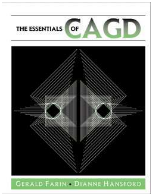
The Essentials of CAGD

Chapter 12: Composite Surfaces

Gerald Farin & Dianne Hansford

CRC Press, Taylor & Francis Group, An A K Peters Book
www.farinhansford.com/books/essentials-cagd



Outline

Introduction to Composite Surfaces

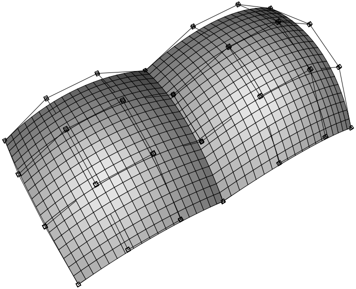
Composite surface: surface composed of more than one patch

One Bézier patch rarely flexible enough to model a real life part

More common: many patches stitched together
$\Rightarrow$ Composite Bézier surface or B-spline surface

Subdivision surfaces are another popular type of composite surface
– Used by many animation studios
– Figure taken from "A Bug's Life" from Pixar Studios.

Composite Bézier Surfaces



"Left" bicubic Bézier patch:

$$\mathbf{b}_{i,j} \quad 0 \le i, j \le 3 \qquad \text{domain: } [u_0, u_1] \times [v_0, v_1]$$

"Right" bicubic Bézier patch:

$$\mathbf{b}_{i,j} \quad 3 \le i \le 6 \quad 0 \le j \le 3 \qquad \text{domain: } [u_1, u_2] \times [v_0, v_1]$$

Both share a common control point and domain boundary

Composite Bézier Surfaces

Smoothness between patches

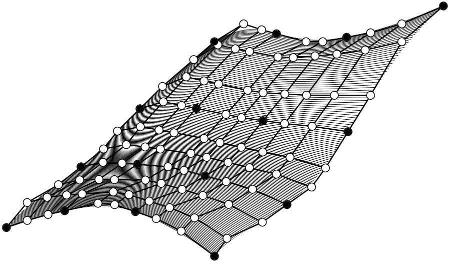
Composite control net contains four rows of control points:

$$\begin{array}{l} \mathbf{b}_{0,0}, \ldots, \mathbf{b}_{6,0} \\ \mathbf{b}_{1,0}, \ldots, \mathbf{b}_{6,1} \\ \mathbf{b}_{2,0}, \ldots, \mathbf{b}_{6,2} \\ \mathbf{b}_{3,0}, \ldots, \mathbf{b}_{6,3} \end{array}$$

Each row interpreted as
the piecewise Bézier polygon of a composite cubic curve
over the knot sequence u_0, u_1, u_2

Surface is C^1 if all rows satisfy curve C^1 conditions

Composite Bézier Surfaces



Knots: $u_i : 0, 1, 3, 4$ "horizontal" $v_j : 0, 1, 2, 3$

For bicubics:

$$\mathbf{b}_{3,j} = \frac{\Delta_1}{\Delta}\mathbf{b}_{2,j} + \frac{\Delta_0}{\Delta}\mathbf{b}_{4,j} \qquad j = 0, 1, 2, 3$$

$$\Delta_0 = u_1 - u_0 \qquad \Delta_1 = u_2 - u_1 \qquad \Delta = u_2 - u_0$$

$\Rightarrow$ Points $\mathbf{b}_{2,j}, \mathbf{b}_{3,j}, \mathbf{b}_{4,j}$ must be collinear *and* in the same ratio:

$$\mathrm{ratio}(\mathbf{b}_{2,j}, \mathbf{b}_{3,j}, \mathbf{b}_{4,j}) = \frac{\Delta_0}{\Delta_1}$$

Composite Bézier Surfaces

C^1 conditions for composite surfaces are quite simple to handle

Rectangular network of patches
with u- and v-knot sequences
$\Rightarrow$ Inflexibility in shape control

