What is subdivision based representation?

Subdivision Surfaces
What is so special?

- Multi-resolution (Scalability)
- One piece representation™ (arbitrary topology)
- Numerical stability
- Code Simplicity
- Covers both polygon form and surface form (Uniformity)
One piece representation™
Multi-resolution (Scalability)
Covers both polygon form and surface form

(Uniformity of representation)
So, just what is a subdivision surface?

- Catmull-Clark
- Quadrilateral
- Doo-Sabin
- Triangular

Loop

Butterfly
Basic Concept (Catmull-Clark Scheme):

- : vertices from mesh $M^0$
- , , : vertices to be generated for $M^1$

Around a vertex $v$ of degree 5
Basic Concept (Catmull-Clark Scheme):

Generating new face points
Face point: centroid of each face
Basic Concept (Catmull-Clark Scheme):

Generating new edge points

\[ e_i^1 = \frac{v^0 + e_i^0 + f_i^1 + f_{i-1}^1}{4} \]
Basic Concept (Catmull-Clark Scheme):

Generating new vertex points

\[ v^1 = \frac{n-2}{n} v^0 + \frac{1}{n^2} \sum e_i^0 + \frac{1}{n^2} \sum f_i^1 \]

: vertex point
Basic Concept (Catmull-Clark Scheme):

Forming new edges

\[ \begin{align*}
  f_1 & : v_0 \\
  f_2 & : v_1 \\
  f_3 & : v_2 \\
  f_4 & : v_3 \\
  f_5 & : v_4 \\
  e_1 & : f_1, f_2 \\
  e_2 & : f_2, f_3 \\
  e_3 & : f_3, f_4 \\
  e_4 & : f_4, f_5 \\
  e_5 & : f_5, f_1 \\
\end{align*} \]

\( v_0 \) : vertex point
Repeatedly refining the control meshes, one gets $M^0, M^1, M^2, M^3, \ldots \rightarrow$ limit surface (subdivision surface)
Modeling made much easier. Why?

- No restrictions on the topology of the control points
- Local refinement is possible
Example of control meshes of Catmull-Clark subdivision surfaces
Can model any kind of special features (by modifying the subdivision rules)
Most importantly, can represent any shape with just one surface

(One piece representation™)

One Piece

Solid Modeling

Multi-Piece
Is One Piece Representation™ Good?

Data Management: Simpler

Rendering: More efficient

Machining: More precise

Animation: Crack free
Does this mean the **solid modeling** area is no longer needed?
What is subdivision based representation?

Subdivision Surfaces

CAD/CAM
What is missing?

1. No parameterization
2. No error control
3. No adaptive tessellation
• Without error control
  No CAD/CAM applications

• Without parameterization
  Difficult to perform picking, rendering, texture mapping

• Without adaptive tessellation
  Too expensive to use
A major **breakthrough** occurred in 1998

- Jos Stam
- Parameterization of Catmull-Clark Subdivision Surfaces
- 1998
Work on Subdivision Surface Parameterization

(Discrete Fourier Transform)

J. Stam  →  Lai/Cheng

Parameterization

The Extended Subdivision Diagram
Applications of the new parameterization technique

- Surface Evaluation
- Texture Mapping
- Boolean Operations
- Surface Trimming
- Adaptive Tessellation
- Animation
Surface Evaluation

Fast, Exact Rendering
Texture Mapping$^1$

1: Lai and Cheng, 2005
Texture Mapping\(^1\)
Texture Mapping\textsuperscript{1}

1: Lai and Cheng, 2005
Boolean Operations\textsuperscript{2}

2: Lai and Cheng, 2005
Surface Trimming$^2$

2: Lai and Cheng, 2005
Adaptive Tessellation

3: Lai and Cheng, 2005
What is error control?
Error Control: Given \( \varepsilon > 0 \), when would \( \| M^n - S \| < \varepsilon \)?
What metric should we use to assess \( \| M^n - S \| \) for an extra-ordinary patch?
A solution is finally available…

- F. Cheng, G. Chen, J. Yong
- Subdivision Depth Computation for Catmull-Clark Subdivision Surfaces
- 2005
This work is also important for adaptive subdivision\textsuperscript{5}.
Basic Idea: Use unbalanced subdivision\textsuperscript{6} to provide smooth transition between areas with different densities

Adaptive subdivision:
input: a piecewise surface P and a subdivision level assignment S
output: a triangular linear approximaiton P** of P

