

Local Fairing with Local Inverse

Javad Sadeghi*

Faramarz Samavati†

Department of Computer Science
University of Calgary, Alberta, Canada

ABSTRACT

Local fairing techniques are extensively used in the geometry processing of curves and surfaces. They also play an important role in the multiresolution shape editing and synthesis applications. However, due to the inter-dependency of the vertices after applying the current fairing techniques, their inverses are not local. Finding a local fairing operation with local inverse provides a well-defined relationship between the smooth vertices and the initial vertices. This paper introduces a new fairing operation for curves and surfaces that is smoothing and local but with a local inverse. In the curve domain, we find a class of banded smoothing matrices with banded inverses. Then, using the geometric interpretation of the corresponding local operation, this class is extended to surfaces. We discuss the advantages of using this new fairing operation in different applications. Also, the resulting operation is used to find novel subdivision schemes with well-defined reverse subdivisions.

Index Terms: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

1 INTRODUCTION

Local fairing techniques are commonly used in Computer Graphics. These techniques use discrete differential operations to minimize the energy of the mesh. This benefits different applications such as denoising, hole-filling and designing high-quality surfaces. In addition, these fairing techniques have important role in the multiresolution (MR) applications. One of the main operations used for surface fairing is discrete Laplacian smoothing. Many applications including mesh editing, 3D morphing and mesh synthesis benefit from this operation in their surface representation and geometry processing. We may achieve similar fairness using other averaging mechanisms, such as *chasing game* [20].

One of the major challenges in the applications that use local fairing operations is to find the inverse of the fairing efficiently. By applying current fairing operations on the vertices of a mesh, or on the points of a curve, these points become subsequently dependent on each other. This inter-dependency between points eliminates the possibility of having a universal local operation for the inverse process. A common workaround is to fix one or more *anchor points* [18] to restore the rest of the points with respect to them. In the curve domain, the act of the local fairing operations can be represented by banded matrices. However, the simplicity and bandedness of the corresponding matrices does not guarantee that their inverses be banded, and more often the resulting inverses are full. Finding a banded fairing matrix with banded inverse provides a well-defined relationship between the perturbed points and the initial points.

In the surface domain, a local inverse for the fairing operation represents an operation that applies within a local neighborhood,

*e-mail: jsadeghi@ucalgary.ca

†e-mail: samavati@ucalgary.ca

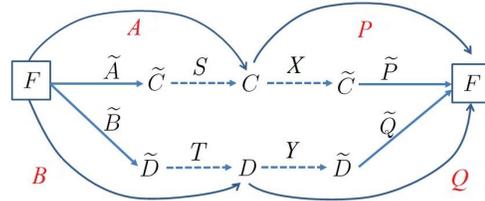


Figure 1: Smooth reverse subdivision framework [16, 17].

rather than influencing the entire mesh as encountered with the use of anchor points. This local inverse operation can address the inter-dependency problem of the current fairing operations. In addition, locality of inverse is important for multiresolution techniques that use fairing operations in their construction. For example, Sadeghi and Samavati [16, 17] present a MR framework that minimizes the energy of the coarse data using discrete Laplacian smoothing. As shown in Figure 1, to achieve smooth coarse points C they start from trial reverse subdivision (RS) operation \tilde{A} and apply discrete Laplacian smoothing matrix S on trial coarse points \tilde{C} . Since the inverse of the discrete Laplacian smoothing matrix ($X = S^{-1}$) is not banded, they cannot find a banded structure for $P = \tilde{P}X$. Similarly, to have a banded reconstruction operation $Q = \tilde{Q}Y$ for the corresponding $B = T\tilde{B}$ in their framework, matrix $Y = T^{-1}$ should have a banded structure.

Recently, Strang [21] presented the necessary conditions for a banded matrix to have a banded inverse. Typical banded matrices have full inverses, and he focuses on banded matrices from the class of banded matrices that can be factorized to multiplication of block-diagonal matrices [22]. This class is divided into four matrix types, namely banded orthogonal matrices, banded permutation matrices, banded plus finite rank and cyclically banded matrices. Raghavan [15] explores each of these classes in detail and finds different settings for factorization of their block matrices.