If not all $u-$isoparametric curves have similar shape, then a common knot sequence for all of them is problematic

Same holds for the v-curves

B-spline Surfaces

B-spline curve:

$$\mathbf{x}(u) = \mathbf{d}_0 N_0^n(u) + \ldots + \mathbf{d}_{D-1} N_{D-1}^n(u) \quad \Rightarrow \quad \mathbf{x}(u) = N^{\mathrm{T}}\mathbf{D}$$

B-spline surface $\mathbf{x}(u, v)$:

$$\mathbf{x}(u, v) = \begin{bmatrix} N_0^m(u) & \ldots & N_{D-1}^m(u) \end{bmatrix} \begin{bmatrix} \mathbf{d}_{0,0} & \ldots & \mathbf{d}_{0,E-1} \\ \vdots & & \vdots \\ \mathbf{d}_{D-1,0} & \ldots & \mathbf{d}_{D-1,E-1} \end{bmatrix} \begin{bmatrix} N_0^n(v) \\ \vdots \\ N_{E-1}^n(v) \end{bmatrix}$$

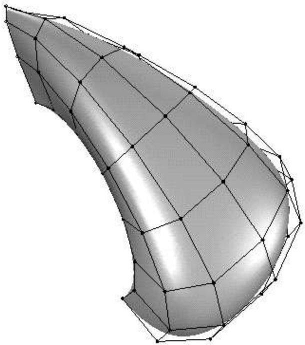
Abbreviated to

$$\mathbf{x}(u, v) = M^{\mathrm{T}}\mathbf{D}N$$

Over knot sequences

$$u_0, u_1, \ldots, u_{R-1} \qquad v_0, v_1, \ldots, v_{S-1}$$

B-spline Surfaces



Bicubic B-spline surface over knot sequences
$u_i = 0, 1, 2, 3, 4, 5$
$v_j = 0, 1, 2, 3, 4$

B-spline surfaces enjoy all the properties of Bézier patches
– Symmetry
– Affine invariance
– Convex hull property
– Etc.

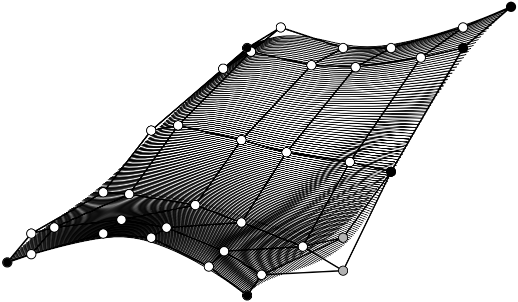
One difference:
Boundary polygons/boundary curves correspondence
– Only with full multiplicity of end knots
– Analogous to endpoint interpolation property of B-spline curves

B-spline Surfaces

Local control:

If one control point is moved

Only up to $(m+1)(n+1)$ patches in vicinity affected



Two control nets differ by only one control point

– Marked in gray for either net

– Surface differences appear through *Moiré patterns*

"waves" not part of either surface

B-spline Surfaces

Isoparametric curve: curve on surface formed by fixing one parameter
– For example: $u = \bar{u}$
Represent *isocurve* as B-spline curve

$$\mathbf{C} = M^{\mathrm{T}}\mathbf{D} = [\mathbf{c}_0, \ldots, \mathbf{c}_{E-1}]$$

Factor $\mathbf{x}(u, v) = M^{\mathrm{T}}\mathbf{D}N$ as

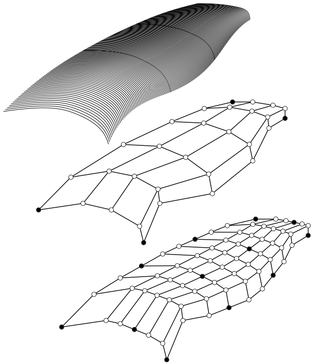
$$\mathbf{x}(\bar{u}, v) = \mathbf{C}N \quad \Rightarrow \quad \text{B-spline curve with variable } v$$

