Progressive Mesh

Hughes Hoppe
Microsoft Research

Author

- Hughes Hoppe

Outline

- Introduction
- Progressive mesh representation
- Progressive mesh construction
- Result

Introduction

- Complex models are expensive to store, transmit, and render, thus motivating a number of problems:
- Mesh simplification
- LOD approximation
- Progressive transmission
- Mesh compression
- Selective refinement

Introduction

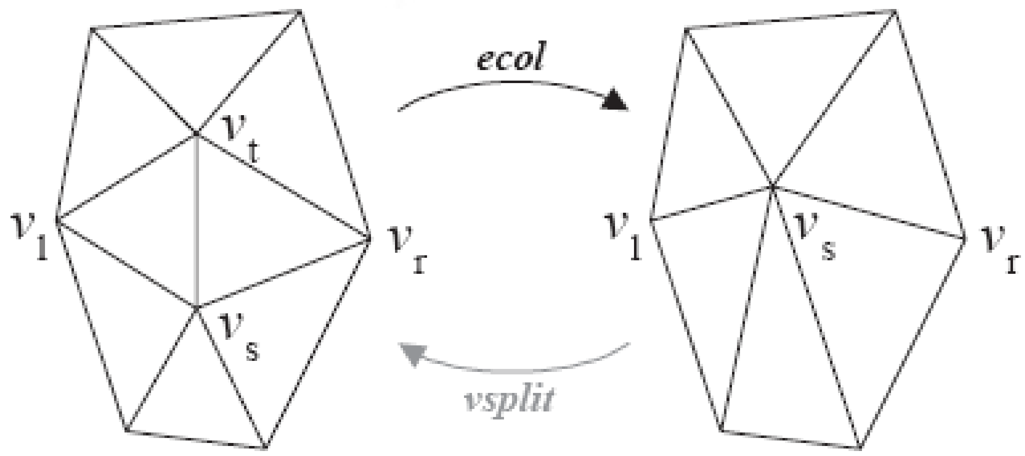
- We can use Progressive Mesh (PM) representation.
- Coarser mesh M_0 + a sequence of n detail records.

Introduction

- Discrete attributes: material identifier (shader function)
- Scalar attributes: diffuse color, normal, texture coordinates.
- M: (K,V,D,S)
- K: simplicial complex

PM representation

- Using edge collapse transformation.



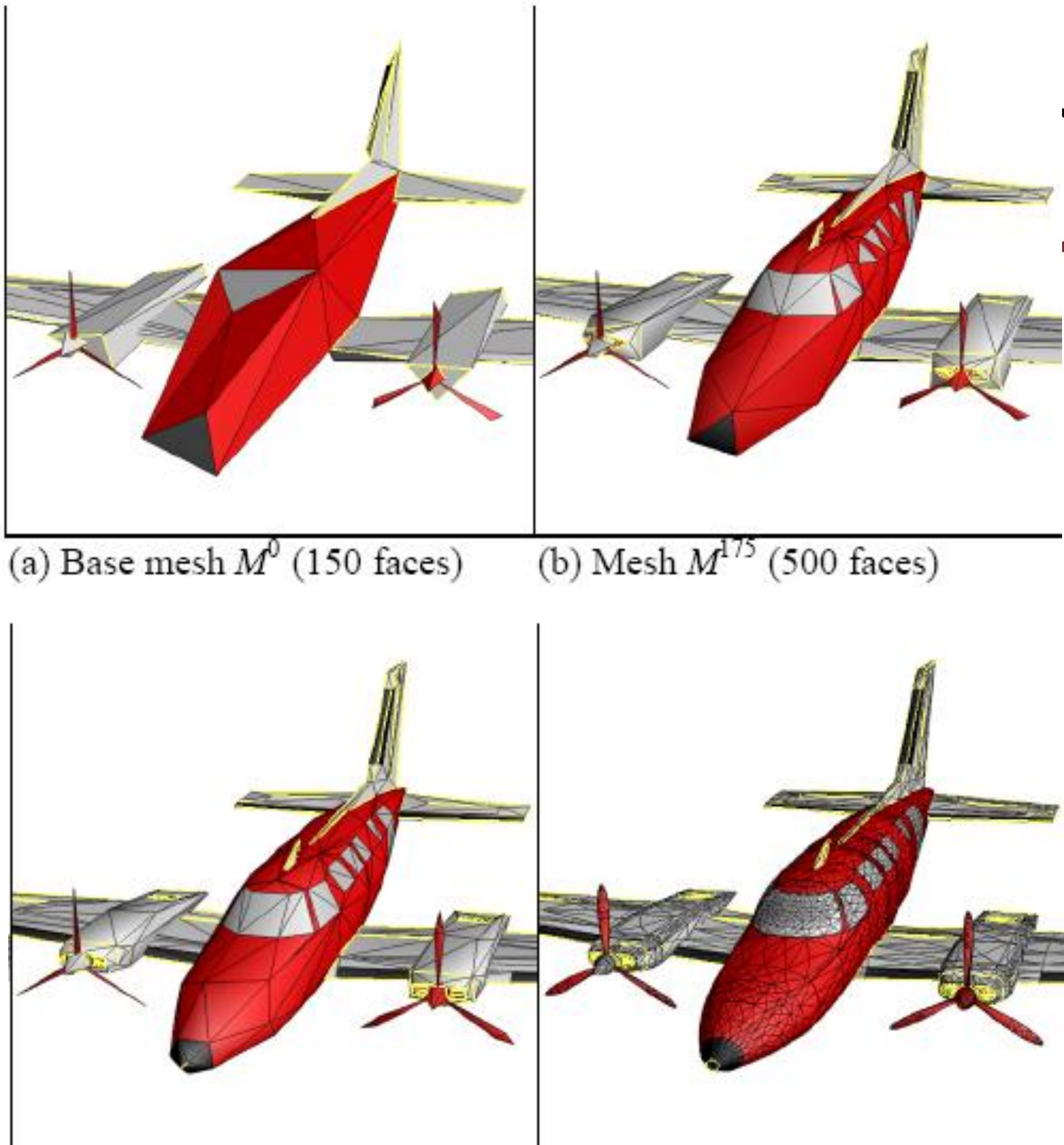
$$(\hat{M}=M^n) \xrightarrow{ecol_{n-1}} \ldots \xrightarrow{ecol_1} M^1 \xrightarrow{ecol_0} M^0$$