In this paper, we first verify the settings of [15] in the construction of atomic smoothing matrices, similar to the curve Laplacian smoothing, but with a banded inverse. The smoothing behavior of these atomic matrices helps to reduce the energy of the curves. By combining these atomic matrices we construct a new class of banded smoothing matrices with banded inverses. Next, we enhance this class using a simple geometric constraint that ignores the inter-dependency between points in the curve Laplacian smoothing. This geometric intuition is then extended to general surfaces by introducing a new mesh fairing operation which is local but with a local inverse. This new operation addresses the limitations of the anchor point solution by localizing the inverse using a mask-like operation that just involves the direct neighbors of a vertex. We show the benefits of using this operation in different examples and discuss how it can be used in MR applications. We also provide a recipe for finding novel subdivision schemes with well-defined reverse subdivisions using this new operation.

The remainder of this paper is organized as follows: section 2 reviews the previous work related to this research. Section 3 dis-

cusses the curve fairing matrices and their inverses. In section 4, our methodology for finding curve fairing matrices with banded inverses is described. Section 5 presents the corresponding local fairing operation for surfaces and discusses its applications. Finally, section 6 concludes the paper and proposes future directions.

2 RELATED WORK

Various techniques are used in discrete differential geometry to achieve fair surface designs. These techniques try to minimize the surface energy of the mesh in a local neighborhood of a vertex. A common approach, which goes back to Duffin’s [8] work in 1959, uses the cotangent representation of the Laplace-Beltrami operator on polygonal meshes. In 1993, Pinkall and Polthier [14] presented a functional representation of the discrete mean curvature vector based on this cotangent representation. Desbrun *et al.* [7] use the cotangent formula to define a new scale-dependent umbrella operator that addresses the drawbacks of Kobbelt *et al.*’s operator [10]. Later, they introduced other discrete operations for piecewise linear surfaces that are consistent, accurate and robust [12]. Recently, Wardetzky [24] has analyzed the convergence and structural properties of the discrete Laplacian vector based on the cotangent formula.

The discrete Laplacian operator for a vertex v_i is defined using its first ring of neighbors $N_1(v_i)$ as

$$\Delta p_i = \sum_{v_j \in N_1(v_i)} \omega_{ij} (p_j - p_i) \quad (1)$$

where p_i and p_j represent the vertex positions and weights are normalized for all vertices v_i

$$\sum_{v_j \in N_1(v_i)} \omega_{ij} = 1. \quad (2)$$

For uniform discretization we can choose the weights $\omega_{ij} = \frac{1}{\#N_1(v_i)}$ to depend definition only on the mesh connectivity [4]. This discretization can be represented as a matrix L with non-zero entries

$$L_{ij} = \begin{cases} -1 & i = j \\ \omega_{ij} & v_j \in N_1(v_i) \end{cases} \quad (3)$$

which is a symmetric, singular and positive semi-definite matrix.

Representation of a point with respect to its discrete Laplacian vector is called *differential* representation. Alexa [1, 2] uses differential representation for 3D morphing and briefly discusses its potential for free-form modeling. Sorkine *et al.* [19] provide an interactive detail preserving surface editing technique based on differential coordinates. This technique is used for feature transfer as well as mixing, and transplanting partial surface meshes. While the local nature of the discrete Laplacian operator facilitates finding differential coordinates, current techniques need to fix the position of at least one anchor point [18] to recover the Cartesian coordinates. Anchor points have global effect on the mesh and by changing their positions the shape of the entire mesh can be manipulated. Finding a local operation for the inverse of the discrete Laplacian operator helps to switch between differential and Cartesian coordinates without using anchor points.

Local fairing based on the discrete Laplacian have been successfully used in a number of multiresolution applications [25, 10, 11, 3]. The smooth coarse surfaces resulted from discrete fairing, improve the characteristics of the detail vectors used in these applications. Sadeghi and Samavati [16, 17] introduce a MR framework that considers the smoothness of the coarse points as part of the decomposition operation. They start their construction from a trial set of multiresolution filters and use a least-squares formulation based on the discrete Laplacian smoothing to minimize the energy of the coarse points while reducing the subdivision error. By enhancing

the results in real-time, their approach improves the curve and surface editing and synthesis applications. However, the absence of a local inverse for the discrete Laplacian smoothing prevents them to find local reconstruction filters. Having a local fairing operation S (see Fig. 1) with a local inverse helps them to address this limitation.

3 CURVE FAIRING MATRICES AND THEIR INVERSES

In order to find the desired local fairing operation with local inverse, we first consider the problem in the curve domain. In this domain, to find banded and smoothing matrices with banded inverses, we evaluate the curve Laplacian smoothing, the chasing game, and their respective inverses.

