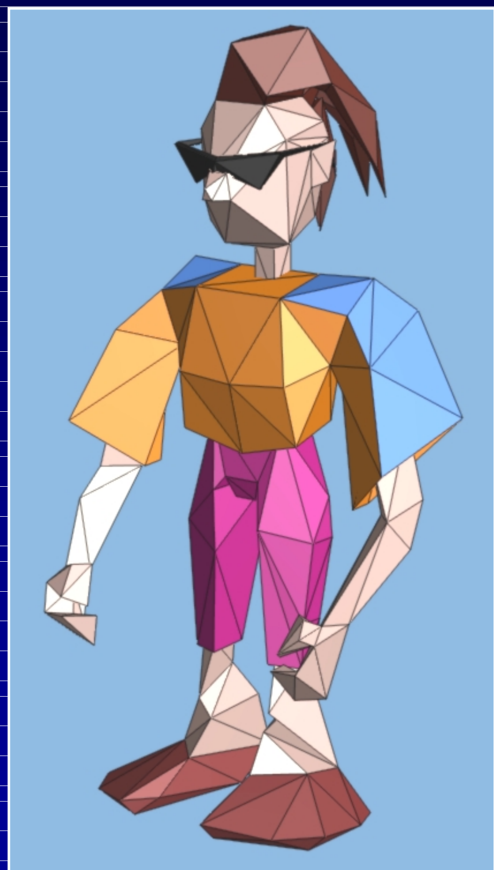
Progressive Meshes and Recent Extensions

Hugues Hoppe
Microsoft Research

SIGGRAPH 97 Course

Multiresolution Surface Modeling

Meshes in computer graphics



mesh M

V	F

Vertex 1 x_1 y_1 z_1	Face 1 2 3
Vertex 2 x_2 y_2 z_2	Face 3 2 4
…	Face 4 2 7
	…

(appearance attributes:
normals, colors, textures, ...)

Complex meshes

Challenges:

- rendering
- storage
- transmission

Talk outline

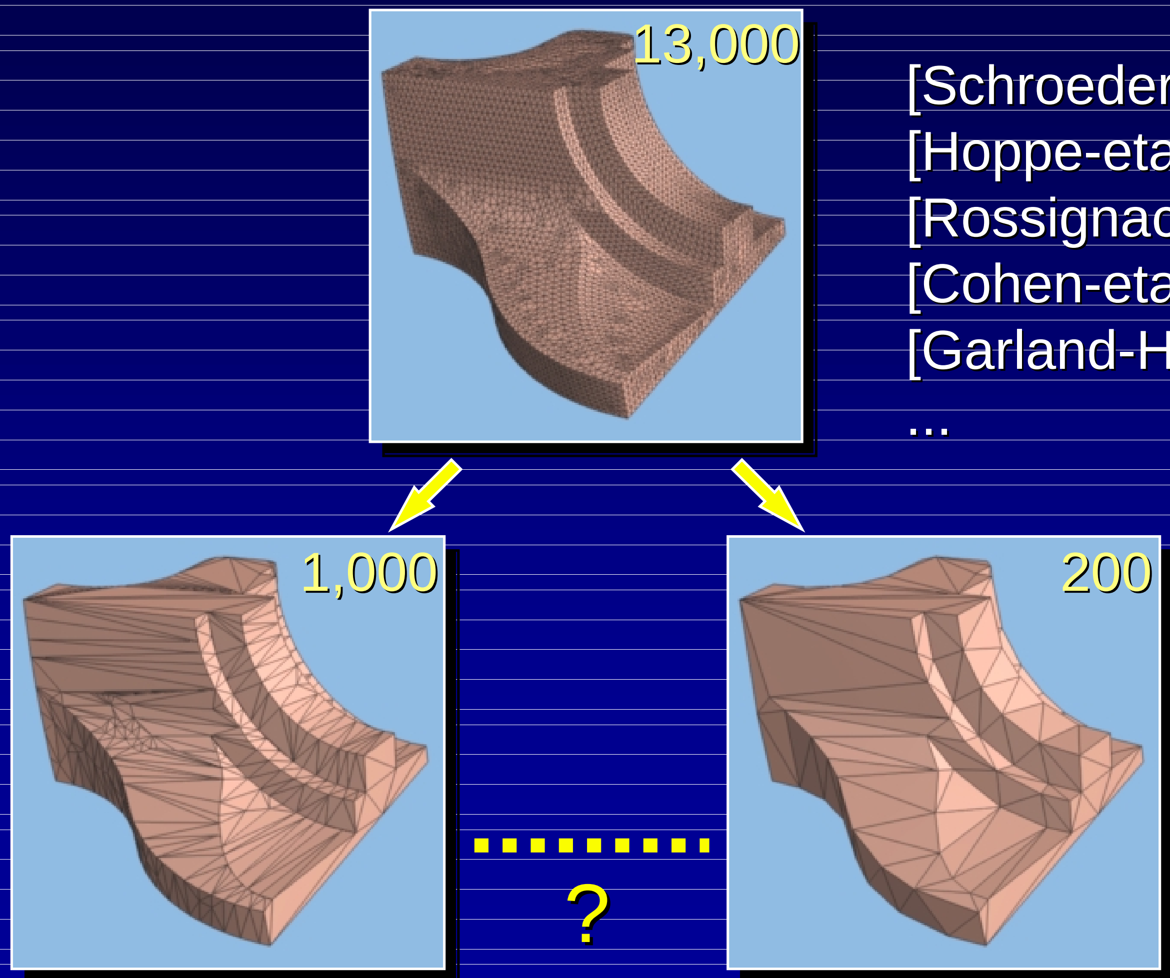
SIGGRAPH

’96 • Progressive mesh (PM) representation
  - continuous-resolution
  - efficient
  - progressive transmission

’97 • View-dependent refinement of PM’s

’97 • Progressive simplicial complex (PSC) repr.

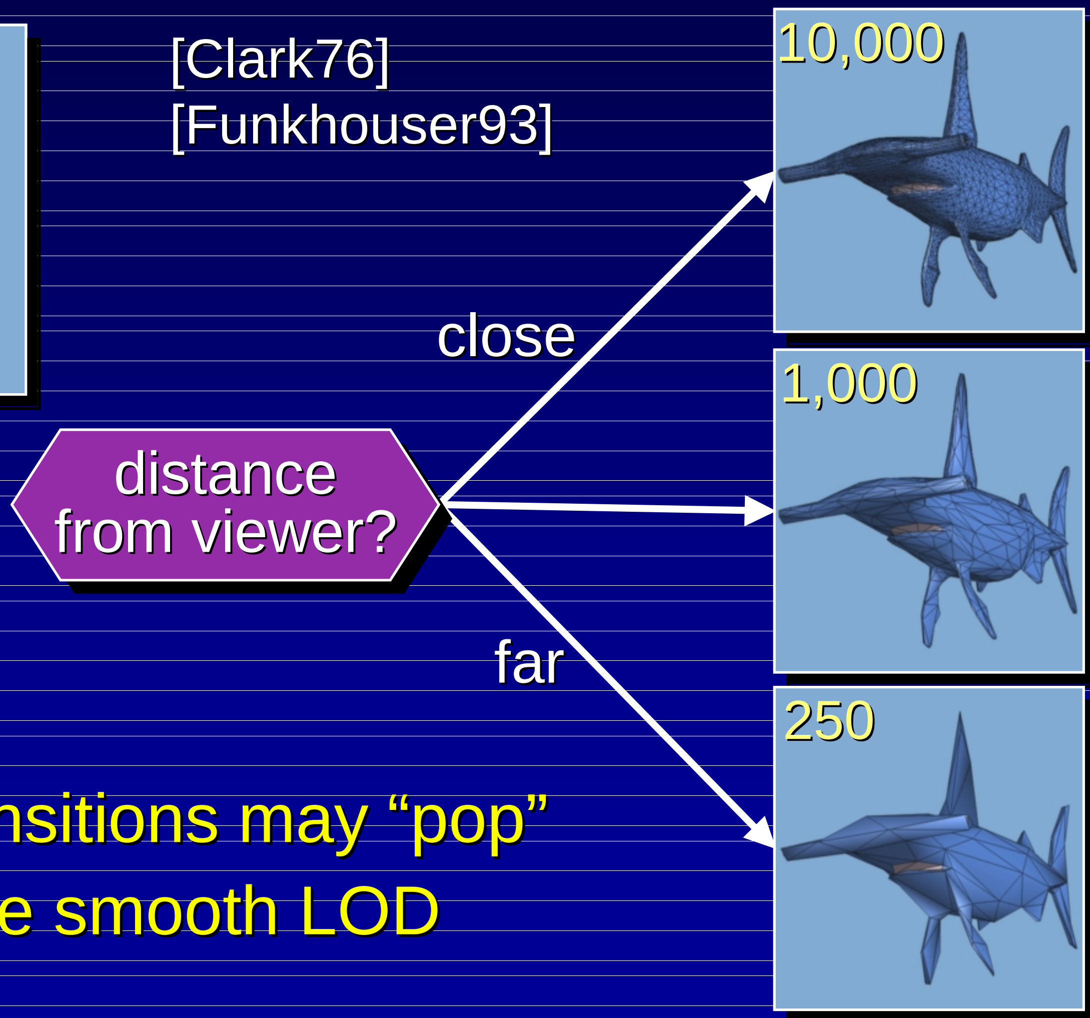
Mesh simplification techniques



[Schroeder-etal92]
[Hoppe-etal93]
[Rossignac-Borrel93]
[Cohen-etal96]
[Garland-Heckbert97]
...

Traditional level-of-detail (LOD)

[Clark76]
[Funkhouser93]



Concern: transitions may “pop”

→ would like smooth LOD

Progressive mesh representation

Basic idea:

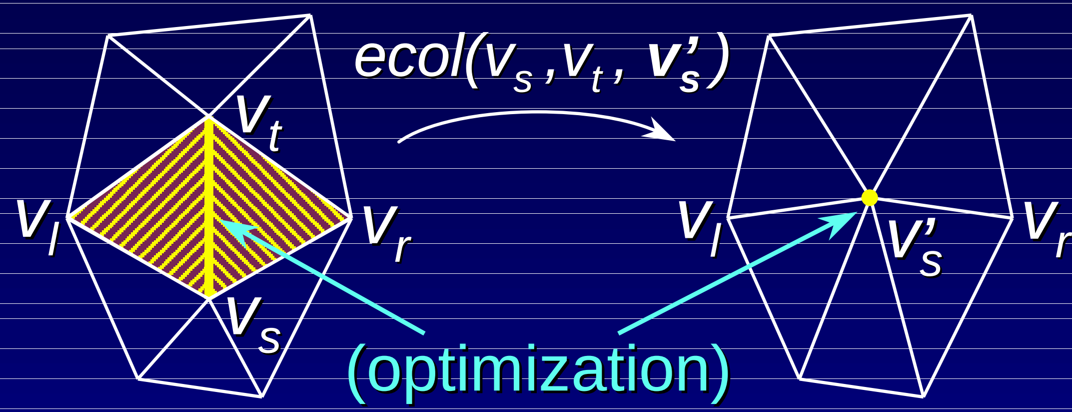
- Simplify mesh through sequence of transformations.
- Record:

