Digital geometry

Digital geometry in 3D
and
Applications using distance transforms

Robin Strand
Outline

• Digital volume (3D) images
• Applications using distance transform
  • Path-planning
  • Matching
  • Skeletonization
  • Measurements
Digital volume images

In the cubic grid:
Each voxel v has three types of neighbors

- 6 voxels are sharing a face with v
- 12 voxels are sharing an edge with v
- 8 voxels are sharing a vertex with v

in total 26 neighbors
Digital volume images

Three different neighborhoods of a voxel

6-neighborhood
face neighbors

18-neighborhood
face and edge neighbors

26-neighborhood
face, edge, and vertex neighbors
“A 3D digital arc is a digital straight line segment if two of its projections onto the principal planes are 2D digital straight lines.”

There should be a 1-1 correspondence between the points in the 3D digital arc and the points in the two projections.
Connectivities

Connectivity paradox also for 3D images
Solution 1:

26-connectedness for object
6-connectedness for background

or

26-connectedness for background
6-connectedness for object
Connectivities

Connectivity paradox also for 3D images

Solution 2:

Cellular complexes
Connectivities

Connectivity paradox also for 3D images

Solution 3:

Use other grids

face-centered cubic grid (FCC)  body-centered cubic grid (BCC)
Why use non-standard grids?

Aliasing is folding in frequency domain. The denser grid in frequency domain, the lower redundancy is needed to avoid aliasing.

The density of a grid is the fraction of the space that is covered by non-overlapping, equally sized balls of maximal radius centred at the grid points.
Why use non-standard grids?

The densest packing is the
• Hexagonal grid in 2D
• FCC grid in 3D
Why use non-standard grids?

Dual grid by the convolution theorem

“Basis $V$ in spatial domain $\Rightarrow$ basis $(V^{-1})^T$ in freq. domain”
Why use non-standard grids?

“Basis $V$ in spatial domain => basis $(V^{-1})^T$ in freq. domain”

Square grid:

$$V = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad V^{-T} = V$$

Hexagonal grid:

$$V = \begin{pmatrix} \sqrt{3}/2 & 0 \\ 1/2 & 1 \end{pmatrix} V^{-T} = \begin{pmatrix} 2/\sqrt{3} & -1/\sqrt{3} \\ 0 & 1 \end{pmatrix}$$
Why use non-standard grids?

“Basis $V$ in spatial domain $\Rightarrow$ basis $(V^{-1})^T$ in freq. domain”

FCC grid:

$$V = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

$$V^{-T} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{pmatrix}$$

BCC grid

BCC in spatial domain $\Leftrightarrow$ FCC in frequency domain
Sampling efficiency
Sampling efficiency, square grid

\[
\frac{1}{\Delta x} \quad \frac{1}{\Delta y}
\]
Sampling efficiency, hexagonal grid
Distance transforms in 3D

Raster scanning

<table>
<thead>
<tr>
<th></th>
<th>Square grid (2D)</th>
<th>Cubic grid (3D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>weighted</td>
<td>2 scans</td>
<td>2 scans</td>
</tr>
<tr>
<td>Euclidean</td>
<td>3 scans</td>
<td>4 scans</td>
</tr>
</tbody>
</table>

Masks for weighted distance

Masks for Euclidean distance
Applications using distance transforms

- Path-planning
  - for robots
- Matching
  - find a subimage in an image
- Skeletonization
  - curve representation of 2D object
  - centers of maximal balls
- Measurements
- Morphological operations (next lecture)
Applications: Path-planning
Path planning: Shortest path between two pixels

- Compute DT: distances from pixel $p_0$
- Search for $p_0$ from $p_n$ in direction of gradient
- Use constrained DT in case of obstacles
Descending mountains using steepest path
Example of path planning

Walk in the direction with the largest slope

\[
\frac{I(p_i) - I(p_i + n_j)}{w_{k(j)}}
\]

\(n_j, j=1,\ldots,8\) are neighbors of \(p_i\)

\(w_{k(j)}\) is the weight used to neighbor \(n_j\)
Applications: Matching
Matching

Used in segmentation to locate known objects in an image, to search for specific patterns etc.