3.1 Curve Laplacian smoothing

Taubin [23] defines the curve Laplacian operator on a set of n periodic points $x = \{x_1, \dots, x_n\}^T$ as

$$\Delta x = -Kx \quad (4)$$

where

$$K = \frac{1}{2} \begin{bmatrix} 2 & -1 & & -1 \\ -1 & 2 & -1 & \\ & \ddots & \ddots & \ddots \\ -1 & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \quad (5)$$

is a banded and symmetric matrix. This fairing refinement is equal to applying the following averaging matrix to x

$$S_l = \begin{bmatrix} 0 & \frac{1}{2} & & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & \\ & \ddots & \ddots & \ddots \\ \frac{1}{2} & & \frac{1}{2} & 0 & \frac{1}{2} \\ & & & \frac{1}{2} & 0 \end{bmatrix}. \quad (6)$$

The problem with using K as the curve Laplacian operator, or equivalently S_l as the final fairing refinement, is that these matrices are singular. It can be observed in Equation 6 that by replacing one of the $\frac{1}{2}$ entries in S_l with 0 it becomes invertible. However, the resulting inverse matrix will be a full matrix. To find a banded inverse for the curve Laplacian smoothing we need to change its matrix form to a well-defined smoothing form.

3.2 Chasing game

The curve Laplacian smoothing provides a way of fairing curves, but other simple possibilities exist. Stam [20] presents a very simple averaging method, which is used for constructing B-spline subdivisions. This averaging can be used in curve fairing by chasing each point on the curve to its midpoint with the next point. Similar to the refinement matrix of the curve Laplacian smoothing (S_l), the chasing game can be represented with a refinement matrix

$$S_c = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & & \\ & \frac{1}{2} & \frac{1}{2} & \\ & & \ddots & \ddots \\ \frac{1}{2} & & \frac{1}{2} & \frac{1}{2} \end{bmatrix}. \quad (7)$$

Figure 2 illustrates the result of applying the S_c matrix of the chasing game on a curve with sharp corners. The benefit of using the chasing game is that, by duplicating the curve points followed by a few steps of S_c , we can create different B-spline subdivision schemes [20]. The key idea behind this property is that B-spline

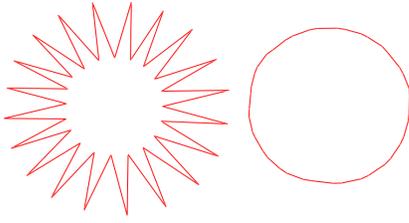


Figure 2: Result of applying the chasing game matrix S_c on a curve with sharp corners.

filters extracted from the k -th row of Pascal's Triangle can be found by multiplying S_c to the B-spline filters extracted from the $(k-1)$ -th row.

Although the chasing game is very simple, again it is not possible to find a banded inverse for S_c . This can be observed by looking at the rows of Pascal's Triangle. It can be proved that there are no universal constants that can linearly combine two or three elements in a row of Pascal's Triangle to generate entries of the upper row. In fact, to find entry i in the k -th row, all of the i elements in the $(k+1)$ -st row are needed. The geometric interpretation of this observation is that in the reverse process of the chasing game, points are subsequently dependent on each other up to a known starting point. However, by removing the $\frac{1}{2}$ from the lower left corner of S_c matrix, we can find an upper-triangular matrix for S_c^{-1} . While there are algorithmic ways to compute the triangular inverse matrices (e.g. backward substitution) we still cannot find a local mask for the inverse of the chasing game.

3.3 Banded matrices with banded inverses

In a banded matrix, non-zero entries are in a band w along the diagonal and $a_{ij} = 0$ for $|i-j| > w$. Strang [21] proves that a banded matrix has a banded inverse if it can be factorized into a product of block diagonal matrices with invertible blocks. Using this property he establishes a group of banded matrices with banded inverses [22]. Consider two block diagonal matrices F_1 and F_2 made up of repeating 2×2 blocks

$$a = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}.$$

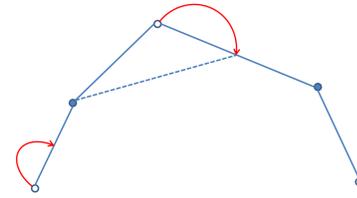
F_1 is created by repeating a along its diagonal

$$F_1 = \begin{bmatrix} a & 0 & 0 & 0 & 0 \\ 0 & a & 0 & 0 & 0 \\ 0 & 0 & a & 0 & 0 \\ 0 & 0 & 0 & a & 0 \\ 0 & 0 & 0 & 0 & a \end{bmatrix}$$

while F_2 first shifts b one down and one to the right and then repeats it along its diagonal

$$F_2 = \begin{bmatrix} b_{22} & 0 & 0 & 0 & 0 & b_{21} \\ 0 & b & 0 & 0 & 0 & 0 \\ 0 & 0 & b & 0 & 0 & 0 \\ 0 & 0 & 0 & b & 0 & 0 \\ 0 & 0 & 0 & 0 & b & 0 \\ b_{12} & 0 & 0 & 0 & 0 & b_{11} \end{bmatrix}.$$