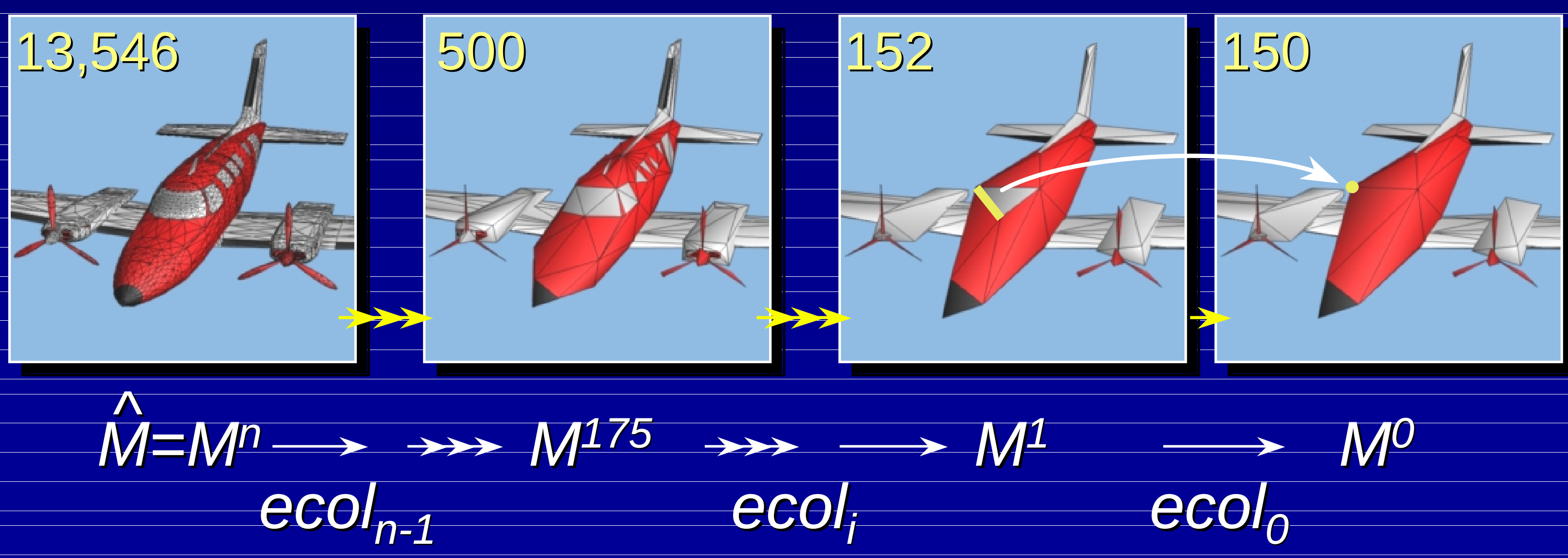
simplified mesh

\+

sequence of inverse transformations

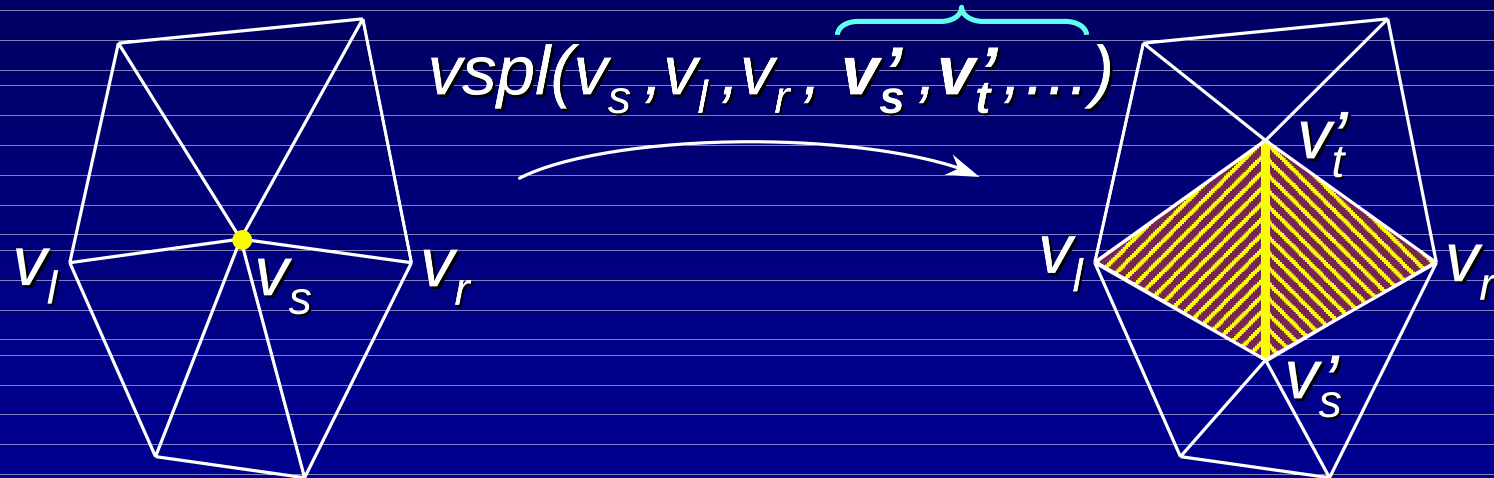
Edge collapse → Simplification



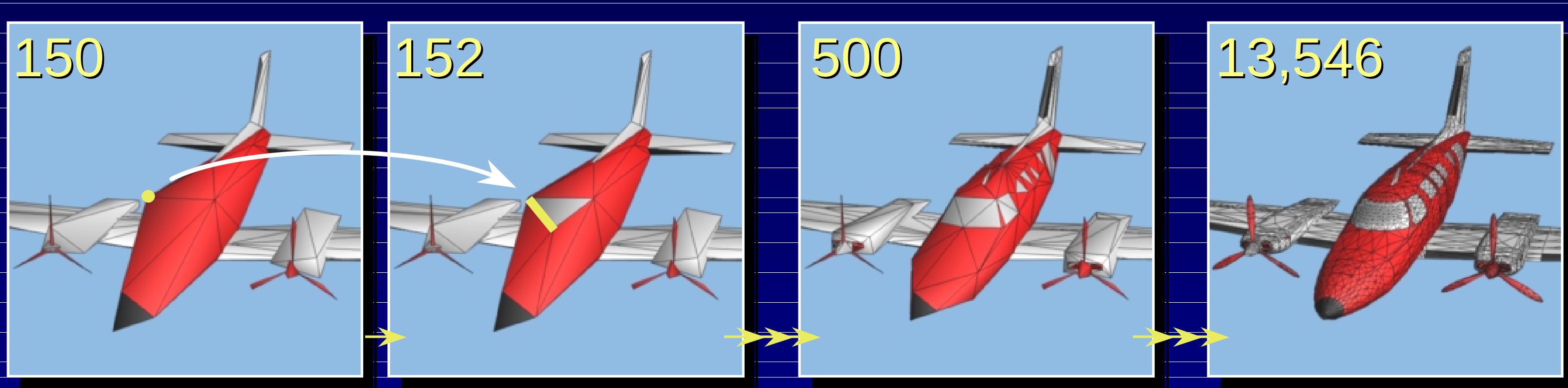


Invertible! Vertex split transformation

$$vspl(v_s, v_l, v_r, \underbrace{\mathbf{v'_s}, \mathbf{v'_t}, \ldots}_{\textit{attributes}})$$



Reconstruction process



$M^0 \longrightarrow M^1 \longrightarrow \cdots \longrightarrow M^{175} \longrightarrow \cdots \longrightarrow M^n = \hat{M}$

$vspl_0$ … $vspl_i$ … $vspl_{n-1}$

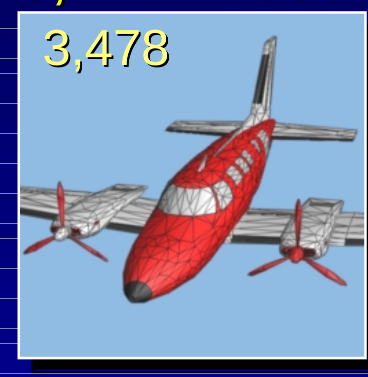
progressive mesh (PM) representation

Application: Continuous-resolution LOD

From PM, extract M^i of any desired complexity.

$\underbrace{M^0 \; vspl_0 \; vspl_1 \rightarrow\rightarrow\rightarrow \; vspl_{i-1}}_{M^i} \rightarrow\rightarrow\rightarrow \; vspl_{n-1}$

3,478 faces?



$M^0 \longleftrightarrow M^n = \hat{M}$

M^i

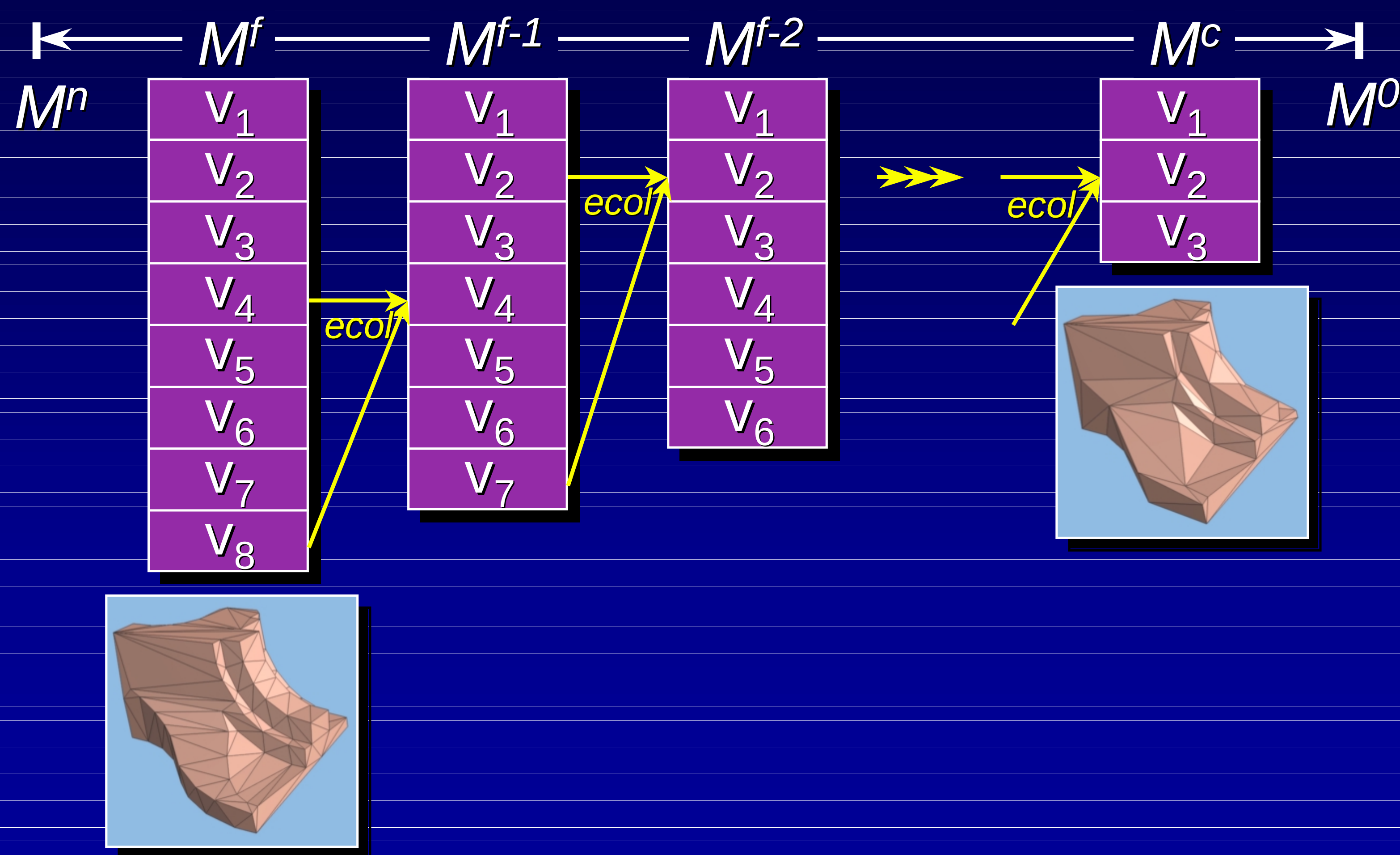
260K faces/sec! 180K faces/sec!

