Segmentation of 3D Meshes through Spectral Clustering

Pacific Graphics 2004, Seoul, Korea
October 8, 2004

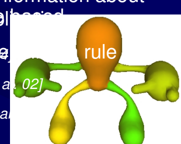
Rong Liu, Hao Zhang
GrUVi Lab, Simon Fraser University
Burnaby, BC, Canada
Email: *lrong*, *haoz*@cs.sfu.ca

Contents

Mesh segmentation

- What ?
- Why ?
- How ?

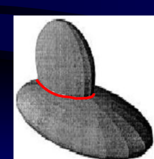
Provides higher-level structural information about



- Geometric morphing [Shlafman et al. 02]
- tion [Li et al. 01]
- Skeleton extraction [Katz & Tal 03]

The minima rule

Definition: all negative minima of the principal curvatures (along their associated lines of curvature) form boundaries between parts. *[Hoffman & Singh 84]*



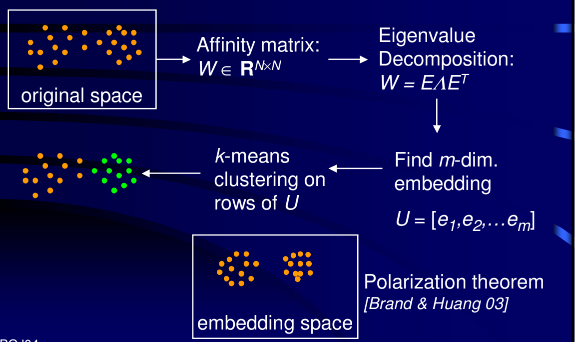
Decompose 3D shapes at concave creases

Previous work

- Watershed *[Mangan & Whittaker 99]*
  - No need to know how many parts a priori
  - Over segmentation & critical region bypass
- Fuzzy clustering *[Katz & Tal 03]*
  - Intrinsically a "fuzzy" *k*-means clustering
- Space sweeping *[Li et al. 01]*
- Convex decomposition *[Chazelle et al. 94]*

Contents

Spectral clustering



PG '04 7

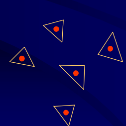
Contents

- Introduction and previous work
- Spectral clustering
- Spectral mesh segmentation algorithm
- Experimental results
- Conclusion + current and future work

PG '04 8

Overview

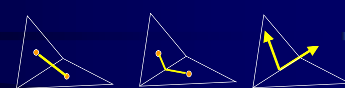
- On a 2D manifold (triangular face assumed)
- Primitives to cluster — faces
  - Faces provide the simplest and most natural tiling of the surface
  - Segmenting along edges fits well with Minima Rule



PG '04 9

Distance between faces

- Adjacent faces:



  - Distance:

    δ <geodesic distance> + $(1 - \delta)$ <angle distance>

  - Angle distance depends on convexity or concavity
  - Concave angle much more emphasized
- Non-adjacent faces i and j :
  - Distance $D_{i,j}$: shortest distance in (dual) graph

PG '04 10

Affinity matrix & normalization

- Affinity matrix W using exponential kernel

$$\mathbf{W}_{i,j} = e^{-\mathbf{D}_{i,j}/2\sigma^2}$$

- Normalization

$$\mathbf{O} = \mathbf{L}^{-1/2}\mathbf{W}\mathbf{L}^{-1/2}$$

L : diagonal matrix of W's row sums

PG '04 11

m-dimensional embedding

- i-th row of U gives m-dimensional spectral embedding of mesh face i

$$U = [e_1\ e_2 \ldots e_m]$$

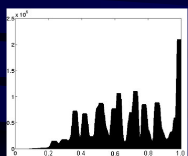
  - $e_1\ e_2 \ldots e_m$: m largest eigenvectors of O
- Normalize each embedding coordinate to have unit length

$$\hat{U}_i = U_i / ||U_i||$$

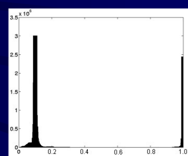
PG '04 12

Association matrix

- Computed from spectral embedding $\hat{O} = \hat{U}\hat{U}^T$



Histogram of *W* entries



Histogram of $\hat{O}$ entries

PG '04 13

k-means clustering

- Performed on embedding coordinates based on Euclidean distance
- Initial cluster centers extracted from $\hat{O}$
  - First two centers chosen to be furthest apart
  - Subsequent centers added in a min-max fashion

PG '04 14

Choice of *k* and *m*

- To determine the number of segments *k*
  - Work with original affinity matrix
  - Add representatives in min-max fashion until max affinity between them increases dramatically
  - Not guaranteed to work well in general *[Everitt 01]*
- How many eigenvectors to use?
  - Well-practiced heuristic: same as number of segments

PG '04 15

Contents

- Introduction and previous work
- Spectral clustering
- Spectral mesh segmentation algorithm
- Experimental results
- Conclusion + current and future work

PG '04 16

Execution time (in seconds)