The product of F_1 and F_2 will be a banded matrix with two block matrices $R = \begin{pmatrix} a_{11}b_{21} & a_{11}b_{22} \\ a_{21}b_{21} & a_{21}b_{22} \end{pmatrix}$ and $R' =$



● Fixed ○ Displaced to the midpoint

Figure 3: Geometric interpretation of F_1 , a corner-cutting atomic operation.

$$\begin{pmatrix} a_{12}b_{11} & a_{12}b_{12} \\ a_{22}b_{11} & a_{22}b_{12} \end{pmatrix} \text{ on every two rows}$$

$$F_{12} = F_1 F_2 = \begin{bmatrix} \ddots & & & & \\ & R & R' & & \\ & & R & R' & \\ & & & & \ddots \end{bmatrix}.$$

Raghavan [15] shows that F_{12} has a banded inverse if $\text{rank}(R) = 1$, $\text{rank}(R') = 1$, $\det(a) \neq 0$ and $\det(b) \neq 0$. In this situation, both F_1^{-1} and F_2^{-1} are banded, thus their multiplication produces a banded matrix F_{12}^{-1} . Raghavan also lists more advanced configurations for banded matrices with banded inverses [15]. Next, we try to find a smoothing matrix using the simplest case of two 2×2 blocks on every two rows.

4 METHODOLOGY

In this section we describe our steps to find an appropriate fairing operation from Strang's group of banded matrices with banded inverses [22].

4.1 Atomic fairing operations with banded inverses

We look for banded matrices that have two block matrices R and R' of 2×2 blocks on every two rows and can be successfully factorized to block diagonal matrices F_1 and F_2 made up of repeating 2×2 blocks along their diagonals. Raghavan [15] proves that the only possible case to have a banded inverse for the product of F_1 and F_2 in this 2×2 setting is when the ranks of R and R' are both one and the determinant of the 2×2 blocks of F_1 and F_2 are non-zero. To narrow down the search space, the 2×2 blocks of the atomic banded block diagonal matrices F_1 and F_2 are enforced to satisfy affinity. This adds two more constraints to the setting for making sure that sum of the entries in each row of a and b are one. In addition, we should construct the atomic matrices in a way that exhibit corner-cutting behavior. The simplest 2×2 block that satisfies all of these conditions is

$$a = b = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{pmatrix}.$$

The resulting 2×2 block has an interesting geometric interpretation. As illustrated in Figure 3, each 2×2 block applies the same averaging mask as the chasing game to the first two points and interpolates the second point. In other words, by fixing alternate points of the chasing game, their inter-dependency is removed. Using this 2×2 block two corner-cutting atomic operations

$$F_1 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & \frac{1}{2} & \frac{1}{2} \\ & & & & 1 \end{bmatrix}, \quad (8)$$

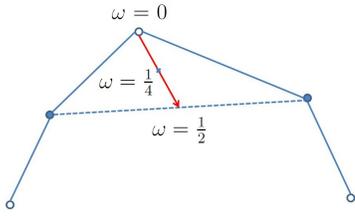


Figure 6: By fixing alternate points and moving the middle point in the direction of its discrete Laplacian smoothing vector, an atomic fairing operation with uniform distribution is achieved.

to

$$S_1 = \begin{bmatrix} 1-2\omega & \omega & & & \omega \\ & \omega & 1-2\omega & \omega & \\ & & \omega & 1-2\omega & \omega \\ & & & \ddots & \ddots \\ & & & & 1 \end{bmatrix} \quad (14)$$

and

$$S_2 = \begin{bmatrix} 1 & & & & \\ \omega & 1-2\omega & \omega & & \\ & \omega & 1-2\omega & \omega & \\ & & & \ddots & \ddots \\ \omega & & & & 1 \end{bmatrix} \quad (15)$$

which both have banded inverses.

