Mesh analysis

Effective techniques for representing, analyzing, searching, and reusing

1

Why now?

Large repositories of 3D data more accessible

- Data storage
- Computing power
- Modeling techniques

2

Why "Shape Extraction"

Examining human image understanding many works indicate that recognition and shape understanding are based on structural decomposition of the shape into smaller parts

HOFFMAN D., RICHARDS W.: Parts of recognition. Cognition 18, *1-3 (December 1984), 65–96.*

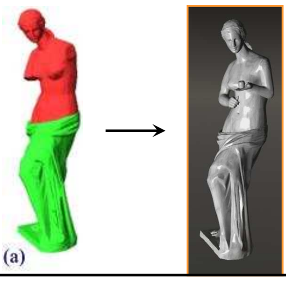
BIEDERMAN I.: Recognition-by-Components: A theory of human image understanding. Psychological Review 94 *(1987), 115–147.*

HOFFMAN D., SIGNH M.: Salience of visual parts. Cognition 63 *(1997), 29–78.*

3

For instance

Modeling by example (Siggraph 2004)



4

Sub-problems

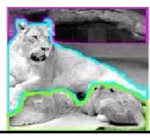
- ❑ Shape-based search
- ❑ Alignment
- ❑ Segmentation
- ❑ "Stitching"

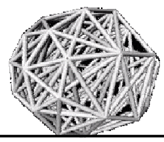
5

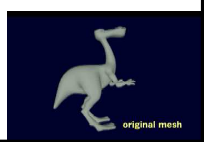
Segmentation

Applies in different domains:

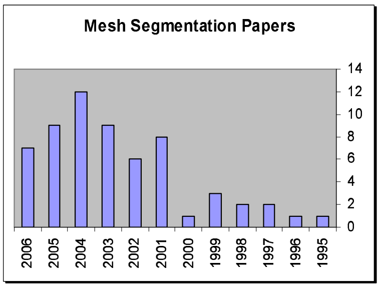
- ❑ Images – "segmentation"
- ❑ Polyhedra – "triangulation" or "convex pieces"
- ❑ Meshes – "decomposition" or "segmentation"







Mesh segmentation



7

Today’s class

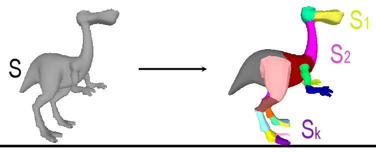
1. Definitions
2. Criteria
3. Applications
4. Algorithms

8

K-way segmrntation

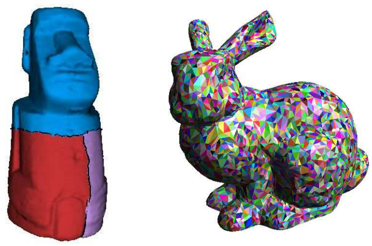
Let S be an orientable mesh.

Goal: decompose S into
connected sub-meshes S_1, S_2... S_k
that are face-wise disjoint, and
whose union gives S.



9

Correct or Not?
Good or Bad?



10

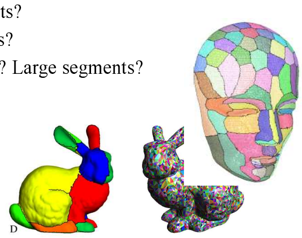
Criteria?

- ❑ Planar segments?
- ❑ Smooth segments?
- ❑ Round segments?

11

Criteria?

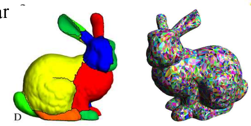
- ❑ Planar segments?
- ❑ Smooth segments?
- ❑ Round segments?
- ❑ Small segments? Large segments?



12

Criteria?

- Planar segments?
- Smooth segments?
- Round segments?
- Small segments? Large segments?
- Small number of segments?
- Smooth boundar



13

Criteria?

- Planar segments?
- Smooth segments?
- Round segments?
- Small segments? Large segments?
- Small number of segments?
- Smooth boundary?
- "Natural" segments?
- More…

14

How to choose criteria?

- What you want / need
- Application in mind

15

Segmentation types

- Geometry-based
- "Meaningful" components

16

Segmentation as optimization definition

Given a mesh M = {V,E,F} and the set of elements $S \in \{V,E,F\}$, find a disjoint partitioning of S into $S_1,\ldots,S_k$ such that the criterion function

$J = J(S_1,\ldots,S_k)$

Be minimized under a set of constraints C.

17

Optimal solution?

If $|S| = n$ and $|\Sigma| = k$, then the search space is of order k^n.

Segmentation must revert to some approximation algorithm:

- Region growing (local greedy)
- Hierarchical clustering (global greedy)
- K-means (iterative)
- Graph Cut
- Spectral Analysis

18

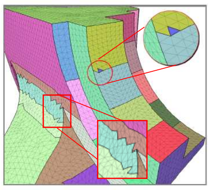
Constraints vs. Attributes

- Constraints:
  - Imposed on the segments, must be preserved
- Elements attributes:
  - Used for the criteria measure in the optimization process

19

Types of constraints

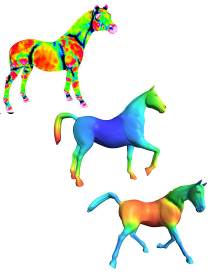
- Cardinality
  - Not too small and not too large or a given number
  - Overall balanced partition
- Geometry
  - Size: area, diameter, radius
  - Convexity, Roundness
  - Boundary smoothness
- Topology
  - Connectivity (single component)
  - Disk topology



