Patching Catmull-Clark Meshes

Jörg Peters[*]

University of Florida

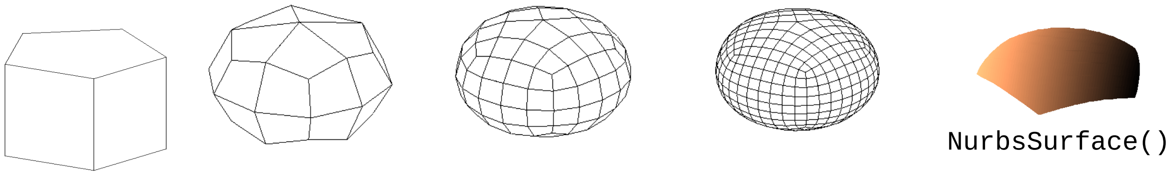


Figure 1: Catmull-Clark subdivision sequence completed as a collection of large Nurbs patches (top five patches are shown).

Abstract

Named after the title, the PCCM transformation is a simple, explicit algorithm that creates large, smoothly joining bicubic Nurbs patches from a refined Catmull-Clark subdivision mesh. The resulting patches are maximally large in the sense that one patch corresponds to one quadrilateral facet of the initial, coarsest quadrilateral mesh before subdivision. The patches join parametrically C^2 and agree with the Catmull-Clark limit surface except in the immediate neighborhood of extraordinary mesh nodes; in such a neighborhood they join at least with tangent continuity and interpolate the limit of the extraordinary mesh node. The PCCM transformation integrates naturally with array-based implementations of subdivision surfaces.

CR Categories: I.3.5 [surface representation, splines]: I.3.6—graphics data structures

Keywords: CAD, Curves & Surfaces, Geometric Modeling

1 Motivation

Catmull-Clark subdivision meshes [2] are an increasingly popular surface representation that comes equipped with tools for adapting shape locally and globally [3, 6]. The limit surfaces can be directly evaluated [13] and an ample body of analysis assures tangent continuity [1, 11, 14]. It would be nice though if we could stop the subdivision process at any point, apply a simple transformation and get a compact, explicit surface representation in the form of a small collection of maximally large, standard spline (Nurbs) patches that join just as smoothly and largely agree with the Catmull-Clark limit surface. The PCCM (Patching Catmull-Clark Meshes) algorithm is such a transformation.

The paper has three parts: a review of basics and literature, the specification of the algorithm and the discussion of the properties of its output:

- Each Nurbs patch covers a maximally large region of the mesh corresponding to the coarsest level quadrilateral mesh facets.
- The patches join C^2 almost everywhere and are at least tangent continuous near the extraordinary mesh nodes. Transitions between patches are almost all parametric.
- The Nurbs patches are polynomial, of order 4 (degree 3) and in interpolating form with 4-fold knots.
- The Nurbs patches differ from the limit surface of the Catmull-Clark subdivision only near the extraordinary mesh nodes. (The patches have generically finite curvature whereas the generic curvature of the Catmull-Clark limit surface at extraordinary mesh nodes is infinite.)
- Nurbs patches, Catmull-Clark subdivision and the PCCM algorithm can use the same array-based data structures.

2 Nurbs, Catmull-Clark and Prior Work

A Nurbs patch $Q(u, v) \in \mathbf{R}^3$ of order 4 (bicubic tensor-product spline) is defined by two nondecreasing sequences of scalars (called *knots*) of length $k+4$, one for the u and one for the v parameter and k^2 control points $Q_{\mathtt{uv}} \in \mathbf{R}^3$. If we connect control points whose indices differ by 1 in exactly one slot, we obtain a *control net* that outlines the patch. A concrete interface for Nurbs patches is the `gluNurbsSurface` specification in OpenGL [4]. By definition, Nurbs do not change geometrically under *knot insertion*. (Repeated insertion of the same knot just transforms to Bézier a.k.a. `glMap` form.) Inserting numbers into the knot sequence subdivides the parameter domain into more pieces between knots. Correspondingly, the control net must be refined or [sic] subdivided by specific rules. For example, if we start with a uniform (evenly spaced) knot sequence consisting of even integers and insert new knots at odd integers we can classify the new control points as (a) facet points, F, corresponding to odd knots, (b) edge points, E, corresponding to one odd and one even knot, and (c) vertex points, V, corresponding to even knots. The weights for new F, E and V type points as

