David Coeurjolly

Table of contents

Distance Transform

David Coeurjolly

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

2006

David
Coeurjolly

Table of contents

Table of contents

Table of contents

Table of contents

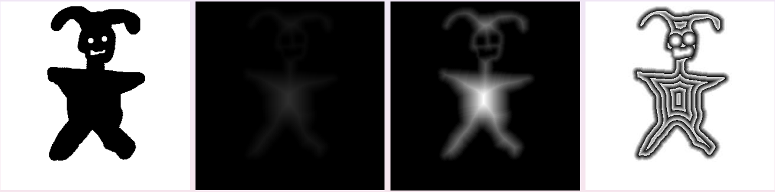
David Coeurjolly

Statement

Context : region based shape analysis

Definition of the DT

Labeling of each pixel p of the object by the distance to the closest point q in the background



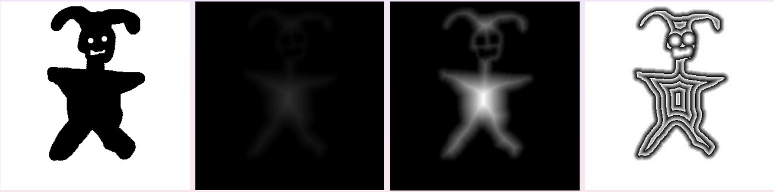
- Classical problem in discrete geometry
- From the DT we can compute:
  - direct measurements (*local width*,...)
  - differential estimators
  - medial axis or skeleton extraction

Statement

Context : region based shape analysis

Definition of the DT

Labeling of each pixel p of the object by the distance to the closest point q in the background



- Classical problem in discrete geometry
- From the DT we can compute:
  - direct measurements (*local width*,...)
  - differential estimators
  - medial axis or skeleton extraction

Table of contents

Approximation of differential estimators

Idea

We consider the boundary of a shape as an implicit surface $f(x,y)=0$ where f is given by the DT

$$f(x,y)=0$$

$$\vec{g}(x,y)=(I_x,I_y)^T$$

$$\vec{t}(x,y)=\frac{(-I_x,I_y)^T}{\sqrt{I_x^2+I_y^2}}$$

$$k=\frac{-t^T Ht}{\|\vec{g}\|},\quad H=\begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix}$$

Approximation of differential estimators

Idea

We consider the boundary of a shape as an implicit surface $f(x,y)=0$ where f is given by the DT


$$\begin{aligned} f(x,y) &= 0 \\ \vec{g}(x,y) &= (I_x, I_y)^T \\ \vec{t}(x,y) &= \frac{(-I_x, I_y)^T}{\sqrt{I_x^2 + I_y^2}} \end{aligned}$$

$$k = \frac{-t^T H t}{\|\vec{g}\|}, \quad H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix}$$

Medial Axis and Skeleton extraction

Shape → DT → Medial Axis

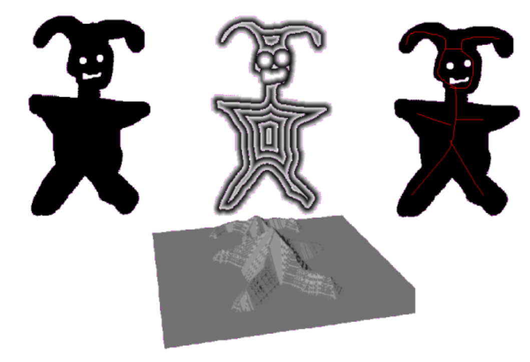


Table of contents

Discrete metrics

Constraint

Definition of a metric based on integer numbers

- Rounded metric $\lceil d \rceil$
- Approximate the Euclidean metric with integers: Chamfer masks
- DT based on sequences of chamfer masks
- Displacement based DT $(d_x, d_y)^T$
- DT based on the squared Euclidean distance SDT

Axioms of a metric

d is a metric on an non-empty set S iff:

$\forall p, q, r \in S$:

- $d(p,p) = 0$
- $d(p,q) = d(q,p)$
- $d(p,r) \leq d(p,q) + d(q,r)$

Ball

A ball of radius r with center p is the set of points q in S such that:

$$d(p,q) < r$$

Rounded Euclidean metric

If d is a metric then $\lceil d \rceil$ is also a metric

- $\lceil d(p,p) \rceil = 0$
- $\lceil d(p,q) \rceil = \lceil d(q,p) \rceil$
- $\lceil d(p,r) \rceil \leq \lceil d(p,q) \rceil + \lceil d(q,r) \rceil$ $(\forall a,b,c \in \mathbb{R}, \quad a+b \geq c \Rightarrow \lceil a \rceil + \lceil b \rceil \geq \lceil c \rceil)$

$\Rightarrow$ Yes !