20

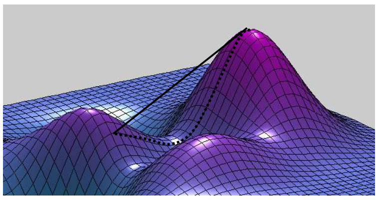
Types of attributes used

- (Geodesic) Distance
- Planarity
- Smoothness, curvature
- Slippage
- Symmetry
- Medial Axis



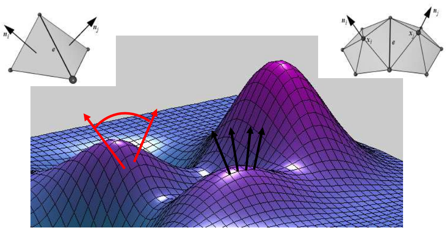
21

Euclidean distances vs. Geodesic distances



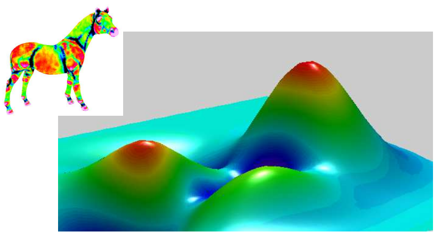
22

Normal direction, Dihedral angles



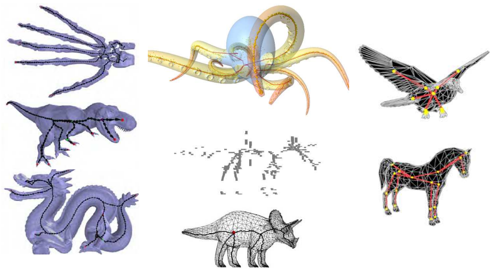
23

Smoothness, Curvature



24

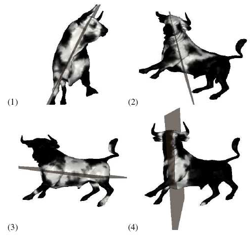
Various skeletons

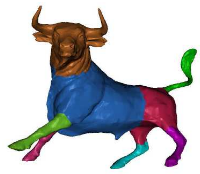


25

Symmetry

(Podolak et al., SIGGRAPH 2006)



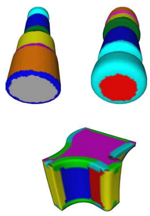


26

Slippage

(Gelfand & Guibas SGP 2004)

- Slippable motions are rigid motions which, when applied to a shape, slide the transformed version against the stationary version without forming any gaps.



27

What for?

- Shape-based retrieval
- Metamorphosis
- 3D puzzles
- Simplification
- Compression
- Collision detection
- Texture mapping
- Object modification, modeling
- Control skeleton extraction

28

Application: Shape-based retrieval

- Signature = decomposition graph with attributes
- Retrieval = sub-graph isomorphism

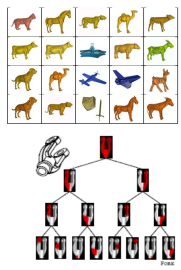
Support - human visual perception

(Biederman, Marr)

29

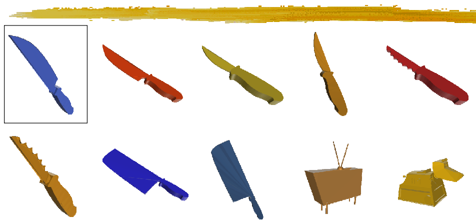
Shape Matching & Retrieval

- ZUCKERBERGER E., TAL A., SHLAFMAN S.: Polyhedral surface decomposition with applications. Computers & Graphics 26, 5 (2002), 733–743.
- PAGE D., ABIDI M., KOSCHAN A., ZHANG Y.: Object representation using the minima rule and superquadrics for under vehicle inspection. In Proceedings of the 1st IEEE Latin American Conference on Robotics and Automation (2003), pp. 91–97.
- BIASOTTI S.: 3d shape matching through topological structures. In Discrete Geometry for Computer Imagery (2003), vol. LNCS 2886, Springer-Verlag, pp. 194–203.



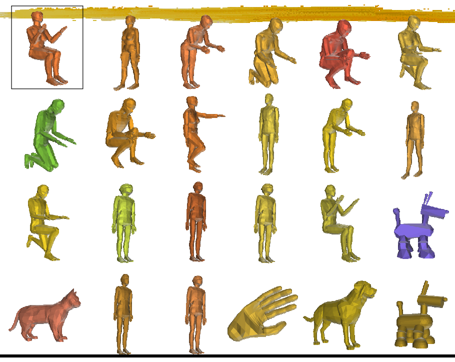
30

Results (8 /9 knives)



31

Results (19 / 19 humans)



32

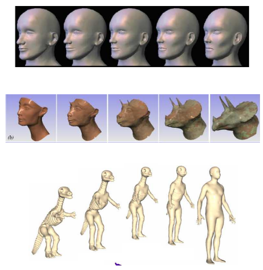
Benefits and drawbacks

+ Invariance to non-rigid transformations
+ No normalization
+ Small signatures
+ No restrictions on topology
- Computation time