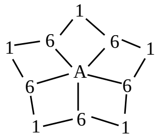
[*] NSF NYI CCR-9457806

averages of the old points are displayed as averaging masks:

$$4F \leftarrow \begin{array}{cc} 1 & 1 \\ 1 & 1 \end{array} \qquad 16E \leftarrow \begin{array}{cc} 1 & 1 \\ 6 & 6 \\ 1 & 1 \end{array} \qquad 64V \leftarrow \begin{array}{ccc} 1 & 6 & 1 \\ 6 & 36 & 6 \\ 1 & 6 & 1 \end{array} \qquad (1)$$

For example, a new F point is the average of four surrounding old points. A central contribution of [2] was the addition of a rule for extraordinary mesh nodes V with $n \neq 4$ neighbors:

$$4n^2 V \leftarrow$$



$$A = 4n^2 - 7n. \qquad (2)$$

The idea of converting from the subdivision mesh back to a spline representation for rendering dates back at least to DeRose, Kass and Truong [3], before the availability of the full Pixar subdivision pipeline. Compared to the large patches derived below, these patches are smaller and more numerous and the conversion is not well-defined in the neighborhood of extraordinary mesh nodes. Nasri and Peters [9] use an approximation to the limit surface of the Doo-Sabin subdivision to get a quickly convergent series of approximations to the volume of the enclosed subdivision object. The approximation surface, however, is only position continuous. Prautzsch [12] gives an elegant solution to the difficult problem of filling n-sided holes. G^2 completion of a Catmull-Clark mesh requires order 7 patches and at least nine times as many as the large patches derived below. Grimm and Hughes [5] use subdivision as a preprocessing step to generating smooth manifolds with which they associate an interesting class of rationally blended spline surfaces. Their approach would generate at least nine times as many rational spline patches either over nonstandard domains or of high degree.

3 From Mesh to Surface

As Figure 1 illustrates, all mesh facets are four-sided either on input or after at most one subdivision step. We refer to each facet of this coarsest quad-mesh as the level 0 of a *quad* and think of ℓth step of Catmull-Clark subdivision as subdividing the level $\ell - 1$ of each quad into 4 times as many subfacets for a total of 4^ℓ subfacets at level ℓ.

Indexing. Since the goal is to transform a repeatedly subdivided mesh we can treat each corner of a quad in isolation as shown in Figures 2 and 3. The quads surrounding the corner point are arranged in counterclockwise order indexed by i. The nodes $P_{\mathtt{uv}}(i)$ of the ith quad lie on a $\mathtt{uv}$ grid of mesh lines. The double subscripts $\mathtt{uv}$ are the Greville abscissae, scaled by 3, of the output Nurbs surface. $P_{00}(i)$ is the corner node for all i and $P_{l0}(i+1) = P_{0l}(i)$. The same indexing is used for the output control points $Q_{\mathtt{uv}}(i)$. If a quad does not have a neighbor due to a manifold boundary, we use the same rule as the Catmull-Clark subdivision, say [8, 15], to provide an additional outside layer of mesh nodes.

The PCCM(ℓ) Algorithm: The *input* is a mesh of a ℓ times subdivided quads with nodes $P_{\mathtt{uv}}(i) \in \mathbf{R}^3$. (If all vertices have 4 or an odd number of neighbors then $\ell > 0$ suffices otherwise $\ell > 1$ should hold.) The *output* is one bicubic (order 4) Nurbs surface (patch) for every quad. The patches are in standard *interpolating form*. That is, the $\mathtt{u}$ and the $\mathtt{v}$ knot sequence start and end with a 4-fold knot (at 0 and at $k = 2^\ell$ where ℓ is the subdivision level). Each sequence can have up to two double knots (at 1 and $2^\ell - 1$) and has single knots otherwise, e.g. $0, 0, 0, 0, 1, 1, 2, 3, 4, 5, 6, 7, 7, 8, 8, 8, 8$. Systems that do not allow internal double knots but do allow uneven knot spacing with minimal distance ϵ can be accommodated by perturbing one inner