(200 MHz Pentium Pro)

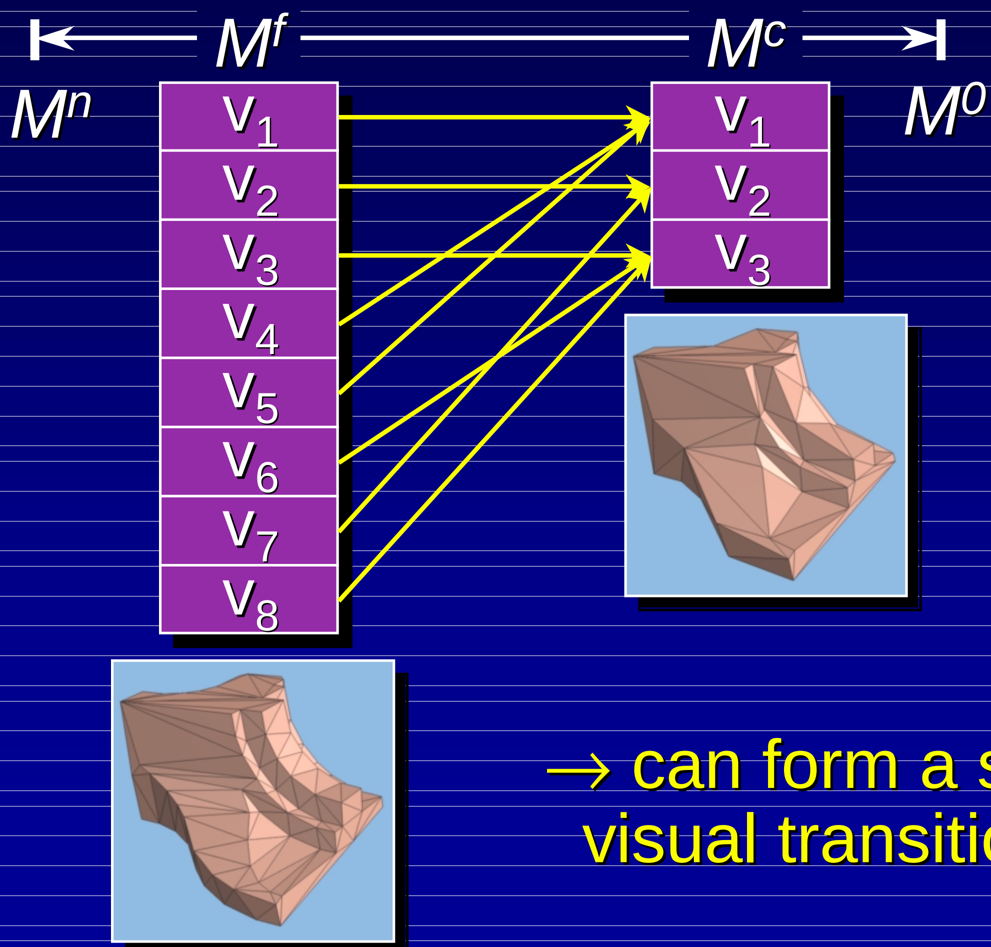
VIDEO: PM construction and LOD

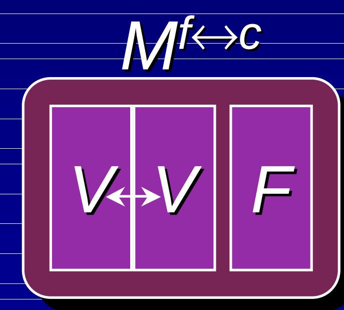
gameguy ecol’s
gameguy vspl’s
gameguy LOD
sandal LOD

Property: Vertex correspondence



Application: Smooth transitions



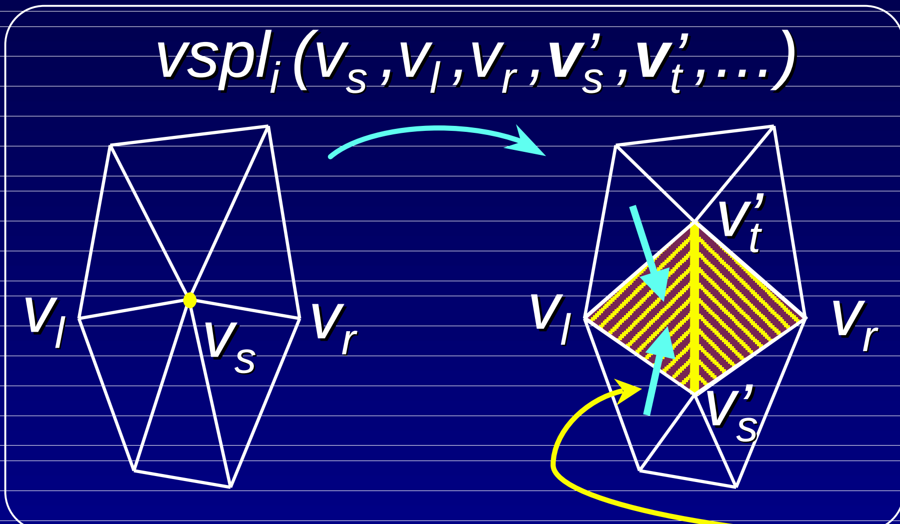


→ can form a smooth visual transition: *geomorph*

VIDEO: PM geomorphs

gameguy 1 geomorph
gameguy 12 geomorphs
gameguy 6 rotating

Application: Mesh compression



Record:

- v_s ($\log_2 i$ bits)
- v_l & v_r (~5 bits)
- $\mathbf{v}_t' - \mathbf{v}_s$ (delta)
- $\mathbf{v}_s' - \mathbf{v}_s$ (delta)
- predict materials
- …

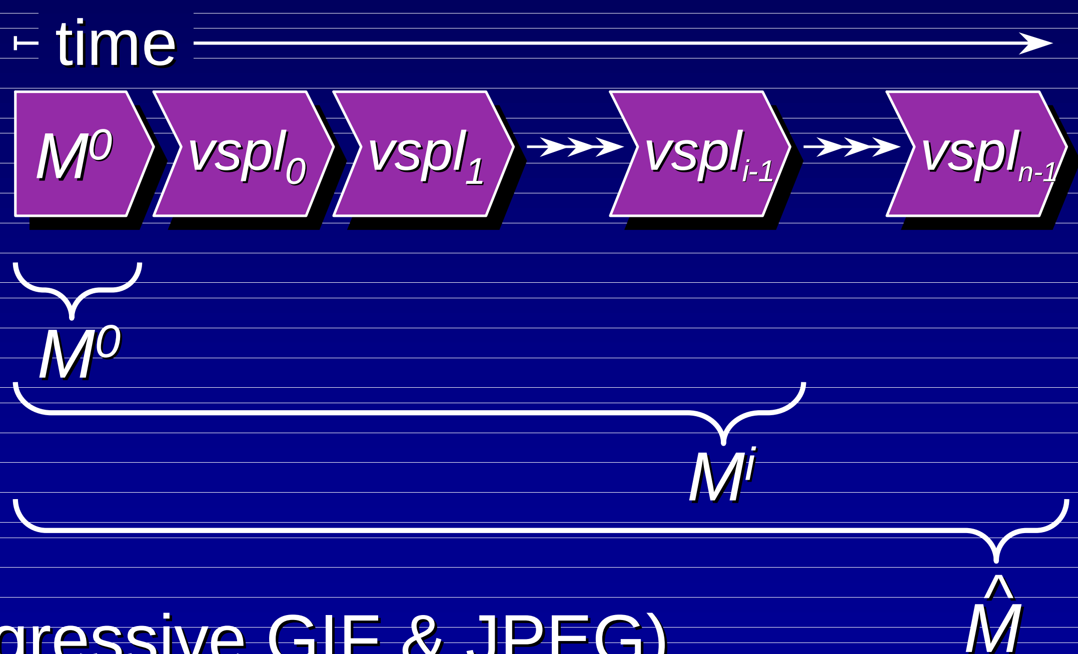
Analysis:

- connectivity: $n(4+\log_2 n)$ bits vs. $n(6\log_2 n)$ bits
- geometry: $\sim 30n$ bits vs. $96n$ bits

[Deering95]

Application: Progressive transmission

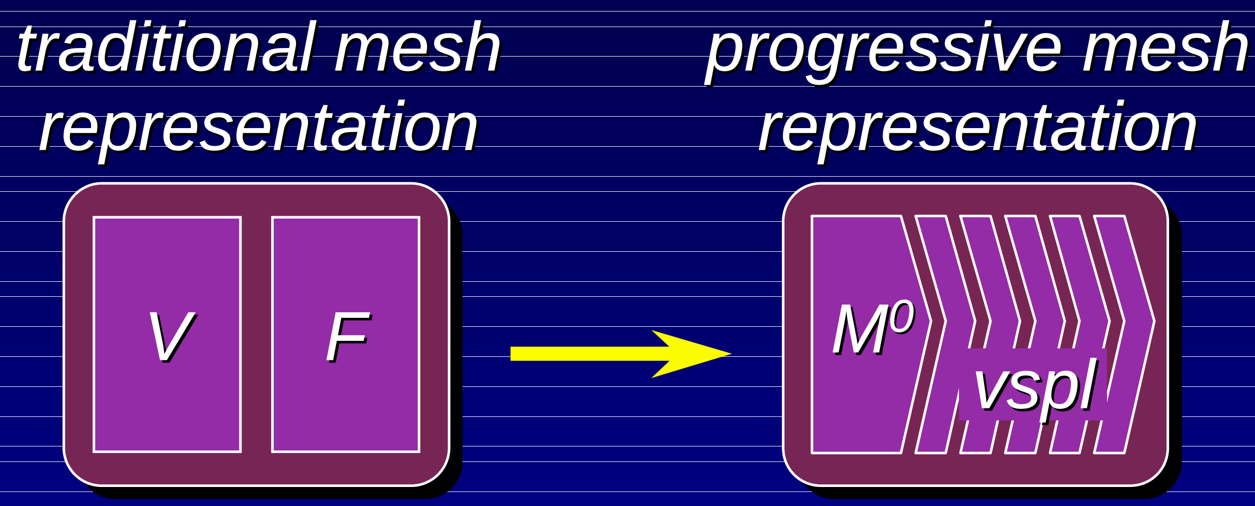
Transmit records progressively:



Receiver displays:

(~ progressive GIF & JPEG)

Conversion

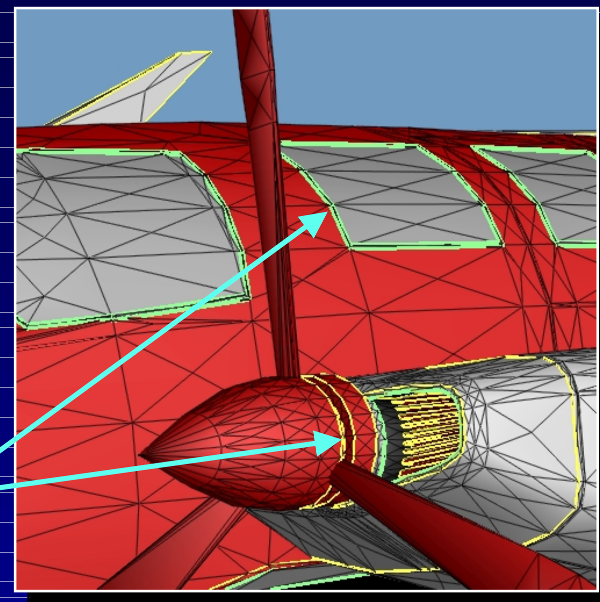


Optimization process

- Various metrics (speed vs. quality)
- Typically performed off-line

How to select edge collapses?