And what about $\lfloor d \rfloor$ or $[d]$?

No!

Chamfer metrics

[Bor86, Ver91, Thi01, FM05]

Idea

First, we fix a set of elementary displacements and then we affect a weight to each step in order to approximate the Euclidean distance.

—	1	—
1	0	1
—	1	—

1	1	1
1	0	1
1	1	1

General masks

$(\rightarrow, \nearrow)$:

b	a	b
a	0	a
b	a	b

$(\rightarrow, \nearrow, \rightarrow + \nearrow)$:

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

$\ldots$

Chamfer metrics

[Bor86, Ver91, Thi01, FM05]

Idea

First, we fix a set of elementary displacements and then we affect a weight to each step in order to approximate the Euclidean distance.

General masks

$(\rightarrow, \nearrow)$:

b	a	b
a	0	a
b	a	b

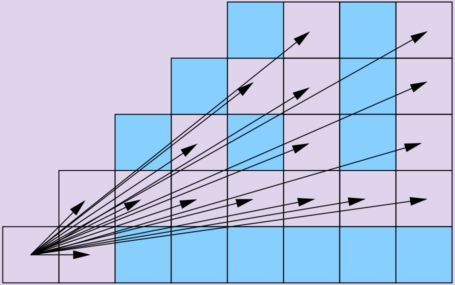
$(\rightarrow, \nearrow, \rightarrow + \nearrow)$:

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

...

Elementary displacements: *Farey series*

(u, v) is valid if $\frac{v}{u}$ is an irreducible fraction



Elementary displacements in a $m \times m$ mask $\Leftrightarrow$ fractions in the Farey series $\mathcal{F}_m$

How to compute the weights ?

Objectives

- The mask must form a metric:
  - $d(p,p) = 0$
  - $d(p,q) = d(q,p)$
  - $d(p,r) \leq d(p,q) + d(q,r)$
- Trade-off between the approximation of the Euclidean distance and the size of the mask

Cost function to minimize

$$d_{mask}(p,q) - d_{euc}(p,q)$$

Over a domain defined by linear constraints

- 3x3 mask : $b < 2a$ and $b > a$
- 5x5 mask : $2a < c$, $3b < 2c$ and $c < a + b$
- ...

How to compute the weights ?

Objectives

- The mask must form a metric:
  - $d(p,p) = 0$
  - $d(p,q) = d(q,p)$
  - $d(p,r) \leq d(p,q) + d(q,r)$
- Trade-off between the approximation of the Euclidean distance and the size of the mask

Cost function to minimize

$$d_{mask}(p,q) - d_{euc}(p,q)$$

Over a domain defined by linear constraints

- 3x3 mask : $b < 2a$ and $b > a$
- 5x5 mask : $2a < c$, $3b < 2c$ and $c < a + b$
- ...

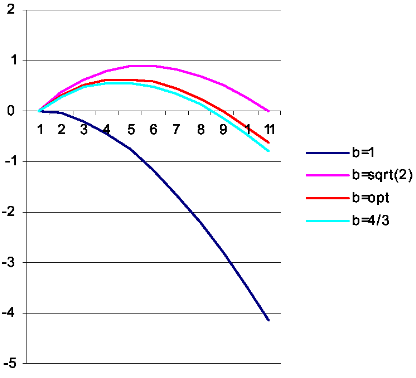
Weight analysis

3x3 masks [Bor86]

$$Diff = yb + (x-y)a - \sqrt{x^2+y^2}$$

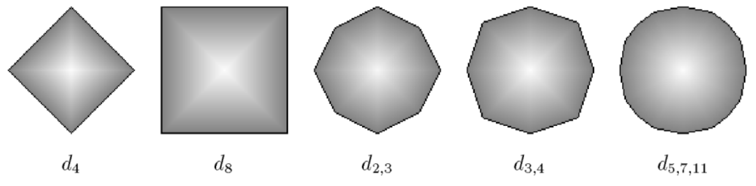
$\Rightarrow a_{opt} = 0.95509$ and $b_{opt} = 1.3693$

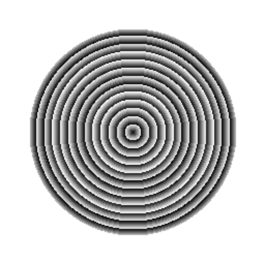
Example: $a = 1$,



- x-axis: the column $x = 10$ and $1 \leq y \leq 10$
- y-axis: the error function $yb + (x-y)a - \sqrt{x^2+y^2}$
- $b = 1$: d_8 or chessboard metric

