Hierarchical Mesh Decomposition using Fuzzy Clustering and Cuts

Sagi Katz and Ayellet
SIGGRAPH 2003

Si Hyung Park
2010 09 15
Korea University
Computer Graphics Lab.
Abstract

• Cutting up a complex object into simpler sub-objects is a fundamental problem
  ▪ In image processing & computational geometry

• This paper
  ▪ The meaningful components
  ▪ Over-segmentation X
  ▪ Jaggy boundaries X
  ▪ Control-skeleton extraction
Flow chart

1. Preprocess
2. Determine $k$
3. Initial decomposition
4. Fuzzy area
5. Compute boundary
1 Introduction(1)

- Solid convex decomposition [Chazelle and Palios 1994] & image segmentation [Sharon et al. 2000; Shi and Malik 2000] have been considered a fundamental problem

[Sharon et al. 2000]
1 Introduction(2)

- A growing interest in mesh decomposition for computer graphics
  - [Chazelle et al. 1997; Gregory et al. 1999; Mangan and Whitaker 1999; Li et al. 2001; Shlafman et al. 2002]

[Chazelle et al. 1997]
1 Introduction(3)

- Applications of mesh decomposition
  - Metamorphosis
    - [Gregory et al. 1999; Zockler et al. 2000; Shlafman et al. 2002]
  - Compression and simplification
    - [Karni and Gotsman 2000; Garland et al. 2001; Zuckerberger et al. 2002]
  - 3D shape retrieval
    - [Zuckerberger et al. 2002]
  - Collision detection
    - [Li et al. 2001]
  - Texture mapping
    - [Levy et al. 2002]
1 Introduction(4)

• Related work
  ▪ [Chazelle and Palios 1992; Chazelle et al. 1997]
    • Convex decomposition schemes
  ▪ [Mangan and Whitaker 1999]
    • Watershed decomposition
1 Introduction(5)

- [Garland et al. 2001]
  - Face clustering
- [Li et al. 2001]
  - Skeletonization and space sweep
- [Shlafman et al. 2002]
  - K–means algorithm
1 Introduction(6)

- Improvements & deviations:
  - Hierarchical
    - No produce “flat” decompositions
  - Boundaries are handled
    - No jaggy boundaries
  - Extracting control-skeletons
    - Skeletal animations

(a) object  (b) skeleton  (c) deformed skeleton  (d) deformed object
2 Overview(1)

- S: orientable mesh

- K-way decomposition
  - \( S_1, S_2, \ldots, S_k \): a k-way decomposition of S
    - All \( i, 1 \leq i \leq k \), \( S_i \subseteq S \)
    - All \( i \), \( S_i \) is connected
    - All \( i \neq j, 1 \leq i, j \leq k \), \( S_i \) and \( S_j \) are face-wise disjoint
    - \( S_i = S \)
    - Binary Decomposition: k-way decomposition with \( k = 2 \)
    - Patch: \( S_i \) is called a patch of S
2 Overview(2)

- Each node = a particular patch
  - In the hierarchy tree
    - Root = Object S

- k-way decomposition
  - Determine “k” & compute at each node
2 Overview(3)

• Key idea :
  ▪ 1. Find the meaningful components
  ▪ 2. Focuses on the fuzzy areas
  ▪ 3. Finds the exact boundaries
2 Overview(4)

- Find fuzzy components
  - Relax the threshold
- Assigning a probability at each faces

- 4 stages
  - 1. Assigning distances
  - 2. Assigning probability
  - 3. Computing a fuzzy decomposition
    - By refining the probability values using an iterative clustering scheme. (K-means algorithm)
  - 4. Constructing boundaries
2 Overview (5)

Fuzzy component

(a) probabilities  (b) fuzzy decomposition  (c) decomposition
3.1 Computing distances(1)

• Angular distance & Geodesic distance

  ▪ Angular distance: \( \text{Ang-Dist}(\alpha_{ij}) = \eta(1 - \cos \alpha_{ij}) \).
  • \( f_i \) and \( f_j \): two adjacent faces
  • \( \alpha_{ij} \): dihedral angle
  • \( \eta \): small positive (convex angle) or 1 (concave angle)

  ▪ Geodesic distance: Center of mass
3.1 Computing distances(2)

- The dual graph of the mesh

\[ \text{Weight}(\text{dual}(f_i), \text{dual}(f_j)) = \delta \cdot \frac{\text{Geod}(f_i, f_j)}{\text{avg}(\text{Geod})} + (1 - \delta) \cdot \frac{\text{Ang-Dist}(\alpha_{ij})}{\text{avg}(\text{Ang-Dist})}. \]

- \text{avg}(\text{Geod}) : average geodesic distance

- \text{avg}(\text{Ang-Dist}) : average angular distance
  
  The denominator reduces effects different sampling rates
3.1 Computing distances(3)

- $Dist(f_i, f_m)$: the shortest path in dual graph

- Calculating in a pre-processing step

- $Dist = \infty$, when different connected components
3.2 Initialization and assigning probabilities