– As v varies, $\mathbf{x}(\bar{u}, v)$ traces out the desired isocurve

Try forming isocurve $\mathbf{x}(u, \bar{v})$

An isocurve control polygon may be treated as any other curve

B-spline Surfaces



Top: bicubic B-spline surface
Middle: B-spline control polygon
Bottom: piecewise Bézier control net

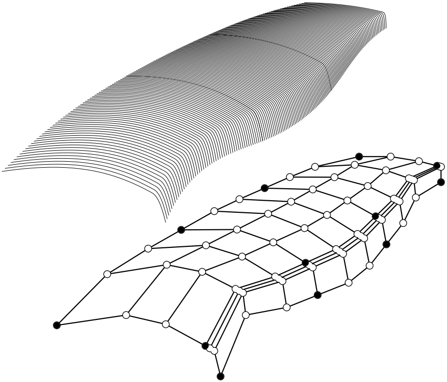
B-spline surface consists of a collection of individual polynomial patches

Each may be written in Bézier form

Obtain patch control nets:

1. Convert each row of control points into piecewise Bézier form
2. Convert each column of result into piecewise Bézier form

B-spline Surfaces



Another example:

B-spline control net the same as previous figure

Different knot sequence in the $u-$direction

Knot sequences: u_i : $0, 3, 4$ $\quad$ v_j : $0, 1, 2, 3$

B-Spline Surface Approximation

Given: Data points $\mathbf{p}_k$; $k = 0, \ldots, K-1$

Find: a B-spline surface that approximates the data

Need more information to solve problem:

B-spline surface specifications

u- and v-knot sequences

u- and v-degrees

Each data point $\mathbf{p}_k$ associated with a pair of parameters (u_k, v_k)
– Parameters expected to be in domain of B-spline surface

B-spline surface (written with linearized ordering of terms)

$$\mathbf{x}(u,v) = \left[N_0^m(u)N_0^n(v), \ldots, N_{D-1}^m(u)N_{E-1}^n(v)\right] \begin{bmatrix} \mathbf{d}_{0,0} \\ \vdots \\ \mathbf{d}_{D-1,E-1} \end{bmatrix}$$

B-Spline Surface Approximation

For the k^{th} data point: $\mathbf{p}_k = \mathbf{x}(u_k, v_k)$

$$\mathbf{x}(u_k, v_k) = \left[N_0^m(u_k)N_0^n(v_k), \ldots, N_{D-1}^m(u_k)N_{E-1}^n(v_k)\right] \begin{bmatrix} \mathbf{d}_{0,0} \\ \vdots \\ \mathbf{d}_{D-1,E-1} \end{bmatrix}$$

Combining all K of these equations

$$\begin{bmatrix} \mathbf{p}_0 \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{p}_{K-1} \end{bmatrix} = \begin{bmatrix} N_0^m(u_0)N_0^n(v_0) & \ldots & N_{D-1}^m(u_0)N_{E-1}^n(v_0) \\ & \vdots & \\ & \vdots & \\ & \vdots & \\ N_0^m(u_{K-1})N_0^n(v_{K-1}) & \ldots & N_{D-1}^m(u_{K-1})N_{E-1}^n(v_{K-1}) \end{bmatrix} \begin{bmatrix} \mathbf{d}_{0,0} \\ \vdots \\ \mathbf{d}_{D-1,E-1} \end{bmatrix}$$

$$\mathbf{P} = M\mathbf{D}$$

Least squares solution found by solving $M^{\mathrm{T}}\mathbf{P} = M^{\mathrm{T}}M\mathbf{D}$

$\Rightarrow$ System of normal equations

B-Spline Surface Approximation

Least squares solution
– may have unsatisfactory shape in some cases
– may not be solvable if "holes" exist in data distribution