Each of the S_1 and S_2 matrices just refines half of the points on the curve. Thus, to refine the whole curve, product matrix of $S = S_1.S_2$ is used which is also banded with banded inverse. In addition, the repeating rows of this matrix

$$\begin{bmatrix} \omega & 1-2\omega & \omega \end{bmatrix}$$

and

$$\begin{bmatrix} \omega^2 & (1-2\omega)\omega & 1-2\omega+2\omega^2 & (1-2\omega)\omega & \omega^2 \end{bmatrix}$$

exhibit symmetry with respect to the diagonal entry. This symmetry addresses the non-uniform distribution issue we had in the previous fairing operation (F_{12}). The matrices produced by $S_1.S_2$ or $S_2.S_1$ are our final solution for the symmetric fairing with banded inverses. They produce banded smoothing matrices with banded inverses and improved distribution of points. For example, the following matrix S results from setting $\omega = \frac{1}{4}$ in $S_1.S_2$

$$S = \begin{bmatrix} \frac{5}{8} & \frac{1}{8} & \frac{1}{16} & & & \frac{1}{16} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & & & \\ \frac{1}{16} & \frac{1}{8} & \frac{5}{8} & \frac{1}{8} & & & \\ & & \frac{1}{8} & \frac{1}{2} & \frac{1}{8} & & \\ & & & \frac{1}{16} & \frac{1}{4} & & \\ & & & & \ddots & \ddots & \ddots \\ & & & & & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{16} & & & & & \frac{1}{16} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{4} & & & & & & \frac{1}{4} & \frac{1}{2} \end{bmatrix} \quad (16)$$

Figure 7 illustrates the results of three applications of this S matrix on a coarse curve. It can be visually observed that the resulting curves are smooth and their points are better distributed on the curve. Also, the resulting matrix S can be used instead of the averaging matrix of the chasing game (S_c) to create novel subdivision schemes. Figure 8 illustrates three applications of a subdivision scheme created by a duplication followed by applying the matrix S three times. The visual smoothness of resulting curves is comparable to the cubic B-spline subdivision smoothness.

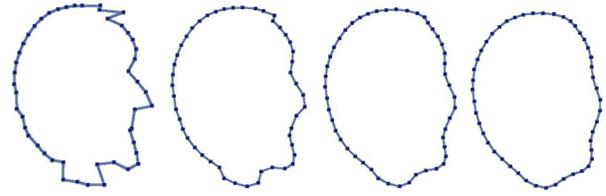


Figure 7: Three applications of matrix S in Equation 16 on a coarse curve.

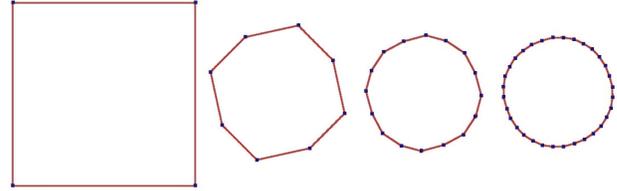


Figure 8: Three applications of the new subdivision scheme created by a duplication and three steps of S in Equation 16.

4.3 Extension to tensor product surfaces

Because of the regular structure of tensor product surfaces, our fairing operation can be easily extended to these surfaces. A tensor product surface is represented by two classes of curves usually denoted by u - and v -curves. For example, in terrains u -curves and v -curves are defined based on two dimensional grid of elevations. To extend our fairing operation to these surfaces, we need to apply S on both u and v directions. For terrains, both u - and v -curves are open curves; thus, an interpolating version of S is found by fixing the boundary vertices and fairing the rest of the vertices as periodic case. In Figure 9, we represent a coarse terrain and two applications of the interpolating version of S along both directions. The results show that our fairing operation can improve the smoothness of the multivariate surfaces. Next, we see how this operation can be extended to general topology surfaces.

5 LOCAL FAIRING WITH LOCAL INVERSE

In this section we introduce our local fairing operation with a local inverse for general topology surfaces. We also discuss the advantages of this operation in practice.

