Curve-Skeleton Applications

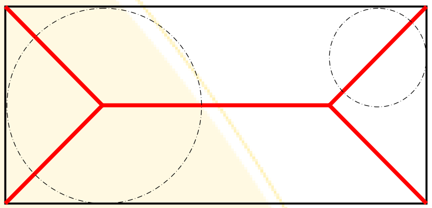
Nicu D. Cornea,
Deborah Silver,
Rutgers University, New Jersey

Patrick Min
John Cabot University, Rome, Italy

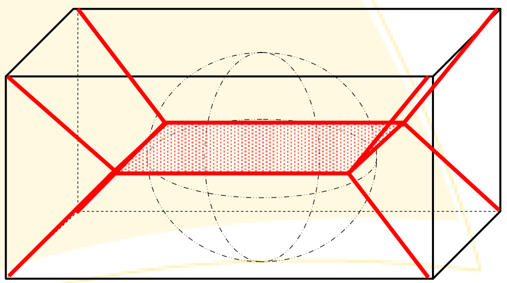
VIS 05
MINNEAPOLIS, MN USA

Types of skeletons

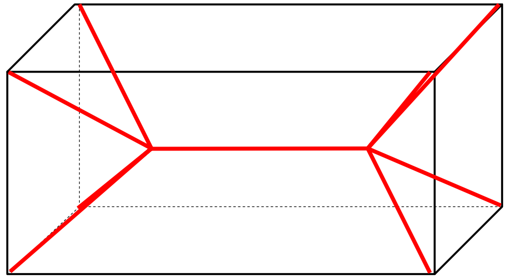
- Skeleton in 2D
  - Locus of centers of maximal inscribed disks
  - Medial axis (Blum, 1967)
  - Set of curves



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

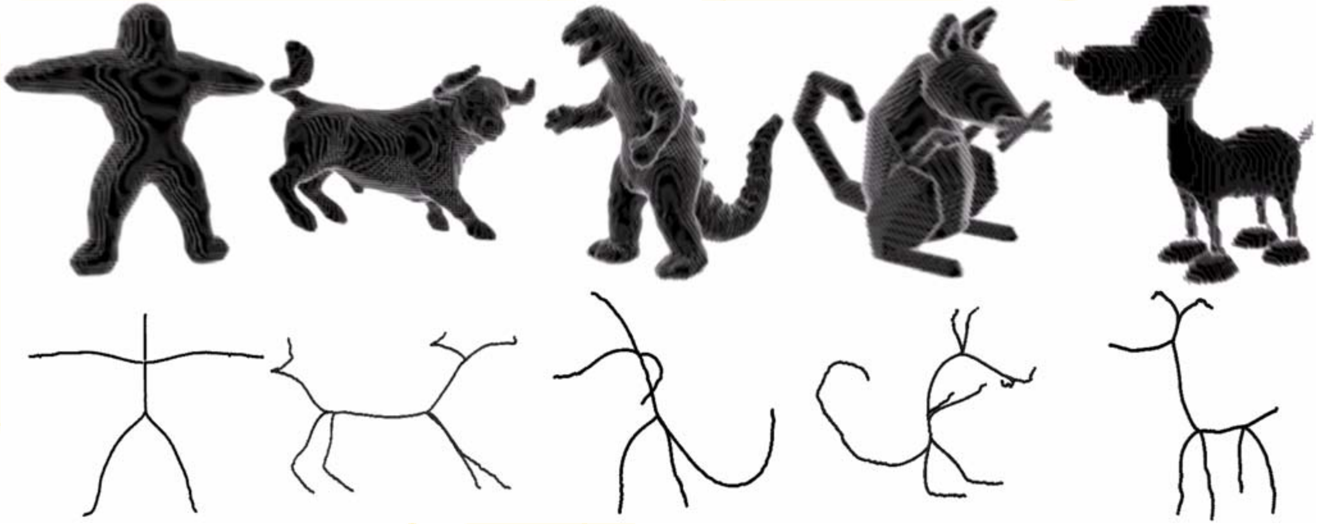


- Sometimes we want a “line-like” 1D skeleton in 3D



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

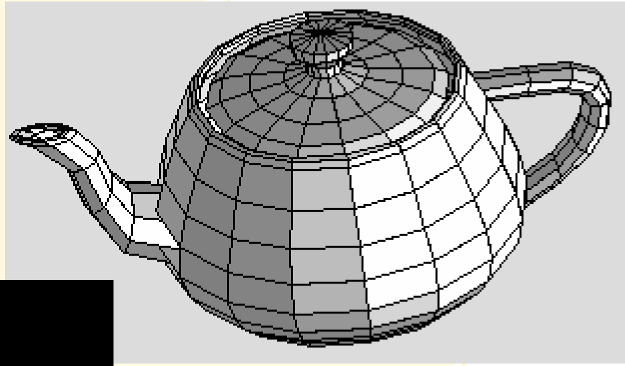
Curve-skeleton properties found in literature

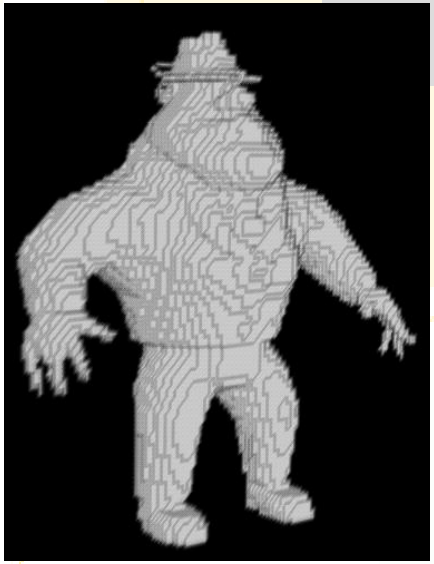
- General properties
  - Centered
  - Homotopic
  - Connected
  - Invariant under isometric transformations
  - Robust
  - Thin
- Application specific
  - Reconstruction
  - Reliability
  - Junction detection and component-wise differentiation
- Properties of the skeletonization process
  - Efficient, 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

Curve-skeleton properties (general) …

- 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
  - *Sk(O)* should be connected if *O* is connected.
  - Consequence of homotopy