Shape equations are a tool to overcome these problems

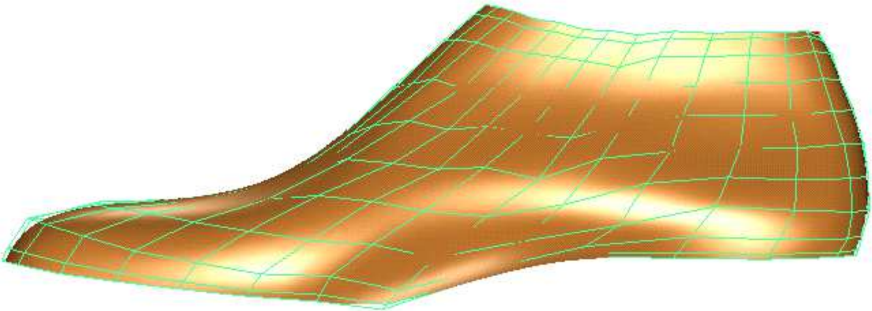
Motivation:
In a "nice" mesh, each control mesh quadrilateral is a parallelogram

$$\mathbf{d}_{i,j} + \mathbf{d}_{i+1,j+1} - \mathbf{d}_{i+1,j} - \mathbf{d}_{i,j+1} = \mathbf{0}$$

Add each of the equations to the overdetermined system

Solve system using normal equations

B-Spline Surface Approximation



Example: least squares B-spline surface fit to a shoe last

B-Spline Surface Interpolation

Bicubic B-spline surfaces interpolation problem

Given: a $P \times Q$ rectangular array of data points $\mathbf{p}_{i,j}$

Find: an interpolating bicubic B-spline surface
Corners of the patch go through the given data points

Surface knot sequences

$$u_i \text{ for } 0 \le i < R \quad \text{and} \quad v_i \text{ for } 0 \le i < S$$

where $R = P + 4$ and $S = Q + 4$

Surface control net

$$\mathbf{d}_{i,j} \qquad 0 \le i < D, \quad 0 \le j < E$$

must have $D = P + 2$ and $E = Q + 2$ control points

B-Spline Surface Interpolation

Solution: reduce it to a series of curve interpolation problems

Interpret the given $P \times Q$ array of data points as a set of P rows of points
To each row with Q points, fit a B-spline curve
$\Rightarrow$ $Q + 2$ control points in each row
Produces a $P \times (Q + 2)$ net of control points $\mathbf{c}_{i,j}$

$\mathbf{c}_{i,j}$ treated in a column-by-column fashion:
To each of these $(Q + 2)$ columns, fit a B-spline curve through P points
$\Rightarrow$ results in $P + 2$ control points in each column

Final result: $(P + 2) \times (Q + 2)$ control net $\mathbf{d}_{i,j}$
$\Rightarrow$Surface interpolating the given $P \times Q$ array of data points

This curve-based approach saves computing time:
Solve tridiagonal linear systems
– One matrix for all row problems and one matrix for all column problems

B-Spline Surface Interpolation

Example:

Given: a 2×3 array of data points

$$\begin{bmatrix} \mathbf{p}_{0,0} & \mathbf{p}_{0,1} & \mathbf{p}_{0,2} \\ \mathbf{p}_{1,0} & \mathbf{p}_{1,1} & \mathbf{p}_{1,2} \end{bmatrix}$$

To each row, fit a B-spline curve $\Rightarrow$ resulting in two control polygons

$$\begin{bmatrix} \mathbf{c}_{0,0} & \mathbf{c}_{0,1} & \mathbf{c}_{0,2} & \mathbf{c}_{0,3} & \mathbf{c}_{0,4} \\ \mathbf{c}_{1,0} & \mathbf{c}_{1,1} & \mathbf{c}_{1,2} & \mathbf{c}_{1,3} & \mathbf{c}_{1,4} \end{bmatrix}$$