33

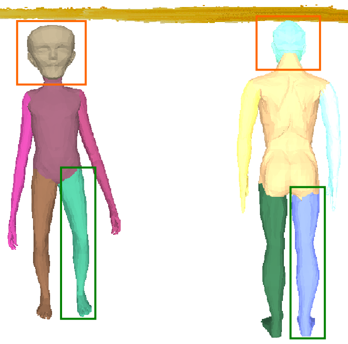
Metamorphosis

- GREGORY A., STATE A., LIN M., MANOCHA D., LIVINGSTON M.: Interactive surface decomposition for polyhedral morphing. The Visual Computer 15 (1999), 453–470.
- ZOCKLER M., STALLING D., HEGE H.-C.: Fast and intuitive generation of geometric shape transitions. The Visual Computer 16, 5 (2000), 241–253.
- ZUCKERBERGER E., TAL A., SHLAFMAN S.: Polyhedral surface decomposition with applications. Computers & Graphics 26, 5 (2002), 733–743.
- SHLAFMAN S., TAL A., KATZ S.: Metamorphosis of polyhedral surfaces using decomposition. Computer Graphics forum 21, 3 (2002). Proceedings Eurographics 2002.



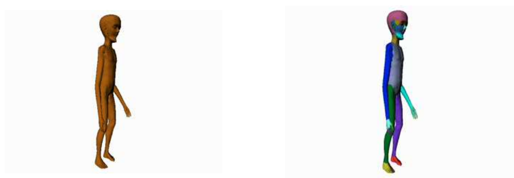
34

Metamorphosis



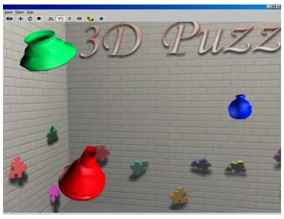
35

Metamorphosis



36

3D puzzles



Simplification

- We want to approximate a complex model (shape) with a simpler one.
- Replacing complex mathematical objects with simpler ones, while keeping the primal information content.

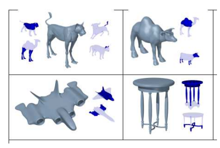
Segmentation context?

- Segment the mesh into regions which will be replaced by simpler elements (planes, cylinders etc.) while the geometric distance between the approximation elements and the original mesh will be small.

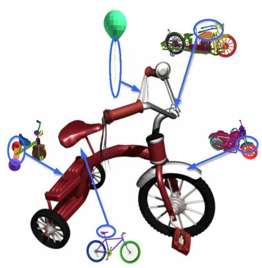
COHEN-STEINER D., ALLIEZ P., DESBRUN M.: Variational shape approximation. ACM Trans. Graph. 23, 3 (2004), 905–914.

Shape modeling

- FUNKHOUSER T., KAZHDAN M., SHILANE P., MIN P., KIEFER W., TAL A., RUSINKIEWICZ S., DOBKIN D.: Modeling by example. ACM Transactions on Graphics (Proceedings SIGGRAPH 2004) 23 (2004), 652-663.
- Vladislav Kraevoy, Dan Julius, Alla Sheffer, Shuffler: Modeling with Interchangeable Parts, Technical sketch, Siggraph 2006

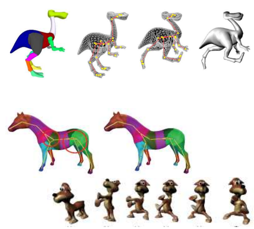


Modeling

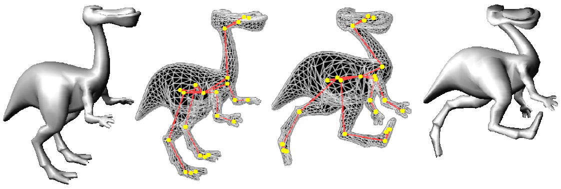


Skeleton extraction & Animation

- MORTARA M., PATAN, G., SPAGNUOLO M., FALCIDIENO B., ROSSIGNAC J.: Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. Algorithmica 38, 1 (2003), 227–248.
- KATZ S., TAL A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. ACM Transactions on Graphics (Proceedings SIGGRAPH 2003) 22, 3 (2003), 954–961.
- WU F.-C., MA W.-C., LIANG R.-H., CHEN B.-Y., OUHYOUNG M.: Domain connected graph: the skeleton of a closed 3d shape for animation. The Visual Computer 22, 2 (2006), 117–135.



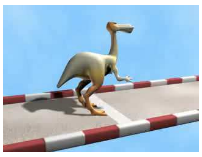
Control skeleton extraction

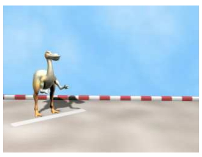


43

44

Control skeleton extraction





45

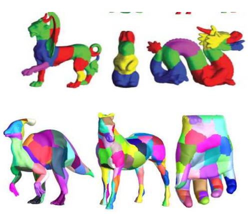
Strips & Quasi-developable surfaces

- MITANI J., SUZUKI H.: Making papercraft toys from meshes using strip-based approximate unfolding. ACM Transaction on Graphics (Proceedings SIGGRAPH 2004) 23, 3 (2004), 259–263.
- JULIUS D., KRAEVOY V., SHEFFER A.: D-charts: Quasi-developable mesh segmentation. Computer Graphics Forum (Proceedings Eurographics 2005) 24, 3 (2005), 981–990.
- SHATZ I., TAL A., LEIFMAN G.: Paper craft models from meshes. The Visual Computer (Proceedings Pacific Graphics 2006) to appear (2006).