Curve-skeleton properties (general) …

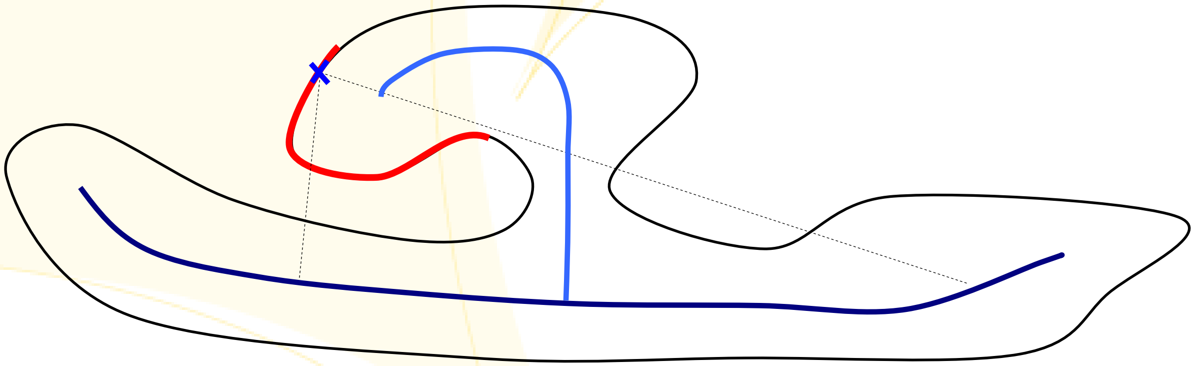
- 4. Invariant under isometric transformations
  - Skeleton of transformed object = transformed skeleton of original object
  - *Sk(T(O)) = T(Sk(O))*

- 5. Robust
  - Weak sensitivity to noise

- 6. Thin
  - 1D – one voxel thick in all directions

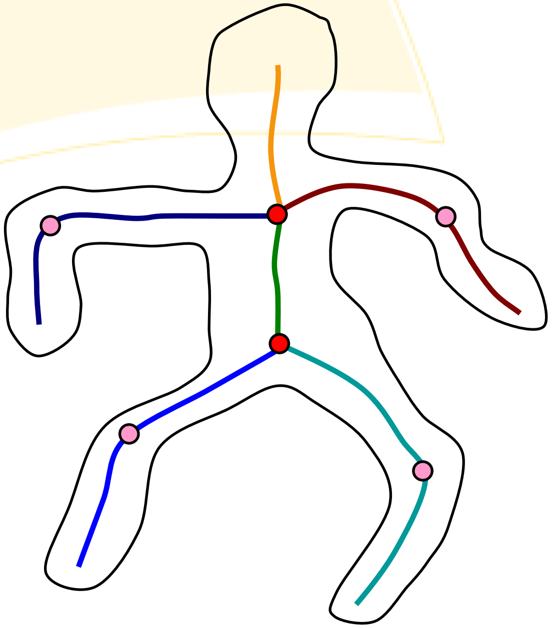
Curve-skeleton properties (app. specific) …

- 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.



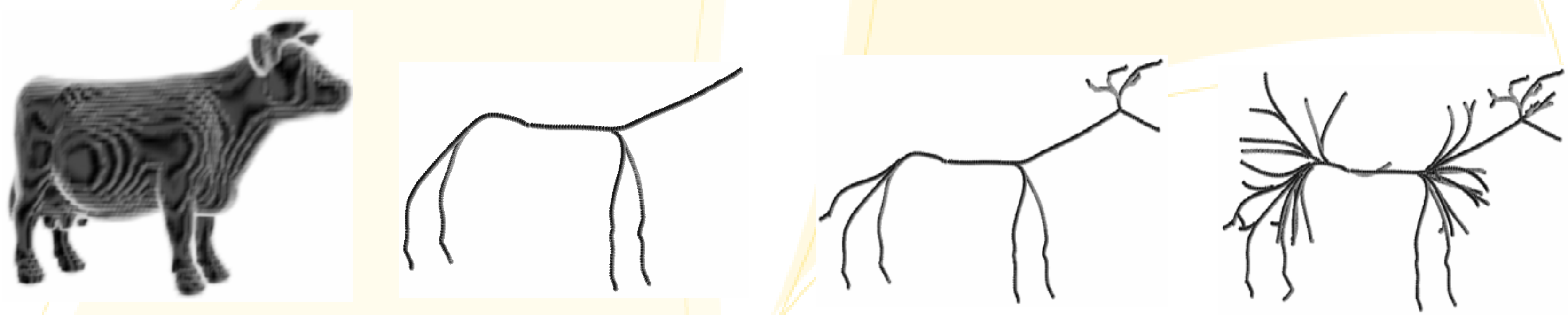
Curve-skeleton properties (app. specific) …

- 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
      - Tal and Katz, 2003; Katz and Pizer, 2003
  - 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

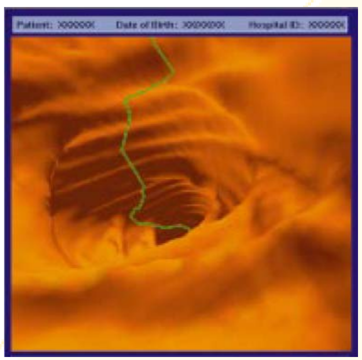
- Not all properties are essential to all applications
- Some properties may be conflicting
  - Thinness and reconstruction
  - Reliable and robust

Curve-skeleton applications

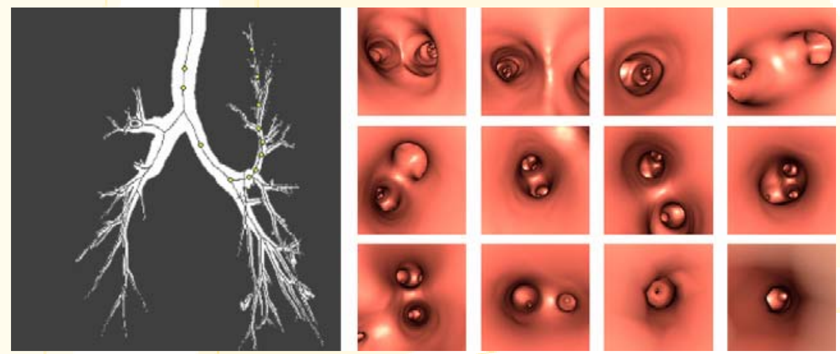
- Virtual navigation and virtual endoscopy
- Computer graphics - animation
- Medical applications
  - Segmentation, registration, quantification of anatomical structures, surgical planning, radiation treatment, curved planar reformation, stenosis detection, aneurism and vessel wall calcification detection, 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

