

Pattern Mapping with Quad-Pattern-Coverable Quad-Meshes

Shiyu Hu^{1a}, Qing Xing^a, Ergun Akleman^a, Jianer Chen^a, Jonathan Gross^b

^aTexas A&M University

^bColumbia University

Abstract

We show that for every surface of positive genus, there exist many quadrilateral manifold meshes that can be texture-mapped with locally translated copies of a single square-texture pattern. This implies, for instance, that every positive-genus surface can be covered seamlessly with any of the 17 plane symmetric wallpaper patterns. We identify sufficient conditions for meshes to be classified as “quad-pattern-coverable”, and we present several methods to construct such meshes. Moreover, we identify some mesh operations that preserve the quad-pattern-coverability property. For instance, since vertex insertion remeshing, which is the remeshing operation behind Catmull-Clark subdivision, preserves quad-pattern-coverability, it is possible to cover any surface of positive genus with iteratively finer versions of the same texture.

Keywords: Modeling



(a) Escher-like pattern (b) twill-weave pattern

Figure 1: Examples of anisotropic/periodic pattern mapping with quad-pattern-coverable meshes, where surfaces are covered by a single anisotropic, periodic wallpaper pattern from Figure 2.

1. Introduction

Texture mapping [1] is very popular in computer graphics applications, since it allows the creation of complicated-looking images without increasing the complexity of the surface geometry. Texturing arbitrary surfaces with repetitive patterns (i.e, pattern mapping [2]) is particularly useful, since repeating a pattern reduces the memory cost, by mapping the same texture image to all faces. Moreover, pattern mapping does not

require painting a texture image for each surface or generating a global texture map on each surface. Pattern mapping can potentially provide natural-looking materials, such as stone, wood, or marble, as well as human-made materials, such as wallpapers or repeating tiles.

One of the main challenges arising when mapping patterns to arbitrary polyhedral meshes is to avoid texture discontinuities caused by singularities in the mesh structure. Texture discontinuities differ from shape discontinuities, although the source of both kinds of discontinuities is the same. In quad-meshes, non-4-valent vertices correspond to mesh singularities. For instance, in Catmull-Clark subdivision, non-4-valent vertices cause C^2 discontinuities, but only at points that correspond to non-4-valent vertices [3]. However, discontinuities can also appear at seams along the edges, which can be visually distracting. Unfortunately, it is not always possible to avoid non-4-valent vertices, since 4-regular quad-meshes exist only for genus-1 surfaces.

In this paper, we show that for any surface of positive genus, there exist quad-meshes that do not cause texture

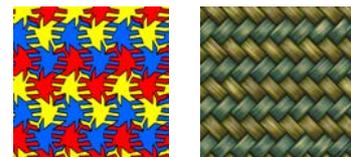


Figure 2: Wallpaper patterns that are used in Figures 1(a) and 1(b).

¹Email address: syhu@cse.tamu.edu (S.Hu)

discontinuities. Using such quad-meshes, which we call *quad-pattern-coverable meshes* (abbr. QPC), it is possible to seamlessly cover a surface of positive genus periodically with any plane symmetric wallpaper pattern. Figures 1(a) and 1(b) are examples of a QPC mesh covered periodically by one of the two wallpaper patterns shown in Figure 2.

QPC meshes can also be covered aperiodically, by using more than one quad pattern. Periodic and aperiodic patterns are contrasted in §2. After introducing some definitions, we establish sufficient conditions in §3 for a mesh to be classified as QPC.

Our results imply that a quad-mesh is not QPC if the valence of at least one vertex is not divisible by 4. This observation implies, in turn, that there exists no genus-0 QPC mesh, since a genus-0 quad-mesh always has some vertices with valences smaller than 4. For positive genus surfaces, by way of contrast, there exist a wide variety of mesh structures that can satisfy the sufficiency conditions. Theoretical and practical algorithms for construction of QPC meshes are provided in §6 and §5.

One obvious problem with vertex valences that are large multiples of 4 is that it is hard to avoid texture distortions near such a vertex. It is preferable, therefore, to reduce the large multiples to valence 8, the smallest non-trivial multiple of 4. We introduce an operation in §6 that can transform the $4k$ -valent vertices in a mesh into 8-valent vertices, while preserving the QPC property, regardless of surface genus. Note that 8 valence vertices in saddle regions do not produce significant texture distortions as it can be seen in our examples such as the ones shown in Figures 1 and 4.

Vertex insertion remeshing that replaces each quad by four smaller ones, as the Catmull-Clark subdivision algorithm being the prime example, preserve quad-pattern-coverability (see §3). It is possible, therefore, to cover any surface of positive genus with iteratively finer versions of a given texture. Catmull-Clark subdivision is also useful in creating smooth models. In our examples such as Figure 1, starting with very coarse QPC meshes, we obtain smooth versions by Catmull-Clark subdivision. Instead of re-texturing the mesh, we bi-linearly interpolate texture coordinates. The result is equivalent to using curved quads, as shown in Figure 3.

We also show in §2 that any wallpaper pattern can be created by translations of one rectangular image, and that such a rectangular pattern can be directly mapped to a toroidal surface, using a $(4, 4)$ mesh. Thus, wallpa-

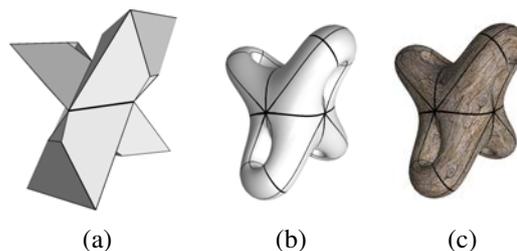


Figure 3: (a) a quad-pattern-coverable mesh; (b) the same mesh after Catmull-Clark subdivision; (c) the pattern superimposed on the quad-mesh.

per patterns can seamlessly cover any toroidal surface. However, other surfaces do not have such a toroidal parameterization and cannot be unfolded onto the Euclidean plane. QPC meshes provide an alternative parameterization that allow mapping of such patterns to any positive-genus surface.

A significant advantage of using wallpaper patterns is that the seamless texturing does not require any unique solution. Cyclic translations of wallpaper patterns are also wallpaper patterns. Therefore, one can control the results, by cyclic translations of the wallpaper pattern in each quadrilateral. Using this property, it is also possible to create seamless texture animations.

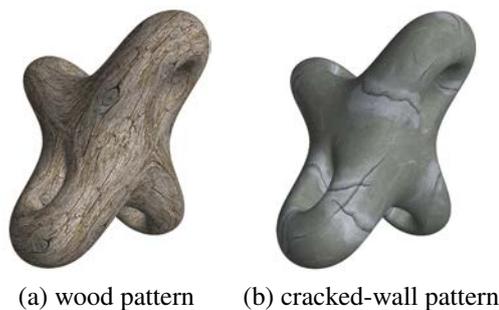


Figure 4: Examples of anisotropic and aperiodic pattern mapping with quad-pattern-coverable meshes, where surfaces are covered aperiodically by more than one quad pattern, with shared boundaries from Figure 5.

Quad-pattern-coverability does not require using a single pattern. We also show that if the boundaries of the patterns match [4], then it is possible to obtain aperiodic covering. For example, the aperiodic covering in Figures 4(a) and 4(b) use the four quad patterns in Figures 5(a) and 5(b) respectively.

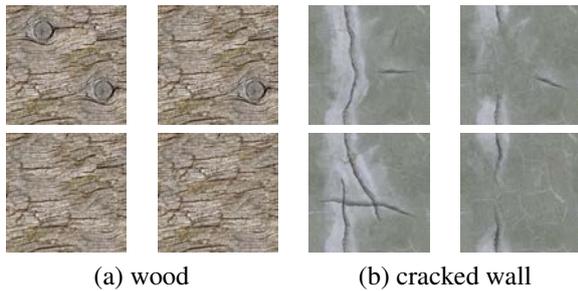


Figure 5: Four quad patterns that are used to texture the object in Figures 1.(c) and (d).

2. Pattern Mapping

Pattern mapping has been used by artists in cultures all around the world to cover planar surfaces. The most popular patterns are wallpaper patterns that provide periodic covering of the plane.