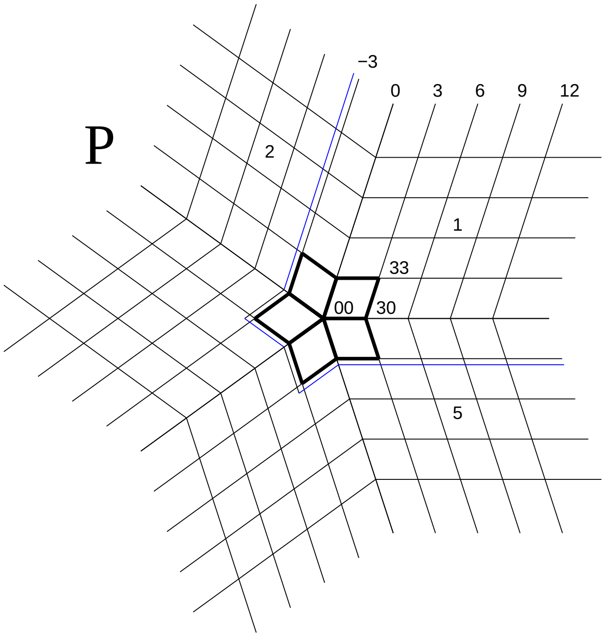


Figure 2: Catmull-Clark mesh in the neighborhood of an extraordinary mesh node of degree 5. Quads are labeled in counterclockwise order from 1 to n around the extraordinary mesh node so that all indices i are interpreted $(i + n - 1) \mod n + 1$. The double subscripts 00,30 and 33 in quad 1 belong to the extraordinary mesh node $P_{00}(1)$, its direct neighbor $P_{30}(1)$ and its diagonal facet neighbor $P_{33}(1)$. The double line delineates (one of the four corners of) a submesh used to define one quad level.

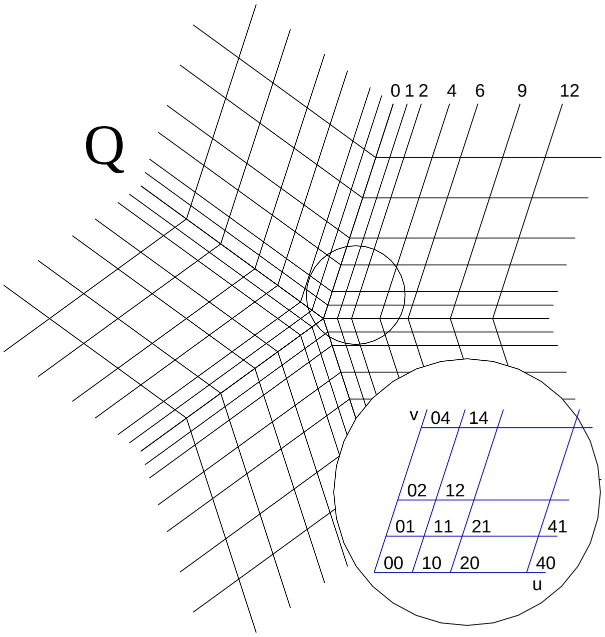


Figure 3: After Knot Insertion the Nurbs patches abut sharing the nodes $Q_{l0}(i+1) = Q_{0l}(i)$. The enlargement shows the indices of the control points relevant for Corner Smoothing.

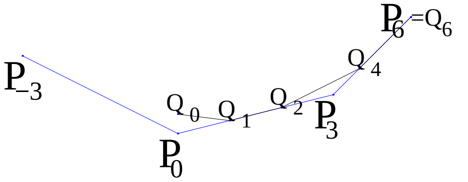


Figure 4: Knot insertion at $\mathtt{u} = 0$; P_{-3} is a mesh node borrowed from the neighboring quad.

double knot by ϵ. PCCM surfaces of level ℓ preserve the blend ratios (or smoothed creases) of the Catmull-Clark mesh [10, 3] up to level ℓ. The surface can additionally be pinched (flattened) at the extraordinary mesh node by decreasing the multiplier a of A_n to less than 1 (more than 1). The control points $Q_{\mathtt{uv}}(i)$ of the Nurbs patches are derived from the input mesh points $P_{\mathtt{uv}}(i)$ in two steps.