Curve-skeleton applications …

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



Hong et.al., 1997

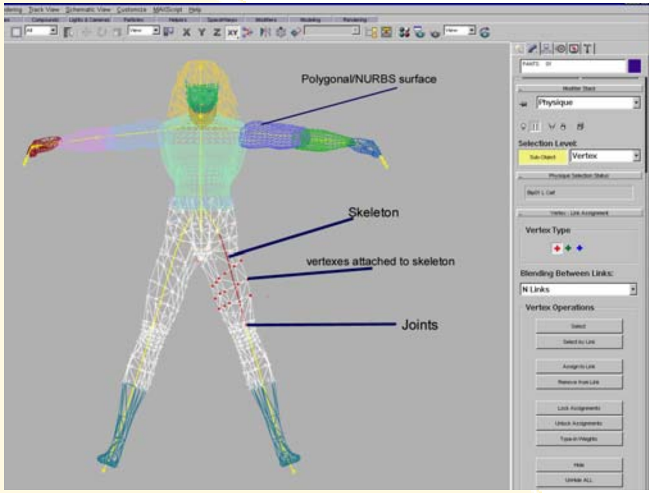


Perchet et.al., 2004

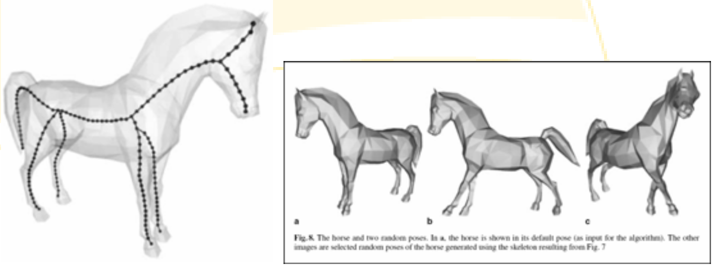
  - Medical applications:
    - Colonoscopy, bronchoscopy, angioscopy
    - Reliability ensures that the physician has the possibility to fully examine the interior of the organ
  - Exploits the centeredness property

Curve-skeleton applications …

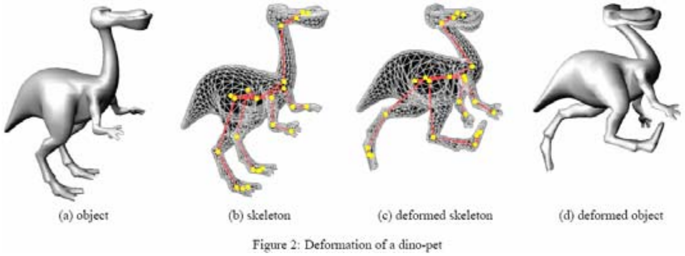
- Traditional computer graphics – animation
  - Maya, 3D Studio Max
  - Bloomenthal, 2002;
  - IK (inverse-kinematics) skeleton
    - a 1D representation of the animated object
    - manipulated by the animator
    - IK skeleton transformations transferred to object polygons
    - usually created manually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



Character Studio



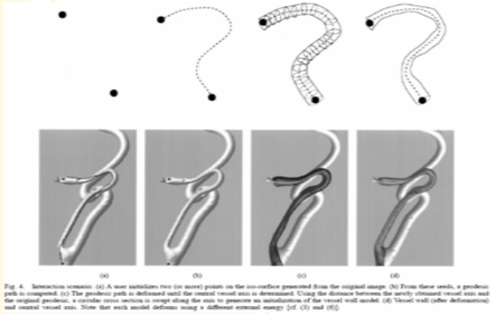
Wade and Parent, 2002



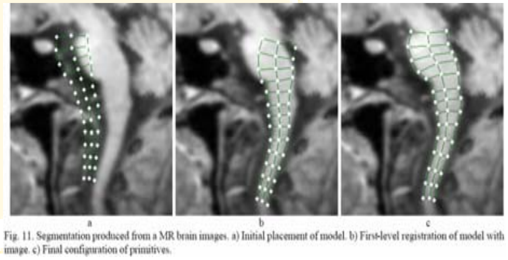
Katz and Tal, 2003

Curve-skeleton applications …

- 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



Frangi et.al., 1999



Pizer et.al., 1999

Sorantin et.al., 2002

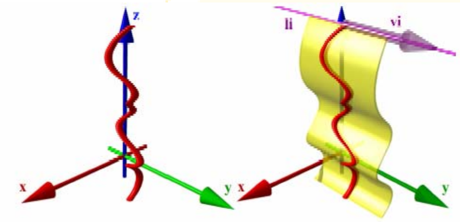


Figure 1: Principle of the CPR visualization: The vector-of-interest (*vi*) and the line-of-interest (*li*) defining the re-sampled plane.

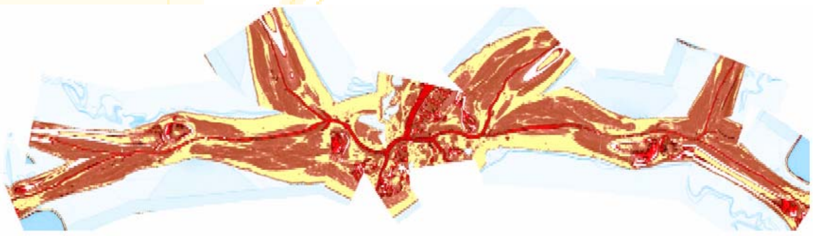


Figure 1: An untangled vascular tree of the peripheral arteries.