2.1. Periodic Patterns

The periodic patterns in the Alhambra in Granada are probably the best-known historical examples of pattern mapping. Such periodic patterns have appeared in architectural design throughout history, and they are frequently used by almost every civilization in their wallpapers and wall decorations, and also in their ceilings, floor tiles, street pavements, and facades of buildings. Despite their widespread occurrence in art and architecture, the theoretical classification of periodic symmetric patterns did not begin until the early twentieth century. Fedorov in 1891 established that there are only 17 mathematically distinct types of patterns with which to cover the plane. These pattern types today are also known as *wallpaper groups*, *periodic groups*, or *(plane) crystallographic groups*.

Each of these groups is a collection of symmetry operations: translation, rotation, reflection, and glide reflection. A rotation can be of period 2, 3, 4 or 6. These operations are called isometries, since they preserve the distance between any two points. In group-theoretic literature, the periodic symmetry wallpaper groups are usually denoted $p1$, $p2$, $p4$, pm , pmm , $p4m$, $p4g$, cm , cmm , pg , pmg , pgg , $p4g$, $p3$, $p6$, $p3m1$, $p31m$ and $p6m$. These groups can be partitioned into rectangular symmetries and hexagonal symmetries. The last five of them ($p3$, $p6$, $p3m1$, $p31m$, $p6m$) have hexagonal symmetries, and the first 12 have rectangular symmetries. (See [5] for detailed discussion.)

From a graphic designer's perspective, the most important implication of the symmetry wallpaper groups is

their identification of symmetry operations. This identification encouraged artists such as Escher, Briss and Mehmedov to discover new and interesting patterns. Knowledge of the 17 symmetry wallpaper groups facilitates the design of symmetric patterns, even with only paper and pen as tools. It is now very easy to find and create wallpaper patterns since there exist many interactive open-source and commercial systems.

One of the key ideas in this paper is to encapsulate all 17 symmetries within a single $p1$ rectangular pattern. This is not unexpected, in view of the fact that every symmetric tiling is a topological covering space of a torus, which is representable by a unit rectangle with identified opposite sides (see [6]). As shown in Figure 6, such a unit rectangle can be drawn with its sides labeled North (N), East (E), South (S), and West (W), which we call a rectangle with NESW-boundary. By pasting N to S and E to W, we construct a torus.

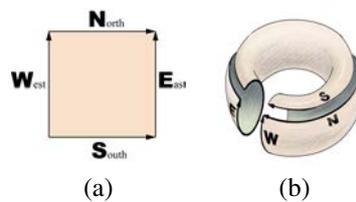


Figure 6: (a) the unit cell; and (b) the embedding the unit cell to a torus.

It is also possible to create larger units, with m unit cells in the North-South direction and n unit cells East-West, as in Figure 7. These $m \times n$ -unit cells can also cover a toroidal surface.

The structure of a toroidal mesh that corresponds to such a quad-pattern-covering is called $(4, 4)$, which indicates each face is a quadrilateral and that 4 quadrilaterals meet at each vertex. This single rectangle corresponds to $p1$ symmetries since it only requires translation in North-South and East-West directions to obtain the wallpaper.

Using the idea of a single $p1$ rectangle to represent all symmetries, the rectangular unit will consist of subunits that are created by rotation, reflection, and glide reflection operations. Perhaps surprisingly, even a hexagonal symmetry can be represented by repeating a $p1$ rectangular unit [7], as illustrated in Figure 8. In other words, regardless of its symmetry wallpaper group, any wallpaper pattern can be represented by a *rectangular texture image* mapped on a toroidal surface. Note that our QPC meshes is only necessary and sufficient for

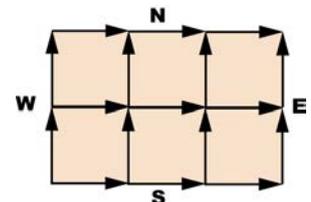


Figure 7: Unit cells can combine to form a larger unit cell.

p1. However, as mentioned above, we can create all 17 symmetric wallpaper patterns based on QPC meshes by carefully designing the texture patterns, as shown in Figure 9. Moreover, with the similar idea for QPC meshes (presented later), it is not hard to obtain the necessary and sufficient conditions for specified wallpaper groups.

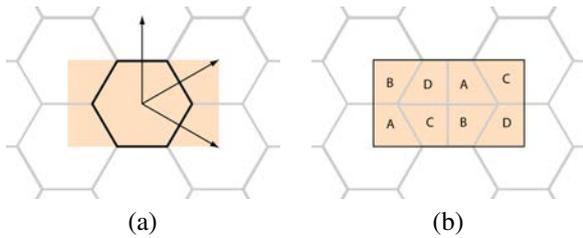


Figure 8: Obtaining hexagonal symmetries using one rectangular unit cell by using subunits.

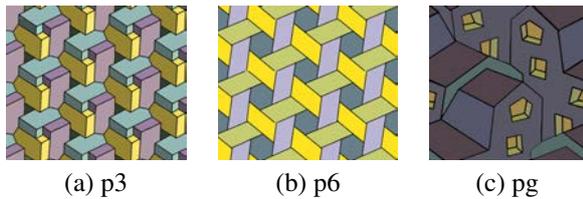


Figure 9: Single p1 rectangles can create any 17 symmetries. These three examples show patterns that provide p3, p6 and pg symmetries.

Representation of any wallpaper pattern by a rectangular texture image allows us to ignore the underlying symmetry pattern. Mapping the same texture image to every quadrilateral simplifies texture mapping. Care is needed only for matching the North-South and the East-West boundaries. With QPC meshes, a unique solution follows after a single edge of the mesh is labeled as either the N-S or the E-W boundary, by consistently labeling the other edges. This can be implemented with a few lines of code. The anisotropic periodic texture-mapped surfaces in Figures 1(a) and 1(b) are obtained in this way, each by using one of the two texture images from Figure 2. Although periodic tiling is useful, if the tile sizes are very small, the repetitiveness of the patterns becomes visually less interesting. Accordingly, we include aperiodic patterns in our study.

2.2. Aperiodic Patterns

Wang tiles [8], which are one of the best-known sources of aperiodic patterns to cover the infinite plane, are related to our work. They consist of a set of square tiles that cannot be rotated, i.e., they have fixed North-South

and East-West orientations with color-coded edges. Berger [9] constructed aperiodic sets of 20426 and 104 Wang tiles, thereby demonstrating the existence of such sets. There now exists sets with 13 and 6 tiles. Wang tiles were introduced to computer graphics by Stam [10] and later again by Cohen et al.[11], for aperiodically texturing planes and toroids.

For extending our work to include Wang tiles, we assign colors to NESW-boundaries such as **NE****SW**, where blue North must be matched to blue South. Although this approach is feasible, it is preferable from a practical perspective if aperiodicity can be achieved with a set of tiles, each of which has exactly the same NESW-boundaries, in which case the designer can focus exclusively on internal parts of the tiles. Neyret and Cani [4] used a continuous-boundary approach to create such a set of tiles. Figure 10 shows a Wang tile constructed from a set of four rectangles with the same NESW-boundaries.

In practice, we do not need Wang tiles for aperiodicity. Having just a few tiles with matching boundaries is sufficient to create visual aperiodicity. Such tiles can effectively be designed using a method similar to image quilting [12]. In this case, the only difference from periodic pattern creation is that we apply a randomly chosen texture image to each quadrilateral. With QPC meshes, the orientation of each texture image is always uniquely defined, after fixing only one edge of the mesh. The anisotropic aperiodic texture-mapped surfaces in Figures 1(c) and 1(d) are obtained in this way using texture images shown in Figures 5(a) and 5(b).

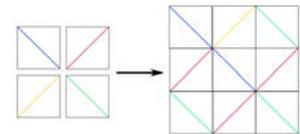


Figure 10: How to create Wang tiles from a set of rectangles with the same NESW boundaries.

In what follows, we show that for every surface of positive genus, there exist mesh-structures that can be covered with a pattern similar to a (4, 4) covering. Using such mesh-structures, any surface of positive genus can be covered seamlessly with periodic and aperiodic patterns. In the next section, we introduce the conditions that a mesh must satisfy, in order to allow such a quad-pattern-covering.