1. [Knot Insertion]
   For each quad, we define a submesh of the Catmull-Clark mesh that includes all subfacets of the quad and those sharing at least one node with the quad: we borrow one layer of nodes from all direct and diagonal neighbor quads to arrive at the submesh delineated by the double line in Fig. 2.
   We interpret the two perpendicular families of grid lines of quad i (without $P_{30}(j)$ for $j \notin \{i-1, i, i+1, i+2\}$) as the control net of an order 4 Nurbs patch with uniform u and v knot sequences $-4, -3, -2, -1, 0, 1, 2, \ldots$. To bring this Nurbs patch into standard interpolating form, we insert three knots first at $u = 0$ and $u = k$ then at $v = 0$ and $v = k$. Figure 4 illustrates the (standard) knot insertion procedure at $u = 0$ for a grid line indexed by $\mathtt{u}$. The new points $Q_{\mathtt{u}}$ are obtained from the old points $P_{\mathtt{u}}$ via

   $$Q_0 = (P_{-3} + 4P_0 + P_3)/6, \quad Q_1 = (2P_0 + P_3)/3.$$

   For every edge with at least one extraordinary mesh node we insert a second knot, at 1 for $u = 0$ (at $k-1$ for $u = k$):

   $$Q_2 = (P_0 + 2P_3)/3, \quad Q_4 = (2P_3 + P_6)/3.$$

   All remaining $Q_{3i,3j} = P_{3i,3j}$, except for the corner point $Q_{00}(1) = Q_{00}(2) = \ldots = Q_{00}(n)$ which we place directly on the Catmull-Clark limit surface [6]:

   $$Q_{00}(1) = \ldots = Q_{00}(n) = \frac{\sum nP_{00}(i) + 4P_{30}(i) + P_{33}(i)}{n(n+5)}.$$

2. [Corner Smoothing]
   If we were to stop at this point, the Nurbs patches would only meet with position continuity close to extraordinary mesh nodes – although with C^2 continuity everywhere else. To obtain tangent continuity, we modify the control points near each extraordinary mesh node. The enlargement of Figure 3 shows the relevant double subscripts. We define, for every n, two n by n matrices A_n and B_n with rows $i = 1, \ldots, n$ and columns $j = 1, \ldots, n$ and entries

   $$A_n(i,j) = \frac{2a}{n}\cos(\frac{2\pi}{n}(i-j)) \qquad a = 1\text{(default) and}$$
   $$B_n(i,j) = \begin{cases} (-1)^{n_{i-j}} & \text{if } n \text{ is odd,} \\ (-1)^j - 2n_{i-j}(-1)^{j-i}/n & \text{if } n \text{ is even,} \end{cases}$$
   $$n_{i-j} = \mod(n+i-j, n).$$

   For example, $B_6(3,3) = -1$. We collect the points $Q_{\mathtt{uv}}(i) \in \mathbf{R}^3$ generated by Knot Insertion for $i = 1, \ldots, n$ and $\mathtt{uv} \in \{00, 10, 20, 40\}$ into $\bar{Q}_{\mathtt{uv}} \in \mathbf{R}^{n\times 3}$. Only if n is even and greater than 4, do we compute $r = \sum_{i=1}^{n}(-1)^i\bar{Q}_{40}(i)/n$ and if $r \neq 0$ we add, for each i, $h_i = -(-1)^i r$ to $Q_{40}(i) = Q_{04}(i-1)$, $Q_{41}(i)$ and $Q_{14}(i-1)$ so that $\sum_{i=1}^{n}(-1)^i Q_{40}(i) = 0$ and $Q_{40}(i) = (Q_{41}(i) + Q_{14}(i-1))/2$. Otherwise all coefficients remain unchanged except

   $$\begin{aligned} Q_{10} &= Q_{00} + A_n\bar{Q}_{10}, \\ Q_{20} &= (Q_{40} + 6Q_{10} - 2Q_{00})/5, \\ Q_{11} &= B_n\Big(Q_{10} + \frac{\cos(2\pi/n)}{6}(Q_{40} - Q_{20})\Big). \end{aligned}$$

   For $i = 1, \ldots, n$, we copy $Q_{\mathtt{v}0}(i+1) = Q_{0\mathtt{v}}(i)$ for $\mathtt{v} \in \{1, 2, 4\}$ and add $Q_{20}(i) - \bar{Q}_{20}(i)$ to $Q_{21}(i)$ and $Q_{12}(i-1)$.

4 Smoothness and Approximation

We *claim*: the output Nurbs patches are internally parametrically C^2 and join the neighbors parametrically C^2 except possibly at extraordinary mesh nodes, across edges $u \in \{0, 1\}, v \in [0, 1]$ respectively $v \in \{0, 1\}, u \in [0, 1]$ shown in bold in Figure 2. Across these edges the surface is at least tangent continuous.