Kanistar et.al., 2002, 2003

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

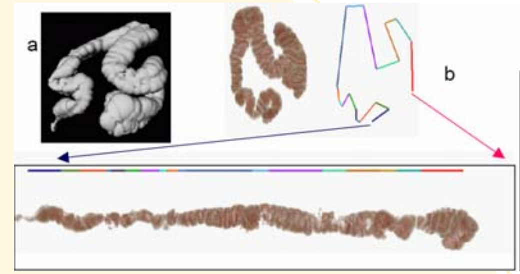


Figure 1: Unwinding the colon. The original colon dataset (a) is 212x150x160. A skeleton was computed (b) and color coded to show the segments of the colon which correspond to the joints in the skeleton. A stretched out colon is shown in (c) and (d). The resulting 3D dataset results is 991x90x94 voxels.

Silver and Gagvani, 2002

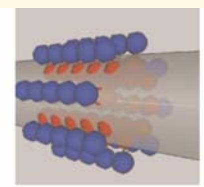


Figure 1. A 3D rendering of the set of kernels distributed along the centerline about $\mathbf{x}_i$ and used to estimate the radius at $\mathbf{x}_i$. Each kernel is a radial sample of a center-on/surround-off binary filter oriented normal to the centerline. The local radius producing a maximum response from the weighted sum of these kernels defines the radius r_i at $\mathbf{x}_i$.

Figure 10. Results of alignment of AVM pre-treatment data (used for registration and shown in dark-gray wireframe) with post-radiation (after gamma-knife treatment of the AVM—models not used in registration, but shown as solid light-gray surfaces). The registration parameters are given in Table 3. There is good correspondence between vessels on the patient's left (image left), but on the right, near the AVM, multiple vessels are misaligned (arrows) and others are missing due to non-rigid deformations that occurred as a result of treatment. Visualizations of such vascular changes helps physicians determine treatment effectiveness.

Figure 8. Registration of the vessel models from a portal CT scan with a hepatic CT scan allows the models of the portal and hepatic vessel to be viewed together. Left: Vessel models from the two scans are shown—vessels from portal scan are shown in light gray; vessels from the hepatic scan are dark gray. Right: To illustrate the limited vascular network correspondence and the non-rigid deformations present, the few vessel models that co-exist in both scans are shown (less than one-third of the total number of model points correspond between models); the data contains non-rigid deformations (arrows), the vascular networks have minimal overlap, the vascular architecture changes, and yet the registration process was effective as indicated by the close correspondence of the surfaces of the models of the common vessels. The difference in registration between including the deformed portal vein and not including it was less than 0.00012 radians and 0.00025 voxels. The identification of the common vessel models shown on the right is merely illustrative—it is not part of the registration process.

Aylward et.al., 2003

Curve-skeleton applications …

- Analysis of scientific data
  - Complex topologies can be easily explained using line drawings
  - Vortex core extraction
  - Feature tracking
  - Plume visualization

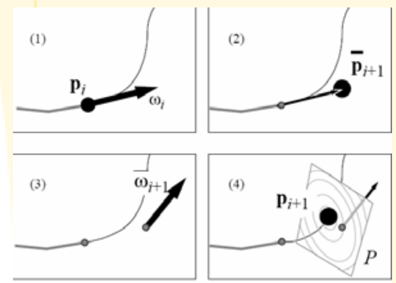


Figure 2. Schematic of predictor-corrector algorithm.

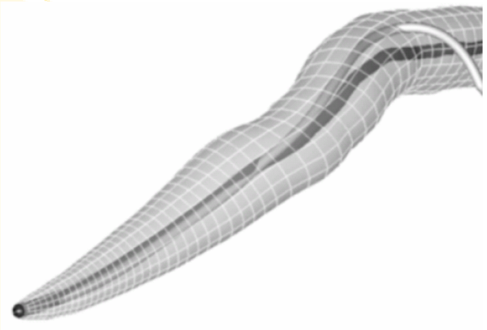
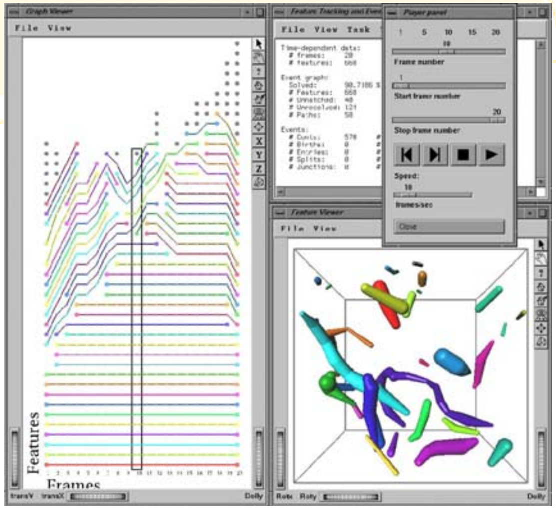
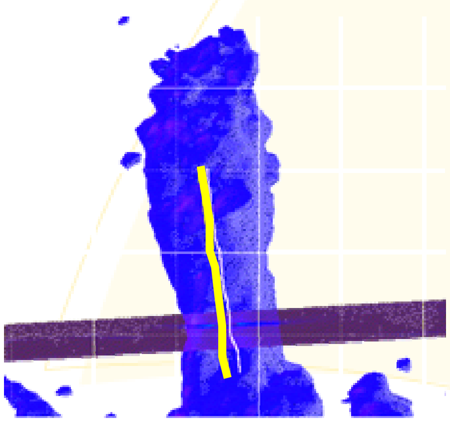


Figure 3. Vorticity line (light) compared to predictor-corrector line (dark). Note that the vorticity line exits from the vortex tube while the predictor-corrector skeleton line follows the core.