Treat each of five columns as a set of curve data points

$$\begin{bmatrix} \mathbf{d}_{0,0} & \mathbf{d}_{0,1} & \mathbf{d}_{0,2} & \mathbf{d}_{0,3} & \mathbf{d}_{0,4} \\ \mathbf{d}_{1,0} & \mathbf{d}_{1,1} & \mathbf{d}_{1,2} & \mathbf{d}_{1,3} & \mathbf{d}_{1,4} \\ \mathbf{d}_{2,0} & \mathbf{d}_{2,1} & \mathbf{d}_{2,2} & \mathbf{d}_{2,3} & \mathbf{d}_{2,4} \\ \mathbf{d}_{3,0} & \mathbf{d}_{3,1} & \mathbf{d}_{3,2} & \mathbf{d}_{3,3} & \mathbf{d}_{3,4} \end{bmatrix}$$

$\mathbf{d}_{i,j}$ form the interpolating surface control mesh

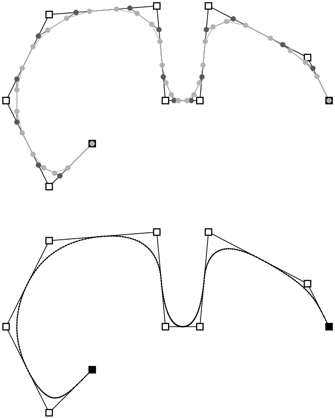
Subdivision Surfaces: Doo-Sabin

de Casteljau algorithm and de Boor algorithm

Two examples of subdivision schemes
Refine a polygon $\Rightarrow$ polygon locally approximates a smooth curve

Both algorithms are actually repeated instances of *knot insertion*

Subdivision Surfaces: Doo-Sabin



Chaikin's algorithm:

Input: a polygon (squares) $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_n$

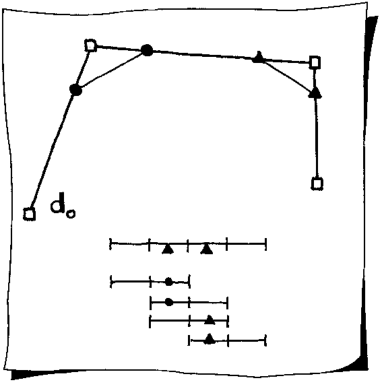
Output: a refined polygon approximating a smooth curve

One step produces

$$\mathbf{d}_0^1 = \mathbf{d}_0$$
$$\mathbf{d}_{2i-1}^1 = \frac{3}{4}\mathbf{d}_i + \frac{1}{4}\mathbf{d}_{i-1}$$
$$\mathbf{d}_{2i}^1 = \frac{3}{4}\mathbf{d}_i + \frac{1}{4}\mathbf{d}_{i+1}$$
$$\mathbf{d}_{2n-1}^1 = \mathbf{d}_n$$

for $i = 1, \ldots, n-1$

Subdivision Surfaces: Doo-Sabin



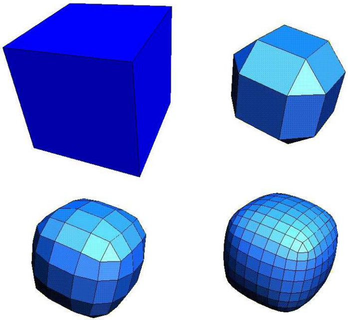
Chaikin's algorithm is a special application of knot insertion

Input polygon consists of de Boor points $\mathbf{d}_i$ of a *quadratic B-spline*

Knot sequence: uniform

One step of algorithm equivalent to inserting a knot at the midpoint of each domain knot interval

Subdivision Surfaces: Doo-Sabin

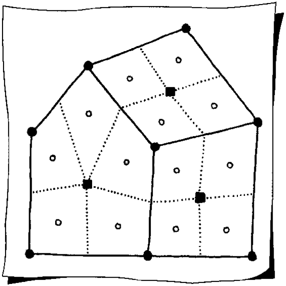


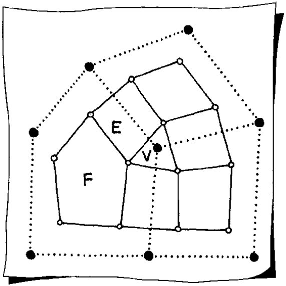
Carry curve concept to surfaces

Chaikin's algorithm generalized to the Doo-Sabin algorithm
– Converges to biquadratic B-splines
– Defined over uniform knots

