Geometry Processing

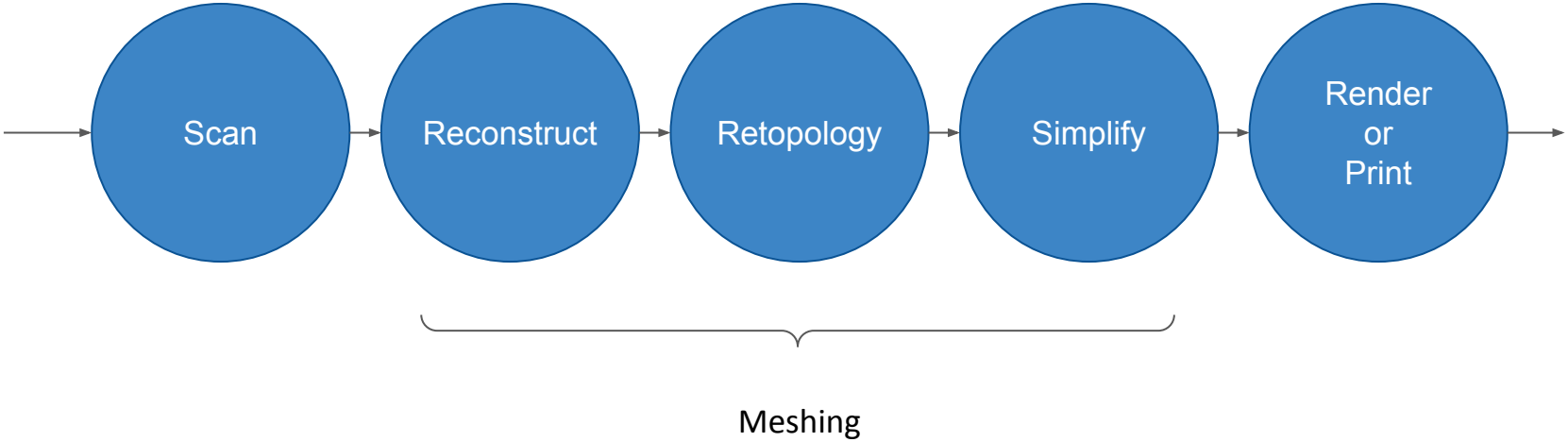
5 Remeshing

Ludwig-Maximilians-Universität München

Session 5: Remeshing

- Motivation
- Mesh Sampling
  - Up-sampling: Subdivision
  - Down-sampling: Simplification
  - Re-sampling: Redescritization
- Summary
- Discussion

Meshing Pipeline



Classification of Terminologies

Meshing

- Up-sampling: appears more often in post processing or runtime phase
  - Local: Subdivision, tessellation
- Down-sampling: appears more often in processing phase
  - Local: Decimation, simplification
- Re-sampling: appears more often in pre-processing phase
  - Global: Retopology, *remeshing*, reconstruction

The Meshing Dilemma

General goal of meshing: Approximate the input well while

- changing mesh connectivity to improve quality
- replacing an arbitrarily structured mesh by a structured one

The Meshing Dilemma: Do you want a target mesh with

high quality, *high render efficiency*, or *high customizability*?

(high polycount, low polycount, complex editing UI)

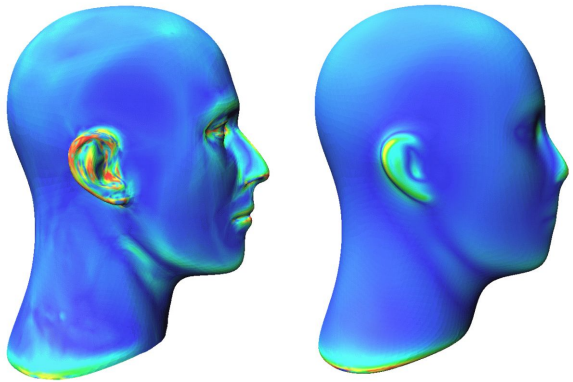
Mesh Quality Inspection

Qualitative Criteria: Visual Perspective

- Subjective concept, such as subtle bumps, wiggles, and inflection points
- Related to human perception, such as applied shading effects
- Dependent on designer's experience
  - Examining by eye
  - No objective measures
  - Cannot do it mathematically
- **Remains an open question**

Quantitative Criteria: Numerical Perspective

- Local structure
- Global structure



Why do we care about mesh quality?

- **Visual artifacts**: allow triangles with arbitrarily large can cause giant jagged spikes
- **Memory and time limitation**: Triangles with small areas use space very inefficiently.
- **Geometric queries**: Triangle with poor quality can create serious robustness problem for topology changes, boundary integration, collision detection, etc.
- **Numerical stability**: Good Laplacian!