Banks and Singer, 1994



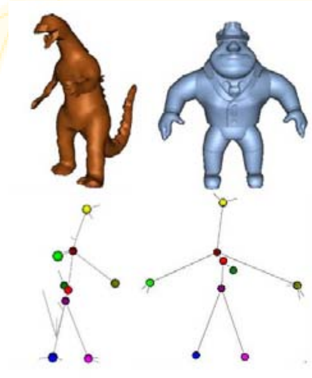
Vrolijk et.al., 2003



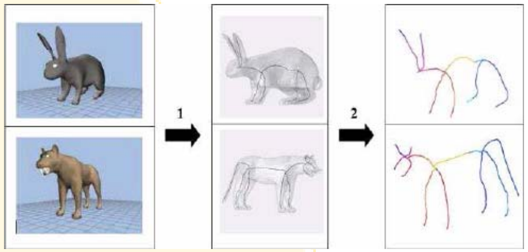
Santilli et.al., 2004

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



Sundar et.al., 2003



Cornea et.al., 2005

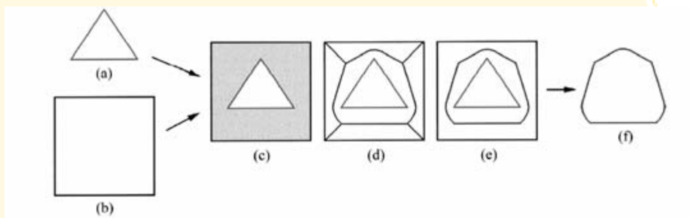


Fig. 3. (a) A triangle and (b) a square are (c) combined by taking their symmetric difference. (d) The resulting object is skeletonized and (e) trimmed to yield (f) the intermediate shape.

Blanding et.al., 2000

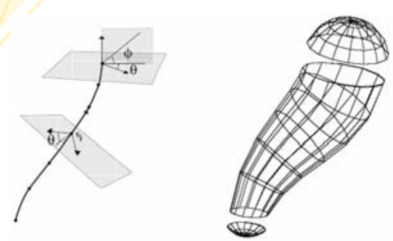


Figure 1. The three sheets

Lazarus and Verrroust, 1998

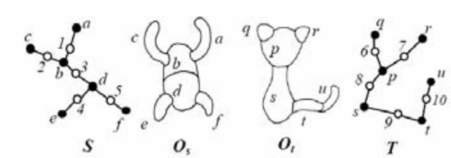


Figure 2: *The source object O_s, the target object O_t, and their respective connectivity graphs S and T. In this figure and subsequent figures, the connections of connectivity graphs are shown explicitly as white nodes for ease of illustration.*

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

re 8: Segmentation results for different models. From left to right the initial mesh, the skeleton graph and the segmented mesh are shown.

Brunner and Brunnet, 2004

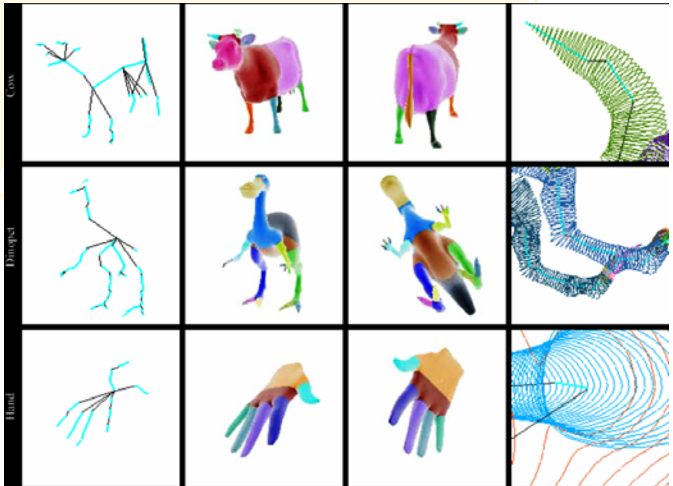


Figure 13: Decomposition Results

Li 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

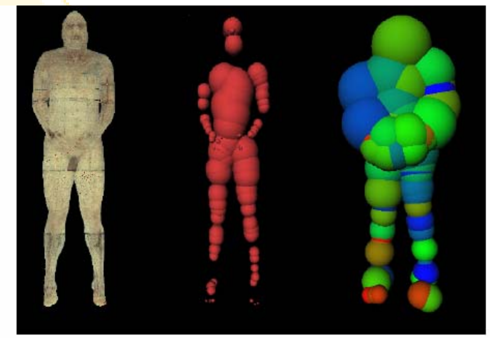


Figure 2: The reconstruction-spheres (center) and the collision-spheres (right) for the Visible Male volume. Only 100 spheres approximate the volume in this case.

Gagvani and Silver, 2000

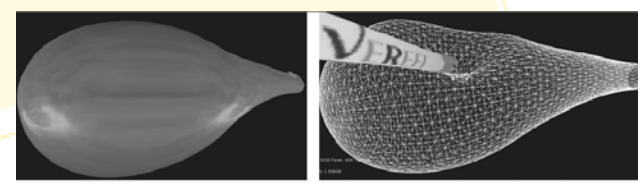


Figure 1. (left) Gallbladder model (right) Wireframe showing real-time tool-tissue deformation.

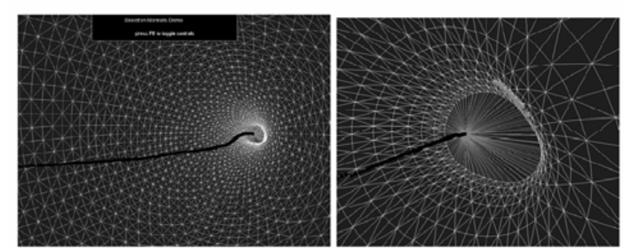
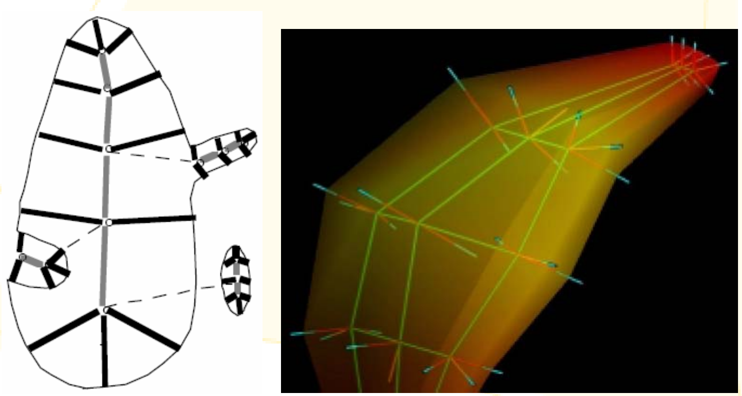


Figure 2. (left) Skeleton points (right) Wireframe showing skeleton points with connector springs attached tc all vertices associated with the skeleton point.

