Curve-Skeleton Applications

Nicu D. Cornea,
Deborah Silver,
*Rutgers University, New Jersey*

Patrick Min
*John Cabot University, Rome, Italy*
Types of skeletons

- **Skeleton in 2D**
  - Locus of centers of maximal inscribed disks
  - Medial axis (Blum, 1967)
  - Set of curves

- **Skeleton in 3D**
  - Surface patches + curves
  - Medial surface

- Sometimes we want a “line-like” 1D skeleton in 3D
The curve-skeleton

- “Compact” 1D representation of 3D objects
  - Call it *curve-skeleton* (Svensson et al., 2002)
    - Idea used earlier in the first thinning algorithms
- Related to the medial surface
  - Reduce surface patches to curves
- Also known as centerline
Motivation

1D representation useful in many applications
• Virtual navigation, data analysis, animation, etc.
  • Reduced dimensionality
  • Simpler algorithms

Issues:
• No formal definition
  • Defined as “I know it when I see it” - In terms of desirable properties
    – application specific
• Large number of algorithms
  • Fine-tuned to specific applications
  • Demonstrated on small set of test objects
  • Unclear how general
• Algorithm classification
  • Existing classifications cannot accommodate some algorithms
    – Some algorithms use techniques from several different classes
      » Ex: distance order thinning
Goal/Outline of presentation

- Analysis of desirable properties of curve-skeletons
  - Extracted from the literature
  - Help in defining the curve-skeleton
- Overview of applications
- Overview of algorithms
  - Classification based on implementation
- Implementation & comparison of different methodologies
  - same set of objects
- Guide for future uses of curve-skeletons
Curve-skeleton properties found in literature

- General properties
  - Centered
  - Homotopic
  - Connected
  - Invariant under isometric transformations
  - Robust
  - Thin

- Application specific
  - Reconstruction
  - Reliability
  - Junction detection and component-wise differentiation

- Properties of the skeletonization process
  - Efficient, hierarchic, handle point sets

Notations:
- $O$ – discrete 3D object
- $Sk(O)$ – curve-skeleton of object $O$
3D Object Representations

- Polygonal mesh
  - Vertices and polygons

- Volume
  - Voxels on a discrete grid

- Unorganized point sets
  - Points with no connectivity information

http://www.cs.utexas.edu/users/amenta/powercrust/unions.html
1. Centered
   - Curves centered within the object

2. Homotopic - preserve original object’s topology
   - (Kong and Rosenfeld, 1989): Same number of
     - Connected components – 6, 18 or 26-connectivity
     - Tunnels – donut hole
     - Cavities – empty space inside object
   - Cavities in a 1D curve?
   - In a strict sense, curve-skeletons cannot preserve topology
   - Relaxed definition for curve-skeleton homotopy
     - Same number of connected components
     - At least one loop around each cavity and tunnel

3. Connected
   - $\text{Sk}(O)$ should be connected if $O$ is connected.
   - Consequence of homotopy
4. Invariant under isometric transformations
   - Skeleton of transformed object = transformed skeleton of original object
   - $Sk(T(O)) = T(Sk(O))$

5. Robust
   - Weak sensitivity to noise

6. Thin
   - 1D – one voxel thick in all directions
7. Reconstruction
   • Ability to recover the original object from the curve-skeleton
     • Compression applications
     • In general not possible

8. Reliable
   • Every boundary point is visible from at least one curve-skeleton location.
   • Can be checked with a line of sight computation.
   • Ensures “reliable” inspection of a 3D object (virtual endoscopy).
   • First introduced by He et.al. in 2001.
9. Junction detection and component-wise differentiation

- Distinguish the different logical components of the object
  - Different components of the curve-skeleton
- Logical components / Mesh Decomposition
  - No precise definition
- Necessary condition: curve-skeleton junctions need to be identified
- Animation, object decomposition
Properties of curve-skeletonization process

- Efficient
  - reduced computational complexity

- Hierarchical
  - can generate a set of skeletons of different complexities
  - same algorithm used for different applications

- Operate on various object representations
  - Polygonal, voxelized, point clouds
Curve-skeleton properties