Doo-Sabin algorithm can be applied to polygonal meshes of arbitrary topology
– Polygons do not have to be be four-sided

Subdivision Surfaces: Doo-Sabin





One step of Doo-Sabin

1. For each face, form new vertices
   a. Centroid
   b. Edge midpoints
   c. New vertex as average of face vertex, centroid, and two edge midpoints
2. Form new faces from new vertices
   a. F-faces
   b. E-faces
   c. V-faces

Repeat until the polygonal mesh is desired smoothness

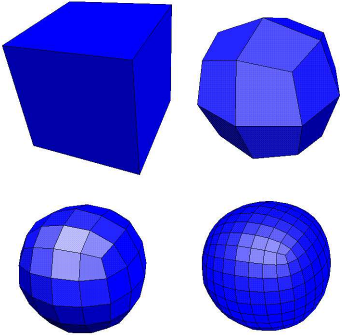
Subdivision Surfaces: Doo-Sabin

Four quadrilaterals – vertices form a 3×3 rectangular net:
$\Rightarrow$ control net of a biquadratic B-spline patch over uniform knot sequences

Neighboring rectangular patches are C^1
– Non-four-sided faces $\Rightarrow$ surface less smooth (in general)

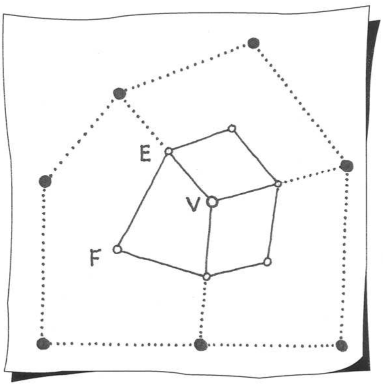
Non-four-sided faces appear if
– In input mesh
– The input mesh has $n \neq 4$ faces around a vertex
    First step of Doo-Sabin creates a face with n vertices
    Extraordinary vertex: non-four-sided face shrinks to a point

Subdivision Surfaces: Catmull-Clark



Catmull-Clark algorithm: Generalization of cubic curve subdivision scheme
– Produce bicubic B-splines
– Generalized to work on polygonal meshes of arbitrary topology

Subdivision Surfaces: Catmull-Clark



One step:

1. Form face points **f**:
   average face vertices
2. Form edge points **e**:
   average edge vertices and two face points
3. Form vertex points **v**:
   n faces around a vertex
4. Form faces of the new mesh:
   $(\mathbf{f}, \mathbf{e}, \mathbf{v}, \mathbf{e})$

$$\mathbf{v} = [\frac{(n-3)}{n}](\text{old } \mathbf{v}) + [\frac{1}{n}](\text{ave } \mathbf{f}) + [\frac{2}{n}](\text{ave of midpoints of edges})$$

Subdivision Surfaces: Catmull-Clark

Nine quadrilaterals – 4×4 rectangular net
$\Rightarrow$ Control net of a bicubic B-spline surface over uniform knot sequences

Neighboring rectangular surfaces are C^2

First step of Catmull-Clark produces all four-sided faces

Extraordinary vertices: place where continuity diminished
n faces will share a vertex if
– an input face was n-sided
– input mesh had n-faces around a vertex
This non-rectangular element will shrink with more steps

Subdivision Surfaces: Catmull-Clark

Graphics and animation industries have embraced subdivision surfaces

Appeal of subdivision surfaces:

Simple polygonal net easily becomes a smooth surface
– Same general shape

Flexible topology – Handles non-four-sided patches
– Example: sphere difficult to deal with only rectangular patches