Webster et.al., 2005



Pizer et.al., 1999

Curve-skeletonization algorithms

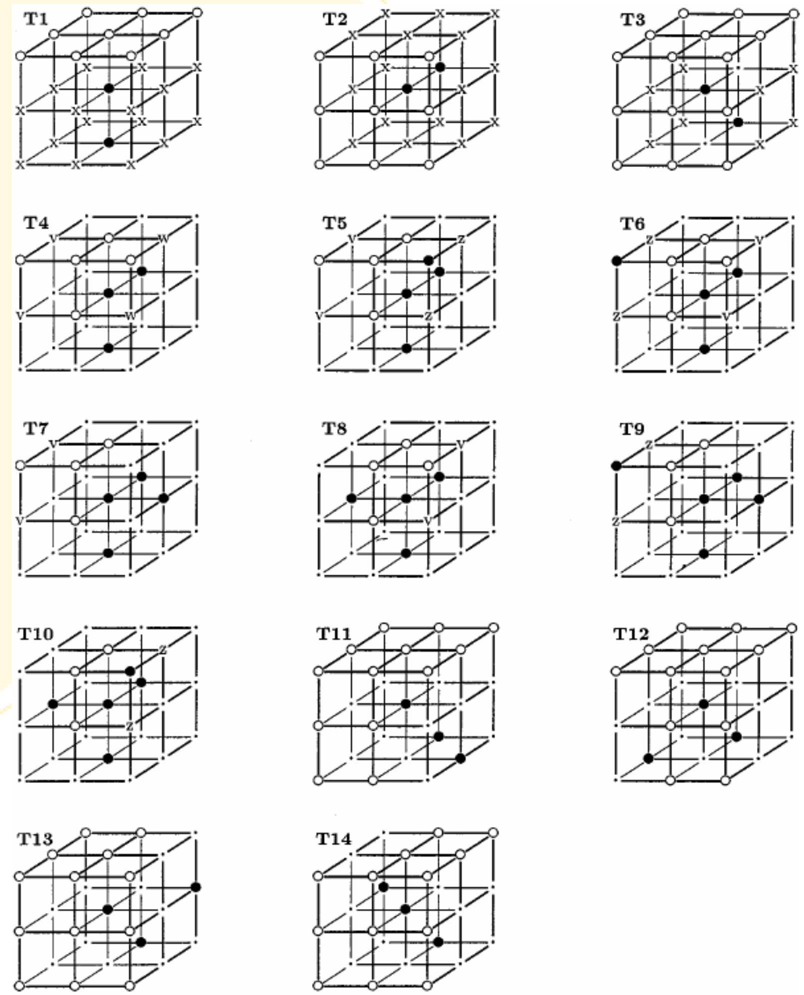
- General algorithms which use only the 3D shape
- Previous classifications based on theory
  - Some algorithms do not fall clearly in one of the categories
- Classification based on underlying implementation
  1. Pure Thinning and boundary propagation
  2. Using a distance field
  3. Geometric methods
  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

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

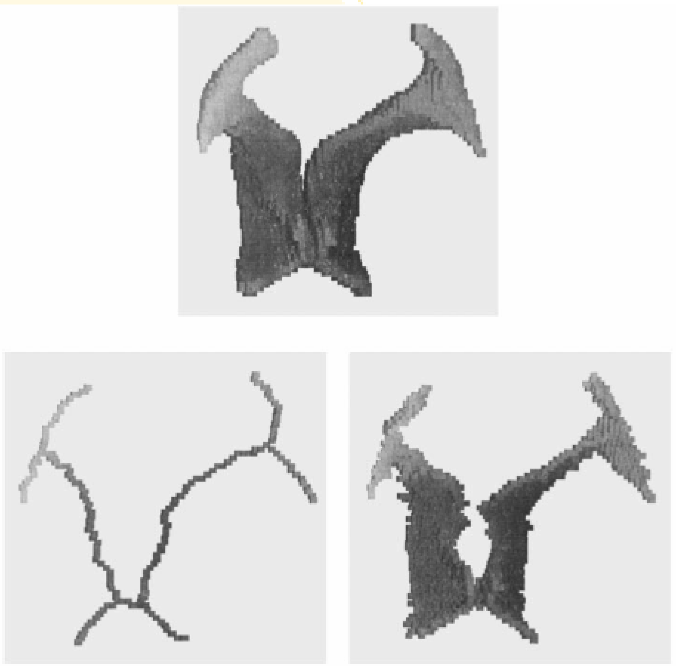


FIG. 11. Thinning of a ventricle extracted from a (greyscale) 3D MR brain study (top); the result of the proposed curve thinning algorithm D12-PK-C (bottom left); the result of the proposed surface thinning algorithm D12-PK-S (bottom right). These pictures are displayed by using the 3DVIEWNIX software system [27].