- Preserve *appearance*:
  - geometric shape
  - scalar fields
    (e.g. color, normals)
  - discontinuity curves



$$E = \underbrace{\cancel{\int}}_{\text{points}}^{\Sigma}(e_{shape} + e_{scalars})dA + \underbrace{\cancel{\int}}_{\text{points}}^{\Sigma}(e_{disc})dL$$

Error metric: point sampling

1600 faces

300 faces

Selecting edge collapses

- Greedy algorithm: always collapse edge resulting in smallest ΔE.
- Optimize position and attributes of resulting vertex.

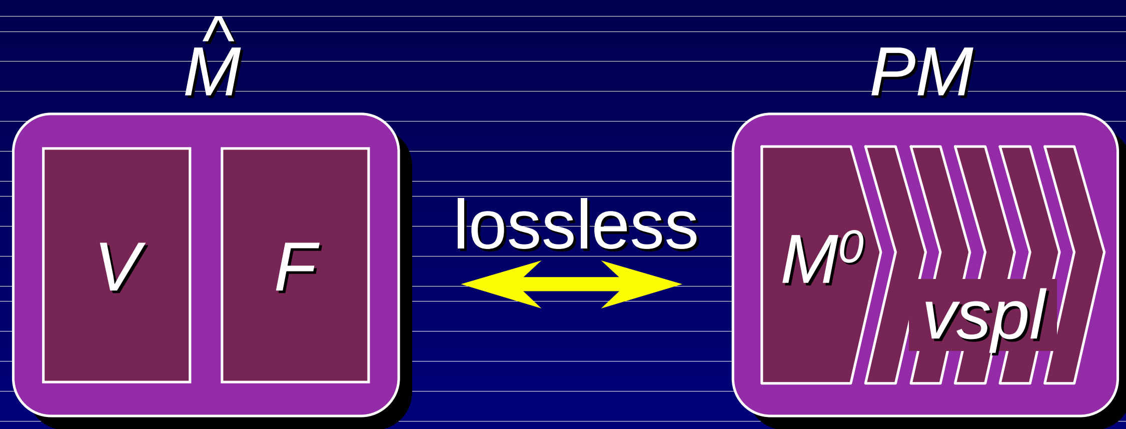
Simplification rates: ~ 30 faces/sec

- off-line process
- could use faster, simpler metrics e.g. [Garland-Heckbert97]

VIDEO/DEMO: PM results

radiosity 6 geomorphs
mandrill image

PM Summary



- single resolution

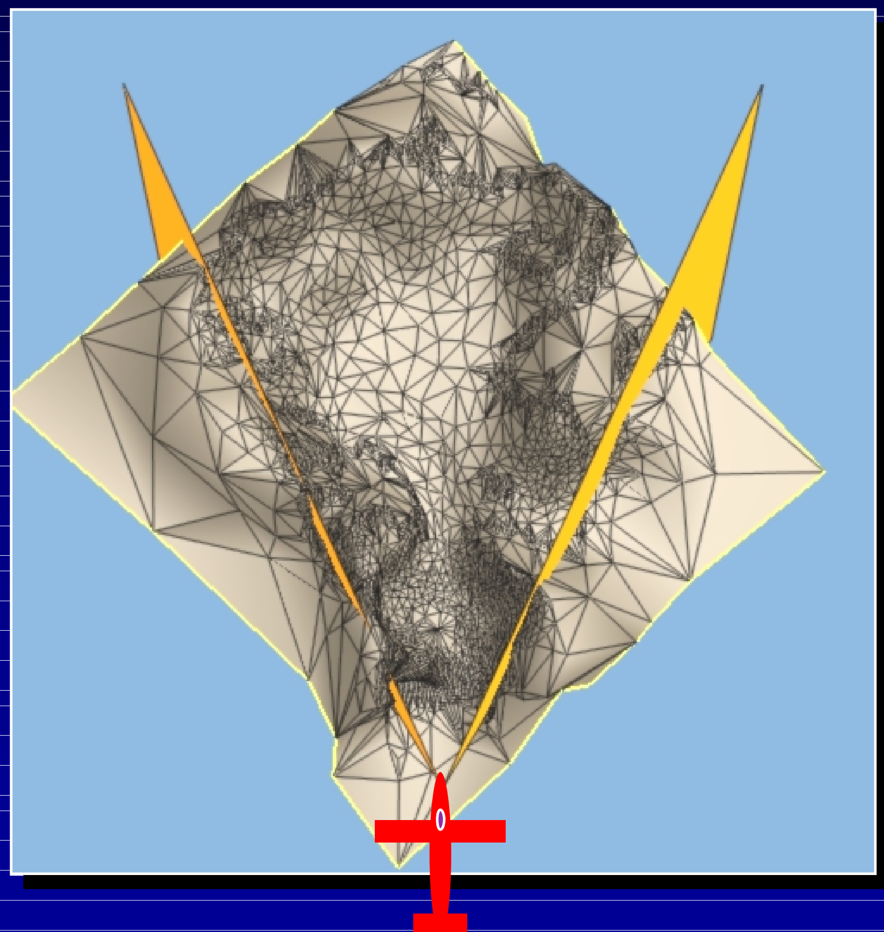
- continuous-resolution
- smooth LOD
- space-efficient
- progressive

[Microsoft DirectX 5.0]

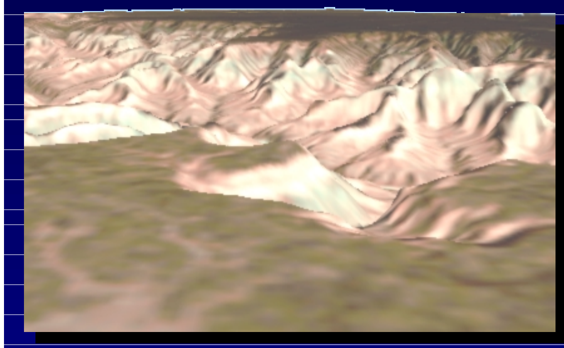
View-Dependent Refinement of Progressive Meshes

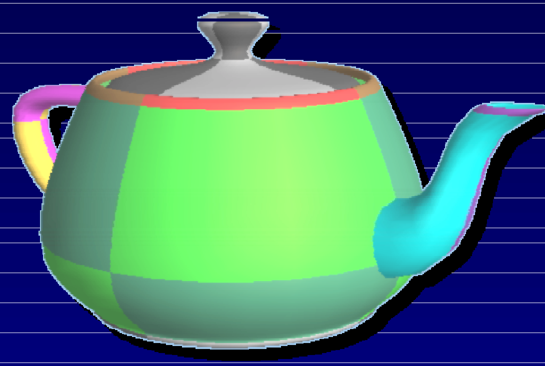
[SIGGRAPH 97]

Adaptive refinement: motivation



Applications & Related work

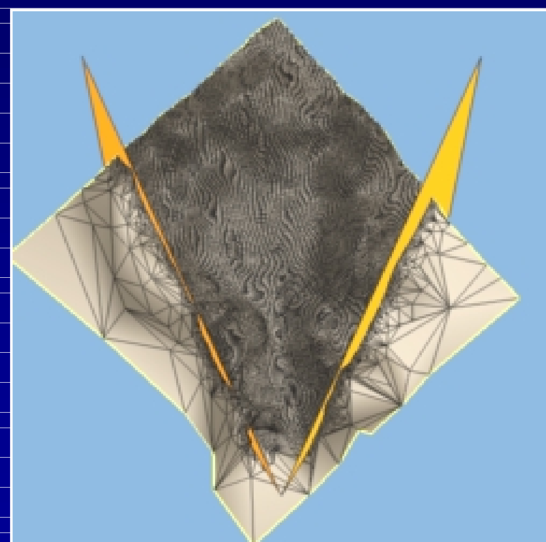


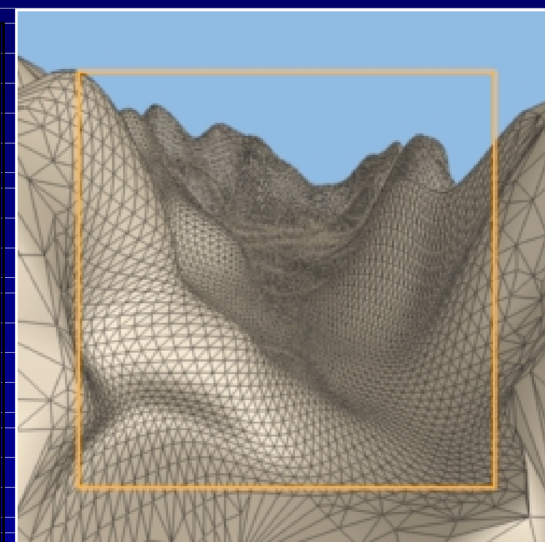


height fields	parametric surfaces	arbitrary meshes
[Cignoni *etal* 95]	[Rockwood *etal* 89]	[Xia-Varshney 96]
[De Floriani *etal* 96]	[Abi Ezzi *etal* 93]	
[Lindstrom *etal* 96]	[Kumar *etal* 95]	
…	…	

Using progressive meshes

M^0 $\cancel{vspl_0}$ $vspl_1$ → → → $\cancel{vspl_{i-1}}$ → → → $vspl_{n-1}$



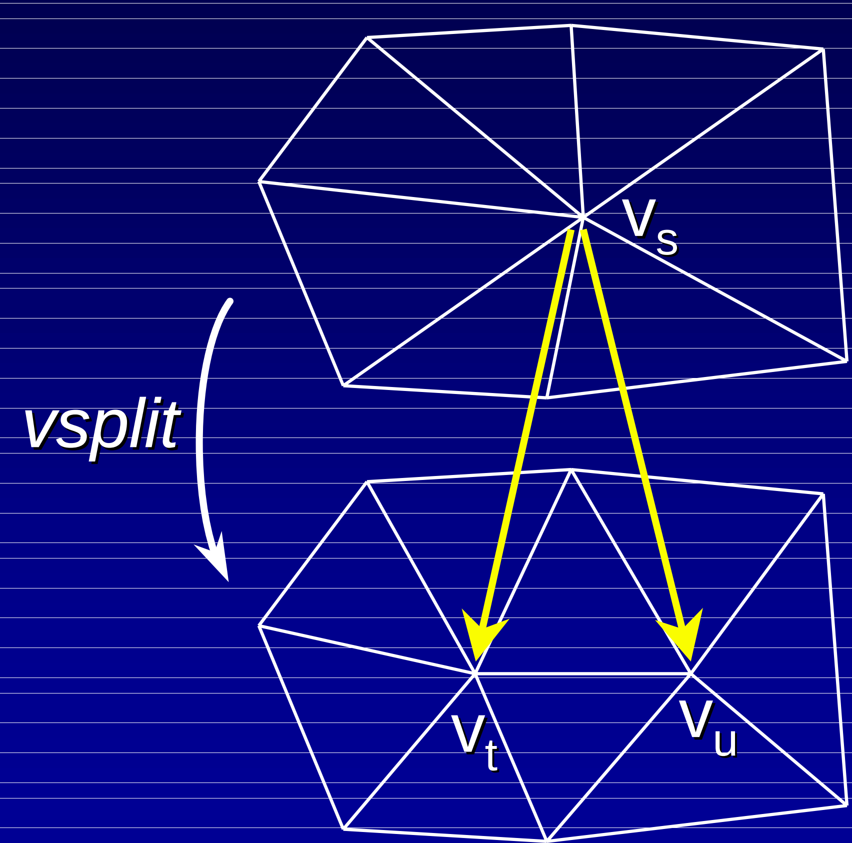


(e.g. view frustum)