3. Quad-Pattern-Coverability of a Mesh

Graphs in our discussion can have multiple edges and self-loops. For an oriented edge $e = \langle v, w \rangle$, (where v and

w are the head and tail, respectively, of e), we denote by \bar{e} the reversed oriented edge $\langle w, v \rangle$ (oriented edges correspond half-edges in Computer Graphics [13]). A graph is n -regular if every vertex in the graph has valence n . A *surface* is a closed orientable 2-manifold. A *mesh* is a graph embedded on a surface, and a *quad-mesh* is a mesh in which all faces are 4-sided. In an n -regular mesh, every vertex has valence n , but the faces need not be all the same size.

Let P be a rectangle pattern (i.e., an image that fills a rectangle). We name its four sides East (E -), South (S -), West (W -), and North (N -), so that a clockwise traversal of its boundary encounters the sides in that order. The pattern P is *periodic* if its mapping to a torus, as in Figure 6(b), is seamless. We are most interested in covering a quad-mesh by a periodic rectangle pattern so that the covering is “seamless”. The QPC property is a formalization of “seamless” covering.

Definition A quad-mesh M is *quad-pattern coverable* if its faces can be covered by a periodic rectangle pattern P , such that for any two adjacent faces of M , the common edge between them is either the E -side of one of the two faces and the W -side of the other, or else the S -side of one of the two faces and the N -side of the other.

Note that the quad-pattern coverability of a quad-mesh M depends only on the mesh M and is independent of any particular periodic rectangle pattern. Here is the central theoretical problem of this section and of the next:

Given a quad-mesh M , is M a QPC mesh?

Recall that for a mesh M on a surface S , the *dual mesh* M' is a graph embedded on the same surface S , such that

- there is a bijective correspondence between the faces of M and the vertices of M' and a bijective correspondence between the edges of M and the edges of M' ;
- if faces f_1 and f_2 have edge e in common, then the dual edge e' joins the vertices f'_1 and f'_2 of M' that are dual to f_1 and f_2 , respectively.

The four oriented edges incident to each vertex f' of M' can be labeled (E -), (S -), (W -), and (N -), in a cyclic ordering consistent with the rotation at f' . A collection of such labelings, one for each vertex of M' , is called an *oriented-edge labeling* of M' . The four labels around each vertex f' induce labels of the face f to which f' is

dual. Based on this observation, we introduce the following definition.

Definition Let M' be a 4-regular mesh embedded on a surface S . An oriented-edge labeling of M' is *consistent* if for every oriented edge labeled (E -), (S -), (W -), or (N -), the oppositely oriented edge is labeled (W -), (N -), (E -), or (S -), respectively.

We immediately infer the following theorem.

Theorem 3.1. *A quad-mesh M is quad-pattern coverable if and only if its dual mesh M' can be labeled consistently.*

The rest of this section is to characterize meshes that have consistent labelings. Toward that goal, let M be a mesh on a surface S , with $\beta = \{e_1, e_2, \dots, e_k\}$ a closed walk in M , and $e_i = \langle v_i, v_{i+1} \rangle$ an oriented edge, for $i = 1, \dots, k$, (with $v_{k+1} = v_1$). The *corner-cost* of the walk β at the vertex v_i is defined to be the number of face corners around v_i from the oriented edge $\bar{e}_{i-1} = \langle v_i, v_{i-1} \rangle$ to the oriented edge $e_i = \langle v_i, v_{i+1} \rangle$ (proceeding around v_i according to the orientation). Figure 11 shows the corner-cost of a walk when it passes through a vertex of degree 4.

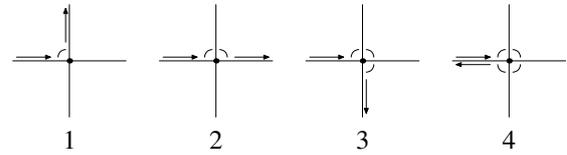


Figure 11: Each number indicates the corner-costs of the corresponding walk as it passes through a vertex.

The *corner-cost* of a walk β is defined to be the sum of corner costs at all vertices traversed by β . In computing the corner-cost, we start along an oriented edge e_1 , we traverse the complete walk β , and we complete the traversal by returning back to the starting oriented edge e_1 .

We also define, naturally, the *length* of the walk β to be the number of traversals of oriented edges along the walk β .

Theorem 3.2. *Let M be a 4-regular mesh embedded on a surface S . Then mesh M can be labeled consistently if and only if for every closed walk in M , the corner-cost of that walk plus twice its length is divisible by 4.*

PROOF. Let β be a closed walk in the mesh M with corner-cost q and length k such that $q + 2k$ is divisible by 4. Quite simply, this means

if k is even, then $q \equiv 0 \pmod{4}$, and

if k is odd, then $q \equiv 2 \pmod{4}$.

To help to analyze the corner-cost of a walk β in the mesh M , we introduce a special directed graph, the *corner-graph* D , which is depicted in Figure 12. The horizontal edges and the vertical edges in D are called *c-edges* (i.e., corner-edges) and the edges between the two rectangles are called *t-edges* (i.e., transition-edges).

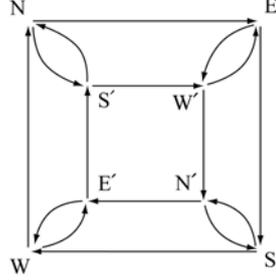


Figure 12: The corner-graph D .

(\Rightarrow) Suppose first that mesh M has a consistent labeling L , and that we have a walk

$$\beta = \{e_1, e_2, \dots, e_k\}$$

in M , where $e_i = \langle v_i, v_{i+1} \rangle$ is an oriented edge for each i (with $v_{k+1} = v_1$). We now construct a longer walk β' in the corner-graph D , which has some ‘‘corner-steps’’ between consecutive β -steps that we denote by e'_i .

We can assume, without loss of generality, that the oriented edge e_1 is the E -edge of vertex v_1 . Then we take vertex E as the starting vertex of the walk β' in D . Since the labeling L is consistent, the oriented edge $\bar{e}_1 = \langle v_2, v_1 \rangle$ must be the W -edge of vertex v_2 . We take the directed edge e'_1 of walk β' in corner-graph D to be the t -edge from vertex E to vertex W' , which signifies that after walk β traverses the E -edge $e_1 = \langle v_1, v_2 \rangle$ out of v_1 , it next takes the W -edge $\bar{e}_1 = \langle v_2, v_1 \rangle$ into v_2 . Now suppose that the walk β has corner-cost h at the vertex v_2 . Then we let walk β' traverse h directed c -edges on the inner rectangle in the corner-graph, after which we observe that the walk β' must be at a vertex of D whose name matches the label of the oriented edge $e_2 = \langle v_2, v_2 \rangle$ of mesh M . In general, each maximal subsequence of consecutive c -edges in β' corresponds to the corner-cost of the walk β as it passes through a vertex in M .

If the walk β is closed, then it returns to the origin of the oriented edge e_1 . As a consequence, the directed walk β' in D ends up either at the vertex E or at the vertex E' (recall that we assumed that e_1 is the E -edge of v_1).

If the length k of walk β is even, then the walk β' contains an even number of t -edges, so it ends up at the vertex E , and it is closed. The structure of the corner-graph D implies that the number of c -edges in a closed directed walk is divisible by 4. Thus, $q + 2k$ is divisible by 4. Alternatively, if the length k is odd, then the

walk β' contains an odd number of t -edges, and thus it ends up at the vertex E' . In this case, the structure of the corner-graph D implies that the number of c -edges in a closed directed walk is congruent to 2 mod 4. Thus, once again, the number $q + 2k$ is divisible by 4. This completes the proof for this direction.

(\Leftarrow) For the other direction, suppose that every closed walk in M satisfies the condition given in the theorem. We show how to construct a consistent labeling on the oriented edges of the mesh M .

Pick a root vertex v of mesh M , and assign the labels E -, S -, W -, and N - to the oriented edges originating at v , in a manner consistent with the orientation of surface S . Proceeding inductively, suppose that a labeling at the vertex w_1 has been assigned and that the labeling for the neighboring vertex w_2 has not been assigned. Then assign to the edges originating at vertex w_2 the unique four labels that are consistent with those at w_1 . Continue until all oriented edges of mesh M are labeled. We must verify that this labeling process L is consistent. For this, it is sufficient to show that the labeling at a vertex w does not depend on the choice of a labeled neighbor vertex.