- Registration, matching of car number plates, ”målsökning”
- Match-based segmentation localizes all image positions at which close to copies of the search pattern is located
Three classes of matching

- Image pixel values directly
  - e.g., correlation methods ("Matching by correlation" 12.2 Gonzalez-Woods)
- Low-level features
  - edges and corners
- High-level features
  - identified (parts of) objects or relations between features, e.g., graph-theoretic methods
Chamfer matching

- Algorithm based on distance transform to locate one-dimensional features (edges)
- Good response in close to correct positions, but poor elsewhere
- Technique for finding best fit of edge points from two different images by minimizing a generalized distance between them

first introduced by Barrow et al. in 1977
Matching

- Find unknown objects
- Hierarchical Chamfer Matching Algorithm
  - Start from edge image
  - DT from edges
  - Search for position giving smallest error

image to find!
Chamfer matching algorithm

- input = search image & template image
- output = image with templates overlayed on the best matching
- start = start positions spread all over the image random covering the image or using a priori knowledge

\[ \sqrt{\frac{1}{n} \sum_{i=1}^{n} v_i^2} \]

- Extract edges in both search image and template image
- Compute DT of the search edge image DT_search
- Superimpose edge_template on DT_search in all start positions (and rotations, translations, scalings)
- Compute root-mean-squares for pixel values that the edges hit → edge distance
- Optimize by small steps in the directions of lower edge distances
Root-mean-square error:

\[
\frac{1}{3} \sqrt{\frac{1}{8} \sum_{i=1}^{8} v_i^2}
\]
HIERARCHICAL
Chamfer matching

- Chamfer matching needs good starting positions
- Embed chamfer matching in a resolution pyramid

HCMA=hierarchical chamfering matching algorithm
Resolution pyramid

- A set of images, $I_0, \ldots, I_n$, of decreasing resolution
- Size of $I_k$ is $\frac{1}{4}$ ($\frac{1}{8}$ for 3D) of $I_{k-1}$
- Lower level by partitioning the array into $2 \times 2(\times 2)$ block of pixels, children, and associate a single pixel, parent
- Parent is set to object or background depending on the color of its children according to some fixed rule (AND, OR, ...)

Resolution pyramid
Resolution pyramids

I_0

I_1

I_2

I_3

"color" of 2x2x2 children gives "color" of parent
Resolution pyramid for HCMA

use OR to preserve edges

128×128  64×64  32×32  16×16  8×8
HCMA

• Chamfer matching
• In resolution pyramid
  – Gives speed up (reduced computations as low-resolution images are used initially)
  – Start positions for original image are reduced as positions are rejected because of too high edge distance value on low levels.
Results

0: edge distance: 0.00
1: edge distance: 1.36
2: edge distance: 1.43
3: edge distance: 1.44
4: edge distance: 1.57
5: edge distance: 1.63
6: edge distance: 1.67
Free camera model

- Example: match lake in aerial photograph with lake edges from a map
- Six parameter problem
  - translation x, y, z
  - rotation
  - scaling
  - perspective
- For every parameter a number of start positions are chosen at highest level (low resolution)
Applications: Skeletonization
Medial axis transform

Often described as being the “locus of local maxima” on a distance map.

Augmented by radial function, the *quench* function.

Blum 1967
Medial axis representation

Compact representation of objects.

Applications:

- Object description
- Object recognition
- Navigation
- Animation
- ...

Medial axis representation

Applications:

• Object description
• Object recognition

• Navigation
• Animation

The object should be fully described by the representation

Only the most important features are needed

• ...

Topology

- Description invariant under "rubber sheet" transformation
- Here, two binary images are "topologically equivalent" if they are homotopic
- Can be described using graphs, homotopy tree. A transformation is homotopic if it does not change the homotopy tree
- Euler number
  \[ E = C - H: \]
  - \( C \) – number of connected components
  - \( H \) – number of holes

Terminology:
- homotopic
- homeomorphic
- topologically equivalent
Medial axis representation in digital images

This can be done in different ways, for example:

- Centers of maximal balls (CMBs)
- Homotopic thinning
- Homotopic thinning keeping the CMBs
- Template matching

Different approaches give different properties of the medial axis.
Medial axis representation in digital images

Centers of maximal balls
Medial axis representation in digital images

- **Maximal ball** – ball in the object that is not covered by any other ball in object.
- **CMB** – its center.

Depends on the distance function!

The *Quench function* associates the radius to each CMB.

Compare with the continuous case.
Pixels as centers of balls

Distance label of pixel $p$ can be interpreted as radius of a ball $B(p,d(p))$, centered on $p$

$B(p,d(p))$ is fully enclosed in the object

chessboard distance
Centers of maximal balls

If not completely covered by any other disc

Note: Not all CMBs needed for reconstruction
Centers of maximal balls (CMB)

- Appear as local maxima in DT for weighted distances(!)
- Union of all discs corresponding to CMBs = object

(cityblock, chessboard, \(<3,4>\), Euclidean)
Centers of maximal balls

A pixel is a center of maximal ball if it is a local maximum in the DT.