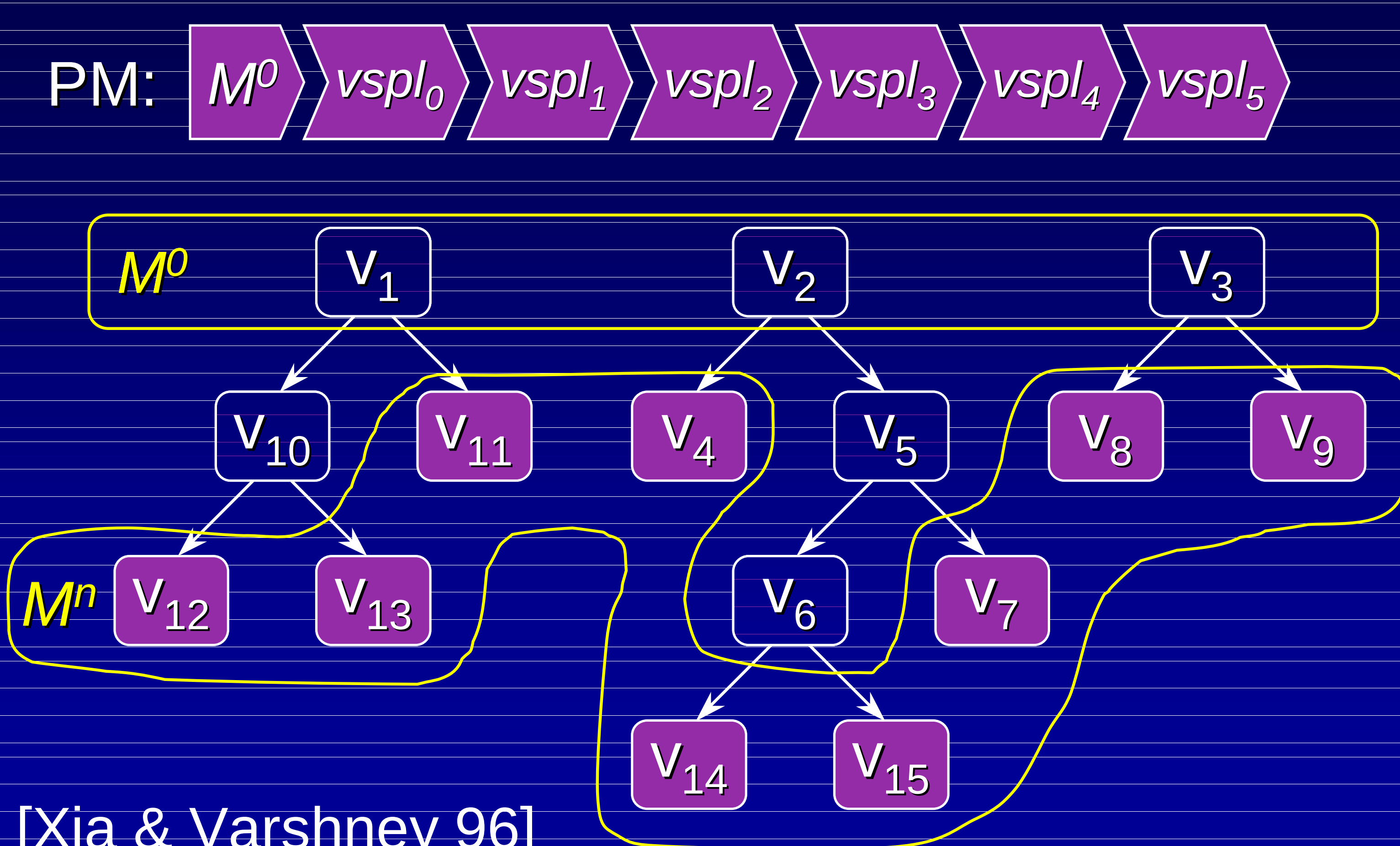
Contributions

- PM → vertex hierarchy
  → selective refinement
- Dependencies → consistent framework
- View-dependent refinement criteria

Parent-child vertex relations

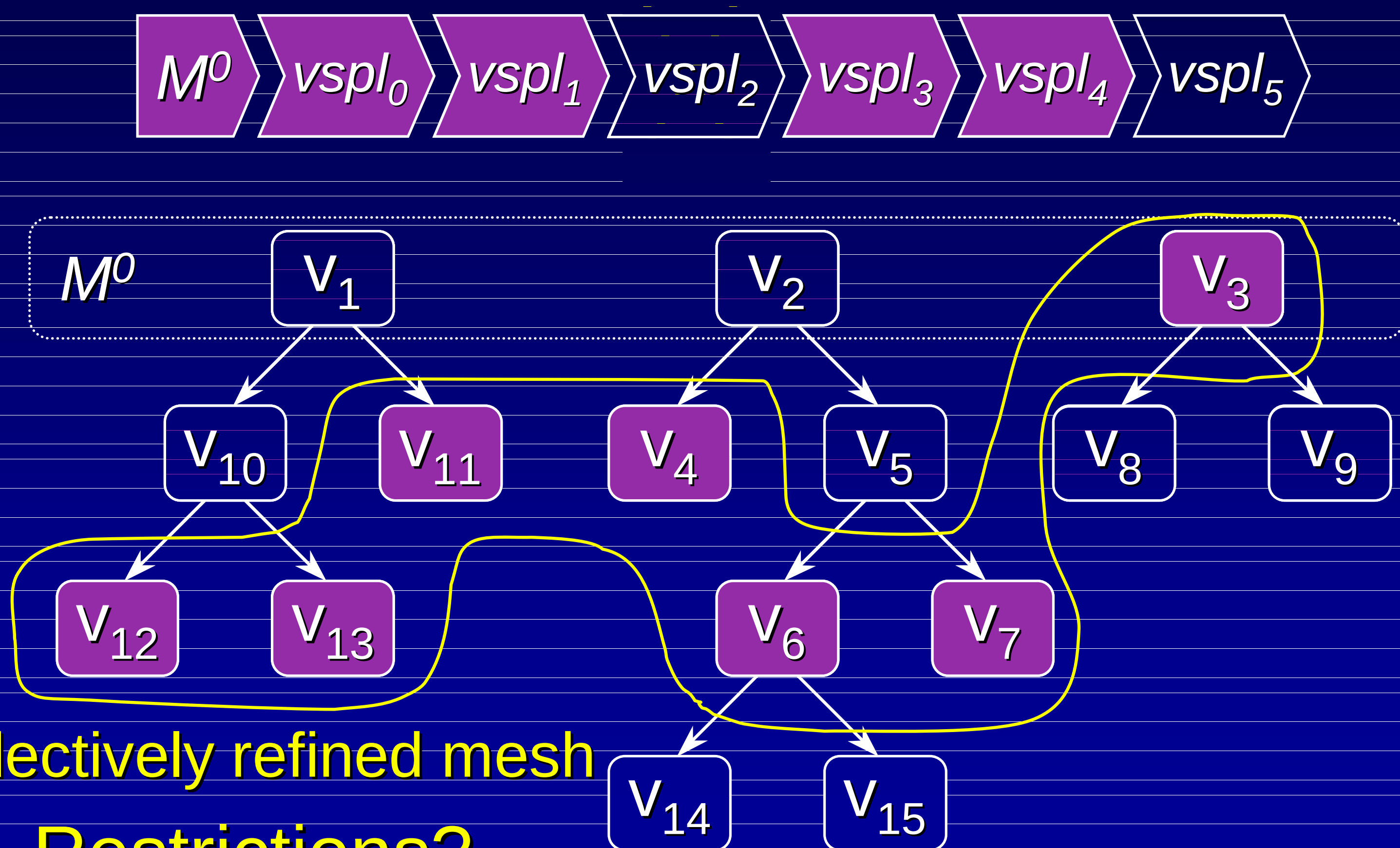


Vertex hierarchy



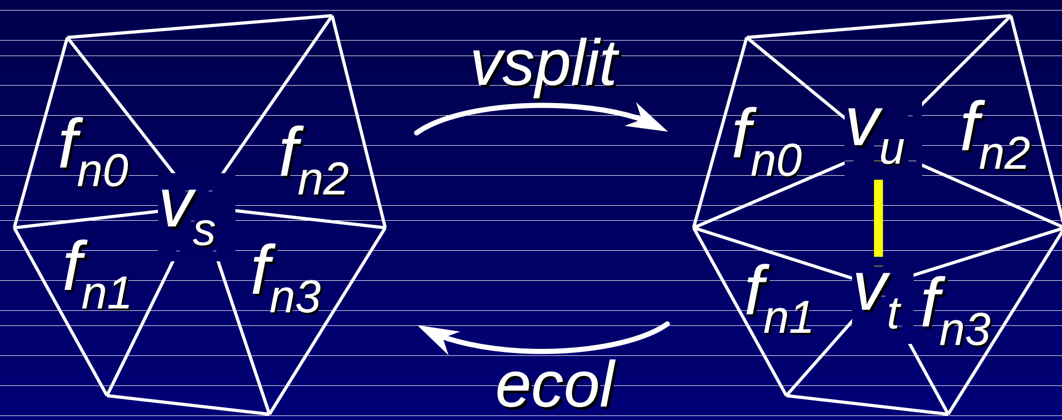
[Xia & Varshney 96]