46

Parameterization

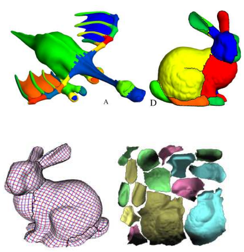
- SANDER P., SNYDER J., GORTLER S., HOPPE H.: Texture mapping progressive meshes. In Proceedings of ACM SIGGRAPH (2001), pp. 409–416.
- SORKINE O., COHEN-OR D., GOLDENTHAL R., LISCHINSKI D.: Bounded-distortion piecewise mesh parameterization. In Proceedings of IEEE Visualization 2002 (2002).
- ZHANG E., MISCHAIKOW K., TURK G.: Feature based surface parameterization and texture mapping. ACM Transaction on Graphics 24, 1 (2005), 1–27.



47

Charts creation

- LEVY B., PETITJEAN S., RAY N., MAILLOT J.: Least squares conformal maps for automatic texture atlas generation. In ACM Computer Graphics, Proc. SIGGRAPH 2002, pp. 362–371.
- ZHOU K., SYNDER J., GUO B., SHUM H.-Y.: Isocharts: stretch-driven mesh parameterization using spectral analysis. In SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing (New York, NY, USA, 2004), ACM Press, pp. 45–54.



48

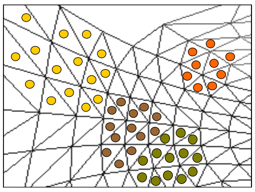
Goal – meaningful components

- Convexity
- Curvatures
- Geodesic distances

Segmentation as a clustering problem

- The basic segmentation problems can viewed as assigning primitive mesh elements to sub meshes.
- This is in fact a clustering problem of primitive elements into groups or clusters.
- This problem is well studied in Machine Learning.
- The different algorithms can be classified as variants of classic clustering algorithms.

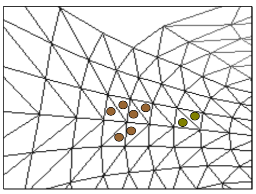
Region growing

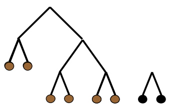


Region growing

```
Region Growing Algorithm
Initialize a priority queue Q of elements
Loop until all elements are clustered
   Choose a seed element and insert to Q
   Create a cluster C from seed
   Loop until Q is empty
      Get the next element s from Q
      If s can be clustered into C
         Cluster s into C
         Insert s neighbors to Q
Merge small clusters into neighboring ones
```


Hierarchical clustering

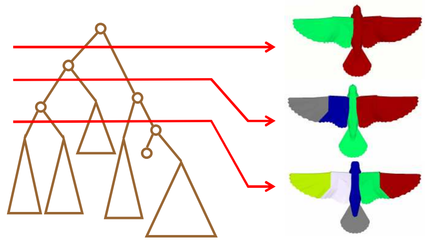




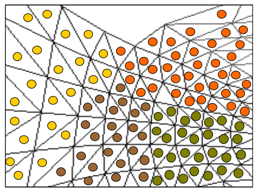
Hierarchical clustering

```
Hierarchical Clustering Algorithm
Initialize a priority queue Q of pairs
Insert all valid element pairs to Q
Loop until Q is empty
   Get the next pair (u,v) from Q
   If (u,v) can be merged
      Merge (u,v) into w
      Insert all valid pairs of w to Q
```

Hierarchy



Iterative clustering

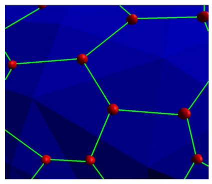


Lloyd (k-means)

```
Iterative Clustering Algorithm
Initialize k representatives of k clusters
Loop until representatives do not change
   For each element s
      Find the best representative i for s
      Assign s to the i^th cluster
   For each cluster i
      Compute a new representative
```


Graph cuts

- Define a graph where each node is an element and the edges hold weights according to the distances between the elements.
- Example: dual graph and the weight is the dihedral angle.



Finding min-cut

Mostly used on portions of the mesh for refinement of borders between segments and smoothing.

Today’s algorithms

- Convex decomposition – Chazelle et al, 97
- Wateshed – Mangan & Whitaker, 99
- Two-phase – Katz & Tal, 03
- Feature-point & Core extraction, Katz et al, 05

Convex decomposition, (Chazelle et al, 97)

- Easiest to represent, manipulate and render
- The human visual system tends to segment complex objects at regions of deep concavities (Biederman)

61

Convex decomposition

- Goal: decompose into convex patches
- Convex patch – lies entirely on the boundary of its convex hull

62

Algorithms

1. Space partitioning
2. Space sweeping
3. Flooding

63

Flooding algorithm

1. Let G be the dual graph
2. Traversing G, collecting vertices (faces), as long as a pre-defined property is not violated
3. When traversal cannot be continued, a new patch is started and the traversal is resumed.