Suppose, as an inductive hypothesis, that so far, the labelings for a set of vertices are consistent. Let w be an as yet unlabeled vertex, with neighbors w_1 and w_2 of w that have been assigned the labelings $l(w_1)$ and $l(w_2)$, respectively. We will show that the labeling $l(w_1)$ and the labeling $l(w_2)$ would induce the same labeling for the vertex w .

For this, we let $\beta_1 = \{v_0 = v, v_1, \dots, v_r = w\}$ be a walk of corner-cost p from the root vertex v to w_1 and $\beta_2 = \{u_s = w, u_{s-1}, \dots, u_0 = v\}$ a walk of corner cost q , such that all vertices in β_1 and β_2 are already labeled. Then

$$\beta_3 = \{v_0 = v, v_1, \dots, v_r = w = u_s, u_{s-1}, \dots, u_0 = v\} \quad (1)$$

is a closed walk in M . By the premise for this direction of the theorem, if the length $r + s$ of β_3 is even, then the corner-cost $p + q$ of β_3 is divisible by 4; and if $r + s$ is odd, then $p + q$ is divisible by 2, but not by 4.

Case 1. We first suppose that the length $r + s$ of walk β_3 is even and that $p + q \equiv 0 \pmod{4}$. Since $p + q \equiv 0 \pmod{4}$, it follows that the walk β_2^{-1} has corner cost $4 - p \pmod{4}$, and in turn, that the walk β_2 has corner cost $p \pmod{4}$. Since, furthermore, the lengths r and q are of the same parity, it follows that the induced walks β_1' and $(\beta_2^{-1})'$ terminate at the same vertex of the corner graph, and accordingly, that they induce the same labelings at w .

Case 2. Now suppose that the length $r + s$ of walk β_3 is odd and that $p + q \equiv 2 \pmod{4}$. Since $r + s$ is odd, it

follows that r and s are of different parity, which implies that one of the induced walks β'_1 and $(\beta_2^{-1})'$ in the corner graph terminates on the inner 4-cycle and the other on the outer 4-cycle. Since $p + q \equiv 2 \pmod{4}$, it follows that the locations of these termination vertices are diagonally opposite. Thus, these termination vertices have the same label, except that one of them is marked *prime*. It follows that the induced walks β'_1 and $(\beta_2^{-1})'$ induce the same labelings at w .

This completes the proof of the theorem. \square

Theorem 3.2 immediately implies our main result regarding the quad-pattern coverability of a mesh.

Corollary 3.3. *A quad-mesh M is quad-pattern coverable if and only if for every closed walk β in the dual mesh M' , the corner-cost of β plus twice its length is divisible by 4.*

Note that the labeling process L in the proof of Theorem 3.2 can be used to test whether a mesh has a consistent labeling: if the mesh has a consistent labeling, then the labeling process L will successfully construct such a labeling, while if the mesh has no consistent labeling, then the labeling process L will be blocked at some stage by an inconsistent structure. Combining this with Theorem 3.1 gives the following corollary.

Corollary 3.4. *There is a linear-time algorithm that tests whether a quad-mesh M is quad-pattern coverable, and when M is quad-pattern coverable, constructs a quad-pattern covering for the mesh.*

Another useful result is that vertex insertion/remeshing, Catmull-Clark subdivision being most widely used example, preserves quad-pattern coverability.

Lemma 3.5. *If a mesh M is quad-pattern coverable, then the mesh M_c after vertex insertion operation on M is also quad-pattern coverable.*

We will not provide a complete proof for this lemma. Vertex insertion operation turns each quad into 2×2 . Therefore, applying vertex insertion operation is the same as using 2×2 version of a given quad-pattern. It is, therefore, the boundaries will still match. The lemma is useful since it shows that any surface of positive genus can iteratively be covered by finer versions of a given texture using Catmull-Clark subdivision.

By Theorem 3.1, the quad-pattern coverability of a quad-mesh M on a surface can be characterized by the consistent labelability of its dual mesh M' , which is a 4-regular graph embedded on the same surface. Moreover, each vertex of degree d in the mesh M corresponds

to a face of size d in the dual mesh M' . Therefore, instead of working on the original quad-pattern coverable mesh M , we can work on the dual mesh M' that has a consistent labeling and reduce the face size for large faces in M' , while keeping the consistent labelability. When we take the dual of the resulting mesh, which has consistent labelings and smaller face sizes, we will get a quad-pattern coverable mesh with reduced vertex degrees.

Lemma 3.6. *Let M be a 4-regular mesh embedded on a surface, with consistent labelings. Then the size of each face is divisible by 4.*

PROOF. Let F be an s -sided face in the mesh M . Take a closed walk β that follows the boundary of the face F in counterclockwise order (i.e., when one traverses along the walk β , the face F is always on the left). The walk β has corner-cost 1 at each vertex of F . Thus, the corner-cost c of the walk β is s . The length l of the walk β is also s . Since M has consistent labelings, it follows from Theorem 3.2 that the sum $c + 2l = 3s$ is divisible by 4. This implies immediately that the size s of the face F is divisible by 4. \square

Most surfaces cannot have a 4-regular M with all faces 4-sided: by the Euler polyhedral equation [6], such a mesh can be embedded only on the torus. On a surface with genus $g \geq 1$, the sum of the valences of all irregular vertices is $8(g - 1) + 4n$, where $1 \leq n \leq 2(g - 1)$ is the number of irregular vertices. The number of irregular vertices can be at most $2(g - 1)$ when all of them are 8-valent. Therefore, we necessarily allow meshes in which some vertex valences are various multiples of 4. Since it is hard to avoid texture distortions near high-valence vertices, it is preferable to reduce the larger multiples of 4 to valence 8, the smallest non-trivial multiple of 4.

4. QPC-Preserving Valence Reduction

Unlike the general quadrilateral mesh editing operations [14, 15], we use dual meshes to reduce vertex valences in a QPC mesh, while keeping the QPC property. Let F be a $4k$ -sided face in a 4-regular mesh M on a surface S , where $k \geq 2$ is a positive integer. Let

$$\beta_F = \{v_1, v_2, \dots, v_{4h-1}, v_{4h}, v_{4h+1}, v_{4h+2}, \dots, v_{4k-1}, v_{4k}\}$$

be the boundary walk of face F , in counterclockwise order, where $1 \leq h \leq k - 1$. The (F, k, h) -operation on the mesh M , illustrated in Figure 13, is defined as follows:

- insert two non-crossing edges $[v_1, v_{4h}]$ and $[v_{4h+1}, v_{4k}]$ into the face F so as to split face F into three faces F_1, F_2 , and F_3 , where the “middle face” F_2 has size 4; and
- delete the edges $[v_{4h}, v_{4h+1}]$ and $[v_{4k}, v_1]$ so as to merge the new middle face F_2 with two other faces F' and F'' .

Note that if the mesh M is 4-regular, then the resulting mesh is also 4-regular.

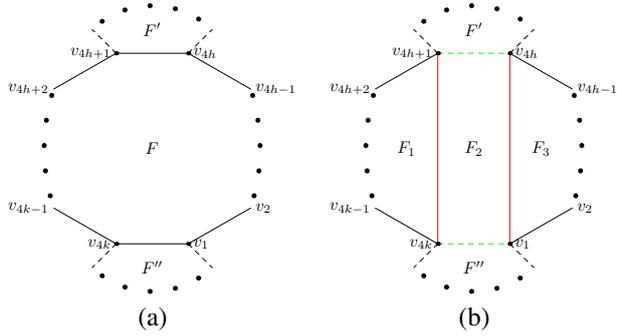


Figure 13: Reducing face-size: (a) before; (b) after.

The (F, k, h) -operation eliminates the face F of large size. However, merging the new middle face F_2 with faces F' and F'' may create another large face with $n' + n'' - 2 + 2$ faces, where n', n'' are the face-sizes of F' and F'' . Nonetheless, if F is an “isolated” large face (i.e., if the faces adjacent to face F are all small), or if there is even a single appropriately located pair F' and F'' with small n', n'' , then the (F, k, h) -operation effectively reduces the large-face problem. For example, suppose that the face F is 16-sided (i.e., $k = 4$) and that for $h = 2$, faces F' and F'' are both 4-sided. Then the (F, k, h) -operation replaces F, F' , and F'' by three 8-sided faces (see Figure 14).