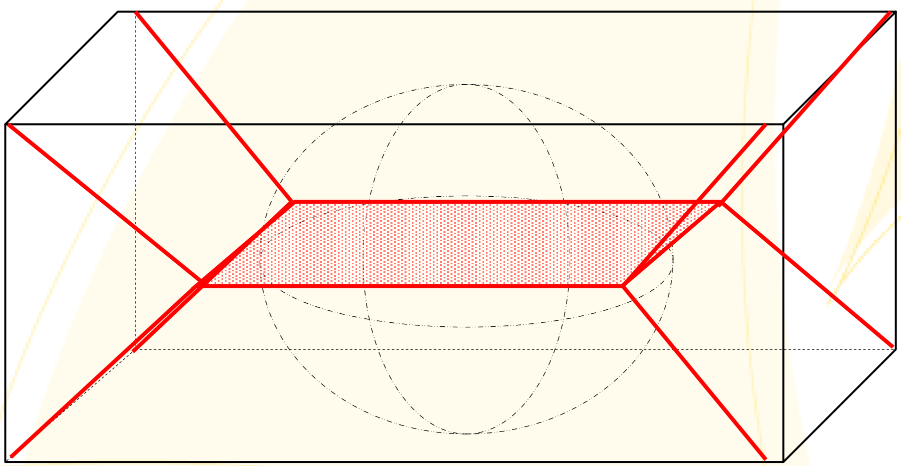
Palagyi and Kuba, 1999

Curve-skeletonization algorithms ...
2. Using a distance field

- Distance transform – distance to closest boundary voxel

$$D(P)_{P \in O} = \min_{Q \in B(O)} (d(P,Q))$$

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



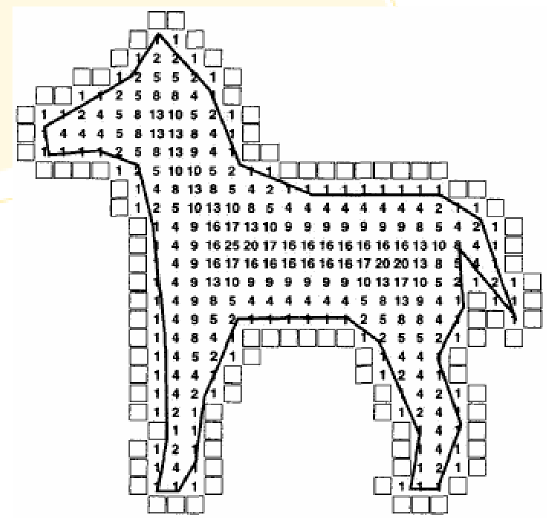


Figure 1. The 2D Euclidean distance map for a voxelized animal shape. Each cell's value is the square of the Euclidean distance to the nearest exterior cell.

Wade and Parent, 2002

Curve-skeletonization algorithms ...
2. Using a distance field

- 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

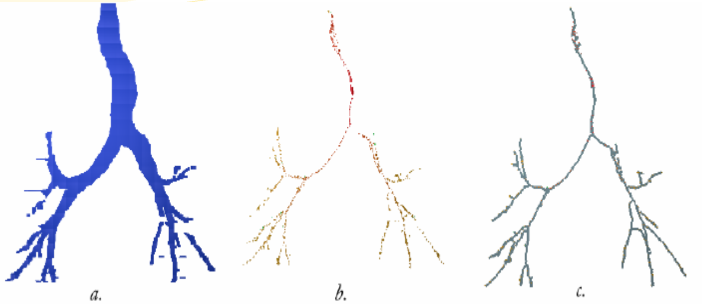


Figure 8: MRI dataset, 512x512x281 of the trachea. (a) The segmented trachea(159k voxels). (b) Skeleton with a thinness = 2.5. (c) The minimum spanning tree of the skeletal voxels.

Gagvani and Silver, 1999

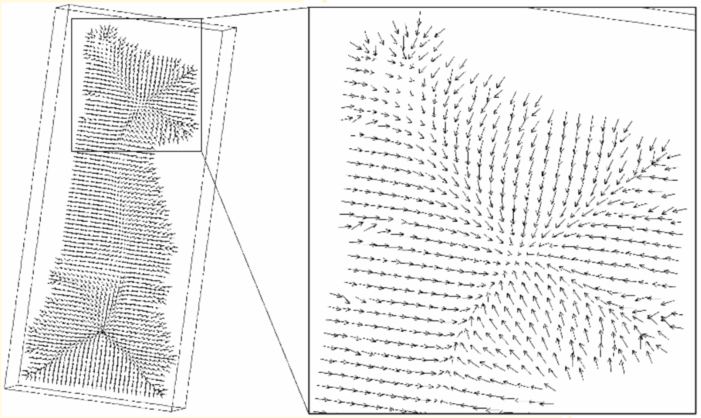
Curve-skeletonization algorithms ...
3. Geometric methods

- 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)

Amenta et.al., 2001

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

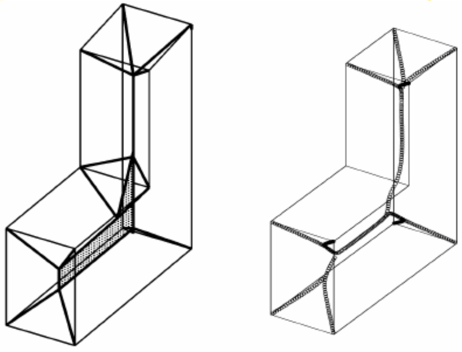


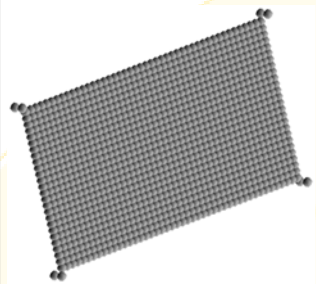
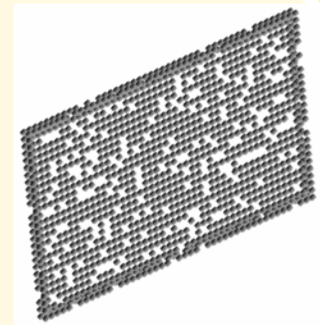
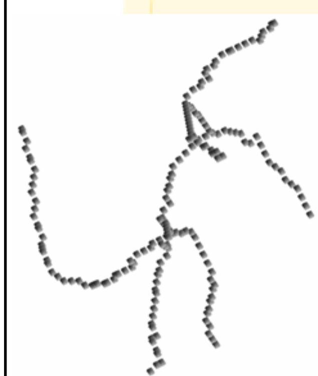
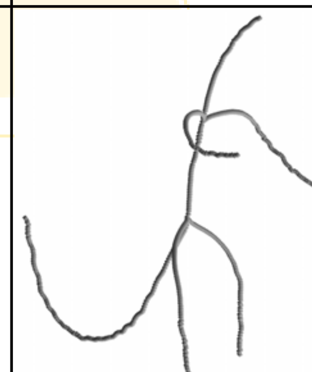
Fig. 12. (a) An L-shaped polyhedron and its MA representation. (b) The skeleton derived by the potential-based method.