Unit balls





(Chamfer ball images from [Thi01])

Sequence of chamfer masks

[RP66, MDKC00, Nag05]

Idea

We consider a set of chamfer masks (*e.g.* d_4 and d_8) and we switch the masks to find a better approximation of d_{euc} in a DT problem

Problems to solve

- Find the set of chamfer masks
- Find the best sequence of the masks to approximate d_{euc}

Chamfer metrics - summary

+ Simple computations
- Approximation of the Euclidean distance
- Not an isotropic representation of an object
+ DT is easy to implement

Main drawbacks

If you:

- change the shape of the pixels (*e.g.* elongated grids with factor λ),
- change size of the mask, or
- change the dimension of the image

you have to update the weights with a new optimization process

Exact Euclidean metrics

DT based on displacement vectors (d_x, d_y)

+ Error free representation
- Complex DT to obtain an error free computation
- We have to store two coordinates

DT based on the squared Euclidean distance (d^2_{euc})

+ Error free representation
+ Fast error free DT
- We have to store the square of integer numbers

Table of contents

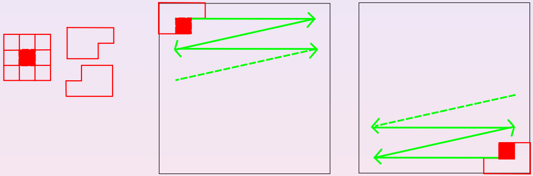
Table of contents

Computation of the Chamfer DT

[RP66, RP68, Bor86]

Sketch of the algorithm

Decomposition of the mask into two parts and double scan of the image to update the *min* distance



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

Initialization :

$$DT(i,j) = 0 \quad \text{if} \quad (i,j) \notin Object$$
$$DT(i,j) = +\infty \quad \text{if} \quad (i,j) \in Object$$

Rem: the displacement $(0,0)$ with weight 0 belongs to the mask...

Example

Using the d_{3-4} mask

3	3	3	3	3	3
3	6	6	6	6	4
3	6	9	9	8	4
3	6	9	12	8	
		3	6	4	

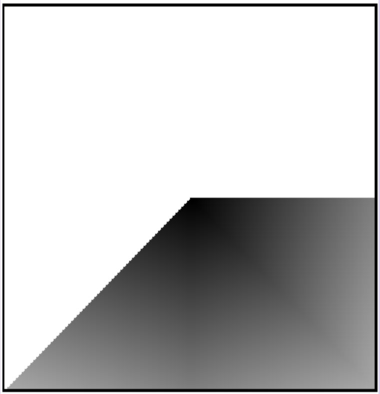
$\longrightarrow$

3	3	3	3	3	3
3	6	6	6	6	3
3	6	7	7	4	3
3	3	4	6	3	
		3	3	3	

(image from [Thi01])

Example

Single background pixel at the center of the image



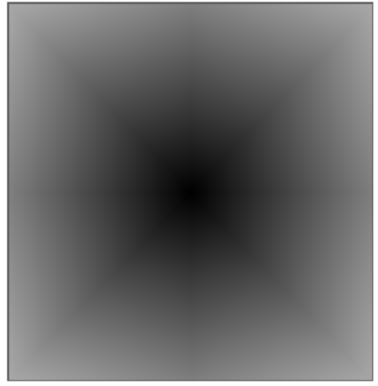


Table of contents

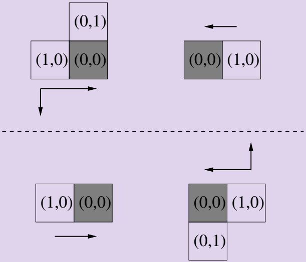
Vector DT

Idea - [Dan80, Mul92]

Store at each pixel the vector $\vec{v} = (x_p, y_p)$ such that $DT(p) = |v(p)|$

Danielson's algorithm [Dan80]

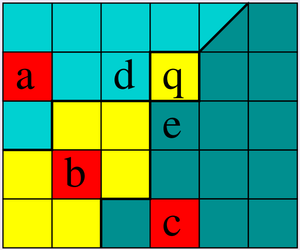
Multiple scan process with directional masks (4-connected). At each step we update $\vec{v} = (x_p, y_p)$ with the vector with the minimal distance.