- Not all properties are essential to all applications
- Some properties may be conflicting
  - Thinness and reconstruction
  - Reliable and robust
Curve-skeleton applications

- Virtual navigation and virtual endoscopy
- Computer graphics - animation
- Medical applications
  - Segmentation, registration, quantification of anatomical structures, surgical planning, radiation treatment, curved planar reformation, stenosis detection, aneurism and vessel wall calcification detection, deforming volumes
- Analysis of scientific data
  - Vortex core extraction, Feature Tracking, Plume visualization
- Matching and retrieval, Morphing
- Mesh decomposition, Mesh repair, Surface reconstruction
- CAD, Collision detection
Virtual navigation and virtual endoscopy

- Collision-free path through a scene or inside an object
  - Virtual camera translated along the skeleton path

Medical applications:
- Colonoscopy, bronchoscopy, angioscopy
- Reliability ensures that the physician has the possibility to fully examine the interior of the organ
- Exploits the centeredness property

Hong et al., 1997
Perchet et al., 2004
Traditional computer graphics – animation

- Maya, 3D Studio Max
- Bloomenthal, 2002;
- IK (inverse-kinematics) skeleton
  - a 1D representation of the animated object
  - manipulated by the animator
  - IK skeleton transformations transferred to object polygons
  - usually created manually by the animator
- Recent attempts to automate the process

Character Studio

Wade and Parent, 2002

Katz and Tal, 2003
More medical applications
- Segmentation and quantification of anatomical structures
  - extract skeletons from tubular objects in medial images:
    - Blood vessels, nerve structures
- Surgical planning, radiation treatment

Stenosis, aneurism and vessel wall calcification detection
- Nystrom et.al., 2001; Sorantin et.al., 2002, Straka et.al., 2004

Curved planar reformation
- flattening of 3D structures
Curve-skeleton applications ...

- Even more medical applications
- Deforming volumes
  - unwinding convoluted structures for easy inspection
    - colon straightening
- Registration
  - aligning two images of the same patient taken with different imaging modalities (MRI, CT, MRA)
    - Use of curve-skeleton reduces the dimensionality of the problem
Analysis of scientific data
- Complex topologies can be easily explained using line drawings
- Vortex core extraction
- Feature tracking
- Plume visualization

Banks and Singer, 1994
Vrolijk et.al., 2003
Santilli et.al., 2004
Curve-skeleton applications …

- Matching and retrieval
  - Given a query object, find similar objects in a database
  - Curve-skeleton used as shape descriptor
  - Can allow part-matching
    - can provide registration of the part in the larger object

- Morphing
  - Smoothly transform one object into another
  - Curve-skeleton used to control the transition process
    - Correspondences between object parts are specified on the skeletons

Fig. 3. (a) A triangle and (b) a square are (c) combined by taking their symmetrical difference. (d) The resulting object is skeletonized and (e) trimmed to yield (f) the intermediate shape.

Blanding et.al., 2000

Lazarus and Verrroust, 1998
Zhao et.al., 2003
Curve-skeleton applications …

- Mesh decomposition
  - Decompose a polygonal mesh into meaningful components
  - Curve-skeleton drives the decomposition process
  - Inverse approach
    - Curve skeleton extracted from mesh decomposition results
      - Katz and Tal, 2003

- Mesh repair
  - Leymarie, 2003

- Surface reconstruction
  - Verroust et.al., 2000; Amenta et.al., 2001
Curve-skeleton applications …

- CAD
  - dimensional reduction of various engineering problems
    • Suresh, 2003
- Collision detection
  - Improve efficiency of the process
- General data structure for graphical objects

Figure 1. (left) Visual feedback model (right) Wireframe showing real-time tissue deformation.

Pizer et.al., 1999

Gagvani and Silver, 2000

Webster et.al., 2005
Curve-skeletonization algorithms

- General algorithms which use only the 3D shape
- Previous classifications based on theory
  - Some algorithms do not fall clearly in one of the categories
- Classification based on underlying implementation
  1. Pure Thinning and boundary propagation
  2. Using a distance field
  3. Geometric methods
  4. Using general-field functions
- Implemented the “core” part of each of these classes
  - Code and test objects available at:
    - http://www.caip.rutgers.edu/~cornea/CurveSkelApp
Curve-skeletonization algorithms ...