Local Structures

Shape

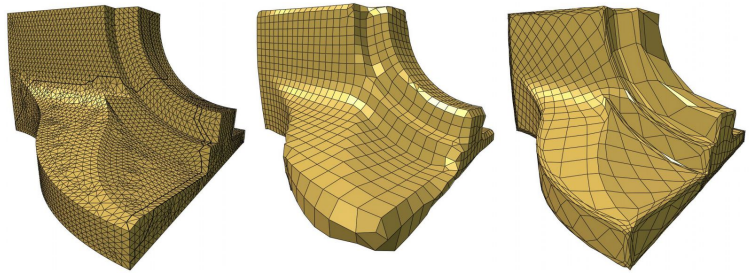
- Isotropic
- Anisotropic

Sampling density

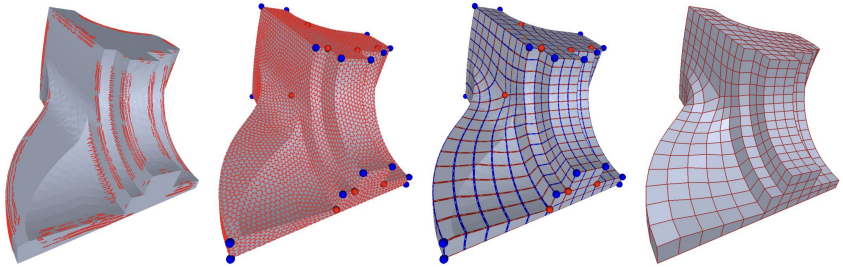
- Uniform
- Adaptive

Alignment and orientation

- Singularity
- Field-alignment



[Kovacs et al 2011]



[Bommes et al 2009]

Global Structure

Regularity of Vertex: Recall valence of triangle mesh and quad mesh

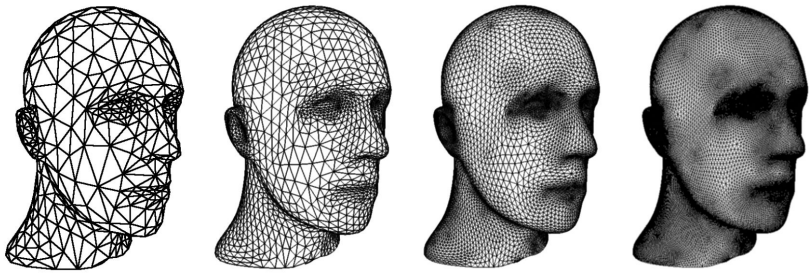
- Regular (degree, 6 for tri-mesh and 4 for quad-mesh)
- Irregular (Singular)

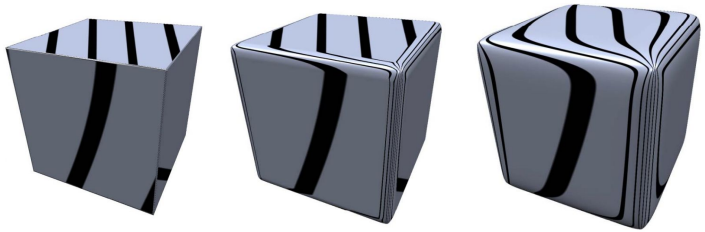
Regularity of Mesh

- Irregular: most vertices are irregular
- Semi-regular: regular subdivision of a coarse initial mesh
- High regular: most vertices are regular
- Regular: all vertices are regular

Fairness: Different continuity (position/normal/curvature)

...





C_0 C_1 C_2

Session 5: Remeshing

- Motivation
- Mesh Sampling
  - Up-sampling: Subdivision
  - Down-sampling: Simplification
  - Re-sampling: Redescritization
- Summary
- Discussion

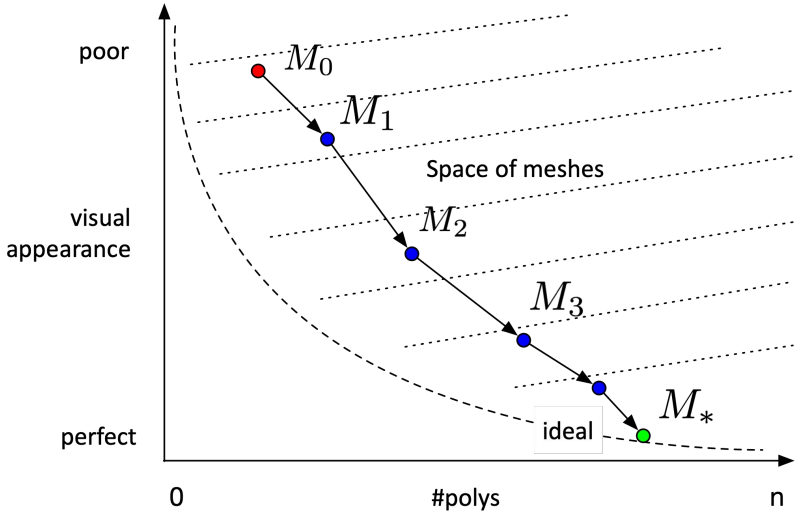
Up-Sampling: Subdivision

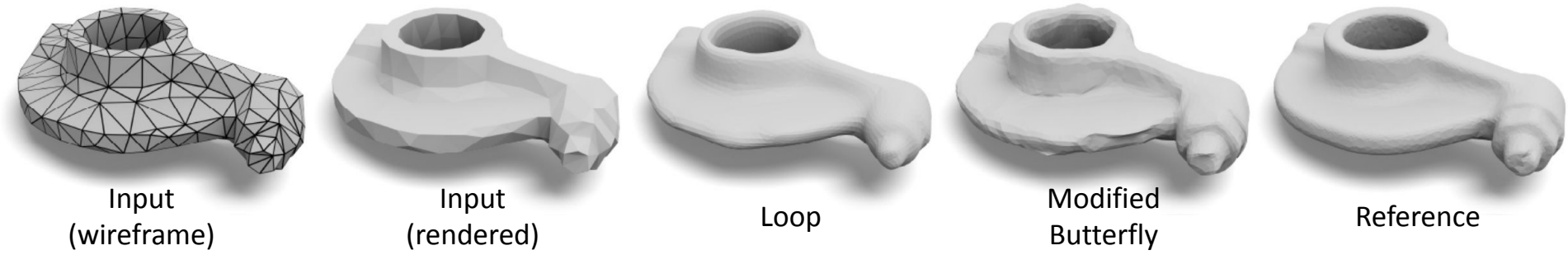
Increase polycount that smoothly approximate *its shape*