- Top-down scan
- Bottom-up scan

Errors in Danielson's VDT

Local updates can lead to incorrect DT

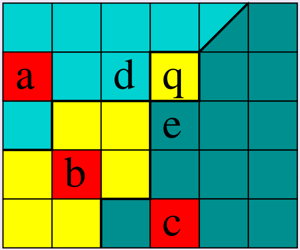


- a, b and c are background pixels
- Danielson's algorithm: $a - d$, $c - e$ but $c - q$ or $a - q$
- Correct algorithm: $a - d$, $c - e$ and $b - q$

Algorithms exist to correct these pathological cases leading to error-free VDT [Cui99, CM99]

Errors in Danielson's VDT

Local updates can lead to incorrect DT



- a, b and c are background pixels
- Danielson's algorithm: $a - d$, $c - e$ but $c - q$ or $a - q$
- Correct algorithm: $a - d$, $c - e$ and $b - q$

Algorithms exist to correct these pathological cases leading to error-free VDT [Cui99, CM99]

Squared Euclidean Distance Transform

Idea

Store the square of the EDT

△ d_{euc}^2 is not a metric

Squared Euclidean Distance Transform

Idea

Store the square of the EDT

$\triangle$ d_{euc}^2 is not a metric

Squared Euclidean Distance Transform

[ST94, Hir96, MRH00]
Let P be the background of the object F, the SEDT at q in F is given by:

$$s(q) = \min_{p \in P}\{d^2_{euc}(p,q)\}$$

If we decompose the problem with $q(i,j)$, we have:

$$s(q) = \min_{p(x,y) \in P}\{(x-i)^2 + (y-j)^2\}$$

and:

$$g(i,j) = \min_x\{(x-i)^2\}$$

with

$$s(i,j) = \min_y\{(y-j)^2 + g(i,y)\}$$

⇒ dimensional decomposition of the DT computation

⇒ separable technique

Step 1: $g(i,j) = \min_x\{(x-i)^2\}$

Simple 2 scan algorithm

- Input row:

∞	■	∞	∞	∞	■	∞	∞

- $\rightarrow$:

∞	■	1	4	9	■	1	4

- $\leftarrow$:

1	■	1	4	1	■	1	4

Computational cost

- $O(n^2)$ for a $n \times n$ image
- $O(n^d)$ for a n^d image

Step 1: $g(i,j) = \min_x\{(x-i)^2\}$

Simple 2 scan algorithm

- Input row:

∞	■	∞	∞	∞	■	∞	∞

- $\rightarrow$:

∞	■	1	4	9	■	1	4

- $\leftarrow$:

1	■	1	4	1	■	1	4

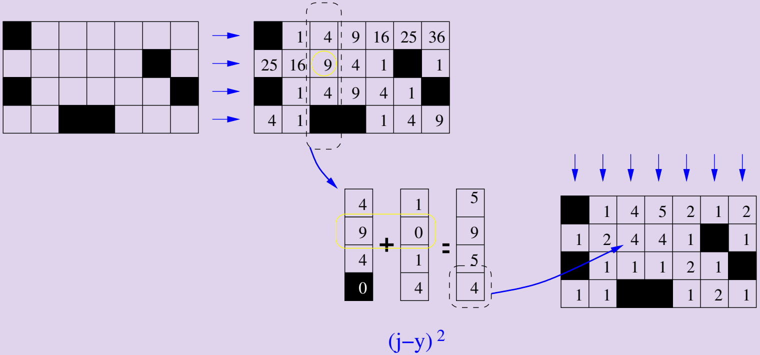
Computational cost

- $O(n^2)$ for a $n \times n$ image
- $O(n^d)$ for a n^d image

Step 2: $s(q) = \min_y \{(y-j)^2 + g(i,y)\}$

Straightforward algorithm

For each cell (i,j), scan the y to compute the *min*

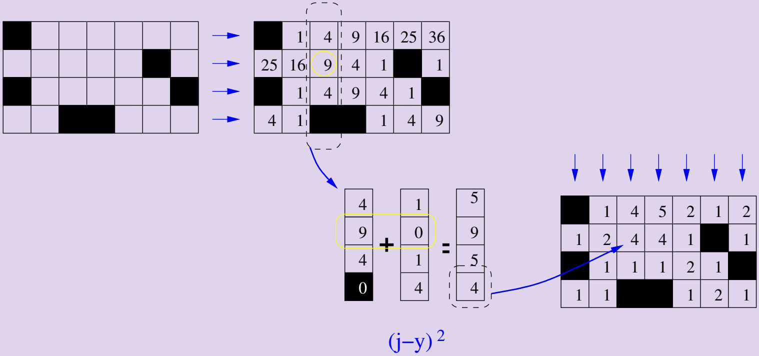


$O(n^3)$ for a 2D image but we can design $O(n^2)$ algorithms [ST94]