Before the Knot Insertion step, submeshes of adjacent patches overlap in three layers of cubic spline control points with a single knot associated with the center control point. Standard spline theory then guarantees that the patches join parametrically C^2 across the boundary edges $u \in \{0, k\}, v \in [1, k-1]$ and $v \in \{0, k\}, u \in [1, k-1]$. Knot insertion does not change the Nurbs surface and the smoothness of the transition. Therefore each patch is also internally C^2 before the Corner Smoothing step.

Corner Smoothing enforces tangent continuity for the remaining parameter interval at the corner from 0 to 1: with u_i, v_i the parameters of patch i and $v_i = u_{i+1} = t \in [0, 1]$ the parameter along the boundary between patch q_i and patch q_{i+1}, and $c = \cos(\frac{2\pi}{n})$, Corner Smoothing enforces the polynomial equation

$$2c(1-t)^2\frac{\partial}{\partial v_i}q_i(0,t) = \frac{\partial}{\partial u_i}q_i(0,t) + \frac{\partial}{\partial v_{i+1}}q_{i+1}(t,0).$$

The equation is easily verified by inserting two more knots at 1 and equating the four resulting Bézier coefficients in the variables Q. For example, at a corner point ($t = 0$) with $n = 3$ neighbors $c = -0.5$ and $-\partial q_i/\partial t = \partial q_i/\partial u_i + \partial q_{i+1}/\partial v_{i+1}$. This relation is enforced by applying A_n. The double root of $(1-t)^2$ at $t = 1$ is equivalent to the collinearity of $Q_{l1}(i)$, $Q_{l0}(i)$ and $Q_{1l}(i-1)$ for $l = 2$ and $l = 4$. The one remaining equation, for the mixed derivatives at $t = 0$, holds due to the choice of B_n and, if n even and greater than 4, the perturbation by h_i. The matrix B_n is derived following [10] but has slightly different entries in the even case.

The continuity across the knot line $u_i = 1$ and $v_i \in [0, 1]$ is that of an order 4 spline with a double knot and therefore is at least C^1. If $r \neq 0$ the joint movement of the nodes $Q_{41}(i)$, $Q_{40}(i)$ and $Q_{14}(i-1)$ preserves collinearity and thereby C^1-ness of the transition between adjacent Nurbs patches.

Unless $r \neq 0$, the Nurbs patches differ from the Catmull-Clark limit surface only near the corners for $(u, v) \in (0, 1]^2$. The difference increases with oscillation of the Catmull-Clark mesh at the extraordinary mesh node since application of A_n and B_n average the data and the choice of Q_{20} removes inflections. Each step of subdivision halves the extent of the region.

5 Conclusion

The strengths of the PCCM algorithm are that it (a) converts Catmull-Clark meshes to closed-form, smoothly-connected, standard Nurbs patches, (b) does so with simple, explicit formulas, (c) integrates seamlessly with the array-based view of subdivision (see the Appendix) and (d) remains local so that almost all patch transitions across patch boundaries are parametrically C^2.

Acknowledgement: I thank the referees, David Lutterkort, Malcolm Sabin, Andy Shiue and Georg Umlauf at SurfLab for their constructive comments.

References

[1] A. A. Ball and D. J. T. Storry. Conditions for tangent plane continuity over recursively generated B-spline surfaces. *ACM Trans. on Graphics*, 7:83–102, July 1988.

[2] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10:350–355, Oct 1978.

[3] Tony DeRose, Michael Kass, and Tien Truong. Subdivision surfaces in character animation. *Proceedings of SIGGRAPH 98*, pages 85–94, July 1998.

[4] OpenGL Foundation. http://trant.sgi.com/opengl/docs/man_pages/hardcopy/GL/html/glu/nurbssurface.html.

[5] Cindy M. Grimm and John F. Hughes. Modeling surfaces of arbitrary topology using manifolds. *Proceedings of SIGGRAPH 95*, pages 359–368, August 1995.

[6] Mark Halstead, Michael Kass, and Tony DeRose. Efficient, fair interpolation using Catmull-Clark surfaces. *Proceedings of SIGGRAPH 93*, pages 35–44, August 1993.

[7] Adi Levin. Interpolating nets of curves by smooth subdivision surfaces. *Computer Graphics*, 33:57–64, 1999. Appendix.

[8] A. H. Nasri. Boundary-corner control in recusive subdivision surfaces. *CAD*, 23(6):405–410, 1991.

[9] A. H. Nasri and J. Peters. Computing volumes of solids enclosed by recursive subdivision surfaces. *Comp. Gr. Forum*, 16(3), September 1997.