Algorithms:

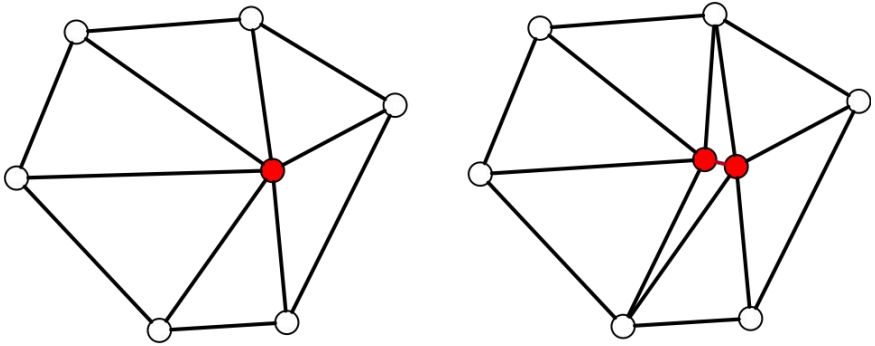
- Loop Subdivision [Loop 1987]
- Modified Butterfly [Zorin et al 1996]
- Catmull-Clark Subdivision [Catmull and Clark 1998]

Recent example: Neural Subdivision [Liu et al 2020]