Lemma 4.1. *Let M be a 4-regular mesh embedded on a surface S with consistent labelings, and let F be a $4k$ -sided face of M , where $k \geq 2$. Then for any h with $1 \leq h \leq k - 1$, the (F, k, h) -operation on M will result in a 4-regular mesh with consistent labelings.*

PROOF. Let L_M be a consistent labeling for the mesh M . Consider the boundary walk (in counterclockwise order)

$$\beta_F = \{v_1, v_2, \dots, v_{4h-1}, v_{4h}, v_{4h+1}, v_{4h+2}, \dots, v_{4k-1}, v_{4k}\}$$

as in Figure 13(a). Suppose that the oriented edge $\langle v_1, v_{4k} \rangle$ is the E -edge for the vertex v_1 under the labeling L_M . (The other three cases have the same proof.) Then the oriented edge $\langle v_{4k}, v_1 \rangle$ is the W -edge at vertex

v_{4k} . From this we know that $\langle v_1, v_2 \rangle$ is the S -edge at vertex v_1 , so $\langle v_2, v_1 \rangle$ is the N -edge for the vertex v_2 , and so on. In general, we have for any integer i :

$\langle v_{4i}, v_{4i+1} \rangle$ is the W -edge for the vertex v_{4i} ;

$\langle v_{4i+1}, v_{4i+2} \rangle$ is the S -edge for the vertex v_{4i+1} ;

$\langle v_{4i+2}, v_{4i+3} \rangle$ is the E -edge for the vertex v_{4i+2} ; and

$\langle v_{4i+3}, v_{4i+4} \rangle$ is the N -edge for the vertex v_{4i+3} .

This implies immediately that the oriented edge $\langle v_{4h}, v_{4h+1} \rangle$ is the W -edge at vertex v_{4h} , so the oriented edge $\langle v_{4h+1}, v_{4h} \rangle$ is the E -edge at vertex v_{4h+1} under the labeling L_M .

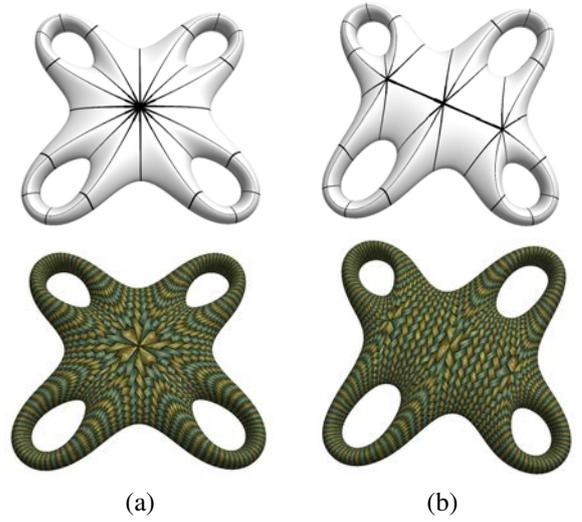


Figure 14: Reduction of a 16-valent vertex into three 8-valent vertices, while preserving QPC property. In this example, we have located high-valence vertices in relatively planar regions to demonstrate the effect of valence reduction better. The positions of such high-valence vertices can be moved to saddle regions to reduce distortion.

Now let M_F be the mesh obtained from mesh M by the (F, k, h) -operation, that is, by inserting edges $[v_1, v_{4h}]$ and $[v_{4h+1}, v_{4k}]$ and deleting the edges $[v_{4h}, v_{4h+1}]$ and $[v_{4k}, v_1]$. As we discussed above, the mesh M_F is a 4-regular graph. Let the mesh M_F retain the labels assigned by L_M to all the other oriented edges, and let $\langle v_1, v_{4h} \rangle$ be the E -edge of vertex v_1 , $\langle v_{4h}, v_1 \rangle$ the W -edge of vertex v_{4h} , $\langle v_{4h+1}, v_{4k} \rangle$ the E -edge of vertex v_{4h+1} , and $\langle v_{4k}, v_{4h+1} \rangle$ be the W -edge of vertex v_{4k} . Clearly, this is a consistent labeling for the mesh M_F . \square

This provides an effective way to eliminate faces of large size, particularly for a mesh with relatively isolated large faces. In QPC meshes, these operations can reduce vertex valences to release the distortion even at

non-saddle points. Here, we do not provide the operation, however It is also possible to move irregular vertices into saddle regions to further reduce the distortions.

5. Constructing QPC Meshes with Permutation Voltage Graphs

In this section, we show that all QPC meshes can be constructed from a simple permutation voltage graph. QPC meshes enjoy the symmetric structure, such that each face can be covered by locally translated copies of periodic rectangle. This inspires us to present QPC meshes with embedded permutation voltage graph[16, 6], which holds the similar property.

We present the definition of the permutation voltage graph first.

Let $G = (V, E)$ be a digraph, which is a graph with directed edges, and X be a symmetric wallpaper group. A *permutation voltage assignment* for G is a function $\alpha : E \rightarrow X$ that labels each edge with a permutation in X . We say the pair $\langle G, \alpha \rangle$ is a *permutation voltage graph*, where G is called the *base graph*.

The *permutation derived graph* $G^\alpha = (V^\alpha, E^\alpha)$ is associated with a permutation voltage graph $\langle G = (V, E), \alpha \rangle$, where $V^\alpha = V \times \{1, \dots, n\}$, and $E^\alpha = E \times \{1, \dots, n\}$. If the edge e is from vertex u to vertex v in G then the edge $e_j = (e, j)$ is from the vertex $u_j = (u, j)$ to the vertex $v_{\alpha(j)} = (v, \alpha(j))$.

The permutation voltage graph provides powerful techniques to construct large and complicated graphs with simple ones. A natural extension is to embed the base graph G on a surface.

Let G be a digraph. A permutation voltage graph (G, α) is *embedded* on a surface, if the underlying undirected graph of G is cellularly embedded on a surface. We correspond a permutation voltage assignment α to the directed edges. Then we consider the embedded permutation derived graph G^α . The digraph G^α is derived from the permutation voltage graph $\langle G, \alpha \rangle$. We embed the undirected G^α on a surface with the rotation system [17, 6] inheriting from the embedded undirected G . Assume there are k edges incident to v in undirected G . Let the rotation at v be e_1, e_2, \dots, e_k , where e_j ($1 \leq j \leq k$) is an oriented edge originating at v in undirected G . It is easy to verify that there are k oriented edges e'_1, e'_2, \dots, e'_k originating at v_i in the undirected G^α , where e'_i is corresponding to e_i . We set the cyclic

order of the oriented edges e'_1, e'_2, \dots, e'_k in the oration at v_i in the embedded undirected G^α is the same with the cyclic order of their corresponding oriented edges at v in embedded undirected G .

We construct an embedded base graph as follows. A *bouquet* B_2 is a undirected graph which consists of one vertex v and two undirected edges, c_1, c_2 , which are self-loops. We embed B_2 on a torus such that the two oriented edges from c_1 (resp. c_2) originating at v are not adjacent in the rotation at v (see Figure 15). Since we have two undirected edges, the graph consists of four oriented edges. These four oriented edges originate at v and they are labeled with $(E-)$, $(S-)$, $(W-)$ and $(N-)$ in clockwise order. The oriented edges $(E-)$ and $(W-)$ share the same edge c_1 , while the oriented edges $(S-)$ and $(N-)$ share the same as c_2 . We set the $(E-)$ oriented edge and $(N-)$ oriented edge as the directed edges in directed B_2 . We will use the directed B_2 to construct the permutation voltage graph. Let $\alpha = (\pi_1, \pi_2)$ be a n -elements permutation voltage assignment to embedded B_2 , such that π_1 is assigned to $(E-)$ directed edge, and π_2 is assigned to $(N-)$ directed edge. We want to show that the embedded undirected permutation derived graph B_2^α is the dual of a QPC mesh, as shown in an example in Figure 16. We show these results in two theorems.