Three phases:
Phase 1: define a label for each vertex of P
Phase 2: generate a gradrilateral subdivision mesh P* of P
Phase 3: convert P* to a triangular linear approximation P** of P
Phase 1:
/* F \equiv \{ f \mid f \text{ is a patch of } P \} */
for each vertex v of P do
    L(v) := \max( \{1\} \cup \{ S(f) \mid f \subseteq F, v \text{ is a vertex of } f \} )
Phase 2:

1. for each vertex \( v \) of \( P \) do
   \[
   \text{LABEL}(v) := L(v);
   \]

2. for each patch \( f \) of \( P \) do
   \[
   \text{Subdivide}(f);
   \]
Subdivide\((f : \text{ quadrilateral surface patch})\);

\[
\begin{align*}
\text{if} & \quad (\text{LABEL}(v) > 0 \text{ for more than one vertex of } f) \\
\text{then} & \\
\text{balanced_sub}(f, f_1, f_2, f_3, f_4); \\
\text{for } i := 1 \text{ to } 4 & \text{ do} \\
\text{subdivide}(f_i); \\
\end{align*}
\]

\[
\begin{align*}
\text{else if} & \quad (\text{LABEL}(v) > 0 \text{ for only one vertex of } f) \\
\text{then} & \\
\text{unbalanced_sub}(f, f_1, f_2, f_3); \\
\text{for } i := 1 \text{ to } 3 & \text{ do} \\
\text{subdivide}(f_i); \\
\end{align*}
\]
balanced_sub(f):

Perform mid-point subdivision on f to get four new subpatches: \( r_1 r_2 r_3 r_4, s_1 s_2 s_3 s_4, t_1 t_2 t_3 t_4, q_1 q_2 q_3 q_4 \), and assign new labels as follows:

\[
\begin{align*}
LABEL(r_1) &= \max\{0, LABEL(v_1) - 1\} \\
LABEL(s_2) &= \max\{0, LABEL(v_2) - 1\} \\
LABEL(t_3) &= \max\{0, LABEL(v_3) - 1\} \\
LABEL(q_4) &= \max\{0, LABEL(v_4) - 1\} \\
LABEL(r_2) &= LABEL(s_1) = \min\{LABEL(r_1), LABEL(s_2)\} \\
LABEL(s_3) &= LABEL(t_2) = \min\{LABEL(s_2), LABEL(t_3)\} \\
LABEL(t_4) &= LABEL(q_3) = \min\{LABEL(t_3), LABEL(q_4)\} \\
LABEL(q_1) &= LABEL(r_4) = \min\{LABEL(q_4), LABEL(r_1)\}
\end{align*}
\]
\[ \text{LABEL}(r_3) = \text{LABEL}(s_4) = \text{LABEL}(t_1) = \text{LABEL}(q_2) \]

\[
\begin{cases} 
0, & \text{if } r_2, s_3, t_4, \text{ and } q_1 \text{ are assigned zero label} \\
\min\{\text{LABEL}(v) \mid v \in \{r_2, t_3, t_4, q_1\}, \text{LABEL}(v) > 0\}, & \text{otherwise}
\end{cases}
\]
unbalanced_sub(f):

If \( \text{LABEL}(v_1) > 0 \), subdivide \( f \) as above to get three new subpatches: \( r_1r_2r_3r_4, s_1s_2s_3s_4, t_1t_2t_3t_4 \), and assign new labels as follows:

\[
\text{LABEL}(r_1) = \text{LABEL}(v_1) - 1
\]

\[
\text{LABEL}(r_i) = 0, \quad i = 2, 3, 4;
\]

\[
\text{LABEL}(s_i) = 0, \quad i = 1, 2, 3, 4
\]

\[
\text{LABEL}(t_i) = 0, \quad i = 1, 2, 3, 4.
\]
Example of adaptive subdivision

Significant savings
Subdivision surfaces have already been used in

- Pixar’s Renderman
- Alias|Wavefront’s Maya
- Nichimen’s Mirai
- Newtek’s Lightwave 3D
Question:

“Is subdivision the representation scheme for future visualization & animation applications?”
The End
Acknowledgement:

- Research work presented here is supported by NSF (DMS-0310645, DMI-0422126).
- Some datasets are taken from P. Schroeder, D. Zorin, L. Kobbelt, H. Hoppe.