PM representation

- Edge collapse transform is invertible.
- Edge collapse ←→ Vertex split

$$M^0 \xrightarrow{vsplit_0} M^1 \xrightarrow{vsplit_1} \dots \xrightarrow{vsplit_{n-1}} (M^n = \hat{M})$$

- We call (M_0,{$vsplit_0$,…,$vsplit_{n-1}$}) a progressive mesh representation of M.

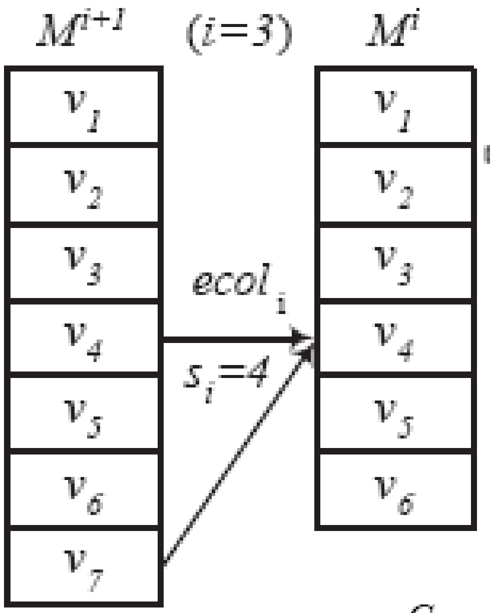


(a) Base mesh M^0 (150 faces) (b) Mesh M^{175} (500 faces)

(c) Mesh M^{425} (1,000 faces) (d) Original $\hat{M}=M^n$ (13,546 faces)

PM representation

For any two mesh M^i & M^{i+1},we can create a smooth visual transition.



Geomorph $M^G(\alpha)$

$$M^G(\alpha) = (K^{i+1}, V^G(\alpha))$$

$$\mathbf{v}_j^G(\alpha) = \begin{cases} (\alpha)\mathbf{v}_j^{i+1} + (1-\alpha)\mathbf{v}_{s_i}^i & , j \in \{s_i, m_0+i+1\} \\ \mathbf{v}_j^{i+1} = \mathbf{v}_j^i & , j \notin \{s_i, m_0+i+1\} \end{cases}$$

PM representation

- Indeed, given a finer mesh M^f and a coarser mesh M^c, $0 \leq c < f \leq n$, we can also have a geomorph $M^G(\alpha)$

$$M^G(\alpha) = (K^f, V^G(\alpha))$$

$$\mathbf{v}_j^G(\alpha) = (\alpha)\mathbf{v}_j^f + (1-\alpha)\mathbf{v}_{A^c(j)}^c$$

$$A^c(j) = \begin{cases} j & , j \leq m_0 + c \\ A^c(s_{j-m_0-1}) & , j > m_0 + c \end{cases}$$

PM construction

- Mesh optimization

$$E(M) = E_{dist}(M) + E_{rep}(M) + E_{spring}(M)$$

$$E_{dist}(M) = \sum_{i} d^2(\mathbf{x}_i, \phi_V(|K|))$$

$$E_{spring}(M) = \sum_{\{j,k\} \in K} \kappa \|\mathbf{v}_j - \mathbf{v}_k\|^2$$