64

Failures

- Local failure – the edge at which the facet is attached to the patch exhibits non-convexity
- Global failure – the patch is locally convex everywhere, but some facet fails to be on the boundary of the convex hull

65

Flood & Retract

1. Flood the surface by covers – patches might overlap
2. Transform the covers into partition - retracting each patch

66

Examples

67

Drawbacks

- The optimization problem is NP-complete
- Over segmentation
- Jagged boundaries

68

Watershed (Mangan and Whitaker, 99)

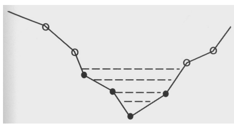
Extension to 3D a known 2D algorithm in image processing

Key idea - Regions are segmented into *catchment-basins (watersheds)*

69

Watersheds

- *Catchment-basin* - set of points whose path of steepest descent terminates in the same local minimum of a height function
- Height function - depends on the application



70

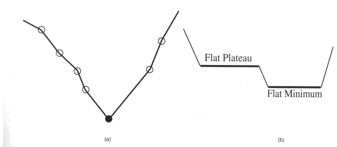
Watershed segmentation algorithm

1. Compute the height function (curvature) at each vertex
2. Find the local minima and assign each a unique label
3. Find each flat area and classify it as a minimum or a plateau
4. Loop through plateaus and allow each one to descend until a labeled region is encountered
5. Allow all remaining unlabeled vertices to similarly descend and join labeled regions
6. Merge regions whose watershed depth is below a preset threshold

71

Initial labeling

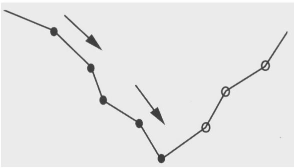
1. Local minima consisted of single vertex
2. Flat minimum
3. Flat plateau



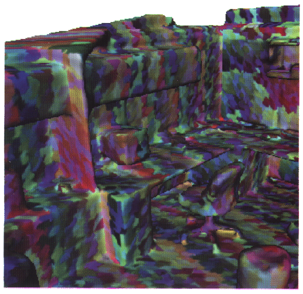
72

Descent

Imagine a drop of water placed at the starting vertex, flowing downhill on the height function



Region merging (why?)



Region merging

Metric – greatest depth of water that segment can hold before it "spills over"

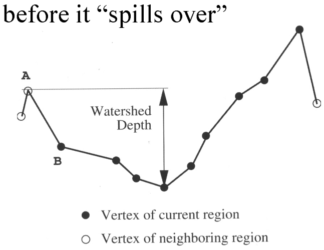


Fig. 5. Defining the depth of a region based on its lowest vertex and lowest boundary vertex.

Region merging algorithm

1. For each region, find its lowest point, neighbors, and lowest boundary point with each neighbor
2. Find depth of region, the difference between the lowest point to lowest boundary point
3. If depth is below predefined threshold, merge this region to region adjacent to lowest boundary point and update new region's information accordingly
4. Repeat until no regions exist that are below the minimum depth

Curvature calculation

Depends on type of data and the level of noise

- Inputs:
  - Volumes (voxels) - data is used to compute curvature
  - Meshes – several possibilities

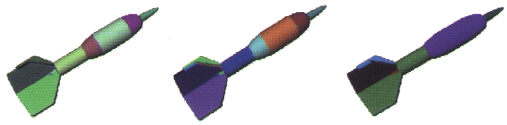
Results

1. Over-segmentation
2. Noise – partitions might fail dramatically
3. Threshold sensibily

Threshold sensitivity

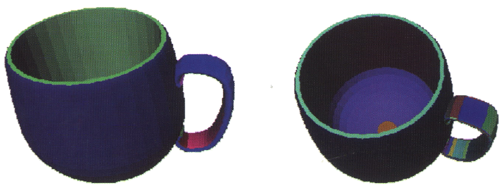
T=0.4 (498R)
T=0.425 (441R)
T=0.275 (957R)

Threshold sensitivity



T = 0.15 (103R) 0.2 (64R) 0.4 (36R)

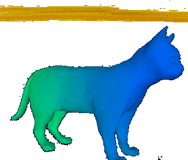
Segmentation based on mesh input



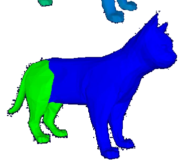
T = 0.05 (42 Regions) 0.0025 (126 Regions)

Two-phase algorithm (Katz and Tal, 03)

1. Find major components with fuzzy boundaries

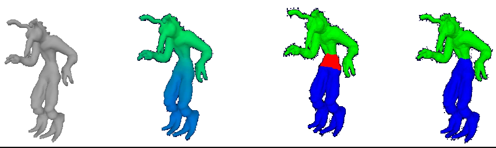


2. Find exact boundaries



Algorithm outline

1. Construct fuzzy decomposition
   a. Assign distances to pairs of faces
   b. Assign probabilities of belonging to patches
   c. Compute a fuzzy decomposition
2. Construct exact boundaries



Distance function

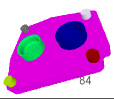
Distant faces less likely to belong to same patch

- Initially, adjacent faces:

$$Dist(f_i, f_j) = (1-\delta)\cdot\frac{AngDist(\alpha_{ij})}{avg(AngDist)} + \delta\cdot\frac{GeodDist(f_i, f_j)}{avg(GeodDist)}$$

$$AngDist(\alpha_{ij}) = \eta(1-\cos\alpha_{ij})$$

- Final distances - shortest paths

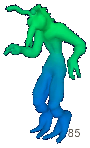


Probabilities

$$P_B(fi) = \frac{Dist(fi, REP_A)}{Dist(fi, REP_A) + Dist(fi, REP_B)} = \frac{afi}{afi + bfi}$$