1. Pure thinning

- Iteratively remove simple points from the boundary
  - Simple point = a point that can be removed without changing topology.
  - Stops when no more simple points exist

- Removal conditions implemented as templates (3x3x3 or larger)

Palagyi and Kuba, 1999
25

Curve-skeletonization algorithms ...  
1. Pure thinning

- Special simple points are kept to preserve object geometry
  - Surface and curve endpoints

- Several flavors
  - Directional
    - voxels removed in one direction at a time
  - Fully parallel
    - all simple points removed at once
  - Non-directional
    - one voxel removed at one step

- Two approaches
  - Get surface skeleton then continue to thin to a curve-skeleton
  - Get curve-skeleton directly

**FIG. 11.** Thinning of a vasculature extracted from a (mouse) 3D MR brain study (top); the result of the proposed curve thinning algorithm D11-PKS (bottom left); the result of the proposed surface thinning algorithm D11-PKS (bottom right). These pictures are displayed by using the 3DVIEW/NEX software system [27].  
Palagyi and Kuba, 1999
Curve-skeletonization algorithms...

2. Using a distance field

- Distance transform – distance to closest boundary voxel

\[ D(P) = \min_{P \in O} (d(P, Q)) \]

- Ridges of the distance function (incl. local max, saddles)
  - Locally centered voxels

Figure 1. The 2D Euclidean distance map for a voxelized animal shape. Each cell's value is the square of the Euclidean distance to the nearest exterior cell.

Wade and Parent, 2002
General structure of a distance field-based algorithm

- Find voxel candidates (ridges)
  - Distance ordered thinning
  - Gradient searching
  - Divergence
  - Geodesic front propagation
  - Threshold bisector angle
- Prune
  - Cluster
- Connect
  - MST, shortest path
  - Some algorithms maintain connectivity while pruning

Figure 8: MRI dataset, 512x512x281 of the trachea. (a) The segmented trachea (159k voxels). (b) Skeleton with a thinness = 2.5. (c) The minimum spanning tree of the skeletal voxels.

Gagvani and Silver, 1999
3. Geometric methods

- Usually apply to objects represented by polygonal meshes
  - continuous space

- Medial surface approaches
  - Voronoi diagram based (Amenta, et.al.)
    - generator elements – boundary elements (points, polygons)
  - Cores and m-reps (Pizer)
    - position, radius, orientations, object angle
  - Shock scaffold (Leymarie)
    - shock curves

- Non-medial surface approaches
  - Level sets of geodesic graph (Verroust et.al. 2000)
  - Edge contraction (Li et.al. 2001)
  - Using the mesh decomposition results (Katz and Tal, 2003)
Curve-skeletonization algorithms ...
4. Using general-field functions

- Generalized potential field function
  - function is a sum of potentials generated by boundary elements

- Visible repulsive force function
  - Newtonian potential function using visibility

- Electrostatic field function
  - electrostatically charged boundary

- Radial basis function
  - boundary samples source of radial basis functions

- Averaging
  - Less sensitive to noise

- Detect sinks in the resulting field and connect them
  - Force-following, active contours

Cornea et. al., 2005

Chuang et.al., 2000
Experimental results
Implementation – “core” of each class

- Implemented the “core” part of each algorithm class
  - Core = First step of each class of algorithms
    - no post-processing
    - used to classify the algorithms

- Thinning
  - Curve-thinning templates from Palagyi and Kuba, 1999

- Distance field
  - Distance function filtering by Gagvani and Silver, 1999

- Geometric
  - Power Crust, Amenta et.al., 2001

- General field
  - Core skeleton using generalized potential field, Chuang et.al., 2000.
<table>
<thead>
<tr>
<th>Test Object</th>
<th>Distance Field</th>
<th>Thinning</th>
<th>Geometric</th>
<th>Potential Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thin block</td>
<td><img src="image1" alt="Distance Field" /></td>
<td><img src="image2" alt="Thinning" /></td>
<td><img src="image3" alt="Geometric" /></td>
<td><img src="image4" alt="Potential Field" /></td>
</tr>
<tr>
<td>(204x132x260)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monster</td>
<td><img src="image5" alt="Distance Field" /></td>
<td><img src="image6" alt="Thinning" /></td>
<td><img src="image7" alt="Geometric" /></td>
<td><img src="image8" alt="Potential Field" /></td>
</tr>
<tr>
<td>(54x87x75)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

continued on next slide …
Experimental results ...

