UFL CISE TR 01-001: Modifications of PCCM

Jörg Peters*

University of Florida

1 Introduction

This note discusses some finer points of Patching Catmull-Clark Meshes pointed out in the Siggraph talk. The modifications have been implemented and examples are posted at [1].
(1) The normal of the PCCM surface at an extraordinary point is free to choose. In particular, we may choose it as the normal of the Catmull-Clark limit surface in the extraordinary point. We give the corresponding formula below. (2) The perturbation of the mesh for higher-order saddle points of even valence can lead to undesirable flatness of the surface in the neighborhood of the extraordinary point. There is a remedy. (3) Pulling and pushing control meshes after a subdivision step allows the distribution of curvature e.g. creating sharper features. This may be viewed,as is common in computer aided design, as adjusting the blend radius between primary surfaces. This note shows how to include blend ratios, i.e. control of the sharpness of transitions, into the PCCM framework.

2 Interpolation of the Catmull-Clark limit normal

To make the normal of PCCM equal to the Catmull-Clark normal we set the tangent coefficient Q_{10} to

$$
\begin{aligned}
Q_{10} &= Q_{00} + \frac{a}{3(2+w_n)}\big(w_n A_n P_{30} + (A_n + A_n^+)P_{33}\big),\\
w_n &= 1 + c + \cos(\frac{\pi}{n}\sqrt{2(9+c)}),\\
a_{\text{default}} &= 1,\\
A_n^+(i,j) &= A_n(i,j+1).
\end{aligned}
$$

3 Higher-order saddle points of even valence

See [1] for ilustrations.

A (simple) saddle point is surrounded by two dips and two raises. A *higher-order saddle point* of even valence is surrounded by three or more dips and raises. Such points occur for example in objects of higher genus, e.g. where three or more handles meet. Since the curvature of a smooth surface has a unique maximum and minimum direction if it is not zero, higher-order saddle points have zero curvature.

Since the first piece of the boundary of a bicubic spline patch generated by PCCM is of degree 2, zero curvature implies that the first piece should be a line segment. This is correctly enforced by enforcing the alternating sum condition, $r = \sum_{i=1}^{n}(-1)^i Q_{40}^{\text{old}}(i)/n = 0$. However, generating the correct straight line segment has a sometimes undesirable shape effect by pushing the flatness that should be local to the higher-order saddle point out into the surrounding geometry. Note that the *issue here is beauty, not smoothness*.

*NSF NYI CCR-9457806

In the following we assume that $r \neq 0$, e.g. $r > 10^{-4}$.

One, often reasonable approach, is to simply accept a tangent discontinuity and not enforce $r = 0$ noting that r decreases by factor of $1/4$ with each subdivision.

A second approach requires inserting an additional knot at $u = 1$ and defining a G^1 rather than a C^1 join across the edges of the polynomial pieces attached to the extraordinary point.

A third approach is to render the PCCM patch without the square $[0,1]^2$ (solid squares in Figure 1 of PCCM), i.e. trim out the polynomial piece attached to the higher-order saddle point. In place of the hole we insert n Bézier patches of degree bi-4. This increases the degree of the boundary piece to 3 so that the curve is not necessarily a straight line segment if the first three coefficients are collinear. This approach is explained below.
(1) We adjust the boundary curve in the Knot Insertion stage:

$$
\begin{aligned}
Q_{10} &\leftarrow Q_{00} + A_n Q_{10},\\
r_{20} &= \sum_{i=1}^{n}(-1)^i Q_{20}(i)/n, \in R\\
q(i) &= -(-1)^i r_{20},\\
Q_{20} &\leftarrow Q_{20} + q,\\
Q_{21} &\leftarrow Q_{21} + q, Q_{12}^- \leftarrow Q_{12}^- + q.
\end{aligned}
$$