5.1 Extension to polygonal meshes

In order to extend our fairing operation to polygonal meshes the geometric interpretation of this corner-cutting operation is used. As illustrated in Figure 10, given a vertex P of the mesh, $M(P)$ is defined as the average of the neighbors of P ($N(P)$). The discrete Laplacian operator is defined by the vector $P - M(P)$ and the discrete Laplacian smoothing is defined by moving P to $M(P)$ concurrently for all points. To have a local inverse, we modify this definition such that interdependency between points are avoided. This happens by fixing $N(P)$ and only moving P . However, if we move P to $M(P)$, the original direction of the discrete Laplacian vector is lost and it is not possible to invert $M(P)$ back to P . Notice that only internal points $V(P)$ on the line segment between P and $M(P)$ can increase the smoothing at P . The advantage of $V(P)$ over $M(P)$ is that the vector $M(P)$ to $V(P)$ is non-zero. We can control the smoothness of our new operation by changing the position of $V(P)$ on the line segment by

$$V(P) = (1 - \omega)P + \omega M(P) \quad (17)$$

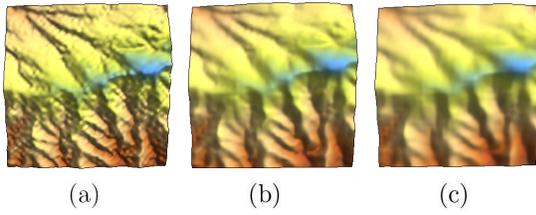


Figure 9: (a) A 53×53 coarse terrain. (b) and (c) two steps of applying interpolating S on both directions of the terrain.

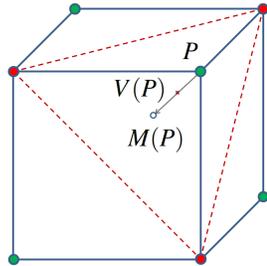


Figure 10: Suitable labeling of a mesh facilitates finding the local inverse of our local fairing operation.

where $0 \leq \omega < 1$. As long as the positions of the $N(P)$ are fixed, we can find a local inverse for this local fairing operation as

$$P = \frac{1}{1-\omega} V(P) - \frac{\omega}{1-\omega} M(P). \quad (18)$$

However, fixing $N(P)$ prevents us from applying this type of fairing concurrently on all vertices of the mesh (see Figure 10). One possible strategy for addressing this issue is to split the vertices of the mesh to two disjoint subsets, fixed (red) and active (green). Then, we can concurrently perturb all of the green vertices knowing that the position of all red vertices are fixed. By applying a similar procedure on the red vertices, we can complete the fairing operation. This red/green split helps us to recover the original vertices of the mesh using the local inverse of our fairing operation. First, we concurrently invert the red perturbed vertices, assuming green vertices are fixed. In the next stage, the local inverse operation in Equation 18 is applied on the green vertices.

Notice that this red/green labeling of vertices is only one of the ways to apply our local fairing operation. Any graph coloring of mesh vertices provides a strategy for applying this fairing. However, even a red/green condition (vertex coloring 2) is satisfied in some important mesh applications. Meshes resulting from Catmull-Clark subdivision satisfy this condition. It happens because, regardless of the valence of the faces, or connectivity of the vertices, after one level of Catmull-Clark subdivision all of the faces on a mesh will consist only of quads. In addition, there is no cycle of the odd length in the graph of the mesh. Therefore, the graph of the resulting quad mesh is bi-partite and we can separate its vertices into two different partitions (red/green). Figure 11 illustrates a high resolution quad mesh (see Figure 11(a)) and its smoothed version (see Figure 11(b)) after a few applications of our fairing operation. To highlight the red/green separation, application of the operation on coarse version (see Figure 11(c)) of the original model is shown in two steps. Figure 11(d) illustrates the fixed vertices by red dots which indicates the green vertices are moved. Similarly, in Figure 11(e) red vertices are moved and positions of the highlighted green vertices are not changed.

Clearly, not all meshes are bi-partite. However, these meshes can still be faired concurrently by finding a valid graph coloring for

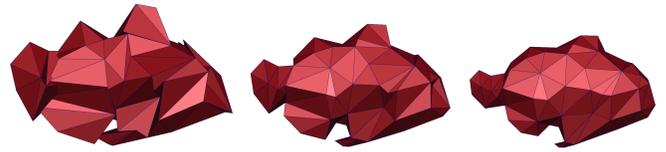


Figure 12: Two applications of our local fairing operation on a coarse triangular mesh.

their vertices. The extreme case of using one color per vertex is also an interesting one. In this case, the fairing operation is used for each vertex individually in an adaptive fashion. We search the graph of the mesh to select specific vertices that need to be smoothed (based on a criterion such as coarseness) and apply our local fairing operation on them. Then we add these perturbed vertices to a stack to be able to retrieve them back in the reverse order. Using this technique we can fair a specific region of the mesh adaptively. Figure 12 illustrates two applications of our fairing operation on a coarse triangular mesh using this approach. Smoothed meshes can be perturbed back to the original coarse mesh in linear time because of the locality of the inverse.