Properties

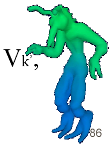
I. $\forall afi < bfi, P_B(f_i) < 0.5$
II. $\forall afi > bfi, P_B(f_i) > 0.5$
III. $\forall afi = bfi, P_B(f_i) = 0.5$
IV. $P_B(f_i) = 1 - P_A(f_i)$



Fuzzy K-means

Goal: optimize $F = \sum_p \sum_f \Pr(f \in patch(p)) \cdot Dist(f, p)$

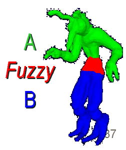
1. Initialization - select set of representatives V_k
2. Compute probabilities
3. Re-compute the set of representatives $V_{k'}$
   $REP_B = \min_f \sum_{fi} P_B(fi) \cdot Dist(f, fi)$
4. If V_k is sufficiently different from $V_{k'}$, set $V_k \longleftarrow V_{k'}$ and go back to 2



Fuzzy decomposition

The surface is decomposed into *A, B, Fuzzy*

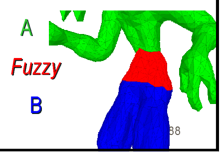
$$A = \{fi \mid P_B(fi) < 0.5 - \varepsilon\}$$
$$B = \{fi \mid P_B(fi) > 0.5 + \varepsilon\}$$
$$Fuzzy = \{fi \mid 0.5 - \varepsilon \le P_B(fi) \le 0.5 + \varepsilon\}$$



Problem: finding boundaries

- Given:
  - G=(V,E) the dual graph of the mesh
  - A,B,Fuzzy
- Partition V into $V_{A'}$ and $V_{B'}$ s.t.

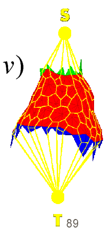
I. $V = V_{A'} \cup V_{B'}$
II. $V_{A'} \cap V_{B'} = \phi$
III. $V_A \subseteq V_{A'}, V_B \subseteq V_{B'}$
IV. Good cut!



Algorithm for finding boundaries

- Assign capacities
- Construct a flow network on *Fuzzy*
- Find the minimum cut

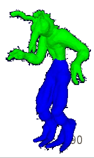
$$weight(Cut(V_{A'}, V_{B'})) = \sum_{u \in V_{A'}, v \in V_{B'}} w(u, v)$$



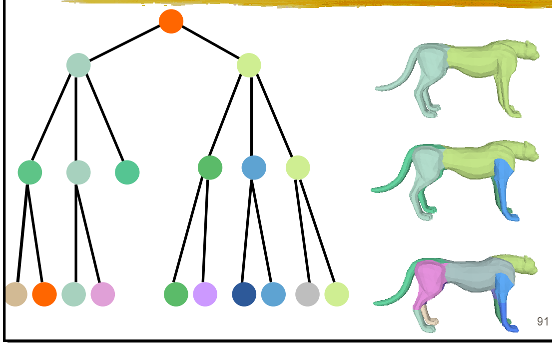
Assigning capacities

Cuts should pass at regions of deep concavities (*Biederman*)

$$Cap(i, j) = \frac{1}{1 + \frac{AngDist(\alpha_{ij})}{avg(AngDist)}}$$



Hierarchical decomposition



91

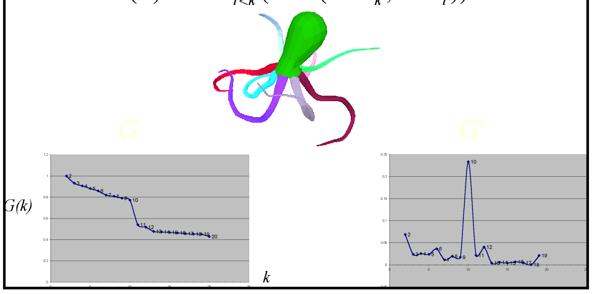
K-way decomposition

- Determining the number of patches
- Selecting initial representatives
- Assigning probability
- Extracting fuzzy areas