- \( P_B(f_i) = \frac{a_{f_i}}{a_{f_i} + b_{f_i}} = \frac{\text{Dist}(f_i, REP_A)}{\text{Dist}(f_i, REP_A) + \text{Dist}(f_i, REP_B)} \)

- \( REP_A \) and \( REP_B \): representing patches A and B
- \( a_{f_i} = \text{Dist}(f_i, REP_A) \)

- \( P_B(f_i) = 1 - P_A(f_i) \)
- \( P_B(REP_B) = 1 \)
- \( P_B(REP_A) = 0 \)
- all other faces \( 0 < P_B(f) < 1 \)
K-means algorithm

- n 개의 객체들의 집합을 k 개의 군집으로 분해
  - 거리에 기반을 둔 clustering
  - 반복/오류 회복
3.3 Generating a fuzzy decomposition(1)

\[ F = \sum_{p} \sum_{f} probability(f \in patch(p)) \cdot Dist(f, p) \]

1. Compute the probabilities of faces to belong to each patch, as described in \( P_B(f_i) = \frac{a_{f_i}}{a_{f_i} + b_{f_i}} \)

2. Re-compute the set of representatives \( V_{k'} \), minimizing the function in \( F \)

\[ REP_A = \min_{f_i} \sum (1 - P_B(f_i)) \cdot Dist(f, f_i), \quad REP_B = \min_{f_i} \sum P_B(f_i) \cdot Dist(f, f_i). \]

3. If \( V_k \) is different from \( V_{k'} \), set \( V_k \leftarrow V_{k'} \) and go back to 1.

\[ V_k : \text{a subset of } k \text{ representatives} \]

\[ \Rightarrow \text{반복/오류 회복} \]
### 3.3 Generating a fuzzy decomposition (2)

- \( A = \{ f_i | P_B(f_i) < 0.5 - \epsilon \} \)
- \( B = \{ f_i | P_B(f_i) > 0.5 + \epsilon \} \)
- \( C = \{ f_i | 0.5 - \epsilon \leq P_B(f_i) \leq 0.5 + \epsilon \} \)
3.3 Generating a fuzzy decomposition (3)

- Problem
  - The dependence on the specific representative

- Solution: the definition improves the results
  - $a_{f_i} = \text{avg}_{f_j \in A}(\text{Dist}(f_i, f_j))$
Min cut/ Max flow

- S에서 T로의 path 중 최대 용량 (Max flow)
- 그래프를 자를 때 최소 arc weights (Min cut)
3.4 Generating the final decomposition (1)

- Dual graph of the mesh: \( G = (V, E) \)
  - \( V_A \) and \( V_B \): dual vertices of A, B
  - \( V_{A'} \) and \( V_{B'} \): final dual vertices

1. \( V = V_{A'} \cup V_{B'} \)
2. \( V_{A'} \cap V_{B'} = \emptyset \)
3. \( V_A \subseteq V_{A'} \), \( V_B \subseteq V_{B'} \)
4. \( \text{weight}(\text{Cut}(V_{A'}, V_{B'})) = \sum_{u \in V_{A'}, v \in V_{B'}} \omega(u, v) \) is minimal.
3.4 Generating the final decomposition(2)

• Dual graph of $C$ : $G_C = (V_C, E_C)$
  - $V_{CA}$ (resp., $V_{CB}$): faces in “A” share an edge with $C$

• $G' = (V', E') = G_C + V_{CA} + V_{CB} + S, T$

\[
V' = V_C \cup V_{CA} \cup V_{CB} \cup \{S, T\}
\]
\[
E' = E_C \cup \{(S, v), \forall v \in V_{CA}\} \cup \{(T, v), \forall v \in V_{CB}\} \cup \{e_{ij} \in E | i \in V_C, j \in \{V_{CA} \cup V_{CB}\}\}
\]
3.4 Generating the final decomposition (3)

- Capacity of the arc

\[
\text{Cap}(i, j) = \begin{cases} 
\frac{1}{1 + \frac{\text{Ang-Dist}(\alpha_{ij})}{\text{avg}(\text{Ang-Dist})}} & \text{if } \{i, j \neq S, T\} \\
\infty & \text{else}
\end{cases}
\]

- where \( \text{Ang-Dist}(\alpha_{ij}) = \eta(1 - \cos \alpha_{ij}) \)

- A boundary: using max flow (min cut) algorithm
  - Pass through concave edges
3.5 Stopping conditions

- 1. Distance between the representatives < threshold
- 2. Difference of the max vs. min dihedral < threshold
- 3. (ave. dist in the patch) / (ave. dist in overall object) < threshold

(a) first level  (b) second level  (c) third level
4 Algorithm – the k-way case(1)

• A k-way decomposition
  ▪ A generalization of the binary case

• 3 issues
  ▪ The number of patches
  ▪ Assignment of probabilities
  ▪ The extraction of the fuzzy area
4 Algorithm – the k-way case(2)