5.2 Applications

Our local fairing operation can benefit various geometry processing applications including mesh deformation, mesh editing and mesh synthesis. In the applications that use differential representation of the mesh, local inverse structure provided by our local fairing operation can be used instead of the anchor points. This helps these applications to achieve local effect with less effort.

Also, our local fairing operation can be used instead of the discrete Laplacian smoothing in different MR techniques. Figure 13 illustrates an example of using our operation in a mesh editing scenario. We first find a coarse approximation of the mesh in Figure 13(a) by applying the Catmull-Clark reverse subdivision filters of [13] (see Figure 13(b)). Next, we apply our local fairing operation on the coarse mesh to find a smooth surface suitable for editing (see Figure 13(c)). Then, we edit some vertices of the mesh (see Figure 13(d)). Next, the edited smooth mesh is returned back to its initial coarse resolution by applying the local inverse of the fairing (see Figure 13(e)). Finally, the edited coarse mesh is subdivided using Catmull-Clark subdivision to reach the original resolution (see Figure 13(f)). The advantage of this approach is that it does not alter the un-edited parts of the mesh because of the locality of the inverse. Also, it can be observed that the scale of the editing in the edited smooth coarse mesh (see Figure 13(d)) is preserved in the edited fine mesh (see Figure 13(f)).

In addition, our fairing operation can address the limitation of Sadeghi and Samavati's MR framework [16, 17] by finding banded reconstruction filters. Using the matrix $S = S_1.S_2$ instead of the discrete Laplacian smoothing matrix S in Figure 1 helps to achieve a banded structure for the inverse matrix $X = S^{-1}$. This provides a banded structure for the reconstruction filter $P = \tilde{P}X$ that can be paired with the optimized decomposition filter $A = \tilde{S}A$. Also, the simple geometric interpretation of the new operation S , helps to find a banded operation T with banded inverse $Y = T^{-1}$ to optimize the trial details in their framework \tilde{D} . This provides a banded reconstruction pair $Q = \tilde{Q}Y$ for optimized decomposition filter $B = T\tilde{B}$.

Furthermore, our new fairing operation can be used to find novel subdivision and reverse subdivision schemes. In the curve domain, the local fairing operation (S) can be used instead of the averaging step of the chasing game to create new subdivision schemes. We just need to duplicate the points of the coarse curve and apply a few steps of the fairing to find subdivision schemes with different levels of smoothness. For example, the subdivision fil-

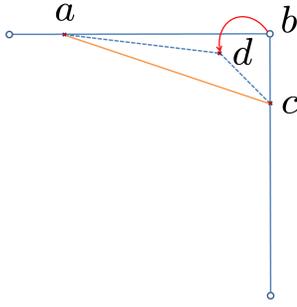


Figure 14: The local effect of applying the new subdivision scheme based on a duplication and one step of the local fairing operation.

Tutorials on Multiresolution in Geometric Modelling, pages 51–68. Springer-Verlag, Heidelberg, 2002.

- [10] L. Kobbelt, S. Campagna, J. Vorsatz, and H. Peter Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 105–114, 1998.
- [11] L. Kobbelt, J. Vorsatz, and H.-P. Seidel. Multiresolution hierarchies on unstructured triangle meshes. *Comput. Geom. Theory Appl.*, 14(1-3):5–24, 1999.
- [12] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In H.-C. Hege and K. Polthier, editors, *Visualization and Mathematics III*, pages 35–57. Springer-Verlag, Heidelberg, 2003.
- [13] L. Olsen and F. Samavati. A discrete approach to multiresolution curves and surfaces. In *Transactions on Computational Science VI*, volume 5730 of *Lecture Notes in Computer Science*, pages 342–361. 2009.
- [14] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, 1993.
- [15] V. Raghavan. Banded matrices with banded inverses. Master's thesis, Massachusetts Institute of Technology, 2010.
- [16] J. Sadeghi and F. F. Samavati. Smooth reverse subdivision. *Comput. Graph.*, 33(3):217–225, 2009.
- [17] J. Sadeghi and F. F. Samavati. Smooth reverse loop and catmull-clark subdivision. *Graphical Models*, 73(5):202–217, 2011.
- [18] O. Sorkine. Differential representations for mesh processing. *Computer Graphics Forum*, 25(4):789–807, 2006.
- [19] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184, 2004.
- [20] J. Stam. On subdivision schemes generalizing uniform b-spline surfaces of arbitrary degree. *Computer Aided Geometric Design*, 18(5):383–396, 2001.
- [21] G. Strang. Fast transforms: Banded matrices with banded inverses. *Proceedings of the National Academy of Sciences*, 107(28):12413–12416, 2010.
- [22] G. Strang. Groups of banded matrices with banded inverses. *Proc. Amer. Math. Soc.*, 139:4255–4264, 2011.
- [23] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of 22nd annual conference on computer graphics and interactive techniques*, 1995.
- [24] M. Wardetzky. Convergence of the cotangent formula: An overview. In A. I. Bobenko, J. M. Sullivan, P. Schröder, and G. M. Ziegler, editors, *Discrete Differential Geometry*, volume 38 of *Oberwolfach Seminars*, pages 275–286. Birkhuser Verlag Basel, 2008.
- [25] D. Zorin, P. Schröder, and W. Sweldens. Interactive multiresolution mesh editing. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 259–268, 1997.