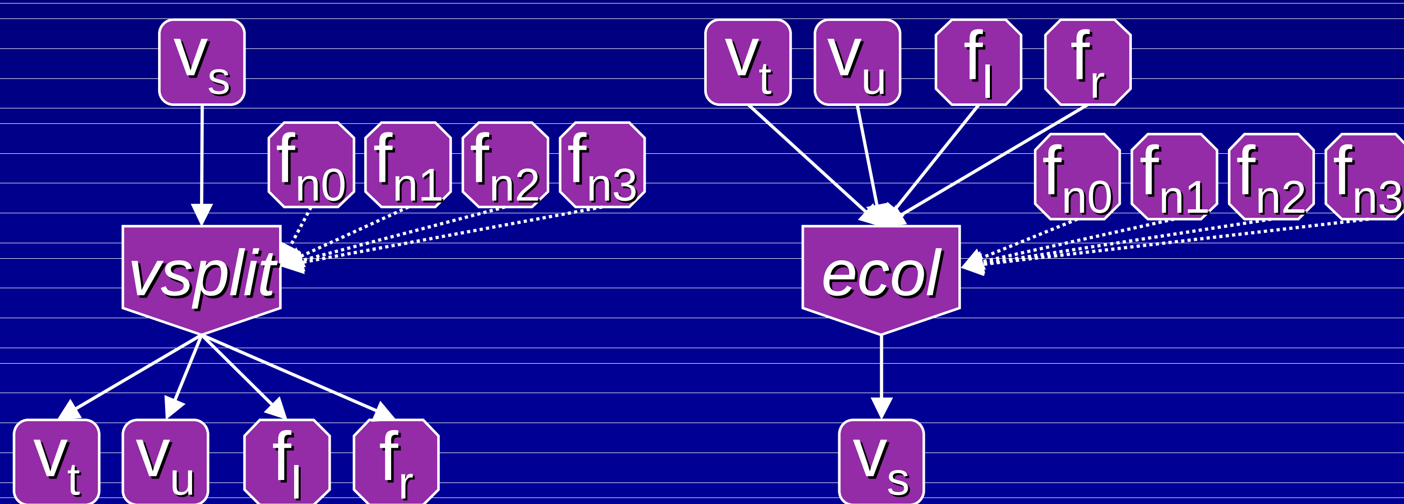
Selective refinement



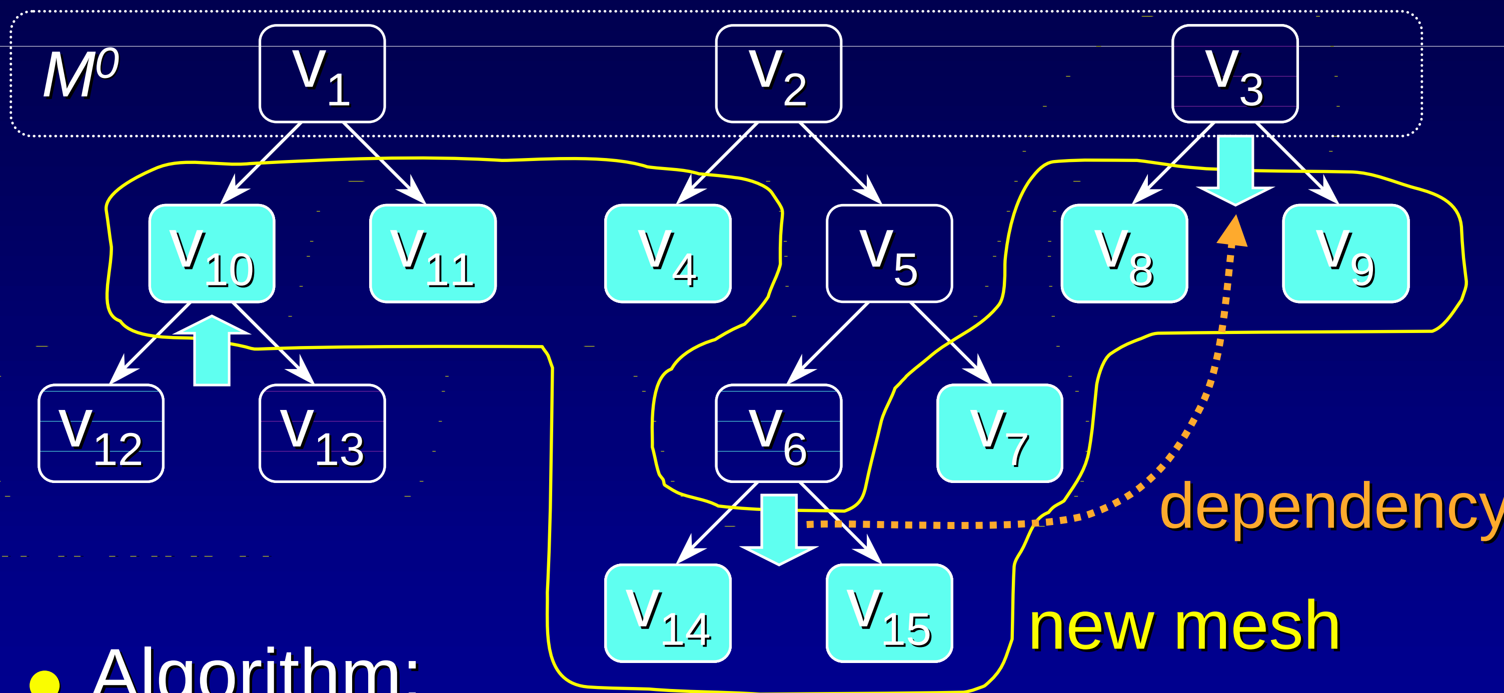
Restrictions?

New vspl/ecol parametrizations





Runtime algorithm



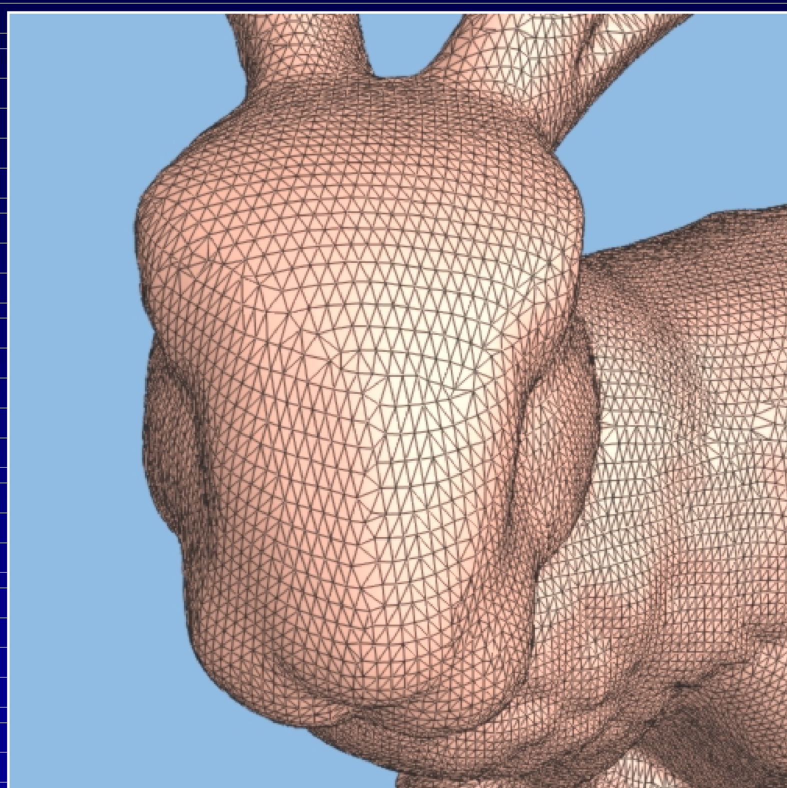
- Algorithm:
  - incremental (frame coherence)
  - efficient (~15% of frame time)
  - amortizable