[10] J. Peters. C^1-surface splines. *SIAM J. Numer. Anal.*, 32(2):645–666, 1995.

[11] J. Peters and U. Reif. Analysis of generalized B-spline subdivision algorithms. *SIAM J. on Numer. Anal.*, 35(2):728–748, April 1998.

[12] H. Prautzsch. Freeform splines. *CAGD*, 14(3):201–206, 1997.

[13] Jos Stam. Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values. *Proceedings of SIGGRAPH 98*, pages 395–404, July 1998.

[14] D. Zorin. *Subdivision and multiresolution surface representation.* PhD thesis, Caltech, 1997.

[15] D. Zorin. Implementing subdivision and multiresolution meshes. *Chapter 6 of Course notes 37 of SIGGRAPH 99*, Aug 1999.

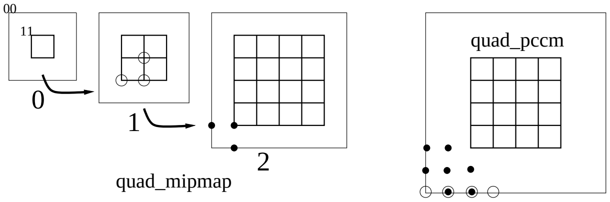


Figure 5: Collection ∘ and distribution • of points; (*left*) the Catmull-Clark mipmap at levels $\ell = 0, 1, 2$, (*right*) PCCM at level $\ell = 2$.

6 Appendix: array-based data structures

Catmull-Clark subdivision and PCCM can be implemented using only the connectivity information of the quads and a mipmap of control point arrays as would be input to `gluNurbsSurface` [4].

- For each quad, store a *mipmap* of arrays. The array at level ℓ is of size $k + 2$ by $k + 2$ by 3, $k = 2^\ell + 1$ and contains the x, y, z node positions of subdivision ℓ . Entry [1][1], short 11 (and symmetrically $1k$, $k1$ and kk), holds the position of a corner node (cf. Figs. 2, 5,*left*). If the corner node is an extraordinary mesh node then 00 is undefined. If $n = 3$ then the entries in 01 and 10 agree. PCCM outputs into an array of size $k + 4$ by $k + 4$ by 3. Entry 00 is the position of a corner coefficient.

- For each extraordinary mesh node, a *connectivity list* stores for each incident quad the global index and the corner of the quad corresponding to the extraordinary mesh node.

Catmull-Clark subdivision with this data structure consists of two parallelizable steps:

a. For each quad, create the mipmap level $\ell + 1$ from level ℓ by the regular subdivision rules (1).

b. For each extraordinary mesh node, use the connectivity list to
– *collect* at level ℓ, $P_{00} = P_{00}(1)$ and $P_{30}(i)$, $P_{33}(i)$ for all i (c.f. Figures 2, 5*left*: if P_{00} is entry 11 then $P_{30}(i)$ and $P_{33}(i)$ are entry 21 and 22).
– *Compute* new locations for $P_{30}(i)$ and $P_{33}(i)$ at level $\ell + 1$ from rules (1) and P_{00} from rule (2) (or use Sabin's curvature bounded rule [7], Appendix).
– *Distribute* the new locations to level $\ell+1$ (Entry 01 receives $P_{30}(i + 2)$ and entry 10 receives $P_{30}(i - 1)$)).

PCCM with this structure consists of two parallelizable steps:

a. For each quad, apply Knot Insertion.

b. For each extraordinary mesh node, use the connectivity list to
–*collect* $Q_{00}(1)$ and $\bar{Q}_{\mathtt{uv}}(i)$, $\mathtt{uv} \in \{10, 20, 40\}$.
–*Compute* $Q_{\mathtt{uv}}(i)$, $\mathtt{uv} \in \{10, 20, 11\}$, $Q_{20} - \bar{Q}_{20}$ and possibly $Q_{40} - \bar{Q}_{40}$.
–*Distribute* $Q_{\mathtt{uv}}(i)$, $\mathtt{uv} \in \{10, 01, 20, 02, 11\}$ and add to $\{21, 12\}$ and possibly $\{04, 40, 14, 41\}$.

All space for subdivision level ℓ can be allocated at the outset, and the connectivity list remains unchanged throughout. The quad-arrays can be input directly to `gluNurbsSurface` or displayed as quad-meshes.