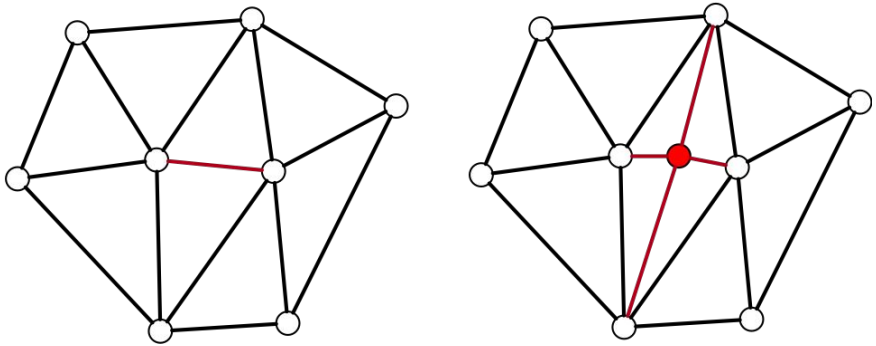




Vertex/Edge Split Operator



Vertex Split

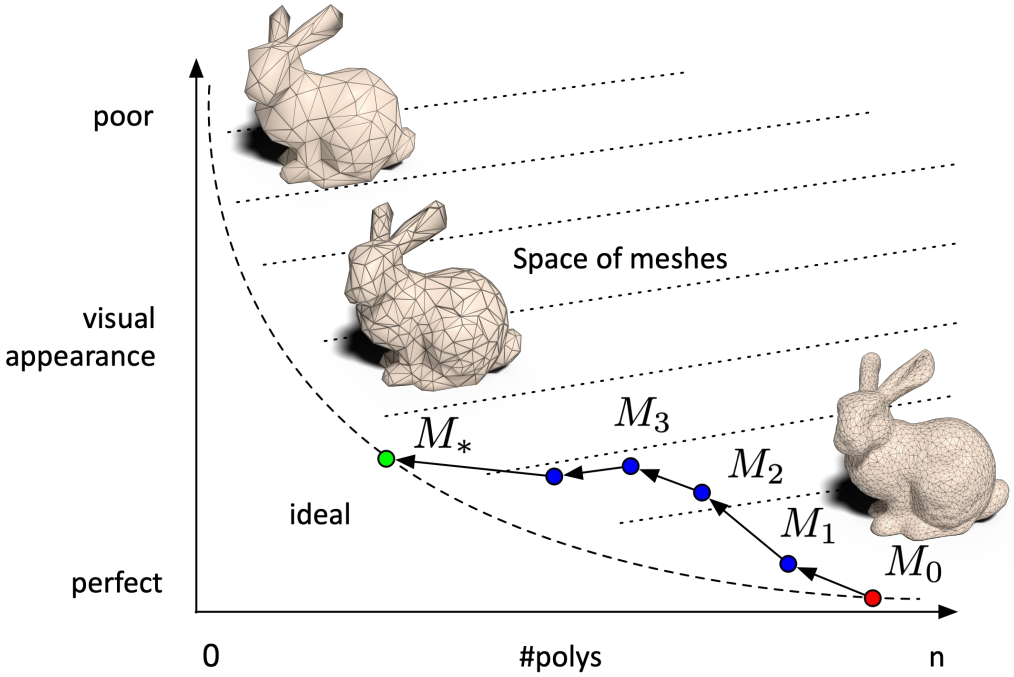


Edge Split

Down-Sampling: Simplification

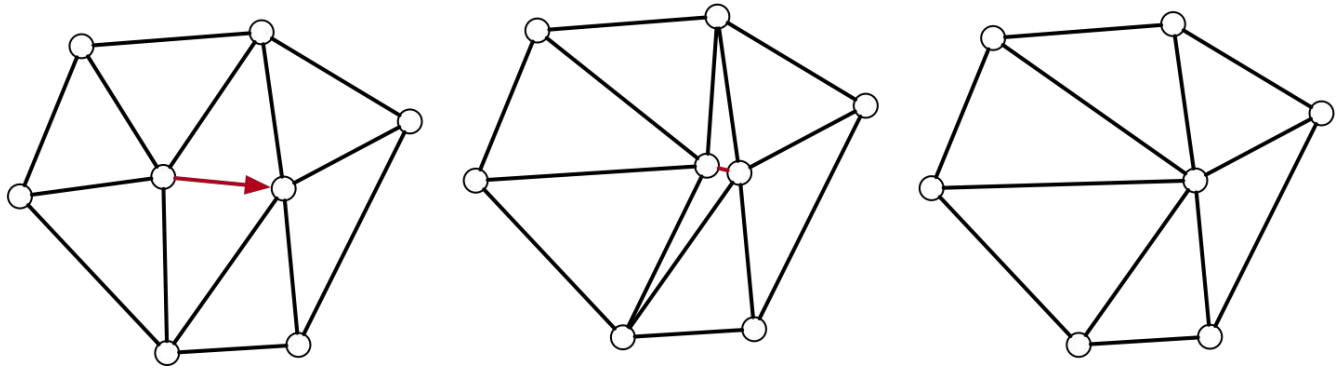
Transforming (altering) a given polygonal mesh into another mesh with fewer faces, edges, and vertices.

The simplification or approximation procedure is usually controlled by user-defined quality criteria (metric).



Edge Collapse (Decimate) Operator

Basic Idea: Collapse an edge then merge one vertex into the other, or replace two vertices by a new one

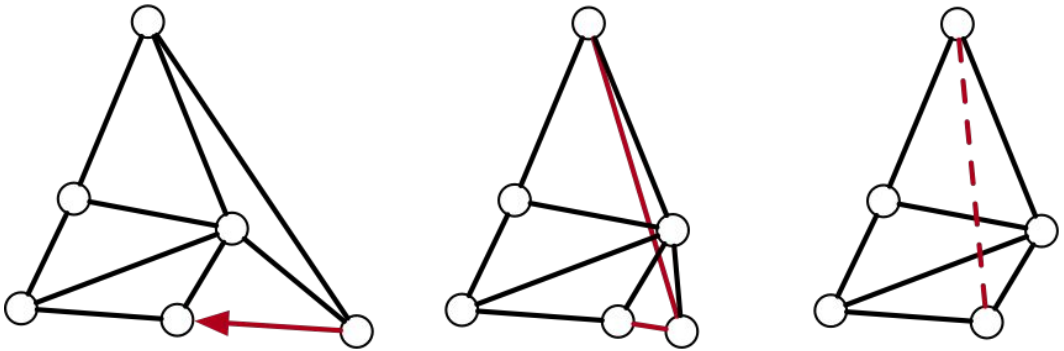


Q: How many vertices, faces and edges are removed in each *edge collapse*?

Q: How to do this in the halfedge representation?

Edge Collapse Operator: Artifacts

Caution: use face normal for checking flipped faces



Euler Operators [Eastman 1979]

Basic Idea: Altering mesh while preserving topology

Name	Description	ΔV	ΔE	ΔF
MBFLV	Make Body-Face-Loop-Vertex	1	0	1
MEV	Make Edge-Vertex	1	1	0
MEFL	Make Edge-Face-Loop	0	1	1
MEKL	Make Edge, Kill Loop	0	1	0
KFLEVB	Kill Faces-Loops-Edges-Vertices-Body	-2	$-n$	$-n$
KFLEVMG	Kill Faces-Loops-Edges-Vertices, Make Genus	-2	$-n$	$-n$

Q: How to represent edge collapse using Euler operators?

Multiresolution Representation

Recall from Laplacian smooth implementation:

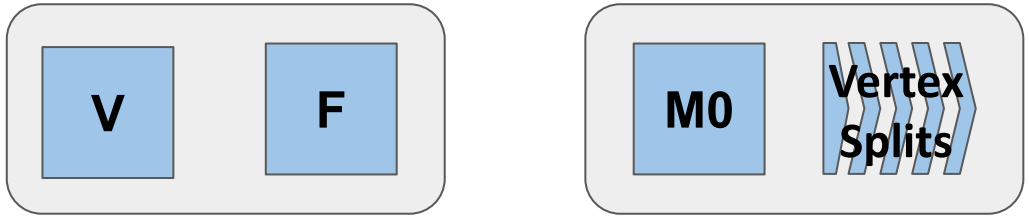
How did you cache smoothed results?

Representation refinement to support multiple resolutions:

- Discrete levels of detail (LOD): Select appropriate model based screen space object size
- Continuous LOD?

Progressive Meshes [Hoope 1996]

Basic idea: replace single resolution via base resolution plus a series of vertex split