PM construction

- simplification algorithm

$$E(M) = E_{dist}(M) + E_{spring}(M) + E_{scalar}(M) + E_{disc}(M)$$

- Placing all candidate edge collapse transformations into a priority queue.
- Priority: $\Delta E = E_{k'} - E_k$ for $K \rightarrow K'$

$$E_{K'} = \min_{V,S} E_{dist}(V) + E_{spring}(V) + E_{scalar}(V,S) + E_{disc}(V)$$

PM construction

- Minimize $E_{dist} + E_{spring}$

$$d^2(\mathbf{x}_i, \phi_V(|K|)) = \min_{\mathbf{b}_i \in |K|} \|\mathbf{x}_i - \phi_V(\mathbf{b}_i)\|^2$$

1. For fixed vertex positions V, compute the optimal parametrizations $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_{|X|}\}$ by projecting the points X onto the mesh.
2. For fixed parametrizations B, compute the optimal vertex positions V by solving a sparse linear least-squares problem.

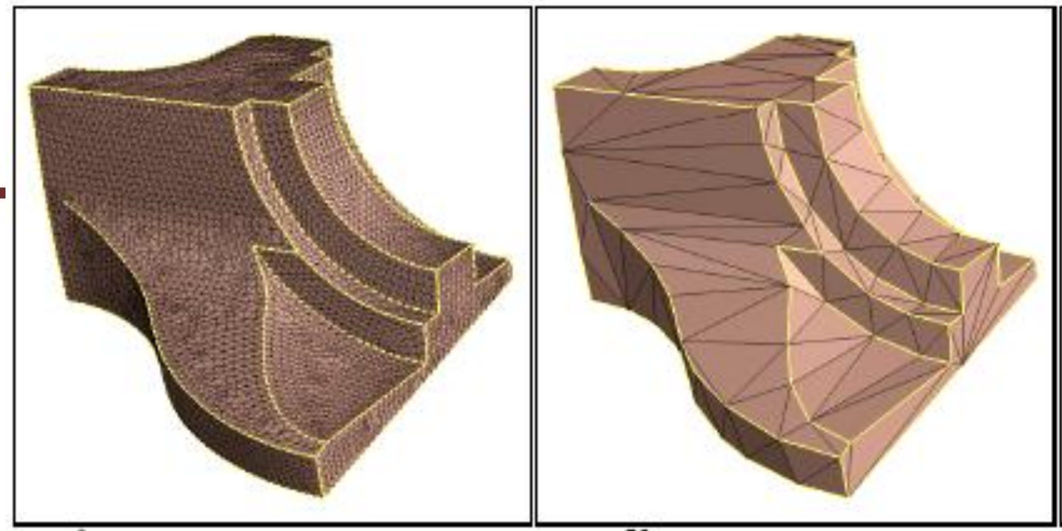
PM construction

- Minimize E_{scalar}

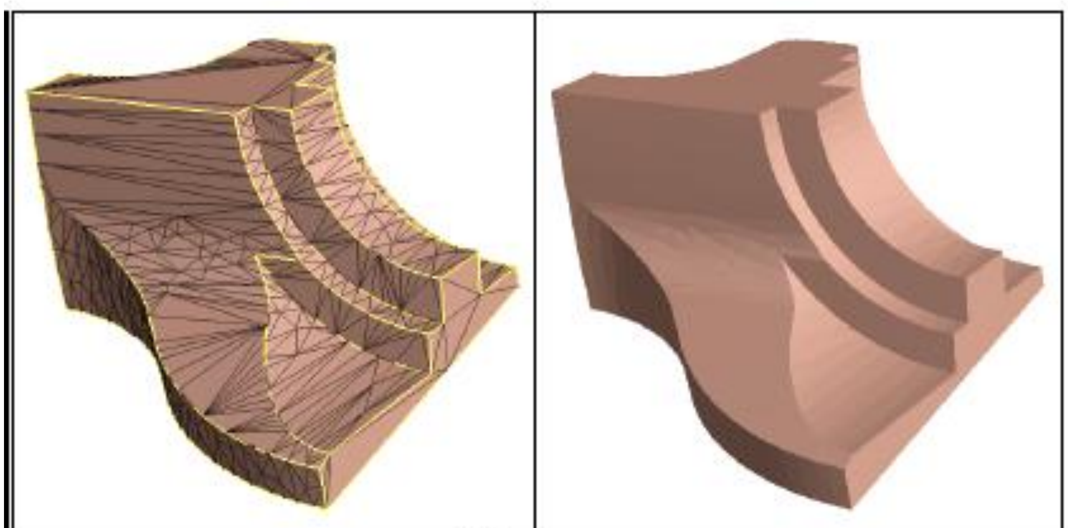
$$E_{scalar}(\underline{V}) = (c_{scalar})^2 \sum_i \|\underline{\mathbf{x}}_i - \phi_{\underline{V}}(\mathbf{b}_i)\|^2$$

PM construction

- Minimize E_{disc} :
- Preserve discontinuity curves



(a) $\hat{M}$ (12,946 faces) (b) M^{75} (200 faces)



(c) M^{475} (1,000 faces)