Step 2: $s(q) = \min_y \{(y - j)^2 + g(i,y)\}$

Straightforward algorithm

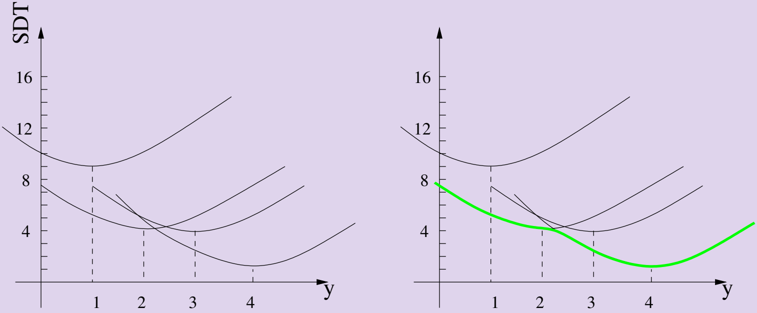
For each cell (i,j), scan the y to compute the *min*



$O(n^3)$ for a 2D image but we can design $O(n^2)$ algorithms [ST94]

Optimal Step 2 algorithm: Parabola lower envelope computation

$\{(y-j)^2 + g(i,y)\}$: family of parabolas



column $[9,4,4,1]$ after step 1 and $[5,4,2,1]$ after step 2

[Hir96, MRH00]

Linear in time lower envelope computation

Sketch of the algorithm [Hir96, MRH00]

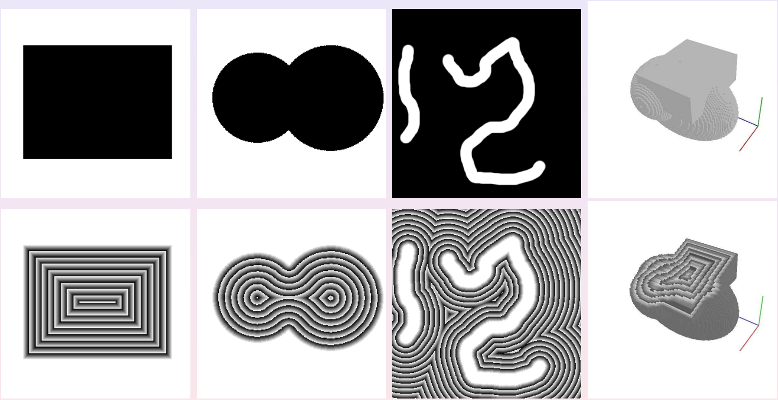
- We scan the parabolas and use a stack to store the parabolas that belong to the lower envelope
- When a new parabola is considered, this parabola may invalidate some parabolas in the stack $\rightarrow$ we pop the parabolas on the stack while the parabola on the top of the stack is invalidated by the new one
- When no more parabolas have to be considered, we compute the SDT map with the heights of the lower envelope parabolas

Computational analysis

Linear process in the number of parabolas

- $O(n^2)$ for a $n \times n$ image
- $O(n^d)$ for a n^d image

Some results of the SEDT



$\Rightarrow$ *optimal in time algorithms and error free DT whatever the dimension*

SDT - summary

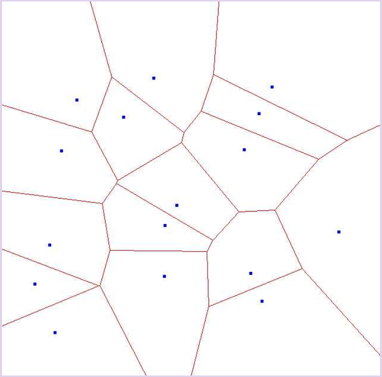
- Optimal algorithms to compute error free SDT
- Trivial generalizations to d-dimensional objects
- Can handle elongated factors
- Based on a isotropic metric

Table of contents

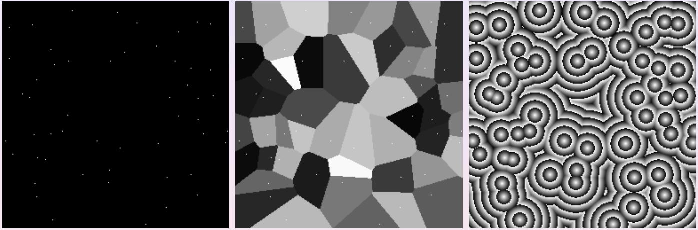
Voronoi diagram in Computational Geometry

Definition in $2-D$

Given a set of sites $\mathcal{S} = \{s_i\}$ in $\mathbb{R}^2$, the Voronoi diagram is a decomposition of the plane into cells $\mathcal{C} = \{c_i\}$ (one cell per site) such that for each point p in the open cell c_i, we have $d(p, s_i) < d(p, s_j)$ for $i \neq j$