92

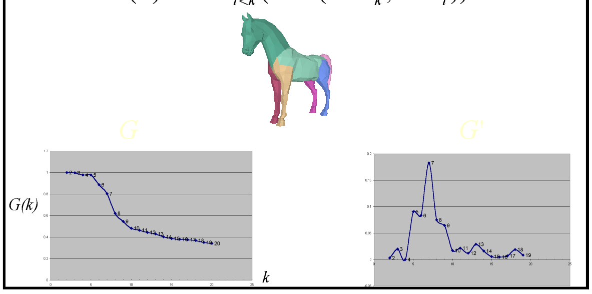
Determining #patches

$$G(k) = \min_{i<k}(Dist(REP_k, REP_i))$$

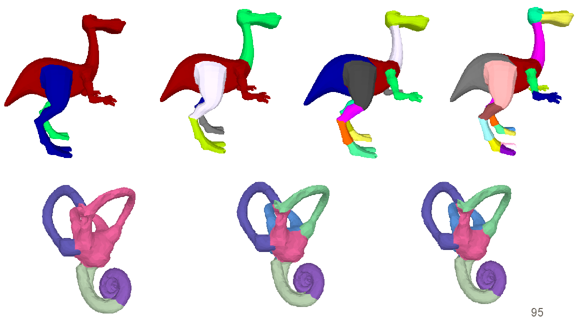


Determining #patches

$$G(k) = \min_{i<k}(Dist(REP_k, REP_i))$$

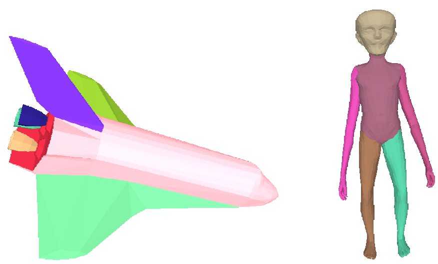


Hierarchical k-way decomposition



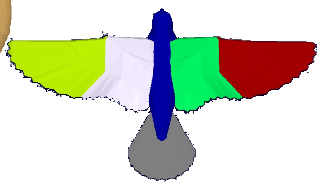
95

Examples



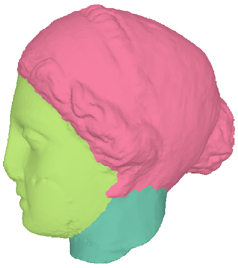
96

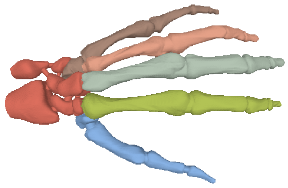
Examples



97

Examples





98

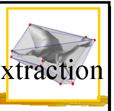
Segmentation by feature point & core extraction

For each hierarchical-level

1. Mesh coarsening
2. Pose invariant representation
3. Feature point detection
4. Core component extraction
5. Mesh segmentation
6. Coarse mesh cut refinement
7. Fine mesh cut refinement

99

Segmentation by feature point & core extraction

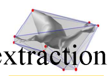


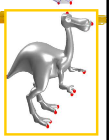
For each hierarchical-level

1. Mesh coarsening
2. Pose invariant representation
3. Feature point detection
4. Core component extraction
5. Mesh segmentation
6. Coarse mesh cut refinement
7. Fine mesh cut refinement

100

Segmentation by feature point & core extraction



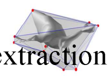


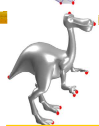
For each hierarchical-level

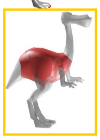
1. Mesh coarsening
2. Pose invariant representation
3. Feature point detection
4. Core component extraction
5. Mesh segmentation
6. Coarse mesh cut refinement
7. Fine mesh cut refinement

101

Segmentation by feature point & core extraction





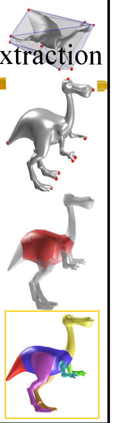


For each hierarchical-level

1. Mesh coarsening
2. Pose invariant representation
3. Feature point detection
4. Core component extraction
5. Mesh segmentation
6. Coarse mesh cut refinement
7. Fine mesh cut refinement

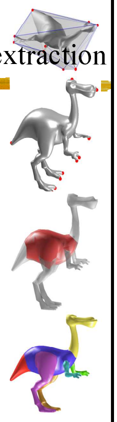
102

Segmentation by feature point & core extraction



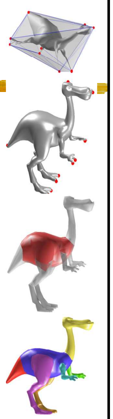
For each hierarchical-level
1. Mesh coarsening
2. Pose invariant representation
3. Feature point detection
4. Core component extraction
5. Mesh segmentation
6. Coarse mesh cut refinement
7. Fine mesh cut refinement

Segmentation by feature point & core extraction



For each hierarchical-level
1. Mesh coarsening
2. Pose invariant representation
3. Feature point detection
4. Core component extraction
5. Mesh segmentation
6. Coarse mesh cut refinement
7. Fine mesh cut refinement

Focus



For each hierarchical-level
1. Mesh coarsening
2. Pose invariant representation
3. Feature point detection
4. Core component extraction
5. Mesh segmentation
6. Coarse mesh cut refinement
7. Fine mesh cut refinement

Pose invariant representation

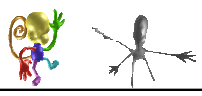
S ⟶ SMDS

Multi-Dimensional Scaling (MDS)

106

Pose invariant representation by MDS

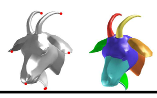
- Transform the vertices such that
  Geodesic dist. in S ≅ Euclidean dist. in SMDS
  $\delta_{ij} = dissimilarity = GeodesicDist(v_i, v_j)$ in S
  $d_{ij} = EuclideanDist(v_i, v_j)$ in SMDS
- Using MDS optimize $F_S = \frac{\sum_{i<j} (f(\delta_{ij}) - d_{ij})^2}{\sum_{i<j} d_{ij}^2}$



107

Feature points

- Should reside on tips of prominent components
- Useful for:
  - Deformation transfer
  - Mesh retrieval
  - Texture mapping
  - Metamorphosis (cross-parameterization)