*(note! take local distance into account)*

for pixel in $<a,b>$ WDT labeled $p$:
- edge neighbors $< p+a$
- vertex neighbors $< p+b$

for city-block:
- edge neighbors have lower or equal label

for chessboard:
- neighbors have lower or equal label
Centers of maximal balls

Original image

Sets of CMBs with different distance functions

- city-block
- chessboard
- $<3,4>$-weighted
cityblock is used for this example
Complete description by CMBs

Object can be represented by its CMBs as it is the union of the maximal balls

Reverse distance transformation can be used to recover the object

- object
- object = grey
- CMBs = black
- reverse DT
Reverse DT from CMBs

max-operation

<3,4> weighted

Propagate from CMBs (in bold)

after forward scan

backward scan
Centers of maximal balls for Euclidean DT

- Not enough to check distance values of neighbors
- Maximal ball: not covered by any other single ball

Remember: the 3x3 neighborhood does not hold enough information about the Euclidean distance

- Simple local comparisons not enough: use look-up tables
Medial axis representation in digital images

Homotopic thinning using simple points
Simple pixels

Pixels that can be removed without altering topology:
• the number of object components and
• the number of background components
are the same before and after removal
Simple pixels
by local neighborhood operations

Decision on whether a pixel is simple or not can be taken based on local neighborhood configuration. For 8-connected object and 4-connected background:

- $N^8(v)$: number of object components in an 8-neighborhood of $v$
- $\overline{N}^8(v)$: number of background components in an 8-neighborhood of $v$, edge connected to $v$

$v$ is simple if

- $N^8(v) = 1$
- $\overline{N}^8(v) = 1$
Homotopic thinning

- Remove border after border if
  - simple pixel
- Number of iterations is dependent on object thickness

Repeat until stability

\{ 
- Find border pixels
- Remove border pixels if simple
\}

OR use distance transform to define borders!
Homotopic thinning

Original image

Result after homotopic thinning (removing only simple points).
Medial axis representation in digital images

Homotopic thinning keeping the CMBs
Homotopic thinning keeping the CMBs

Keep CMBs and remove simple points sequentially

- Compute distance transform
- Remove border after border if
  - not a CMB
  - simple pixel
- Number of iterations is dependent on object thickness
Homotopic thinning keeping the CMBs

Original image

city-block  chessboard  <3,4>-weighted
Homotopic thinning keeping the CMBs with different DTs

Different aspects:
- shape preservation
- compression
- stability under rotation
Medial axis representation in digital images

Homotopic thinning by template matching
Thinning using morphology

- Sequential thinning by a sequence of structuring elements (SE, "masks")
  - Application of hit-or-miss
  - Identify border pixels (use DT)
  - Remove pixels satisfying one SE
  - Composite SEs: object, background, don’t care

\[
L_1 = \begin{bmatrix}
0 & 0 & 0 \\
* & 1 & * \\
1 & 1 & 1
\end{bmatrix} \quad L_2 = \begin{bmatrix}
* & 0 & 0 \\
1 & 1 & 0 \\
* & 1 & *
\end{bmatrix} \quad \ldots
\]

$L$ from Golay alphabet gives homotopic thinning
Thinning by template matching using the templates on the previous slide.
Skeletal properties

- In an image with object O and background B, the skeleton S is categorized by the following properties
  - S is topologically equivalent to O
  - S is centered within O
  - S is unit-wide
  - O is recovered by reversing S

Sometimes skeleton is defined as a transformation having all these properties.
Skeletal properties

- S is topologically equivalent to O: no
- S is centered within O: yes
- S is unit-wide: no
- O is recovered by reversing S: yes
Skeletal properties

Homotopic thinning

- S is topologically equivalent to O yes
- S is centered within O yes
- S is unit-wide yes
- O is recovered by reversing S no
Skeletal properties

- S is topologically equivalent to O  yes
- S is centered within O  yes
- S is unit-wide  no
- O is recovered by reversing S  yes

Homotopic thinning keeping the CMBs
Skeletal properties

- S is topologically equivalent to O  yes
- S is centered within O  yes
- S is unit-wide  yes
- O is recovered by reversing S  no

Thinning by template matching
Classes of points in skeleton in 2D

Equidistance from one boundary component

- end point

Equidistant from two boundary components

- curve point

Equidistant from three or more boundary components

- branch point
Thickness and length measurements

**Thickness:**
highest distance label in object gives maximum thickness

**Length of curve:**
distance propagation along the curve starting from one end-point (for instance by constrained DT)
Skeletons in 3D

Similar methods as in 2D apply to 3D.