Single Resolution

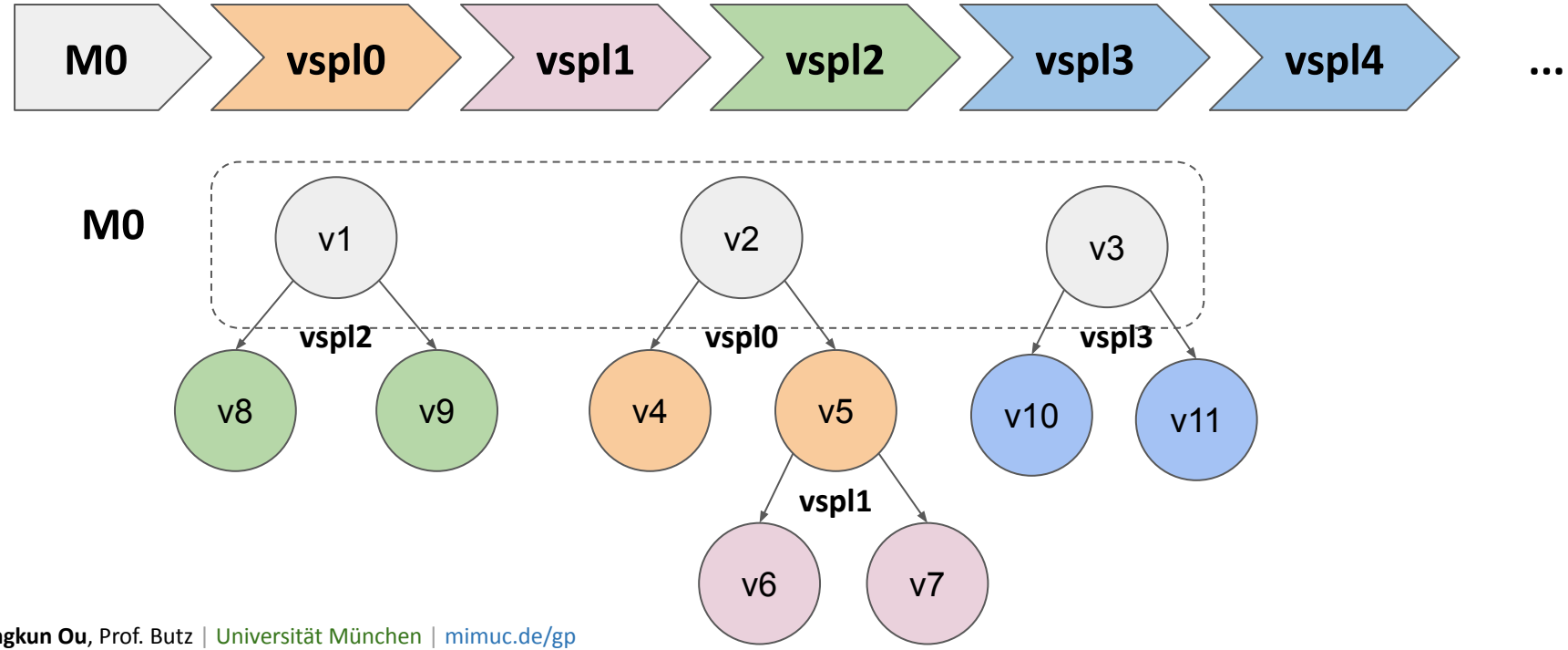
Multiresolution Progressive Mesh

Streaming Scenario: reduce bandwidth when transmitting a mesh over network:

- Mesh = list of vertices + list of faces, versus:
- Mesh = Base mesh (M0) + list of vertex split operations

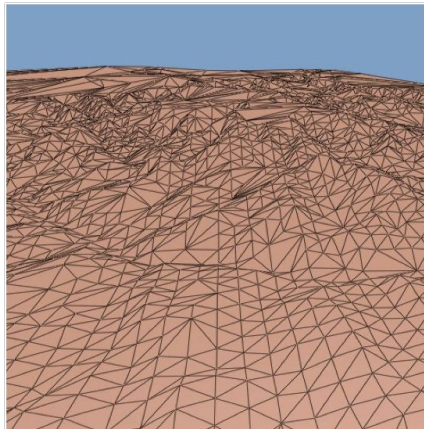
Progressive Meshes [Hoope 1996]

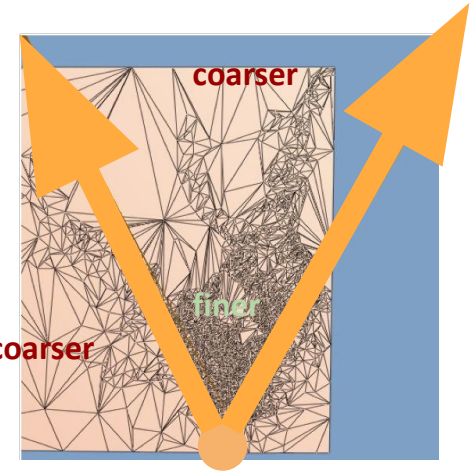
Progressive meshes enable *continuous* levels of detail or multiresolution hierarchy

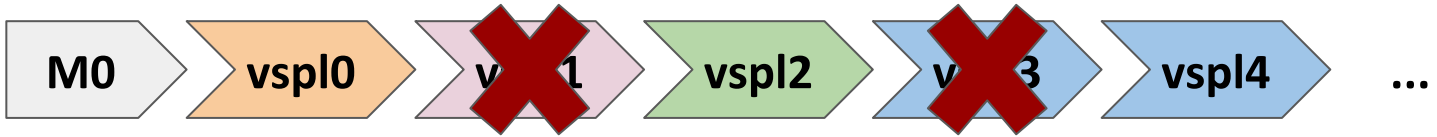


Application: View-dependent Refinement [Hoppe 1997]

Show nearby portions of object at higher resolution than distant portions







Local Measuring: Curvature-base Cost [Melax 1998]

A possible way

$$\text{cost}(u,v) = ||u-v|| \times \max_{f \in T_u} \left\{ \min_{n \in T_{uv}} \{1 - N_u \cdot N_v\} \right\}$$

where ***Tu*** is the set of triangles that contains ***u***, ***Tuv*** is the set of triangles that contains both ***u*** and ***v***.