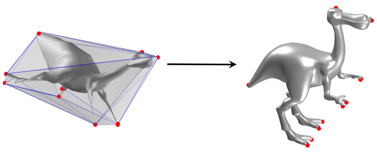
108

Feature point detection

- Local maximum of sum of geodesic distances

$$\sum_{v_i\in S} GeodDist(v,v_i) > \sum_{v_i\in S} GeodDist(v_n,v_i)$$

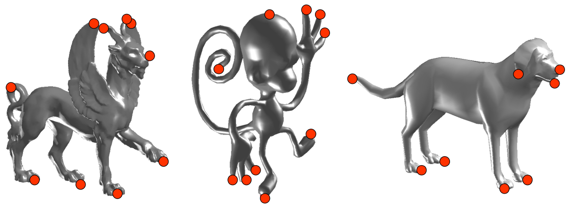
- Resides on the convex-hull of SMDS



Insensitive to noise, does not require user parameters

109

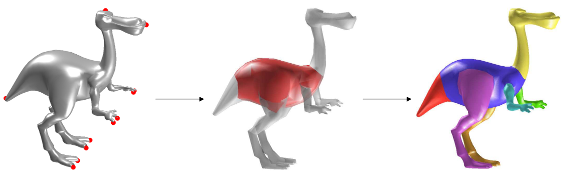
Feature point detection results



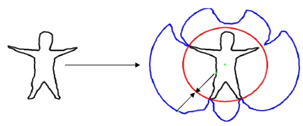
110

Core extraction & Mesh segmentation

1. Spherical mirroring of SMDS
2. Extraction of the core component of S
3. Extraction of the other segments of S



Spherical mirroring



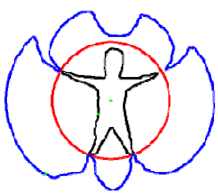
$$v_{mirror} = v + 2(R - \|v - C\|)\frac{(v-C)}{\|v-C\|}$$

$$R = \max_v \|v - C\|$$

$$C = center \quad of \quad S_{MDS}$$

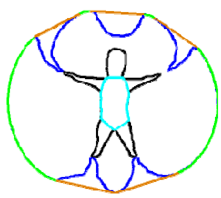
112

Core extraction



113

Core extraction

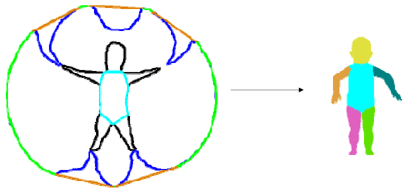


114

Core extraction results

115

Mesh segmentation

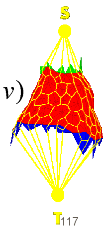


116

Refining the boundaries

- Construct a flow network on *the search region*
- Assign capacities
- Find the minimum cut

$$weight(Cut(V_{A'},V_{B'})) = \sum_{u\in V_{A'},v\in V_{B'}} w(u,v)$$



117

Cut refinement – Minimum cut

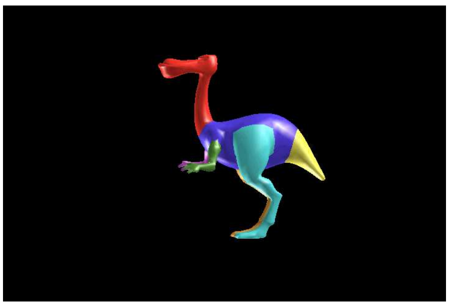
For each coarse boundary between segments:

- Search region - the faces whose distance to the coarse boundary is small
- Arc capacities

$$\omega_{ij} = \alpha\left(\frac{angW_{ij}}{AVG_{angW}}\right) + (1-\alpha)\left(\frac{edge_{ij}}{AVG_{edge}}\right)$$

118

Results – Smooth boundaries



119

Results – Insensitivity to pose

First hierarchical level

Third Hierarchical level

120

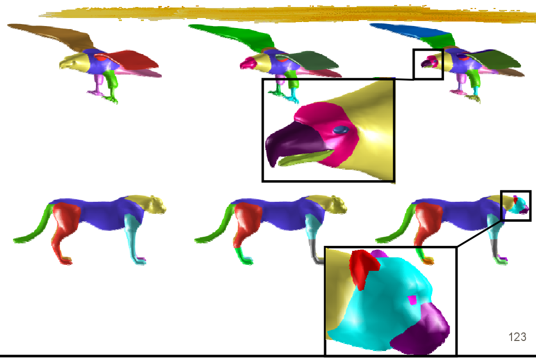
Results – Insensitivity to proportions

121

Results – Meaningful components

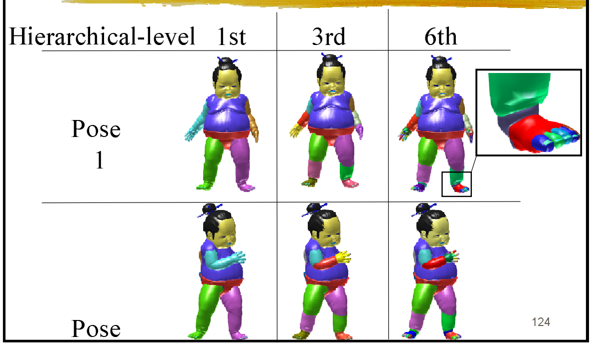
122

Results – Small feature extraction



123

Results – Correct hierarchical levels



124

Advanced Issues

- Can we say it is "correct"?
- What is the notion of "shape"?
- Some interpretation would be that our notion and perception of the shape (our "segmentation") would not change under certain transformations:
  - Rigid body invariant
  - Generally NOT Affine invariant
  - Pose invariant?

125

Final Remarks

- Many applications use mesh segmentation
- Segmentation usually has more effect on the results than seem to be realized
- 3D segmentation is still a very difficult problem – and still in its infancy, e.g. compared to image segmentation
- More advanced coherency issues should be addressed - pose invariance, extracting similar parts and more…

126