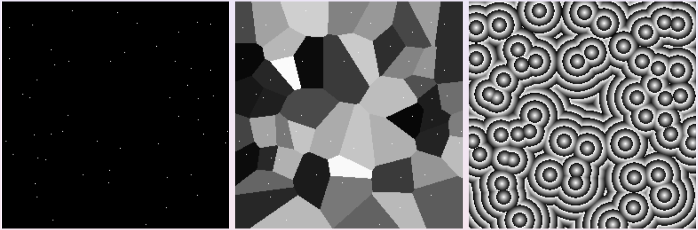
Voronoi diagrams and EDT



EDT $\Leftrightarrow$ rewriting the Voronoi diagram labeling of background points

DT algorithms based on the Voronoi diagram extraction exist to compute the EDT

Voronoi diagrams and EDT

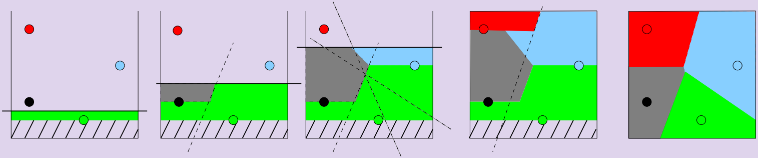


EDT $\Leftrightarrow$ rewriting the Voronoi diagram labeling of background points

DT algorithms based on the Voronoi diagram extraction exist to compute the EDT

Sweep line technique to construct 2-D discrete Voronoi diagrams

[BGKW95, GM98]

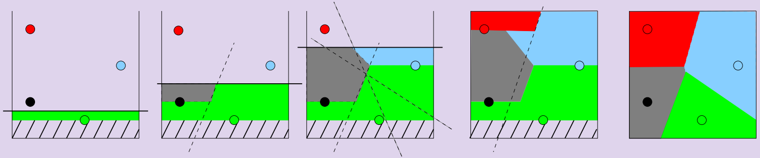


- 2 scan process to construct the diagram
- When we move from a row to the next one, we use a process based on a stack of sites to update the Voronoi diagram on the current row

$O(n^2)$ for a 2-D image

Sweep line technique to construct 2-D discrete Voronoi diagrams

[BGKW95, GM98]



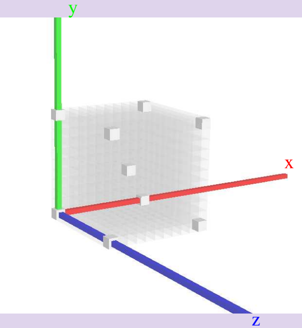
- 2 scan process to construct the diagram
- When we move from a row to the next one, we use a process based on a stack of sites to update the Voronoi diagram on the current row

$O(n^2)$ for a 2-D image

Generalizations to higher dimensions

Idea - [Coe02, CRMQR03]

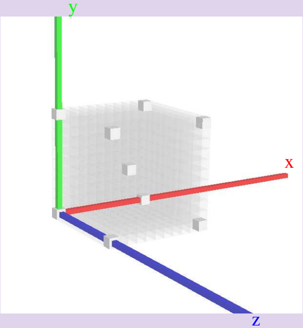
Separable decomposition of the d-dimensional Voronoi diagram

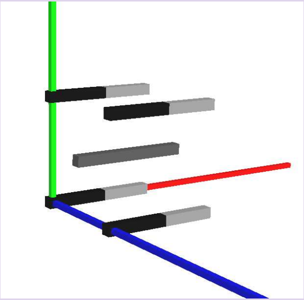


Generalizations to higher dimensions

Idea - [Coe02, CRMQR03]

Separable decomposition of the *d*-dimensional Voronoi diagram

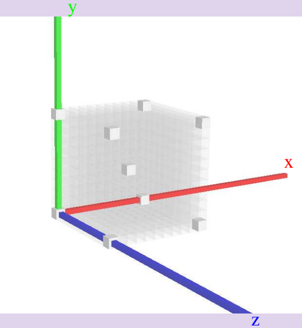


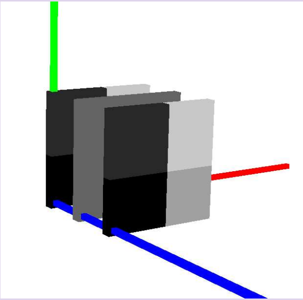


Generalizations to higher dimensions

Idea - [Coe02, CRMQR03]