Here $Q_{12}^-(l) = Q_{12}(l-1)$.
(2) We assemble the n bicubic patches with coefficients $c_{ij}(l), l = 1,\ldots,n$ from the Knot Insertion stage. For simplicity we drop the (l) that indicates the lth patch:

$$
\begin{bmatrix}
c_{03} & c_{13} & c_{23} & c_{33}\\
c_{02} & c_{12} & c_{22} & c_{32}\\
c_{01} & c_{11} & c_{21} & c_{31}\\
c_{00} & c_{10} & c_{20} & c_{30}
\end{bmatrix} =
$$

$$
\begin{bmatrix}
\frac{Q_{04}+Q_{02}}{2} & \frac{Q_{14}+Q_{12}}{2} & \frac{Q_{24}+Q_{22}}{2} & \frac{Q_{24}+Q_{22}+Q_{44}+Q_{42}}{4}\\
Q_{02} & Q_{12} & Q_{22} & \frac{Q_{22}+Q_{42}}{2}\\
Q_{01} & Q_{11} & Q_{21} & \frac{Q_{21}+Q_{41}}{2}\\
Q_{00} & Q_{10} & Q_{20} & \frac{Q_{20}+Q_{40}}{2}
\end{bmatrix}
$$

(3) We degree-raise this patch first in the u-direction then in the v-direction:

$$
\begin{aligned}
h_{i,j} &= \frac{i}{4}c_{i-1,j} + \frac{4-i}{4}c_{i,j}, \quad i = 0,\ldots,4,\ j = 0,\ldots,3,\\
b_{i,j} &= \frac{j}{4}h_{i,j-1} + \frac{4-j}{4}h_{i,j}, \quad i = 0,\ldots,4,\ j = 0,\ldots,4
\end{aligned}
$$

and obtain the bi-4 patch

$$
\begin{bmatrix}
b_{04} & b_{14} & b_{24} & b_{34} & b_{44}\\
b_{03} & b_{13} & b_{23} & b_{33} & b_{43}\\
b_{02} & b_{12} & b_{22} & b_{32} & b_{42}\\
b_{01} & b_{11} & b_{21} & b_{31} & b_{41}\\
b_{00} & b_{10} & b_{20} & b_{30} & b_{40}
\end{bmatrix}
$$

(4) We adjust the biquartic patches. The superscript $^+$ denotes a shift: $b^+_{ij}(l) = b_{ij}(l+1)$ and $c = \cos(2\pi/n)$.

$$
\begin{aligned}
b_{11} &\leftarrow B_n r_{11}, \qquad r_{11} = b_{01} + \frac{3c}{8}(c_{02} - c_{01}),\\
&\quad r_{02} = b^{\text{old}}_{02} - (b^{\text{old}}_{12} + b^{\text{old},+}_{21})/2 + \frac{c}{8}(c_{03} - c_{02}),\\
b_{12} &\leftarrow b^{\text{old}}_{12} + r_{02},\\
b^+_{21} &\leftarrow b^{\text{old},+}_{21} + r_{02},
\end{aligned}
$$

4 Blend ratios for Catmull-Clark Meshes

PCCM does not depend on the actual placement of the coefficients in the Catmull-Clark mesh. In particular, we can choose different rules, say a least squares fit to data, to determine the location of the mesh points. One option is to group the mesh nodes closer together or spread them further apart in certain regions or to pull points to distribute curvature. Since this basic idea was used earlier in the context of biquadratic splines and the application was to define blends between polynomial patches in the sense of geometric design, the numbers governing the process are called 'blend ratios'; semi-smooth creasing would be just as good a name for the mechanism.

The question then is how to move a mesh predictably with the least amount of (programming or user interaction) effort. This section describes such a mechanism in the context of the data structures of PCCM. The work amounts to an *extra pass along the boundaries* of an array for Catmull-Clark subdivision. The effect is to pull (or repell) the boundary coefficients P_{i0} of an edge towards a cubic curve that is defined by the boundary coefficients at the coarsest level. At the same time we pull the tangent coefficients towards the boundary coefficients. These two effects should really have a separate control but that would lead to more notation.

Every blend-edge pq at level ℓ has two associated scalars

$$a^\ell_{pq}, a^\ell_{qp} \in [0,1].$$

If $a^\ell_{pq} = a^\ell_{qp} = 1$ the edge will be sharp. If $a^\ell_{pq} = a^\ell_{qp} = 0$ the edge will be rounded. If $a^\ell_{pq} = 1$ and $a^\ell_{qp} = 0$ the edge will start sharp and become more rounded.