A SUBDIVISION ANALYSIS

The subdivision schemes created using our new fairing operation provide acceptable visual smoothness. However, a proof for the convergence and smoothness properties of these new subdivision schemes is needed. Instead of showing the proof for the general case we choose the subdivision scheme created by one duplication step followed by one time of applying S in Equation 16

$$P = \begin{bmatrix} \frac{3}{4} & \frac{1}{16} & & \frac{3}{16} \\ \frac{3}{4} & \frac{1}{4} & & \\ \frac{3}{4} & \frac{1}{4} & \frac{1}{16} & \\ \frac{1}{16} & & \frac{1}{4} & \\ & & \ddots & \ddots \\ & \frac{1}{16} & & \frac{3}{16} & \frac{3}{4} \\ & & & & \frac{1}{4} \end{bmatrix}. \quad (20)$$

The filters of this subdivision are not symmetric but we can observe that provide a corner-cutting scheme. We can rely on the proof presented at [5] for the convergence of this scheme. Also by forming the local subdivision matrix we can check that the leading Eigen-value is one. However, we choose the formalism of Laurent polynomials for our analysis [9]. It helps to see the difference between the new subdivisions and the general B-spline subdivisions.

If a subdivision scheme be convergent its masks should have affinity property $\sum_j p_{2j} = \sum_j p_{2j+1} = 1$. This property holds for our case because $\frac{3}{4} + \frac{1}{4} = 1$ and $\frac{3}{16} + \frac{3}{4} + \frac{1}{16} = 1$. Using the subdivision filter of the above mentioned subdivision $\mathbf{p} = \{\frac{1}{16}, \frac{1}{4}, \frac{3}{4}, \frac{3}{16}\}$ we form the following Laurent polynomial

$$p(z) = \frac{1}{16} + \frac{1}{4}z + \frac{3}{4}z^2 + \frac{3}{4}z^3 + \frac{3}{16}z^4. \quad (21)$$

Because of the affinity property we should have $p(-1) = 0$ and $p(1) = 2$. Thus the polynomial should be factorized to $p(z) = (1+z)q(z)$ with $q(1) = 1$. It is easy to show this factorization for our case as

$$p(z) = (1+z) \left[\frac{1}{16} (1+3z+9z^2+3z^3) \right]. \quad (22)$$

To check the convergence of subdivision S_p we should check whether the subdivision corresponding to $q(z)$ (denoted by S_q) is contractive or not. It can be proven that subdivision $\mathbf{q} = \{\frac{1}{16}, \frac{3}{16}, \frac{9}{16}, \frac{3}{16}\}$ is contractive because the summation of the rows of the resulting subdivision matrix is less than one. Therefore,

$$\lim_{k \rightarrow \infty} (S_q)^k f^0 = 0$$

for all initial data f^0 . This verifies the convergence of the subdivision scheme S_p .

Since the resulting subdivision scheme can be considered as a corner-cutting method, we use the technique presented by de Boor [6] for smoothness of local corner-cutting schemes. He proves that the limit of a ‘local’ corner-cutting scheme is C^1 -continuous provided the corners of the broken lines become increasingly flatter. As shown in Figure 14 the new corner $\angle adc$ achieved after applying P lies inside the two segments which form the old corner $\angle abc$. This guarantees the locality of the corner. It can also be observed that the new corner’s angle is increased. Based on the proof of de Boor, repeating this process creates a limit curve which is C^1 -continuous.