Implementation thinking: costs of neighbors can also be affected

Use data structure: *priority queue* or *min-heap*.

- cost of access min element: ***O(1)***
- cost of affected elements manipulation: ***O(log(n))***

Local Measuring: Quadric Error Metric (QEM) [Garland 1997]

Basic idea: Minimize distance to neighboring triangles' planes

Target vertex: $\mathbf{x} = (x, y, z)^\top$

Neighboring faces: $\mathbf{p}_i = (\mathbf{n}_i, \mathbf{x}_i)$ where $\mathbf{n}_i$ is unit face normal

Distance: $||\mathbf{x}, \mathbf{p}_i||^2 = (\mathbf{n}_i^\top \mathbf{x} - \mathbf{n}_i^\top \mathbf{x}_i)^2$

Using homogeneous coordinates $\bar{\mathbf{x}} = (\mathbf{x}, 1)^\top, \bar{\mathbf{n}}_i = (\mathbf{n}_i, -\mathbf{n}_i^\top \mathbf{x}_i)^\top$

The distance can be rewritten as: $||\mathbf{x}, \mathbf{p}_i||^2 = (\bar{\mathbf{n}}_i^\top \bar{\mathbf{x}})^2 = \bar{\mathbf{x}}^\top \bar{\mathbf{n}}_i \bar{\mathbf{n}}_i^\top \bar{\mathbf{x}} = \bar{\mathbf{x}}^\top \mathbf{Q}_i \bar{\mathbf{x}}$

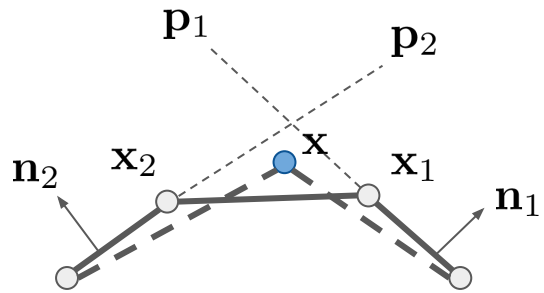
Define $\mathbf{Q}_i = \bar{\mathbf{n}}_i \bar{\mathbf{n}}_i^\top$ (4x4 matrix, irrelevant to the unknown $\mathbf{x}$) as ***face quadrics***

Thus, the sum (vertex quadrics) of squared distance of $\mathbf{x}$ to all neighboring faces: $\sum_i \bar{\mathbf{x}}^\top \mathbf{Q}_i \bar{\mathbf{x}} = \bar{\mathbf{x}}^\top \mathbf{Q} \bar{\mathbf{x}}$ where $\mathbf{Q} = \sum_i \mathbf{Q}_i$

Goal: Minimize sum of squared distance ⇒ Solve linear equation $\bar{\mathbf{x}}^\top \mathbf{Q} \bar{\mathbf{x}} = 0$ and obtain the position of target vertex that can replace a given edge (for edge collapse)

Define ***edge error*** based on the solved optimal target vertex $\bar{\mathbf{x}}^\top (\mathbf{Q}_1 + \mathbf{Q}_2) \bar{\mathbf{x}}$

where $\mathbf{Q}_1 + \mathbf{Q}_2$ is called the ***edge quadrics***



QEM-based Mesh Simplification (QSlim) [Garland 1997]

Basic idea: Do edge collapse on an edge with minimum edge quadric error

Procedure:

1. Pre-compute *vertex quadrics* for all vertices, then compute *edge quadrics* for all edges

2. Compute optimal vertex for each edge, and enqueue edge into a priority queue using the edge error as priority

3. Iteratively pop edges from the priority queue, then do edge collapse, and update neighboring edge errors. Terminate until the number of faces meets the desired number of faces

QEM-based Mesh Simplification (QSlim) [Garland 1997]

Implementation thinking:

- Preserving boundary: assign big edge error for boundary edges, or skip edge collapse directly
- Prevent flipping face: compare normals before/after edge collapse
- Bounded errors: terminate edge collapse process if edge error is greater than expected threshold
- Performance: avoid removing edges immediately from the heap (moving elements from array). Instead, mark edge as removed, and do cleanup of the heap (array-based) at the end of simplification

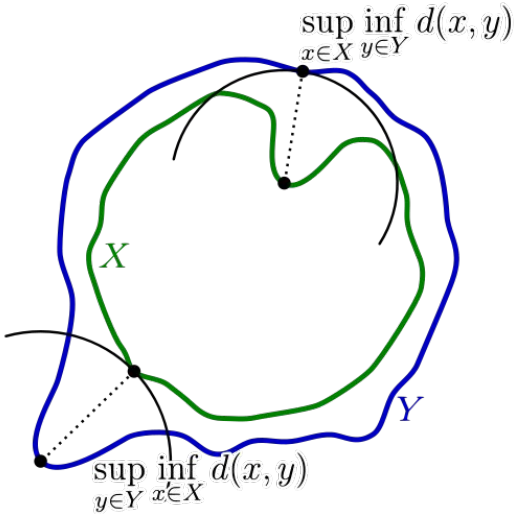
Pseudo-code:

```
const pq = new PriorityQueue()
edges.forEach(e => pq.enqueue(e)) // sort based on quadric error
for (; mesh.numFaces() > targetFaces; ) {
     const edge = pq.dequeue()
     if (edge is valid) edge.collapse()
}
```

Global Measuring: Hausdorff Distance

Basic idea: use hausdorff distance

$$d_H(X,Y) = \max\{\sup_{x\in X}\inf_{y\in Y} d(x,y), \sup_{y\in Y}\inf_{x\in X} d(x,y)\}$$



Generalized Metrics

Generalization required to handle appearance properties, such as color/texture/normals.

