_{r+1-j} > v^{(r)}_{r-j}$.

Lemma

A pixel $p(i,j)$ is an absentee if and only if i^2 lies in $J^{(r)}_{r-j} := \left[v^{(r)}_{r-j}, u^{(r+1)}_{r+1-j}\right)$ for some $r \in \mathbb{Z}^+$.

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

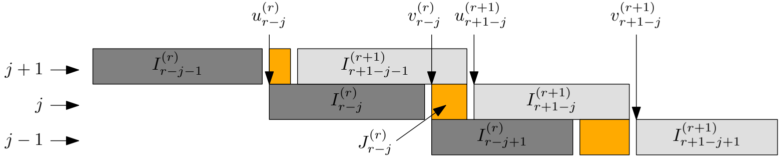
Missing Voxels: Parabolic Characterization XII



Lemma

If $p(i, j)$ is an absentee in Octant 1, then $(i-1, j) \in \mathcal{C}^{\mathbb{Z}}(o, r)$ and $(i+1, j) \in \mathcal{C}^{\mathbb{Z}}(o, r+1)$ for some $r \in \mathbb{Z}^+$.

Missing Voxels: Parabolic Characterization XIII

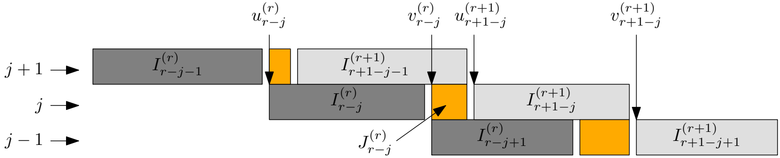


Although the previous lemma provides a way to decide whether or not a given pixel is an absentee, it requires to find for which value(s) of r the existence of square numbers in $J^{(r)}_{r-j}$ has to be checked. *So the following theorem:*

Theorem

(i, j) is an absentee if and only if $i^2 \in J^{(r)}_{r-j}$, where $r = \max\left\{s \in \mathbb{Z} : s^2 < i^2 + j^2\right\}$.

Missing Voxels: Parabolic Family I



$$v^{(r)}_{r-j} = (2k+1)j + k^2, u^{(r+1)}_{r+1-j} = (2k+1)j + (k+1)^2.$$

If $p(i,j)$ lies on kth run of $\mathcal{C}^{\mathbb{Z}}_1(o,r)$, then

$$i^2 < (2k+1)j + k^2;$$

if $p(i,j)$ lies left of $(k+1)$th run of $\mathcal{C}^{\mathbb{Z}}_1(o,r+1)$, then

$$i^2 < (2k+1)j + (k+1)^2.$$

Missing Voxels: Parabolic Family II

Straight line
Time
Gregorian
DSL & DSS
Problem
Periodicity
Properties
Approximate straightness
Circle
Construction
Properties
DCS
DCR & DCH
Segmentation
Properties
DCT
DCG
Surface

The corresponding *open parabolic regions*:

$$\underline{P}_k : x^2 < (2k+1)y + k^2,$$
$$\overline{P}_k : x^2 < (2k+1)y + (k+1)^2.$$

Evidently, the pixels or integer points lying in the region given by $\overline{P}_k \smallsetminus \underline{P}_k$ in Octant 1 for a given pair of j and k — and hence for a given (r, j)-pair — are absentees in Octant 1.

Lemma

Number of square numbers in $J^{(r)}_{r-j} = \left|\left\{(i,j) : (i,j) \in \left(\overline{P}_k \smallsetminus \underline{P}_k\right) \cap \mathbb{Z}^2_1\right\}\right|$.

Missing Voxels: Parabolic Family III

From above lemma, we can derive the region of all absentees for a given value of k by considering all possible values of j for $r \geqslant 0$ so that $r - j = k$. Thus, all the integer points of Octant 1 which are contained in the following *half-open parabolic strip* are absentee points.

$$P_k := \overline{P}_k \setminus \underline{P}_k = (2k+1)y + k^2 \leqslant x^2 < (2k+1)y + (k+1)^2.$$

Lemma

All pixels in $F_k := P_k \cap \mathbb{Z}_1^2$ are absentees.