Separable decomposition of the d-dimensional Voronoi diagram

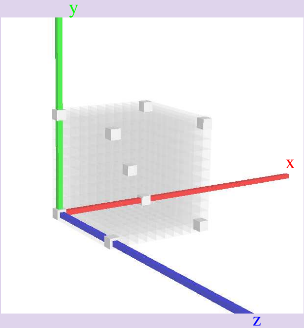


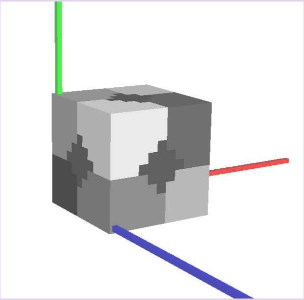


Generalizations to higher dimensions

Idea - [Coe02, CRMQR03]

Separable decomposition of the d-dimensional Voronoi diagram





Computational cost

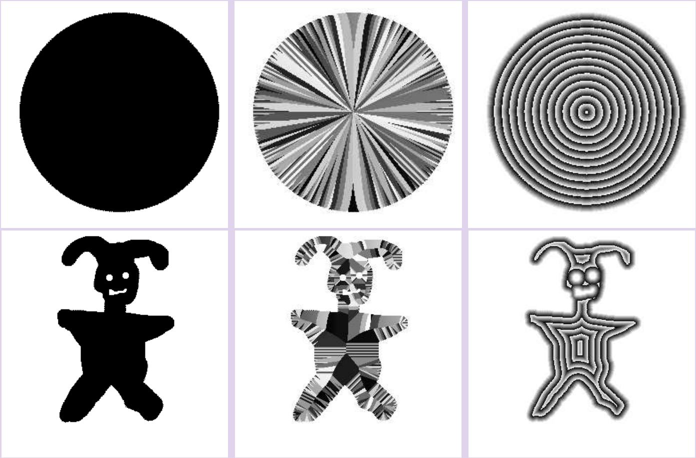
Analysis

- the problem is decomposed into several $1-D$ Voronoi diagram constructions
- each $1-D$ problem can be solved in linear time

$\Rightarrow O(n^2)$ for $2-D$ images and $O(n^d)$ for $d-D$ images

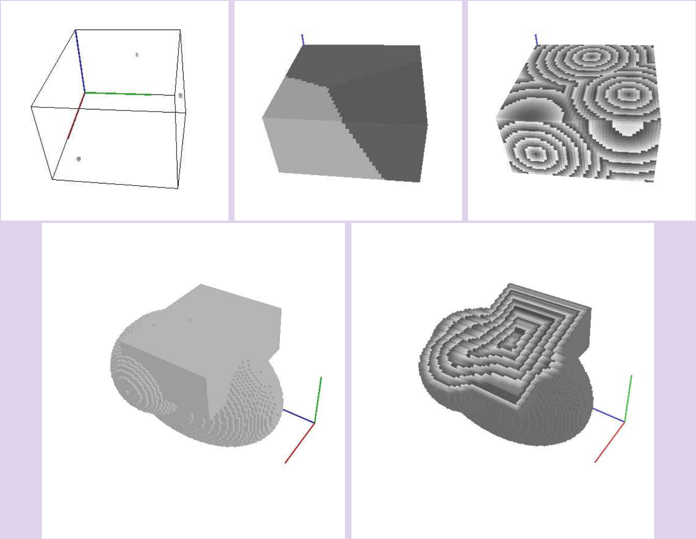
Results

2-D



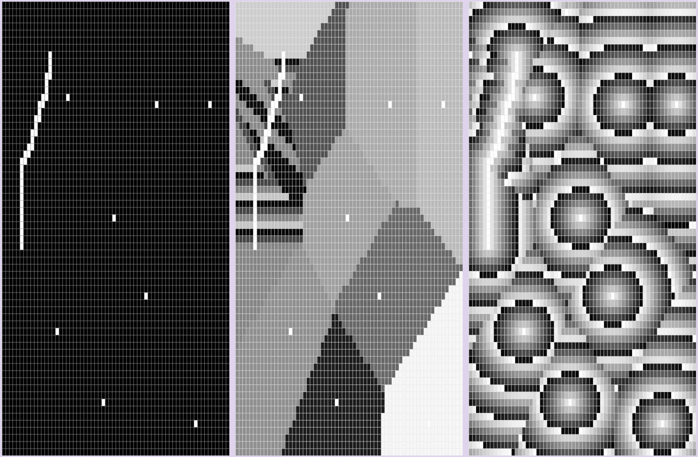
Results

3-D



Generalization for separable techniques

Anisotropic grids - [Coe02]