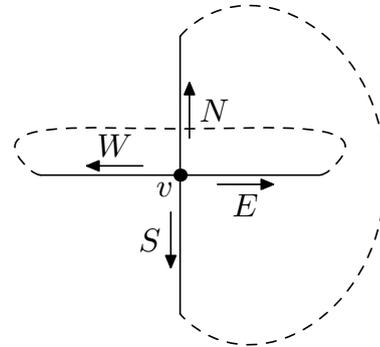


Figure 15: An embedded bouquet B_2 with two self-circles

The edge set of a spanning subgraph for a graph G is called an *2-factor* for G if that spanning subgraph is regular of valence 2.

Lemma 5.1. *If M' is the dual of a QPC mesh, the edge set in M' can be divided into two 2-factors L_1 and L_2 , where the oriented edges in L_1, L_2 are labeled with $(E-)$, $(W-)$ and $(N-)$, $(S-)$ respectively in a consistent oriented-edge labeling.*

PROOF. According to Theorem 3.1, M' can be labeled consistently. Hence, the two oriented edges for each edge in M' are either labelled by $(E-)$, $(W-)$ or labelled

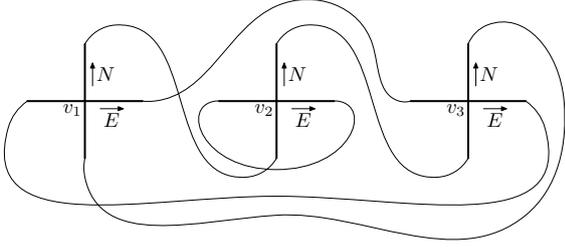


Figure 16: An example of graph rotation systems derived from permutation voltage graph B_2^α , by assigning $\alpha = \{(1\ 2\ 3), (1\ 3)(2)\}$ to $(N-)$, $(E-)$ directed edges of B_2 respectively. This particular mesh has genus 2.

by $(N-)$, $(S-)$. The edges in M' can be separated into two subsets. We collect all the edges whose oriented edges are labelled by $(E-)$, $(W-)$ in set L'_1 , and the edges whose oriented edges are labelled by the $(N-)$, $(S-)$ in set L'_2 . Since the four oriented edges originating at any vertex v in M' are labelled with $(E-)$, $(S-)$, $(W-)$ and $(N-)$, two of the four edges incident to v are in L'_1 and the other two are in L'_2 . Therefore, L'_1 , L'_2 are 2-factors. \square

To show the equivalence between two meshes, we need the definition below.

Definition Two meshes M_1 and M_2 are *embedding isomorphic* if there exists an isomorphism f , such that for each vertex v_i in M_1 , if the oriented edge $\langle v_i, v_j \rangle$ is immediately followed by oriented edge $\langle v_i, v'_j \rangle$ in the rotation at v_i , the oriented edges $\langle f(v_i), f(v_j) \rangle$ and $\langle f(v_i), f(v'_j) \rangle$ exist in M_2 , and $\langle f(v_i), f(v_j) \rangle$ is immediately followed by the oriented edge $\langle f(v_i), f(v'_j) \rangle$ in the rotation at $f(v_i)$ in M_2 .

Theorem 5.2. *There exists an embedded undirected derived permutation graph B_2^α which is embedding isomorphic to the dual of any given QPC mesh M' .*

PROOF. We label the vertices in M' as v_1, v_2, \dots, v_n , where n is the size of the vertex set in M' . According to Lemma 5.1, the edges in M' can be separated into two 2-factors L_1, L_2 , where the oriented edges of L_1, L_2 are labelled with $(E-)$, $(W-)$ and $(N-)$, $(S-)$ respectively.

(1) If L_i ($i = 1, 2$) is a connected 2-factor, which is also a hamiltonian cycle, we start from a vertex v and traverse L_1 (resp. L_2) along $(E-)$ (resp. $(N-)$) oriented edge to obtain a cyclic permutation π_i corresponding to the order of subscripts of the encountered vertices.

(2) If L_i contains several disjoint cycles c_1, c_2, \dots, c_m , we traverse L_i to obtain a permutation for each cycle. The permutation π_i is the product of the cycle permutations.

Therefore, we obtain a permutation voltage assignment $\alpha = \{\pi_1, \pi_2\}$. We assign the permutation π_1, π_2 to $(E-)$ and $(N-)$ directed edges in B_2 . We show the permutation derived graph B_2^α is embedding isomorphic to M' .

The vertices in B_2^α are labelled as v'_1, v'_2, \dots, v'_n . The n vertices are joined with directed edges in the order of the permutations π_1 and π_2 . If we have a bijection function mapping from v_i in M' to v'_i in B_2^α , it is easy to realize the undirected B_2^α is isomorphic to the underlying graph of M' , since if v_j is adjacent to v_i in M' , v'_j is also adjacent to v'_i in undirected B_2^α . We assign $(E-)$ to the directed edges derived from π_1 , and assign $(N-)$ to the directed edges derived from π_2 . In the corresponding undirected B_2^α , we assign the opposite oriented edge of $(E-)$ as $(W-)$, and opposite of $(N-)$ as $(S-)$. Since π_1, π_2 are the permutations of the vertex subscribes along $(E-)$ and $(N-)$ oriented edges in M' , the oriented edge $\langle v_i, v_j \rangle$ has the same label with $\langle v'_i, v'_j \rangle$. According to the definition of embedded derived graph, the $(E-)$, $(S-)$, $(W-)$ and $(N-)$ oriented edges are embedded in clockwise order at each vertex v'_i in B_2^α , which is the same with M' . Therefore, the embedded undirected B_2^α is embedding isomorphic to M' . \square

Theorem 5.2 shows any QPC mesh can be represented by the embedded B_2 associated with a permutation voltage assignment. In the following, we further show any given permutation voltage assignment to embedded B_2 gives a QPC mesh.

Theorem 5.3. *Given any permutation voltage assignment $\alpha = (\pi_1, \pi_2)$ to $(E-)$, $(N-)$ oriented edges in the embedded bouquet B_2 , the embedded undirected B_2^α is the dual of QPC mesh.*

PROOF. We sequentially connect the n vertices $\{v_1, v_2, \dots, v_n\}$ with oriented edges in the order of the permutation π_1 and π_2 , and assign the oriented edges from π_1 with label $(E-)$, the oriented edges from π_2 with label $(N-)$. For each oriented edge along the permutation π_1 (resp. π_2), we draw an opposite oriented edges labeled with $(W-)$ (resp. $(S-)$). Therefore, in B_2^α , the oriented edges labeled with $(E-)$ and $(N-)$ are shared edges with oriented edges labeled with $(W-)$ and $(S-)$ respectively. For each vertex v in embedded undirected B_2^α , there are four oriented edges originating at v , which are labelled as $(E-)$, $(S-)$, $(W-)$ and $(N-)$, where $(E-)$, $(S-)$, $(W-)$ and $(N-)$ are embedded in clockwise order. Therefore, we have a consistent labelling for the embedded undirected graph B_2^α . According to Theorem 3.1, B_2^α is the dual of a QPC mesh. \square

Theorem 5.2 presents a method to compress the infor-

mation for a given QPC mesh. It describes any QPC mesh by a fixed embedded graph associated with a very limited data, i.e., the two n -elements permutations. Theorem 5.3 shows an algorithm to construct an arbitrary QPC mesh with a theoretical view.

6. Construction of QPC Meshes from Regular Meshes

This section provides an practical alternative for constructing QPC Meshes. Motivated by the result of Lemma 3.6 - valences of all vertices of a QPC-mesh must be divisible by 4 - we endeavor to construct QPC-meshes by first constructing quad-meshes with all valences divisible by 4, which we call $4k$ -valent quad-meshes. In this section, we show how to construct $4k$ -valent quad-meshes. After we construct them, we can easily test them for the QPC property with the efficient labeling algorithm used in the proof of Theorem 3.2.

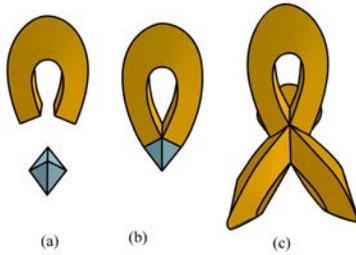


Figure 17: (a) octahedron and prism; (b) adding the prismatic handle increases the genus by one; (c) joining four pairs of faces with handles yields a $(4, 8, 4)$ regular mesh.