Conceptually, after refining the quad to level $\ell+1$, we also compute the refinement of the cubic boundary curve associated with the blend-edge. Then we average the control points of this boundary curve and the control points corresponding to the boundary in the quad using a^ℓ_{pq} and a^ℓ_{qp}. The only challenge is to update consistently in the neighborhood of vertices within the framework of PCCM arrays.

For each quad, we store a_{pq} and a_{qp} for each edge.

Let P^ℓ_i be the ith control point, $i = 0,\dots,k$, $k = 2^\ell$ of the boundary in the quad at level ℓ and $P^\ell_{-,i}$ and $P^\ell_{+,i}$ the control point to the left and right of the boundary traversing from p to q. At level ℓ we first replace the odd-labeled then the even-labeled control points.

For $i = 2,\dots,k-1$ and $j = 2i-1$

$$
\begin{aligned}
&C = (P^\ell_{i-1} + P^\ell_i)/2,\ a = \frac{j}{2k}a^\ell_{pq} + \frac{2k-j}{2k}a^\ell_{qp} \in [0,1]\\
&P^{\ell+1}_j \leftarrow aC + (1-a)P^{\ell+1}_j\\
&P^{\ell+1}_{+,j} \leftarrow aC + (1-a)P^{\ell+1}_{+,j}\\
&P^{\ell+1}_{-,j} \leftarrow aC + (1-a)P^{\ell+1}_{-,j}
\end{aligned}
$$

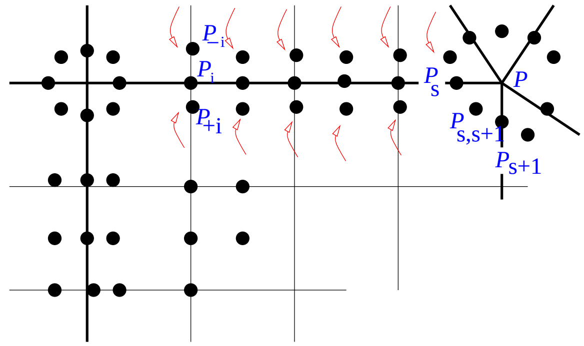

Figure 1: blend ratios

and for $i = 1,\dots,k-1$ and $j = 2i$,

$$
\begin{aligned}
&C = (P^\ell_{i-1} + 6P^\ell_i + P^\ell_{i+1})/8,\ a = \frac{j}{2k}a^\ell_{pq} + \frac{2k-j}{2k}a^\ell_{qp} \in [0,1]\\
&P^{\ell+1}_j \leftarrow aC + (1-a)P^{\ell+1}_j\\
&P^{\ell+1}_{+,j} \leftarrow aC + (1-a)P^{\ell+1}_{+,j}\\
&P^{\ell+1}_{-,j} \leftarrow aC + (1-a)P^{\ell+1}_{-,j}
\end{aligned}
$$

For each vertex p with position P, neighbor node positions P_q and the neighbors locally renamed $\nu = 1,\dots,n$ ($a_{p\nu}$ determines stretching perpendicular to edge $p\nu$) set

$$
\begin{aligned}
&C_\nu = (P^\ell + P^\ell_\nu)/2,\ a_\nu = \frac{1}{2k}a^\ell_{p\nu} + \frac{2k-1}{2k}a^\ell_{\nu p} \in [0,1]\\
&P^{\ell+1} \leftarrow (1 - \sum a_\nu/n)P^{\ell+1} + \sum a_\nu/nP^\ell\\
&P^{\ell+1}_\nu \leftarrow \frac{a_{\nu-1} + a_{\nu+1}}{2}P^\ell\\
&\qquad + (1 - \frac{a_{\nu-1} + a_{\nu+1}}{2})\big(a_\nu C_\nu + (1-a_\nu)P^{\ell+1}_\nu\big)\\
&P^{\ell+1}_{\nu,\nu+1} \leftarrow a_\nu a_{\nu+1}P^\ell + a_\nu(1-a_{\nu+1})C_\nu + (1-a_\nu)a_{\nu+1}C_{\nu+1}\\
&\qquad + (1-a_\nu)(1-a_{\nu+1})P^{\ell+1}_{\nu,\nu+1}.
\end{aligned}
$$

[1] L.J. Shiue, implementation snapshots
http://www.cise.ufl.edu/research/SurfLab/pccm_demo/index.html