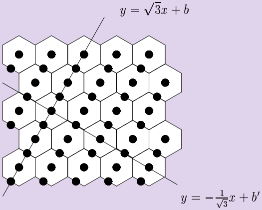


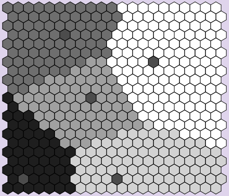
Hexagonal grids - [Coe02]

Generalization for separable techniques

Anisotropic grids - [Coe02]

Hexagonal grids - [Coe02]





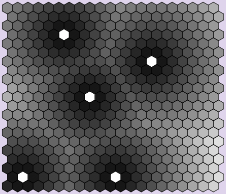


Table of contents

Conclusion

- Optimal algorithms to compute the DT based on the error free Euclidean metric or Chamfer metrics
- Links between DT and classical objects in the Computational Geometry
- We also have *Farey series* in DT problems !

Codes are available on the TC18 webpages

http://www.cb.uu.se/~tc18/

Technical Committee 18 “Discrete Geometry” of the International Association on Pattern Recognition (IAPR)

Conclusion

- Optimal algorithms to compute the DT based on the error free Euclidean metric or Chamfer metrics
- Links between DT and classical objects in the Computational Geometry
- We also have *Farey series* in DT problems !

Codes are available on the TC18 webpages

`http://www.cb.uu.se/~tc18/`

Technical Committee 18 “Discrete Geometry” of the International Association on Pattern Recognition (IAPR)

H. Breu, J. Gil, D. Kirkpatrick, and M. Werman.
Linear time euclidean distance transform algorithms.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 17, 1995.

G. Borgefors.
Distance transformations in digital images.
Computer Vision, Graphics, and Image Processing, 34(3):344–371, June 1986.

O. Cuisenaire and B. Macq.
Fast euclidean distance transformations by propagation using multiple neighbourhoods.
Computer Vision and Image Understanding, 76, November 1999.

D. Coeurjolly.
Algorithmique et géométrie discrète pour la caractérisation des courbes et des surfaces.
PhD thesis, Université Lumière Lyon 2, Bron, Laboratoire ERIC, dec 2002.

Jr. C. R. Maurer, R. Qi, and V. Raghavan.
A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions.
IEEE Trans. on Pattern Analysis and Machine Intelligence, 25(2):285–270, feb 2003.

O. Cuisenaire.
Distance Transformations : Fast Algorithms and Applications to Medical Image Processing.
PhD thesis, Université Catholique de Louvain, oct 1999.

P. E. Danielsson.
Euclidean distance mapping.
CGIP, 14:227–248, 1980.

C. Fouard and G. Malandain.
3-d chamfer distances and norms in anisotropic grids.
Image and Vision Computing, 23(2):143–158, February 2005.

W. Guan and S. Ma.
A list-processing approach to compute voronoi diagrams and the euclidean distance transform.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 20:757–761, 1998.

David Coeurjolly

Definitions

Applications

Discrete metrics

DT Algorithms

Chamfer based DT

Euclidean DT

Voronoi diagram based DT

Conclusion

T. Hirata.
A unified linear-time algorithm for computing distance maps.
Information Processing Letters, 58(3):129–133, May 1996.

J. Mukherjee, P. P. Dasa, M. Aswatha Kumarb, and B. N. Chatterjib.
On approximating euclidean metrics by digital distances in 2d and 3d.
Pattern Recognition Letters, 21(6–7):573–582, 2000.

A. Meijster, J.B.T.M. Roerdink, and W. H. Hesselink.
A general algorithm for computing distance transforms in linear time.
In *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 331–340. Kluwer, 2000.

J. C. Mullikin.
The vector distance transform in two and three dimensions.
Computer Vision, Graphics, and Image Processing. Graphical Models and Image Processing, 54(6):526–535, November 1992.

Benedek Nagy.
A comparison among distances based on neighborhood sequences in regular grids.
In *SCIA*, pages 1027–1036, 2005.

A. Rosenfeld and J. L. Pfaltz.
Sequential operations in digital picture processing.
Journal of the ACM, 13(4):471–494, October 1966.

A. Rosenfeld and J. L. Pfalz.
Distance functions on digital pictures.
Pattern Recognition, 1:33–61, 1968.

T. Saito and J. I. Toriwaki.
New algorithms for Euclidean distance transformations of an *n*-dimensional digitized picture with applications.
Pattern Recognition, 27:1551–1565, 1994.

E. Thiel.
Géométrie des distances de chanfrein.

Habilitation à Diriger des Recherches, Université de la Méditerranée, Aix-Marseille 2, Déc 2001.

B. J. H Verwer.
Local distances for distance transformations in two and three dimensions.
Pattern Recognition Letters, 12:671–682, november 1991.