As a simple example of how to construct $4k$ -valent quad-meshes, we describe how to transform an octahedron into such a mesh. An octahedron O (see Figure 17(a)) has six 4-valent vertices, 12 edges, and eight 3-sided faces. We choose any pair of faces, and we connect them by a deformed prism, as illustrated in Figure 17(b). The resulting mesh has 6 vertices, 15 edges, and 9 faces, and is of genus one. The three “new faces” in the deformed prism are quadrilaterals, and the two chosen faces on O have disappeared.

If we partition the eight faces on the octahedron into pairs, and if we connect each face pair by a prismatic handle, then the resulting mesh O' has six vertices, each 8-valent; it has 24 edges; it has 12 faces, each 4-sided; and it is of genus 4. Thus, the new mesh O' is a regular mesh of type $(4, 8, 4)$ [18].

Figure 17(c) shows a regular mesh that is constructed this way. Since this regular mesh satisfy the conditions in Theorem 3.2, it is a QPC mesh. We observe that obtaining a quad-mesh depends only on being able to pair faces of like size; that it, it is not necessary that all faces be of the same size.

We recall that regular meshes are categorized by triples (n, m, g) , where n is the face size, m is the vertex valence, and g is the genus of the surface. For $g = 0$, the regular meshes include the 1-skeletons of the platonic solids and also the two-sided polygon. Any product graph of the form $C_m \times C_n$, where C_s is the s -cycle, is a $(4, 4, 1)$ -mesh for the torus.

For our purposes, any $(4n, 4m, g)$ mesh can be useful for obtaining $4k$ -valent quad-meshes, since Catmull-Clark subdivision of all the $4n$ -sided faces will result in a quad-mesh in which the valence of every vertex remains a multiple of 4. Importantly for the pursuit of QPC meshes, there exist regular meshes in the form of $(4n, 4m, g)$, for every genus g , including the forms $(4, 8, g)$, $(4, 12, g)$, $(8, 8, g)$, and $(4g, 4g, g)$. Moreover, there exist construction algorithms for regular mesh families in the forms $(4, 4i, 4i-3)$, $(4, 4i, 2i-1)$, $(4, 4i, i)$, $(4, 8i-4, i)$, $(4i, 4i, i)$, $(4i, 4i, 2i-1)$, and $(4i, 4i, 4i-3)$, where i is any positive real number [18]. Of course, the regular meshes for any given triple (n, m, g) are not unique. Moreover, by selecting different pairs of faces to connect with the prismatic handles, we can obtain different regular meshes, usually more than one of which provides the QPC property.

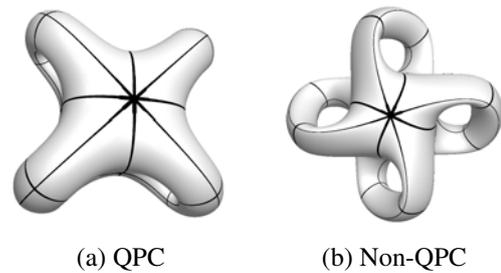


Figure 18: Subdivided versions of two other quad-meshes obtained by the algorithm to produce $(4, 8, 4)$ meshes.

For most regular mesh construction algorithms the vertex valences rise with the genus, and we can reduce the valences using the operation provided in Lemma 4.1. Figure 14(a) shows a genus-4 QPC mesh that is constructed by a $(4, 4i, i)$ algorithm and which includes two 16-valent vertices. Figure 14(b) shows the visual effect of the reduction of a 16-valent vertex to three valence 8 vertices.

We have developed additional algorithms to construct $4k$ -valent quad-meshes by adding prismatic handles that connect two faces. A complete review of all these algorithms is beyond the scope of this paper. Instead, we focus here on one such construction algorithm, which produces $(4, 8, 3k+1)$ -regular meshes, that is, of valence at most 8.

In practice, to construct $4k$ -valent quad meshes with handles, we use multi-segment piecewise-approximate versions of deformed prisms. The resulting QPC meshes are not formally regular, since not all vertices have the same valence, but they closely approximate the shape of regular meshes. The QPC-mesh shown in Figure 3(a) is similar to the regular mesh in Figure 17(c). Selecting different pairs of faces for connecting handles, we obtain different $(4, 8, 4)$ meshes for the surface of genus 4, two of which are shown in Figure 18. Among them, Figure 18(a) is QPC, but Figure 18(b) is not. Generalization of the $(4, 8, 4)$ construction to $(4, 8, 3k+1)$ is straightforward. Consider a set of k octahedra; simply pairing the faces and then joining pairs with prismatic handles yields a $(4, 8, 3k+1)$ mesh. Figures 20 and 20 show examples of higher genus QPC quad meshes with vertex valences at most 8.

7. Implementation and Results

We have shown that for every surface of positive genus, there exist quadrilateral meshes that permit seamless texture-mapping. With such meshes, it is very easy to seamlessly cover a surface with quad textures. All images herein were created with an extremely simple rendering program that does not provide complicated shading, and they are all direct screen captures. For any graduate student who is knowledgeable in computer graphics, the development of such software should not take more than a day. Testing of the software is also straightforward since there is always a solution if the mesh is QPC.

The wallpaper patterns that cover QPC meshes can easily be animated, since cyclic translations of wallpaper patterns are also wallpaper patterns. Texture coordinates can cyclically be translated, simply by using $u = u + \delta u - \lfloor u + \delta u \rfloor$ and $v = v + \delta v - \lfloor v + \delta v \rfloor$ where δu and δv are displacement, and where $\lfloor \cdot \rfloor$ is the floor function (which guarantees cyclic translation). Since the texture is already periodic, any cyclic translation of quad pattern still provides a seamless texture. If quad-patterns are continuously translated cyclicly, then the result becomes an animation of the texture sliding on the surface.

This property is also useful to control the texture mapping, i.e., the texture can be slid over the surface until best results are obtained.

8. Conclusion and Discussion

In this paper we show that for every surface of positive genus, there exist quadrilateral meshes that can seamlessly be texture-mapped over the surface. Using this set of quad-meshes, called QPC meshes, surfaces can also be covered aperiodically by using more than one quad pattern. Quad-pattern coverability is an addition to the repertoire of tools for texture mapping. It does not conflict with any existing tools, and it provides new power.

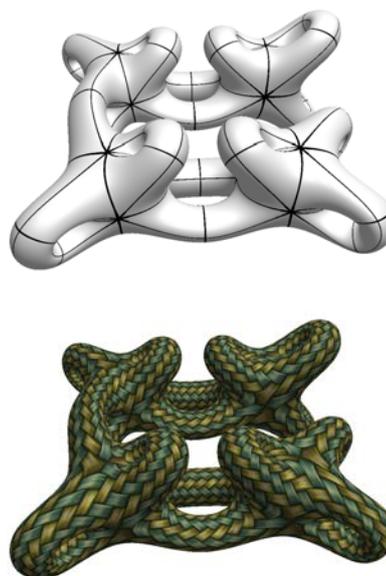


Figure 19: An example of QPC meshes obtained from a $(4, 8, 13)$ regular mesh, which is constructed from four octahedra. second row shows textured surface.

We have provided a theoretical construction algorithm that is based on a very simple permutation voltage graph. This algorithm always guarantees to construct QPC meshes. Our results imply that a quad-mesh is not QPC, unless the valence of every vertex is divisible by 4. Based on this observation, we have provided practical construction algorithms for finding QPC meshes based on regular meshes [18]. Using these algorithms we have constructed a large set of QPC meshes. Similarly, the regular map construction method introduced by Van Wijk can also be used to construct $4k$ -valent

quad meshes [19]. The regular maps that can be used to find QPC meshes include $(4, 8, g)$ and $(4, 12, g)$. These regular maps also does not necessarily have the QPC property. Therefore, there is still a need to verify the presence of the QPC property.

We also show that Catmull-Clark subdivision preserves the QPC property, and therefore it is possible to cover surfaces with iteratively finer versions of the same texture. We have applied Catmull-Clark subdivision to create curved quads. In this paper, we did not particularly try to reduce distortions. Catmull-Clark subdivision partially helps to reduce the distortions that are caused by significant differences in the sizes and shapes of quadrilaterals in the initial meshes, and optimization algorithms can further reduce such distortions.

