Further Graphics

Subdivision Surfaces
Problems with Bezier (NURBS) patches

- Joining spline patches with $C^n$ continuity across an edge is challenging.
- What happens to continuity at corners where the number of patches meeting isn’t exactly four?
- Animation is tricky: bending and blending are doable, but not easy.

Sadly, the world isn’t made up of shapes that can always be made from one smoothly-deformed rectangular surface.
Subdivision surfaces

- Beyond shipbuilding: we want guaranteed continuity, without having to build everything out of rectangular patches.
  - Applications include CAD/CAM, 3D printing, museums and scanning, medicine, movies…

- The solution: subdivision surfaces.

*Geri’s Game*, by Pixar (1997)
Subdivision surfaces

- Instead of ticking a parameter $t$ along a parametric curve (or the parameters $u, v$ over a parametric grid), subdivision surfaces repeatedly refine from a coarse set of control points.

- Each step of refinement adds new faces and vertices.

- The process converges to a smooth limit surface.

(Catmull-Clark in action)
Subdivision surfaces – History

- de Rahm described a 2D (curve) subdivision scheme in 1947; rediscovered in 1974 by Chaikin
- Concept extended to 3D (surface) schemes by two separate groups during 1978:
  - Doo and Sabin found a biquadratic surface
  - Catmull and Clark found a bicubic surface
- Subsequent work in the 1980s (Loop, 1987; Dyn [Butterfly subdivision], 1990) led to tools suitable for CAD/CAM and animation
Subdivision surfaces and the movies

- Pixar first demonstrated subdivision surfaces in 1997 with Geri’s Game.
  - Up until then they’d done everything in NURBS (Toy Story, A Bug’s Life.)
  - From 1999 onwards everything they did was with subdivision surfaces (Toy Story 2, Monsters Inc, Finding Nemo...)
  - Two decades on, it’s all heavily customized.

- It’s not clear what Dreamworks uses, but they have recent patents on subdivision techniques.
Useful terms

- A scheme which describes a 1D curve (even if that curve is travelling in 3D space, or higher) is called **univariate**, referring to the fact that the limit curve can be approximated by a polynomial in one variable \((t)\).
- A scheme which describes a 2D surface is called **bivariate**, the limit surface can be approximated by a \(u,v\) parameterization.
- A scheme which retains and passes through its original control points is called an **interpolating** scheme.
- A scheme which moves away from its original control points, converging to a limit curve or surface nearby, is called an **approximating** scheme.
How it works

Example: Chaikin curve subdivision (2D)

- On each edge, insert new control points at $\frac{1}{4}$ and $\frac{3}{4}$ between old vertices; delete the old points
- The limit curve is $C_1$ everywhere (despite the poor figure.)
Notation

Chaikin can be written programmatically as:

\[ P_{2i}^{k+1} = \left(\frac{3}{4}\right)P_i^k + \left(\frac{1}{4}\right)P_{i+1}^k \quad \leftarrow \text{Even} \]

\[ P_{2i+1}^{k+1} = \left(\frac{1}{4}\right)P_i^k + \left(\frac{3}{4}\right)P_{i+1}^k \quad \leftarrow \text{Odd} \]

...where \( k \) is the ‘generation’; each generation will have twice as many control points as before.

Notice the different treatment of generating odd and even control points.

Borders (terminal points) are a special case.
Notation

Chaikin can be written in vector notation as:

\[
\begin{bmatrix}
\vdots \\
P_{2i-2}^{k+1} \\
P_{2i-1}^{k+1} \\
P_{2i}^{k+1} \\
P_{2i+1}^{k+1} \\
P_{2i+2}^{k+1} \\
P_{2i+3}^{k+1} \\
\vdots \\
\end{bmatrix} = \frac{1}{4}
\begin{bmatrix}
0 & 3 & 1 & 0 & 0 & 0 \\
0 & 1 & 3 & 0 & 0 & 0 \\
0 & 0 & 3 & 1 & 0 & 0 \\
0 & 0 & 1 & 3 & 0 & 0 \\
0 & 0 & 0 & 3 & 1 & 0 \\
0 & 0 & 0 & 1 & 3 & 0 \\
\vdots & & & & & \\
\end{bmatrix}
\begin{bmatrix}
\vdots \\
P_{i-2}^k \\
P_{i-1}^k \\
P_i^k \\
P_{i+1}^k \\
P_{i+2}^k \\
P_{i+3}^k \\
\vdots \\
\end{bmatrix}
\]
Notation

- The standard notation compresses the scheme to a kernel:
  - \( h = (1/4)[\ldots, 0, 0, 1, 3, 3, 1, 0, 0, \ldots] \)
- The kernel interlaces the odd and even rules.
- It also makes matrix analysis possible: eigenanalysis of the matrix form can be used to prove the continuity of the subdivision limit surface.
  - The details of analysis are fascinating, lengthy, and sadly beyond the scope of this course
- The limit curve of Chaikin is a quadratic B-spline!
Reading the kernel

Consider the kernel

\[ h=(1/8)[...,0,0,1,4,6,4,1,0,0,...] \]

You would read this as

\[ P_{2i}^{k+1} = \left(\frac{1}{8}\right)(P_{i-1}^k + 6P_i^k + P_{i+1}^k) \]
\[ P_{2i+1}^{k+1} = \left(\frac{1}{8}\right)(4P_i^k + 4P_{i+1}^k) \]

The limit curve is provably C2-continuous.
Making the jump to 3D: Doo-Sabin

*Doo-Sabin* takes Chaikin to 3D:

\[
P = \frac{9}{16} A + \frac{3}{16} B + \frac{3}{16} C + \frac{1}{16} D
\]

This replaces every old vertex with four new vertices.