Missing Voxels: Parabolic Family IV

The family of all the half-open parabolic strips, $P_0, P_1, P_2, \ldots$, thus contains all the absentees in Octant 1.

Theorem

Only and all the absentees of Octant 1 and Octant 8 lie in

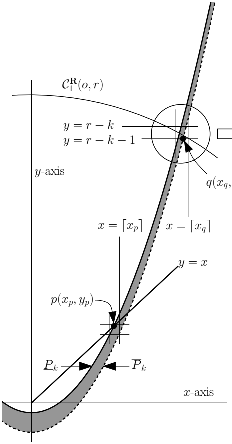
$$\mathcal{F} := \left\{ P_k \cap \mathbb{Z}_1^2 : k = 0, 1, 2, \ldots \right\}.$$

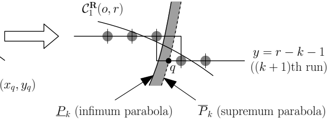
Absentees: Count I

Lemma

For a given k, $P_k \cap \mathbb{Z}_1^2$ contains exactly one absentee on each vertical grid line.

Absentees: Count II





Absentees: Count III

Lemma

The count of absentees contained by the parabolic strip P_k in $\mathcal{D}_1^{\mathbb{Z}}(o, r)$ is given by

$$n_{kr} = \left\lceil \sqrt{(2k+1)r - k(k+1)} \right\rceil - \left\lceil \left((2k+1) + \sqrt{8k^2 + 4k + 1} \right) /2 \right\rceil .$$

Lemma

For a given r, the number of half-open parabolic strips intersecting $\mathcal{C}_1^{\mathbb{Z}}(o, r)$ is given by $m_r = r - \left\lceil r/\sqrt{2} \right\rceil + 1$.

Absentees: Count IV

Theorem

Total count of absentees lying inside $\mathcal{C}^{\mathbb{Z}}(o, r)$ is given by

$$N_r = 8 \sum_{k=0}^{m_r - 1} n_{kr},$$

where $n_{kr} = \left\lceil \sqrt{(2k+1)r - k(k+1)} \right\rceil - \left\lceil 2k + 1 + \frac{1}{2}\sqrt{(8k^2 + 4k + 1)} \right\rceil$ *and* $m_r = r - \left\lceil r/\sqrt{2} \right\rceil + 1$.

Further reading I

- P. Bhowmick and B. B. Bhattacharya.
  Number-theoretic interpretation and construction of a digital circle.
  Discrete Applied Mathematics, 156(12):2381–2399, 2008.
- P. Bhowmick and B. B. Bhattacharya.
  Fast polygonal approximation of digital curves using relaxed straightness properties.
  IEEE Trans. PAMI, 29(9):1590–1602, 2007.
- R. Brons.
  Linguistic methods for description of a straight line on a grid.
  Comput. Graphics Image Process. **2** (1974) 48–62.
- H. Freeman
  Boundary encoding and processing.
  in: B.S. Lipkin, A. Rosenfeld (Eds.), Picture Processing and Psychopictorics, Academic Press, NY (1970) 241–263.

Further reading II

R. Klette and A. Rosenfeld.
Digital Geometry: Geometric Methods for Digital Image Analysis.
Morgan Kaufmann Series in Computer Graphics and Geometric Modeling, 2004.

R. Klette and A. Rosenfeld.
Digital straightness: A review.
Discrete Applied Mathematics, 139(1-3):197–230, 2004.

S. Pal and P. Bhowmick.
Determining Digital Circularity Using Integer Intervals.
Journal of Mathematical Imaging & Vision, 2011 (*in press*).

J.-P. Reveillès.
Géométrie discrète, calcul en nombres entiers et algorithmique.
Thèse d'état, Univ. Louis Pasteur, Strasbourg, 1991.

A. Rosenfeld.
Digital straight line segments.
IEEE Trans. Computers **23** (1974) 1264–1269.

Thank You