Another type of texture distortion caused by large valence vertices needs special attention. We have developed an operation to reduce large valences to valence 8 while preserving the QPC property. In addition, for reducing the distortions around high valence vertices, it is helpful to move the position of such a vertex to a saddle region. Although, we have not provided a formal study in the paper, we want to point out that it is possible to move the position of a high valence vertex to another place with local mesh operations, while keeping the QPC property.

A direction to explore for future research can be meshes for manifolds with boundaries, which may have QPC parameterizations. QPC parameterizations also suggest that QPC meshes may have properties that can be used for shape modeling.

Current quadrangulation schemes, such as Quad-Cover [20], Mixed-Integer quadrangulation [21] and wave-based anisotropic quadrangulation [22], do not provide control for global topological structures of the meshes to convert any positive genus mesh into a QPC. Thus, an important future research direction can be the development of quadrangulation schemes that can produce quad-coverable meshes directly from any positive genus mesh.

Acknowledgments

This work partially supported by the National Science Foundation under Grant No. NSF-CCF-0917288.

References

- [1] E. Catmull, A Subdivision Algorithm for Computer Display of Curved Surfaces, PhD. Thesis, Department of Computer Science, University of Utah, December 1974.
- [2] C. Soler, M. P. Cani, A. Angelidis, Hierarchical pattern mapping, Proceedings of ACM SIGGRAPH 2001 21 (4) (2002) 673–680.
- [3] E. Catmull, J. Clark, Recursively generated b-spline surfaces on arbitrary topological meshes, Computer Aided Design (10) (1978) 350–355.
- [4] F. Neyret, M. Cani, Pattern based texturing revisited, Proceedings of ACM SIGGRAPH 1999 18 (4) (1999) 235–242.
- [5] B. Grunbaum, G. Shephard, Tilings and Patterns, W. H. Freeman and Co, NY, 1987.
- [6] J. L. Gross, T. W. Tucker, Topological Graph Theory, Wiley Interscience, New York, 1987.
- [7] E. Akleman, J. Chen, B. Meric, Intuitive and effective design of periodic symmetric tiles, Proceedings of ACM Multimedia 2000 21 (4) (2000) 100–108.
- [8] H. Wang, Games, logic, and computers, Scientific American (1965) 98–106.
- [9] R. Berger, The undecidability of the domino problem, In Memoirs American Mathematical Society 72 (1966) 98–106.
- [10] J. Stam, Hierarchical pattern mapping, Aperiodic Texture Mapping. Technical Report R046, European Research Consortium for Informatics and Mathematics (ERCIM) (1997).
- [11] M. F. Cohen, J. Shade, S. Hiller, O. Deussen, Wang tiles for image and texture generation, Proceedings of ACM SIGGRAPH 2003 23 (4) (2003) 287294.
- [12] A. A. Efros, W. T. Freeman, Image quilting for texture synthesis and transfer, ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2001 20 (4) (2001) 341–346.
- [13] M. Mantyla, An Introduction to Solid Modeling, Computer Science Press, Rockville, MA., 1988.
- [14] Y. Li, E. Zhang, Y. Kobayashi, P. Wonka, Editing operations for irregular vertices in triangle meshes, in: ACM SIGGRAPH Asia 2010 papers, no. 12 in SIGGRAPH ASIA '10, 2010, pp. 153:1–153:12.
- [15] C. Peng, E. Zhang, Y. Kobayashi, P. Wonka, Connectivity editing for quadrilateral meshes, in: Proceedings of the 2011 SIGGRAPH Asia Conference, no. 12 in SA '11, 2011, pp. 141:1–141:12.
- [16] J. L. Gross, T. Tucker, Generating all graph coverings by permutation voltage assignments, Discrete Mathematics (1997) 273–283.
- [17] J. Edmonds, A combinatorial representation for polyhedral surface, Notices American Mathematics Society (7) (1960) 646.
- [18] E. Akleman, J. Chen, Regular mesh construction algorithms using regular handles, in: Proc. IEEE Int. Conference on Shape Modeling and Applications, (SMI 06), 2006, pp. 171–181.
- [19] J. J. V. Wijk, Symmetric tiling of closed surfaces: Visualization of regular maps, ACM Transactions on Graphics; Proceedings of SIGGRAPH'2009 (2009) 28.1–28.12.
- [20] F. Kalberer, M. Nieser, K. Polthier, Quadcover - surface parameterization using branched coverings, Computer Graphics Forum / Proceedings of 14th Eurographics Workshop on Rendering 26 (3) (2007) 10–19.
- [21] D. Bommers, H. Zimmer, L. Kobbelt, Mixed-integer quadrangulation, ACM Trans. Graph. 28 (2009) 77:1–77:10.
- [22] M. Zhang, J. Huang, X. Liu, H. Bao, A wave-based anisotropic quadrangulation method, SIGGRAPH '10, 2010, pp. 118:1–118:8.

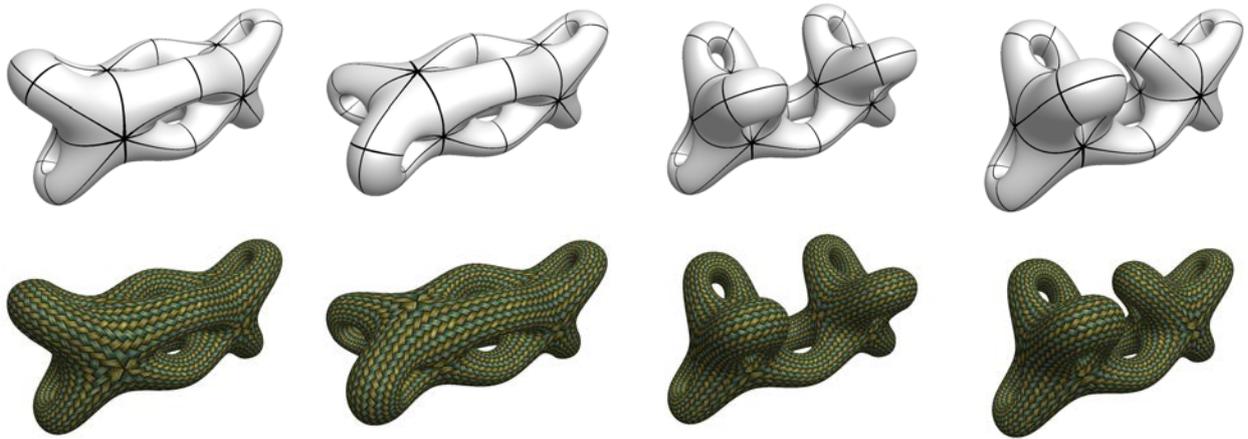


Figure 20: Examples of QPC meshes obtained from $(4, 8, 7)$ regular meshes, which are constructed from two octahedra. Second row shows textured surfaces.

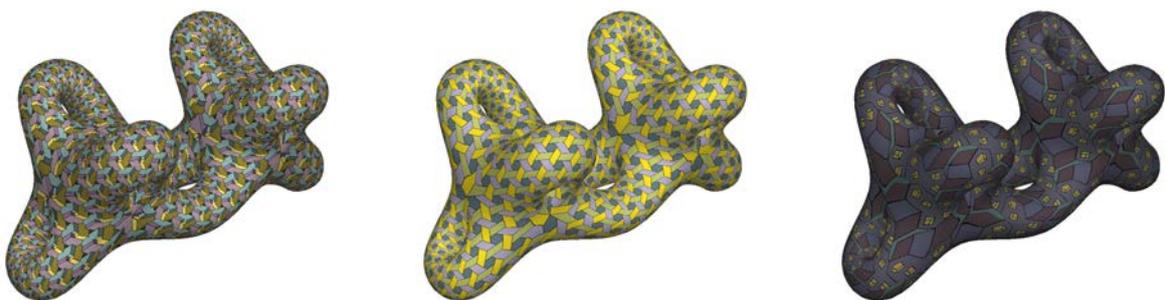


Figure 21: Examples of QPC meshes textured with $p3$, $p6$ and pg patterns shown in Figure 9.