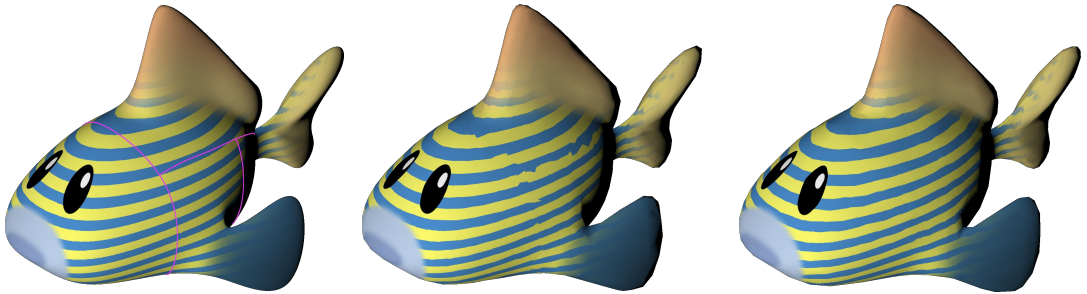
Appearance-base metric: treat each vertex as a 6-vector [x,y,z,r,g,b]

- Assume this 6D space is Euclidean (color space is only roughly Euclidean)

More metrics based on quadrics:

	Vertex	Dimension
Color	(x, y, z, r, g, b)	6x6
Texture	(x, y, z, u, v)	5x5
Normal	(x, y, z, l, m, n)	6x6
Color+Normal	(x, y, z, r, g, b, u, v, w)	9x9

$$Q(\mathbf{v}) = \mathbf{v}^\top \mathbf{A}\mathbf{v} + 2\mathbf{b}^\top \mathbf{v} + \mathbf{c}$$



Ground Truth | QEM [Garland 1997] | Seamless [Liu et al 2017]

Resampling: Energy [Hoppe et al 93]

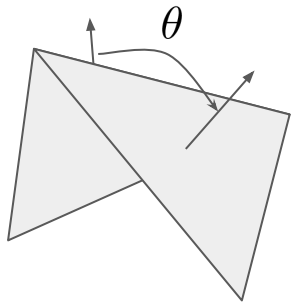
Consider simplification as an optimization problem, minimize an energy function:

$$E(\mathcal{M}) = E_{\text{dist}}(K, V) + E_{\text{rep}}(K) + E_{\text{spring}}(K, V)$$

Distance energy: sum of squared distance from target mesh to input mesh

Representation energy: penalizes meshes with large number of vertices

Spring energy penalizes sharp dihedral angles

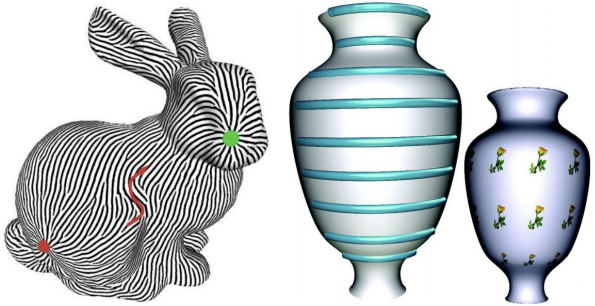


Resampling: Directional Field

Recent increasing interests of quad meshing

Consider constraints regarding alignment quirement

1-vector field	2^2-vector field
2-direction field	2^2-direction field
1^3-vector field	6-direction field
4-vector field	2^3-vector field



[Jakob et al 2015]

Summary

- (Re)Meshing can be a problem of local, global, or *semi-global*
- The dilemma of meshing is to balance the customizability of mesh editing UI, speed of mesh pre-processing and real-time mesh processing
- Dealing with large scale (noising) mesh processing remains challenging
- Improving meshing process and inspecting mesh quality remain open problems

```
prepare Ax = b
solve // expensive global step
update mesh
```

Global Approach

```
for region in mesh {
     if meet defined criteria { break }
     apply local operator // cheap greedy step
}
```

Local Approach

Further Readings: Subdivision

[Loop 1987] Charles Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of Mathematics. 1987.

[Catmull and Clark 1998] E. Catmull and J. Clark. Recursively Generated B-Spline Surfaces on Arbitrary Topological Meshes. Computer-aided design. 1998

[Zorin et al 1996] Denis Zorin, Peter Schröder, and Wim Sweldens. 1996. Interpolating Subdivision for Meshes with Arbitrary Topology. SIGGRAPH 96. 1996.

[Liu et al 2020] Hsueh-Ti Derek Liu, Vladimir G. Kim, Siddhartha Chaudhuri, Noam Aigerman, and Alec Jacobson. Neural subdivision. ACM Trans. Graph. 2020

Further Readings: Simplifications

[Hoppe 1996] Hoppe, Hugues. "Progressive meshes." *SIGGRAPH' 96,* 1996.

[Garland et al 1997] Garland, Michael, and Paul S. Heckbert. "Surface simplification using quadric error metrics." *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. 1997.

[Melax 1998] Melax, Stan. "A simple, fast, and effective polygon reduction algorithm." *Game Developer*. 1998.

[Luebke 2001] Luebke, David P. "A developer's survey of polygonal simplification algorithms." *IEEE Computer Graphics and Applications*. 2001.