View-dependent refinement criteria

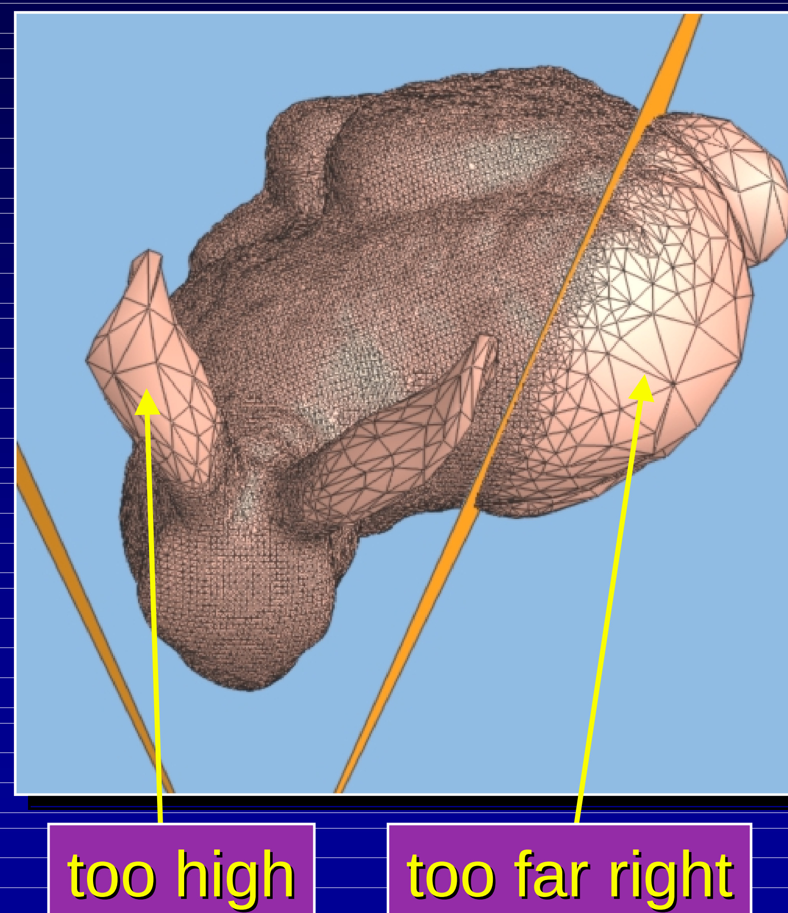
<u>3 criteria:</u>

- view frustum
- surface orientation
- screen-space geometric error

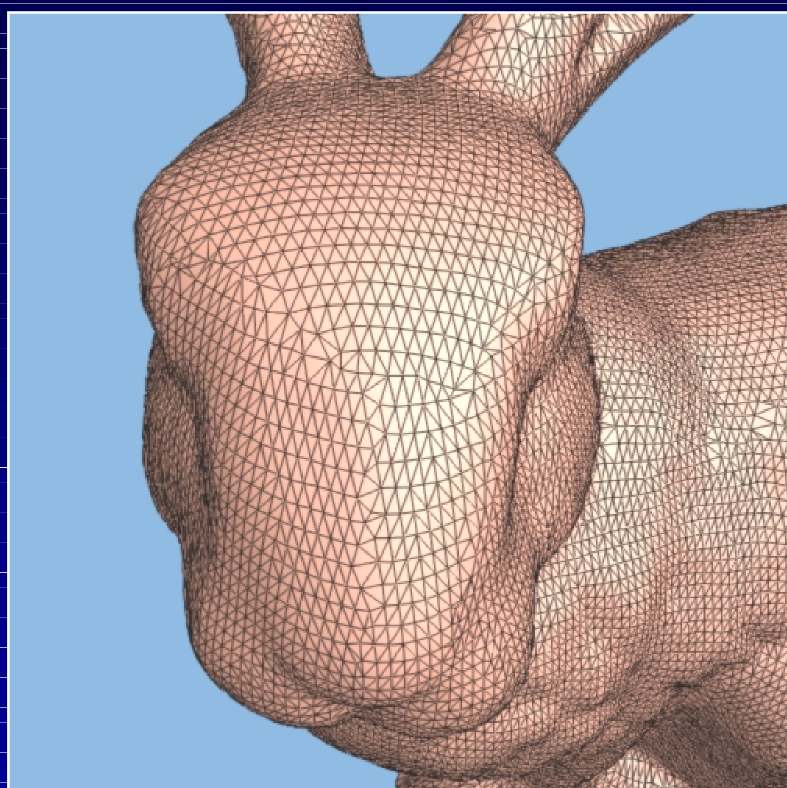
View frustum

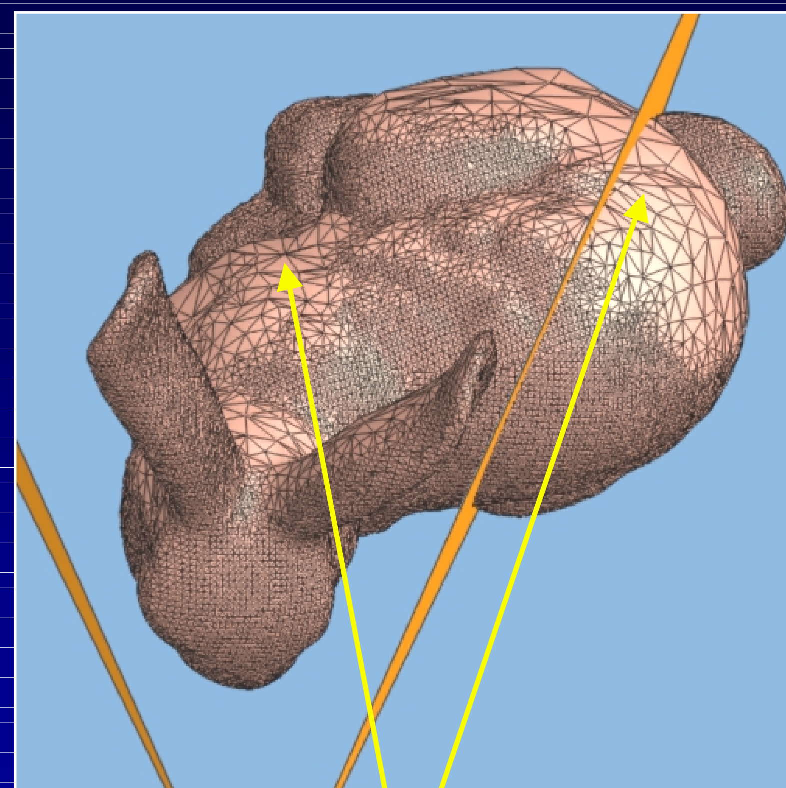


view is unchanged



Surface orientation



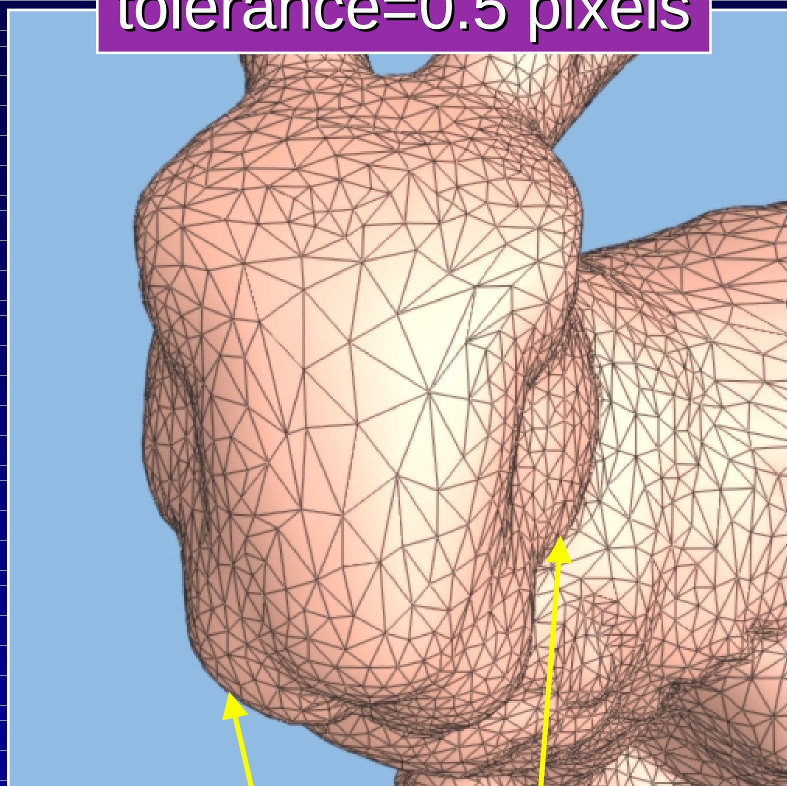


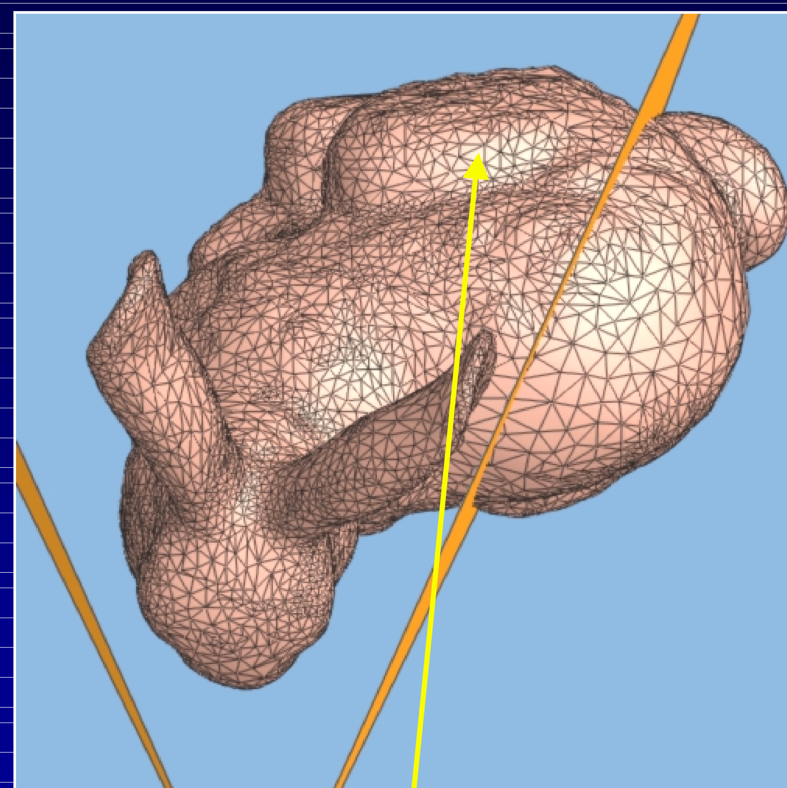
view is unchanged

oriented away

Screen-space geometric error

tolerance=0.5 pixels

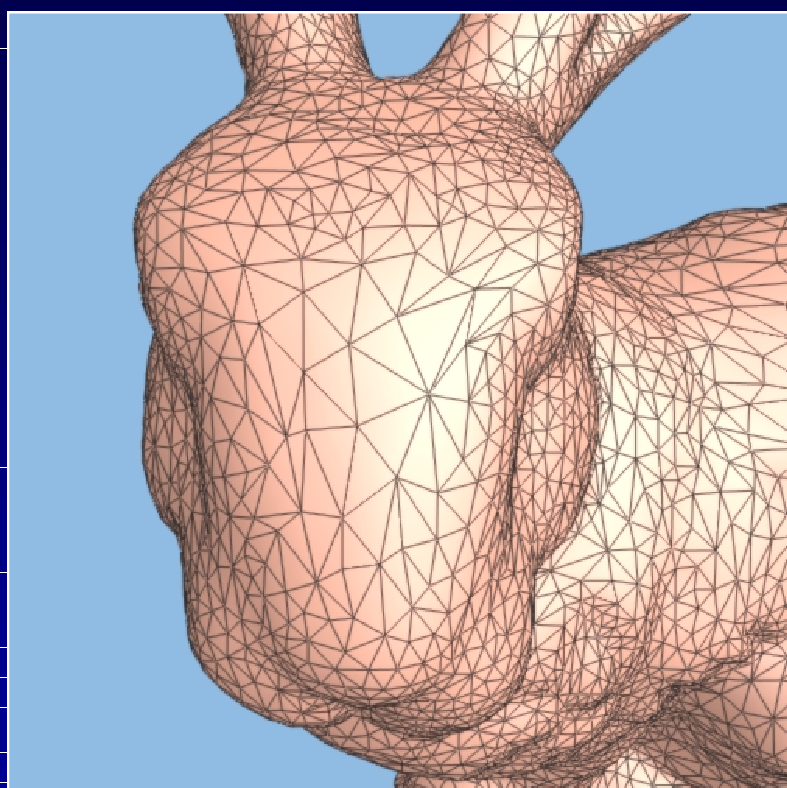


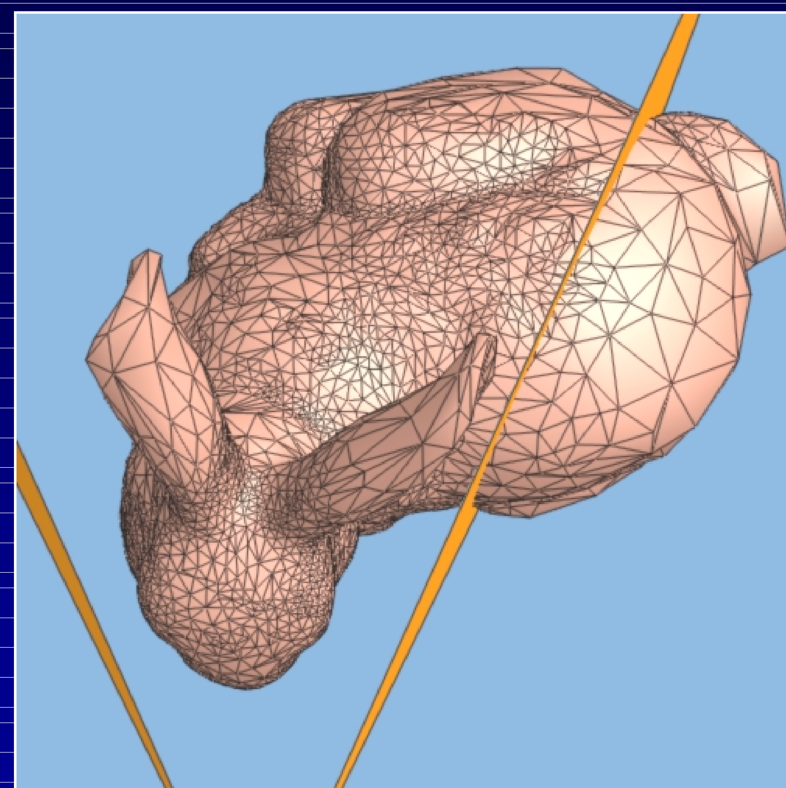


refinement near silhouette

coarser in distance

All three criteria together



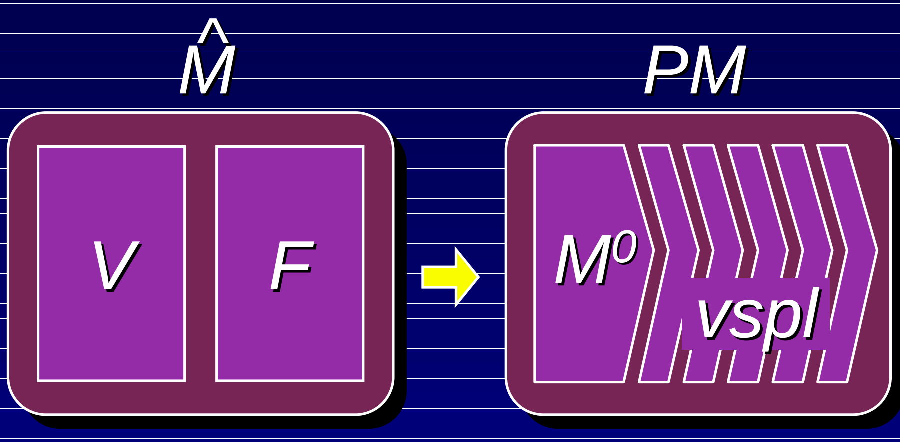


69,473 faces → 10,528 faces
1.9 frame/sec → 6.7 frame/sec

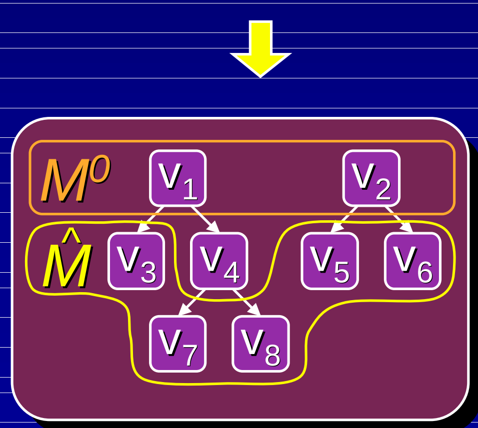
VIDEO: Selective Refinement

view frustum (2-window terrain)
screen-space error (2-window terrain)
silhouette refinement (sphere)
all 3 criteria including orientation (teapot)
arbitrary mesh (gameguy)
geomorph
flythrough of gcanyon