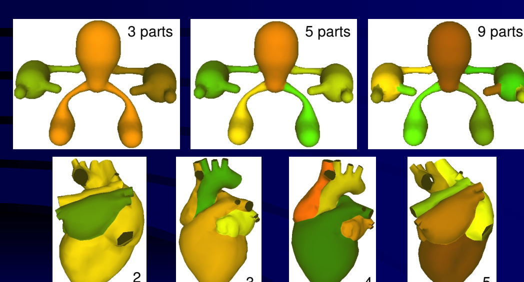
No. Faces	Affinity mat.	Eigen-decompose	Clustering	Total
496	***0.17***	0.16	0.02	0.35
800	***0.51***	0.37	0.03	0.91
1200	***1.7***	0.52	0.09	2.31
1619	***2.7***	1.1	0.17	3.97
2000	***4.76***	1.44	0.29	6.49
4000	***21.35***	6.5	1.72	29.57

Machine configuration: Intel Xeon 2.8GHz, 1GB RAM

Note: Most time spent on constructing affinity matrix

PG '04 17

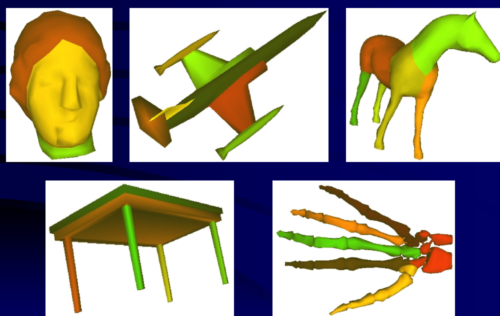
Segmentation results



Hierarchical decomposition via increase in *k*

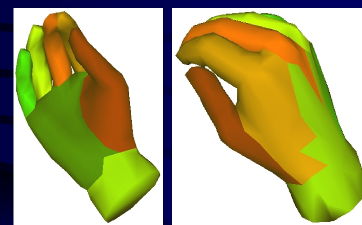
PG '04 18

More results



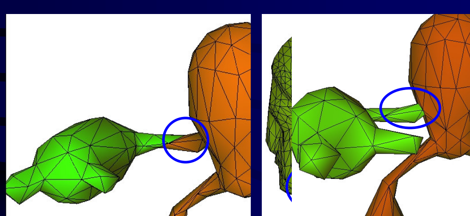
PG '04 19

Problematic examples



PG '04 20

Polarization matters



PG '04 21

Contents

PG '04 22

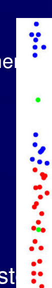
Conclusion

- Feasibility and effectiveness of spectral clustering for 3D mesh segmentation
  - Polarization matters
  - Still much to be improved, e.g., timing, parameters
- Ease of implementation
  - Plug in existing eigensolver (Arpack) and standard *k*-means
- Quality
  - Comparable to fuzzy clustering *[Katz & Tal 03]*

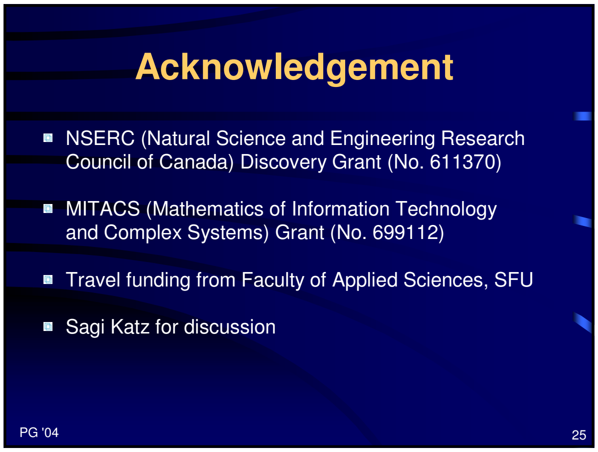
PG '04 23

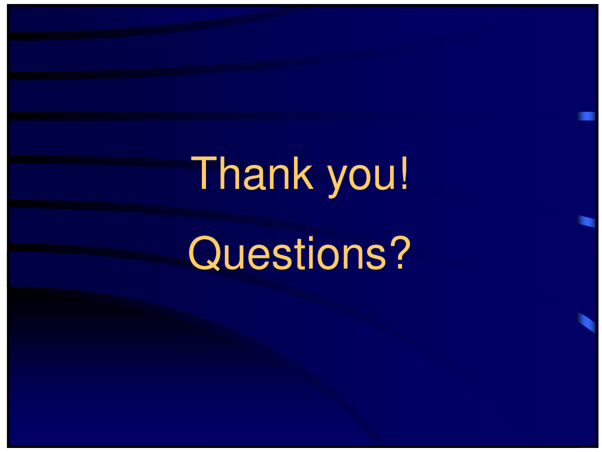
Current and future work

- Sub-sampling algorithm
  - Complexity: $O(|F|^2 log|F|) \rightarrow O(s|F|log|F|)$, whe s number of samples and is really a constant
- Hierarchical bisection
  - Avoids choosing *k*
  - Improves segmentation quality
- Boundary smoothing via morphology
- Replace *k*-means by more advanced clust g



PG '04 24


Acknowledgement
NSERC (Natural Science and Engineering Research Council of Canada) Discovery Grant (No. 611370)
MITACS (Mathematics of Information Technology and Complex Systems) Grant (No. 699112)
Travel funding from Faculty of Applied Sciences, SFU
Sagi Katz for discussion
PG '04
25


Thank you!
Questions?