<table>
<thead>
<tr>
<th>Test Object</th>
<th>Distance Field</th>
<th>Thinning</th>
<th>Geometric</th>
<th>Potential Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mushroom (80x87x59)</td>
<td><img src="image1" alt="Distance Field" /></td>
<td><img src="image2" alt="Thinning" /></td>
<td><img src="image3" alt="Geometric" /></td>
<td><img src="image4" alt="Potential Field" /></td>
</tr>
<tr>
<td>Colon (204x132x260)</td>
<td><img src="image5" alt="Distance Field" /></td>
<td><img src="image6" alt="Thinning" /></td>
<td><img src="image7" alt="Geometric" /></td>
<td><img src="image8" alt="Potential Field" /></td>
</tr>
<tr>
<td>Twist (100x87x58)</td>
<td><img src="image9" alt="Distance Field" /></td>
<td><img src="image10" alt="Thinning" /></td>
<td><img src="image11" alt="Geometric" /></td>
<td><img src="image12" alt="Potential Field" /></td>
</tr>
</tbody>
</table>
**Experimental results …**

<table>
<thead>
<tr>
<th>Test Object</th>
<th>Distance Field</th>
<th>Thinning</th>
<th>Geometric</th>
<th>Potential Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chess piece (40x39x87)</td>
<td><img src="image1" alt="Distance Field" /></td>
<td><img src="image2" alt="Thinning" /></td>
<td><img src="image3" alt="Geometric" /></td>
<td><img src="image4" alt="Potential Field" /></td>
</tr>
<tr>
<td>Chess piece with 10% noise on the surface (39x38x86)</td>
<td><img src="image5" alt="Distance Field" /></td>
<td><img src="image6" alt="Thinning" /></td>
<td><img src="image7" alt="Geometric" /></td>
<td><img src="image8" alt="Potential Field" /></td>
</tr>
</tbody>
</table>

Experimental results ...  
Running time vs. Nr. object voxels
Discussion of comparison results

- No single method is good for everything
- Distance field and geometric methods do not produce a curve-skeleton directly
  - Need additional pruning and connectivity steps
  - Sensitive to noise
- Thinning and potential field directly produce curve-skeletons
  - Thinning
    - Fast but more sensitive to noise and not very smooth
  - Potential field is too slow
- Not a totally fair comparison of different methodologies!
  - Only implemented the “core” of each methodology
  - Additional post-processing steps change the results significantly
- Goal: To show which methodology takes us closer to a curve-skeleton in the first step
  - Gives an idea about how much post-processing needed to get a curve-skeleton
### Discussion of the various methodologies

<table>
<thead>
<tr>
<th>Methodology</th>
<th>Thinning</th>
<th>Distance field</th>
<th>Geometric</th>
<th>General field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centered</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Homotopy</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Connected</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Transf. Invariance</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Robust</td>
<td>✗</td>
<td>-</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Thin</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>Reconstruction*</td>
<td>✗</td>
<td>-</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Reliable</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Junction detection</td>
<td>-</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Efficiency</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>✗</td>
</tr>
<tr>
<td>Hierarchic</td>
<td>✗</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Handle Point Sets</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- ✓ yes
- ✗ no
- - possible
Conclusions

- Compiled a list of desirable properties of curve-skeletons
- Reviewed some applications
- Classification of algorithms
  - based on implementation
- Comparison of methodologies
- Guide for future use of curve-skeletons
  - Think about the required properties
  - Then choose the appropriate methodology
Future work

- Develop algorithms to check properties

- Challenge
  - Test future algorithms on large databases of general objects (for example, The Princeton Shape Benchmark Database: http://shape.cs.princeton.edu/benchmark/).
Acknowledgements

- This work is supported in part by NSF 0118760 and NSF EIA-0205178.

- We would also like to thank Dr. Raman Balasubramanian and Xiaosong Yuan for their help.