Selective Refinement Summary



- continuous-resolution
- smooth LOD
- space-efficient
- progressive



- view-dependent refinement
- real-time algorithm

Progressive Simplicial Complexes

[SIGGRAPH 97]

(Joint work with Jovan Popovic)

PM restrictions:

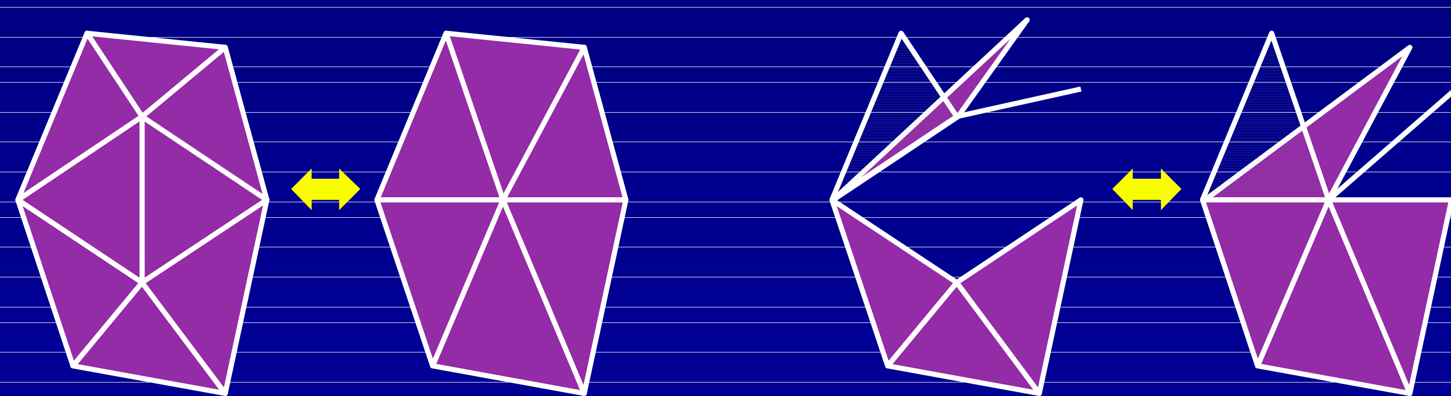
- Supports only "meshes"
  (orientable, 2-dimensional manifolds)
- Preserves topological type

Progressive Simplicial Complexes

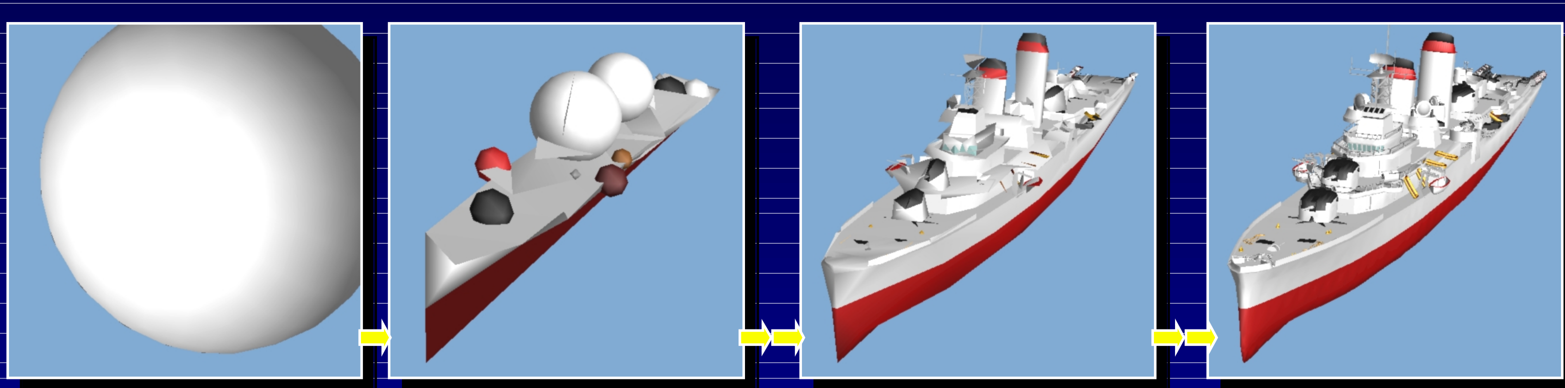
- Represent arbitrary “triangulations”:
  - any dimension,
  - non-orientable,
  - non-manifold,
  - non-regular, …
- Progressively encode both geometry **and topology**.

Generalization

PM		*PSC*
edge collapse (*ecol*)	↔	vertex unification (*vunify*)
vertex split (*vspl*)	↔	generalized vertex split (*gvspl*)



LOD sequence



$M^1 \twoheadrightarrow M^{22} \longrightarrow \twoheadrightarrow M^{116} \twoheadrightarrow \longrightarrow M^n = \hat{M}$

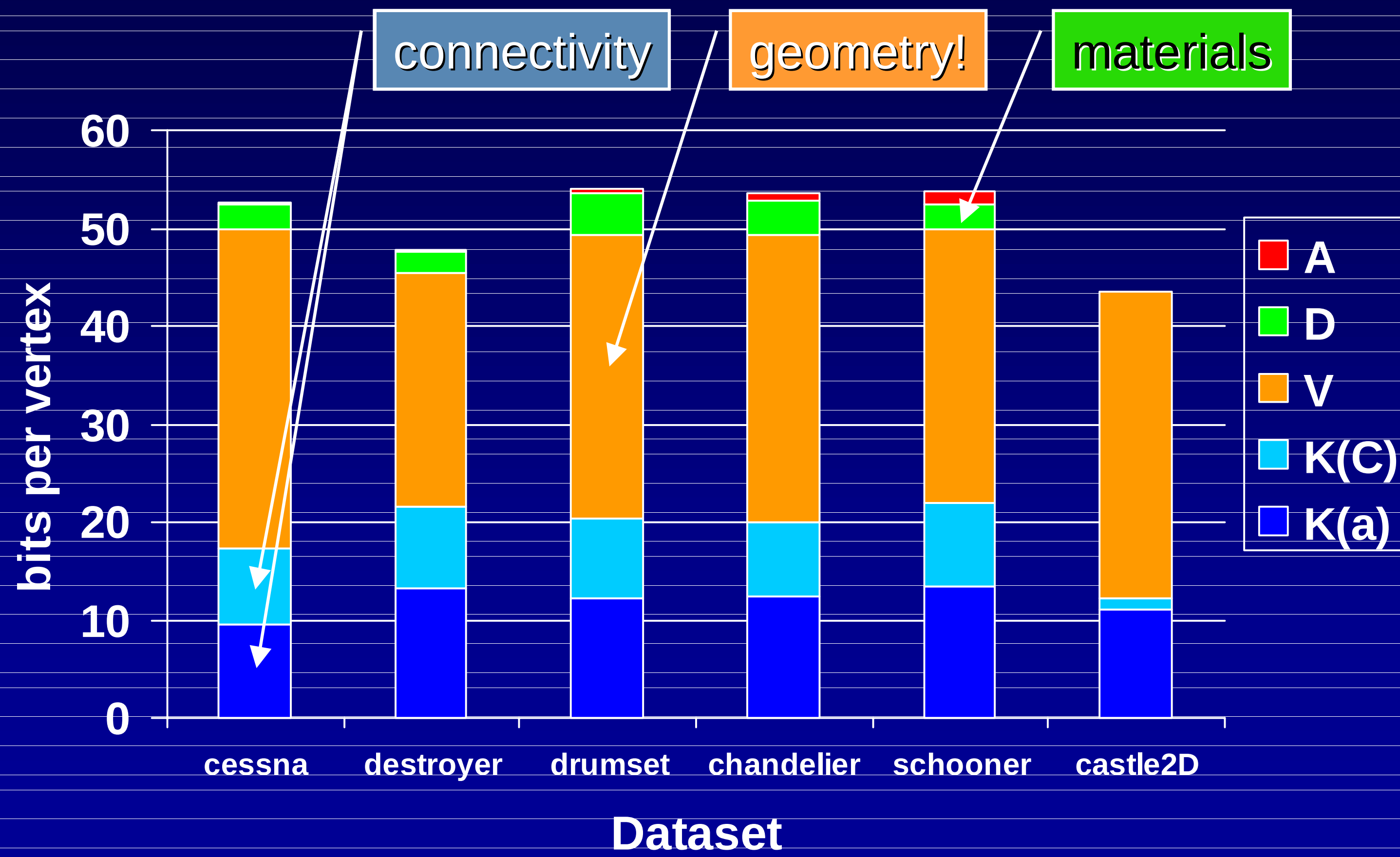
$gvspl_1$ … … $gvspl_i$ … $gvspl_{n-1}$

PSC representation

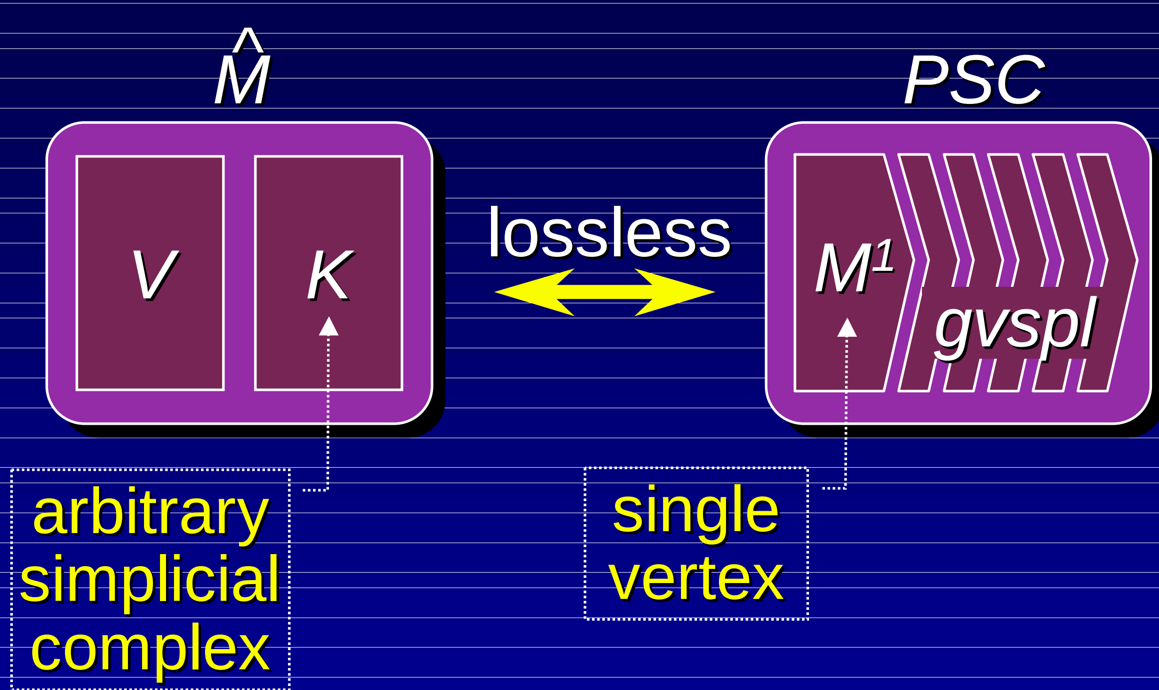
VIDEO: PSC

drumset
schooner
comparison of PM and PSC (davidson)
geomorphs (cessna)
1-d non-manifold curve (castle)

Space analysis



PSC Summary



- any triangulation
- progressive geometry and topology

Future Work

- LOD on volumes
- Memory management for large models
- Refinement criteria for surface shading
- Animated models
- Editing using PM/PSC