The limit surface is biquadratic, C1 continuous everywhere.
Doo-Sabin in action

(0) 18 faces

(1) 54 faces

(2) 190 faces

(3) 702 faces
Catmull-Clark

- *Catmull-Clark* is a bivariate approximating scheme with kernel $h=(1/8)[1, 4, 6, 4, 1]$.
  - Limit surface is bicubic, C2-continuous.
**Catmull-Clark**

Getting tensor again:

\[
\begin{bmatrix}
1 \\
4 \\
6 \\
4 \\
1
\end{bmatrix} \times \frac{1}{8} \begin{bmatrix}
1 \\
4 \\
6 \\
4 \\
1
\end{bmatrix} = \frac{1}{64} \begin{bmatrix}
1 & 4 & 6 & 4 & 1 \\
4 & 16 & 24 & 16 & 4 \\
6 & 24 & 36 & 24 & 6 \\
4 & 16 & 24 & 16 & 4 \\
1 & 4 & 6 & 4 & 1
\end{bmatrix}
\]

Vertex rule  Face rule  Edge rule
Catmull-Clark in action
Catmull-Clark vs Doo-Sabin

Doo-Sabin

Catmull-Clark
Extraordinary vertices

- Catmull-Clark and Doo-Sabin both operate on quadrilateral meshes.
  - All faces have four boundary edges
  - All vertices have four incident edges
- What happens when the mesh contains extraordinary vertices or faces?
  - For many schemes, adaptive weights exist which can continue to guarantee at least some (non-zero) degree of continuity, but not always the best possible.
- CC replaces extraordinary faces with extraordinary vertices; DS replaces extraordinary vertices with extraordinary faces.

Detail of Doo-Sabin at cube corner
Extraordinary vertices: Catmull-Clark

Catmull-Clark vertex rules generalized for extraordinary vertices:

- Original vertex: \( \frac{(4n-7)}{4n} \)
- Immediate neighbors in the one-ring: \( \frac{3}{2n^2} \)
- Interleaved neighbors in the one-ring: \( \frac{1}{4n^2} \)

Schemes for simplicial (triangular) meshes

- **Loop scheme:**

- **Butterfly scheme:**

Split each triangle into four parts

(All weights are /16)
Loop subdivision

Loop subdivision in action. The asymmetry is due to the choice of face diagonals.

Creases

Extensions exist for most schemes to support *creases*, vertices and edges flagged for partial or hybrid subdivision.
Continuous level of detail

For live applications (e.g. games) can compute *continuous* level of detail, typically as a function of distance:
Direct evaluation of the limit surface

- In the 1999 paper *Exact Evaluation Of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter Values*, Jos Stam (now at Alias|Wavefront) describes a method for finding the exact final positions of the CC limit surface.
  - His method is based on calculating the tangent and normal vectors to the limit surface and then shifting the control points out to their final positions.
  - What’s particularly clever is that he gives exact evaluation at the extraordinary vertices. (Non-trivial.)
Bounding boxes and convex hulls for subdivision surfaces

- The limit surface is (the weighted average of (the weighted averages of (the weighted averages of (repeat for eternity…)))) the original control points.
- This implies that for any scheme where all weights are positive and sum to one, the limit surface lies entirely within the convex hull of the original control points.
- For schemes with negative weights:
  - Let $L = \max_t \sum_i |N_i(t)|$ be the greatest sum throughout parameter space of the absolute values of the weights.
  - For a scheme with negative weights, $L$ will exceed 1.
  - Then the limit surface must lie within the convex hull of the original control points, expanded unilaterally by a ratio of $(L-1)$. 
Splitting a subdivision surface

Many algorithms rely on subdividing a surface and examining the bounding boxes of smaller facets.
- Rendering, ray/surface intersections…

It’s not enough just to delete half your control points: the limit surface will change (see right)
- Need to include all control points from the previous generation, which influence the limit surface in this smaller part.

(Top) 5x Catmull-Clark subdivision of a cube
(Bottom) 5x Catmull-Clark subdivision of two halves of a cube; the limit surfaces are clearly different.
Subdivision Schemes—A partial list

- **Approximating**
  - Quadrilateral
    - $(1/2)[1,2,1]$
    - $(1/4)[1,3,3,1]$
      (Doo-Sabin)
    - $(1/8)[1,4,6,4,1]$
      (Catmull-Clark)
    - *Mid-Edge*
  - Triangles
    - *Loop*

- **Interpolating**
  - Quadrilateral
    - *Kobbelt*
  - Triangle
    - *Butterfly*
    - “$\sqrt{3}$” *Subdivision*

Many more exist, some much more complex
This is a major topic of ongoing research
References