We need to define

- Homotopic transformations,
- Simple points, and
- CMBs in 3D.
Skeletons in 3D

Basic notions

**concavity**
- dent on the object

**tunnel**
- background passing through the object

**cavity**
- background component enclosed in the object
Homotopic transformation

Here, a transformation is homotopic (topology preserving) if it can be written as a sequence of adding/removing simple points.

In 2D, the number of components and holes remain unchanged under the transformation.

In 3D, the number of object components, the number of cavities and the number of tunnels remain unchanged.
Topology preserving removal

A point is **simple** iff its removal does not alter the topology

non-simple (object)  non-simple (background)  simple

Can be detected in a similar way as for 2D images.
Simple points

Decision on whether a voxel is simple or not can be taken based on local neighborhood configuration. For 26-connected object and 6-connected background,

\[ N^{26}(v) \] number of object components in a 26-neighborhood of \( v \)

\[ \bar{N}^{18}_{f}(v) \] number of background components in an 18-neighborhood of \( v \), face connected to \( v \)

\( v \) is simple if

\[ N^{26}(v) = 1, \quad \bar{N}^{18}_{f}(v) = 1 \]
\[ N^{26}(v) = 0, \bar{N}^{18}_f(v) = 1 \]

→ object component would be removed

\[ N^{26}(v) = 1, \bar{N}^{18}_f(v) = 0 \]

→ creating a cavity

\[ N^{26}(v) = 2, \bar{N}^{18}_f(v) = 1 \]

→ disconnecting a curve

\[ N^{26}(v) = 1, \bar{N}^{18}_f(v) = 2 \]

→ creating a tunnel
Classification of voxels in surfaces

- Curve: $N^{26}(v) \geq 2$, $\overline{N}^{18}_{f}(v) = 1$
- Junction: $N^{26}(v) = 1$, $\overline{N}^{18}_{f}(v) > 2$
- Inner surface: $N^{26}(v) = 1$, $\overline{N}^{18}_{f}(v) = 2$
- Surface edge (object border): $N^{26}(v) = 1$, $\overline{N}^{18}_{f}(v) = 1$

Compare with the continuous case.
Balls generated by different metrics

- $D^6$: Unit weight to face neighbors
- $D^{26}$: Unit weight to face, edge, and vertex neighbors
- $\langle 3,4,5 \rangle$: Weight 3, 4, 5 to face, edge, and vertex neighbors, respectively
- Euclidean
Centers of maximal balls

As in 2D, a voxel is a center of maximal ball
 if it is a local maximum in the DT
 for weighted distances.

*(note! take local distance into account)*

for voxel in $<a,b,c>$ WDT labeled $v$:
    face neighbors $< v+a$
    edge neighbors $< v+b$
    vertex neighbors $< v+c$

for $D^6$:
    face neighbors have lower or equal label

for $D^{26}$:
    neighbors have lower or equal label
Skeletonization in 3D

Surface skeleton is obtained by keeping CMBs and removing simple points sequentially.

- 3D object $\rightarrow$ 2D surface skeleton $\rightarrow$ 1D curve skeleton
- reversibility can only be guaranteed from surface skeleton
Surface skeleton to curve skeleton

- Classification of voxels
  - edge, inner, curve, junction
- Iterative removal of surface edge voxels
- Preserve topology and keep junction and curve voxels

junctions

curves

sharp corners
D⁶ surface and curve skeletons
Classes of points in skeleton in 3D

CMBs, $D^6$

Equidistance from one boundary component
- end point

Equidistant from two boundary components
- surface point

Equidistant from three or more boundary components
- branch point
Vessel analysis

“blood vessel” with narrowings from two views
Blood vessels & curve skeletons

D⁶ surface skeleton used & pruning applied to curve skeleton
Summary

- Grids, connectivities
- Non-Cartesian grids
- Distance transforms in 3D
- Applications using DT
  - Path-planning
  - Chamfer matching
- Skeletonization in 2D (and 3D)
  - Simple points, Centers of maximal balls (CMBs)
  - Skeletonization by
    - Centers of maximal balls (CMBs)
    - Homotopic thinning
    - Homotopic thinning keeping the CMBs
  - Template matching
  - Skeletal properties