[Liu et al. 2017] Liu, Songrun, et al. "Seamless: seam erasure and seam-aware decoupling of shape from mesh resolution." *ACM Trans. Graph.* 2017.

[Lescoat et al.2020] Lescoat, Thibault, et al. "Spectral Mesh Simplification." *Computer Graphics Forum*. Vol. 39. No. 2. 2020.

Further Readings: Remeshing

[Bommes et al. 2009] Bommes, David, Henrik Zimmer, and Leif Kobbelt. "Mixed-integer quadrangulation." *ACM Trans. Graph.* 2009.

[Felix et al. 2013] Knöppel, Felix, et al. "Globally optimal direction fields." *ACM Trans. Graph.* 2013. (tri-only)

[Jakob et al. 2015] Jakob, Wenzel, et al. "Instant field-aligned meshes." *ACM Trans. Graph.* 2015.

[Huang et al. 2018] Huang, Jingwei, et al. "Quadriflow: A scalable and robust method for quadrangulation." *Computer Graphics Forum*. 2018.

[Lyon et al. 2019] Lyon, Max, et al. "Parametrization quantization with free boundaries for trimmed quad meshing." *ACM Trans. Graph. 2019.*

[Lyon et al. 2020] Lyon, Max, David Bommes et al. "Cost Minimizing Local Anisotropic Quad Mesh Refinement." *Computer Graphics Forum*. 2020.

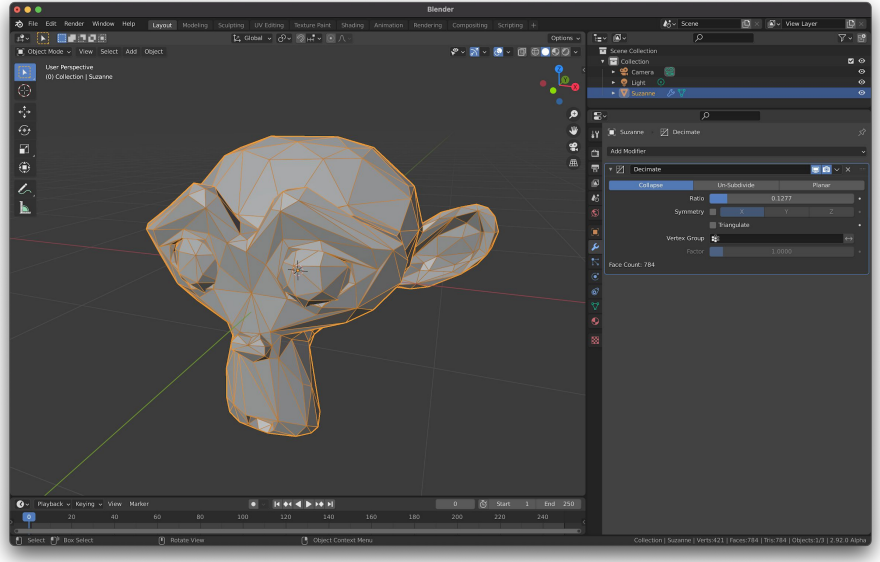
Session 5: Remeshing

- Motivation
- Mesh Sampling
  - Up-sampling: Subdivision
  - Down-sampling: Simplification
  - Re-sampling: Redescritization
- Summary
- Discussion

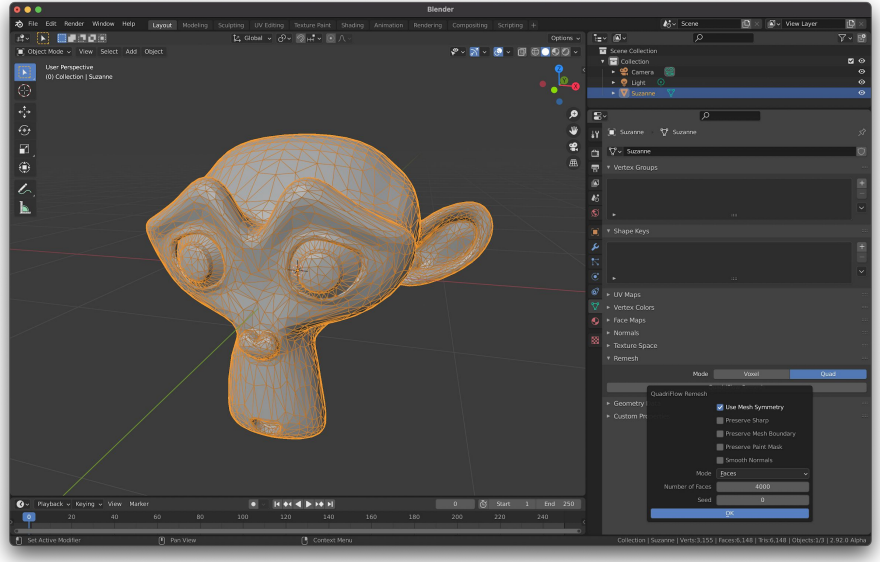
Simplification and Retopology in Blender

https://developer.blender.org/diffusion/B/browse/master/source/blender/bmesh/tools/bmesh_decimate_collapse.c;c2a01a6c118e$1290

https://developer.blender.org/diffusion/B/browse/master/source/blender/editors/object/object_remesh.c;c2a01a6c118e$816



Modifier > Decimate



Object Data > Remesh

Other Softwares Today

Open source softwares:

Instant-meshes

Quadric Flow (integrated in Blender)

Commercial softwares: Zremesher/Quadremesher/...