• The first representative = main “body”
  ➔ Minimum sum of distances from all other faces

• Add representatives
  ▪ Maximize their minimum distance from previously assigned representatives
4 Algorithm – the k-way case(3)

- Determine \( k \)
  - \( G(k) = \min_{i < k} \left( \text{Dist}(\text{REP}_k, \text{REP}_i) \right) \)
  - \( G \) largely decreases when adding one more representative after assigning major parts

- \( \therefore k = \text{maximizes the first derivative of } G \)
4 Algorithm – the k-way case(4)

(a) object

(b) function $G$

(c) first derivative of $G$
4 Algorithm – the k-way case (5)

• Assignment of probabilities

\[ P_{p_j}(f_i) = \frac{1}{\text{Dist}(f_i, REP(p_j))} \sum_{l} \frac{1}{\text{Dist}(f_i, REP(p_l))} \]

• A face \( f_i \), the probability \( P_{p_j}(f_i) \) of belonging to patch \( p_j \)
• \( 0 \leq P(f_i, p_j) \) \( \leq 1 \)
• The sum of the probabilities = 1
• The distance of a face increases = the probability decreases
4 Algorithm – the k-way case(6)

- The extraction of the fuzzy area
  - Each pair of neighboring components
    - Proceed like the binary case
4 Algorithm – the k-way case(7)

- Computational complexity: $O(V^2 \log V + IV^2)$
  - $V$: number of vertices
  - $I$: number of iterations in K-means algorithm

- Distances computation: $O(V^2 \log V)$
  - Using Dijkstra’s algorithm

- Faces are assigned to patches: $O(IV^2)$

- Minimum cut: $O(C^2 \log C)$
Flow chart

Preprocess

Determine k

Initial decomposition

Fuzzy area

Compute boundary

거리 계산
Dual graph(거리.weight)
P₀들과 먼 거리 P_n 추가 초기 representatives 구성
초기 patch 구성
반복/오류 회복(K-means)
Min cut
5 Results(1)

- 4 stages in large models
  - The model is simplified
  - Decomposition
  - Define the *fuzzy regions*
  - The minimum cuts
5 Results(2)

• On P4, 1500 MHz, 512Mb RAM PC

• Mechanical part : 1 sec

• Dino-pet(3999 faces, 4 level) : 57 sec

• Venus : 244 sec

• Skeleton hand : 1654 sec

(a) alien – 3999 faces  
6 patches

(b) camel – 2674 faces  
14 patches

(c) mechanical part – 1270 faces  
7 patches

(d) heart – 1619 faces  
4 patches

(e) Venus – 67,170 faces  
3 patches

(f) skeleton hand – 654,666 faces  
6 patches
5 Results(3)

- Comparison

(a) [Li et al]  (b) [Shlafman et al]  (c) Our algorithm
5 Results(4)

- **Problem**
  - *Minimal cuts* favor small sets of isolated nodes
    - ∴ Weight increases with the number of edges
  - Another option: normalized cuts [Shi and Malik 2000]
    - NP-complete problem
    - The weight is defined by both the geo. & the ang. dist
  ➔ But the results varied
6 Control Skeleton Extraction(1)

- Benefit for matching, retrieval, metamorphosis and computer animation
- Previous are based on medial surface extraction, level set diagrams or Reeb Graphs
6 Control Skeleton Extraction(2)

- A novel control-skeleton extraction algorithm
  - From decomposition
    - general, fully automatic, simple and fast
6 Control Skeleton Extraction(3)

• Algorithm
  - 필수 조건: star-shaped decomposition structure
    - i.e. Elbow joint is a descendant of shoulder
  - Force this star-shaped decomposition structure
    - The patch is merged with a neighboring if it is not adjacent to the central patch
6 Control Skeleton Extraction(4)

- k-way decomposition is computed
  - A tree of joints is generated

- Joint is positioned at the center of mass of the boundary
6 Control Skeleton Extraction(5)

- Animate: \( y_i = \sum_{j=0}^{J-1} (w_{ij} x_{ij} M_j) \)
  - \( J \): number of joints
  - \( x_{ij} \): original vector position of \( v_i \) relative to coordinate of joint \( j \)
  - \( M_j \): transformation matrix
  - \( w_{ij} \): the extent to which it belongs to joint \( j \)
6 Control Skeleton Extraction(6)

(a) object  (b) skeleton  (c) deformed skeleton  (d) deformed object
7 Conclusion(1)

- Hierarchically decomposing meshes
  - No jaggy boundaries, No over-segmentation

- The key idea
  - Find the meaningful components
    - geodesic distances and convexity are considered
  - Exact boundaries
    - by formulating a constrained network flow problem
  - Control skeleton extraction
7 Conclusion(2)

• 개선 과제
  - Different distance functions & different capacity functions can be experimented with
  - Color and texture can be embedded in the algorithm