Chuang et.al., 2000

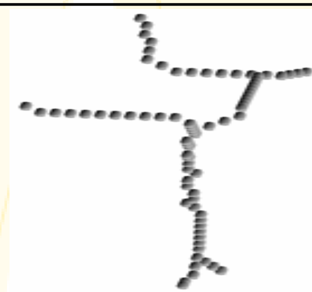
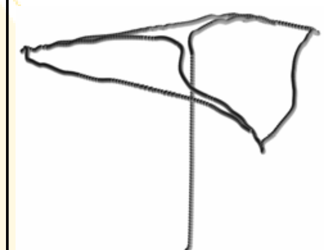
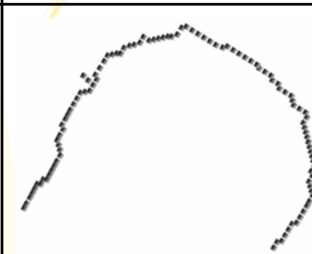
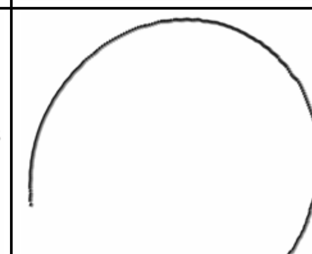
Experimental results
Implementation – “core” of each class

- 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.

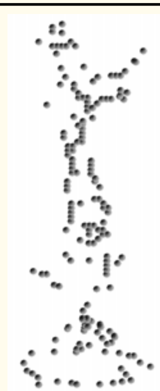
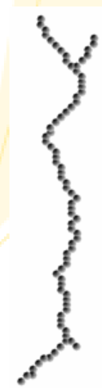
Experimental results …

	Test Object	Distance Field	Thinning	Geometric	Potential Field
Thin block (204x132x260)					
Monster (54x87x75)					
continued on next slide …					

Experimental results …

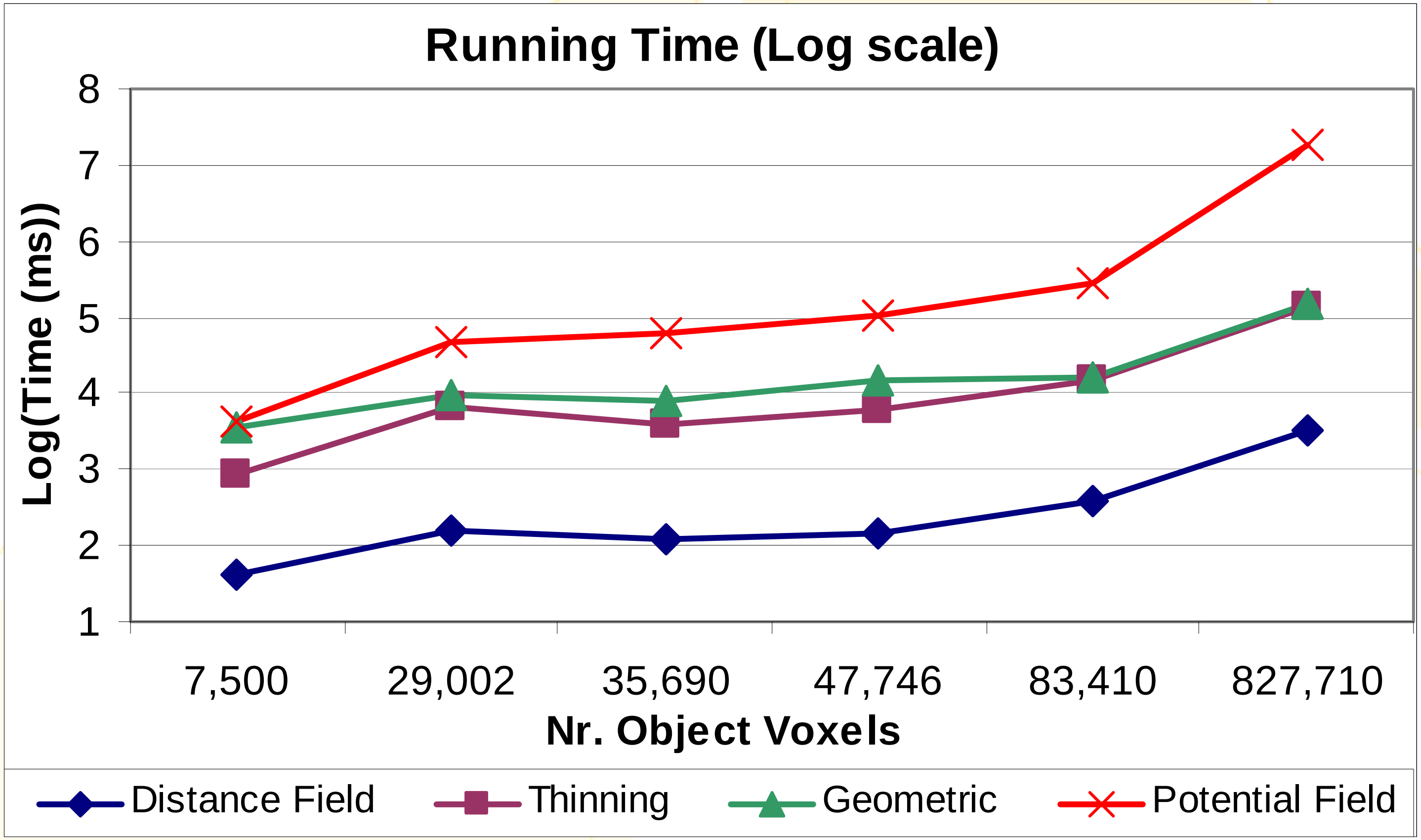
	Test Object	Distance Field	Thinning	Geometric	Potential Field
Mushroom (80x87x59)					
Colon (204x132x260)					
Twist (100x87x58)					

Experimental results …

	Test Object	Distance Field	Thinning	Geometric	Potential Field
Chess piece (40x39x87)					
Chess piece with 10% noise on the surface (39x38x86)					

Objects and code available at: http://www.caip.rutgers.edu/~cornea/CurveSkelApp/index.html#Downloads

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

	Thinning	Distance field	Geometric	General field
Centered	–	✓	✓	–
Homotopy	✓	–	✓	–
Connected	✓	–	✓	–
Transf. Invariance	–	✓	✓	✓
Robust	✗	–	✗	✓
Thin	✓	–	–	✓
Reconstruction*	✗	–	✗	✗
Reliable	–	–	–	–
Junction detection	–	✗	✗	✓
Efficiency	✓	✓	–	✗
Hierarchic	✗	–	✓	✓
Handle Point Sets	✗	✓	✓	✓

✓ yes   – possible   ✗ no

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