Abstract

The dual of a 2-manifold polygonal mesh without boundary is commonly defined as another mesh with the same topology (genus) but different connectivity (vertex-face incidence), in which faces and vertices occupy complementary locations and the position of each dual vertex is computed as the center of mass (barycenter or centroid) of the vertices that support the corresponding face. This barycenter dual mesh operator is connectivity idempotent but not geometrically idempotent for any choice of vertex positions, other than constants. In this paper we construct a new resampling dual mesh operator that is geometrically idempotent for the largest possible linear subspace of vertex positions. We look at the primal and dual mesh connectivities as irregular sampling spaces, and at the rules to determine dual vertex positions as the result of a resampling process that minimizes signal loss. Our formulation, motivated by the duality of Platonic solids, requires the solution of a simple least-squares problem. We introduce a simple and efficient iterative algorithm closely related to Laplacian smoothing, and with the same computational cost. We also characterize the configurations of vertex positions where signal loss does and does not occur during dual mesh resampling, and the asymptotic behavior of iterative dual mesh resampling in the general case. Finally, we describe the close relation existing with discrete fairing and variational subdivision, and define a new primal-dual interpolatory recursive subdivision scheme.

CR Categories and Subject Descriptors:
I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling - surface, solid, and object representations.


1 Introduction

A polygonal mesh is defined by the association between the faces and their sustaining vertices (connectivity), by the vertex positions (geometry), and by optional colors, normals and texture coordinates (properties). Properties can be bound to the vertices, faces, or corners of the mesh, but it is sufficient to consider meshes with vertex positions and no other properties. This is so because: properties bound per vertex can be treated in the same way as vertex positions, and properties bound per face or per corner can be regarded as bound per vertex to a closely related mesh (dual mesh in the per face case, Doo-Sabin [4] connectivity in the per corner case).

In general we look at vertex positions as signals defined on the mesh connectivity. In section 2 we review some basic concepts about meshes and mesh signals, and establish the notation for the rest of the paper. The result of applying the barycenter dual mesh operator to a manifold polygonal mesh without boundary (the primal mesh) is another mesh with the same topology but different

IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 taubin@us.ibm.com. Work performed at the California Institute of Technology during the 2000-2001 academic year, while on sabbatical from IBM Research as Visiting Professor of Electrical Engineering
cient algorithm, described in section 7, to compute the resampling dual vertex positions with the Laplacian operator leads to an efficiently named because the term dual is reserved in Mathematics to operators that are equal to the identity when squared (idempotent operators); and when the square of the barycenter dual mesh operator is applied to a mesh, the original connectivity is recovered but the vertex positions are not. In fact, the linear operator defined by the square of the barycenter dual mesh operator on the primal vertex positions is a second order smoothing operator that displays the same kind of shrinkage behavior as Laplacian smoothing [13], always producing shrinkage when applied to non-constant vertex positions. For example, figures 1-(c) and 1-(d) show the result of applying the square and the fourth power of the dual mesh operator to the primal mesh of figure 1-(a), respectively.

In this paper we look at the construction of the dual vertex positions as a resampling process, where the primal vertex positions, regarded as signals defined on the primal mesh connectivity, are linearly resampled (transferred) according to the dual mesh connectivity. The problem is how to define resampling rules as a function of the connectivity so that loss of information is minimized, i.e., so that in general the original signal is recovered when the same process is applied to the resampled signal (dual vertex positions) on the dual mesh connectivity. Note that this is not the case when the dual vertex positions are defined as the barycenters of the faces. Here the result of resampling back from the dual to the primal sampling space always produces loss of signal, unless the signal is constant (zero frequency).

In section 4 we show that the dual vertex positions of the Platonic solids [17] circumscribed by a common sphere can also be defined as the solution of a least-squares problem with a quadratic energy function linking primal and dual vertex positions, and that this energy function is well defined for any manifold polygonal mesh without boundary. In section 5 we derive explicit expressions for the new resampling dual vertex positions as linear functions of the primal vertex positions. In section 6 we rewrite the formula for the resampling dual vertex positions as a function of the primal and dual Laplacian operators, and we show that the linear operator defined by the square of the resampling dual mesh operator is a smoothing operator that prevents shrinkage as in Taubin’s $\lambda_M^\mu$ smoothing algorithm [13]. The new expression for the resampling dual vertex positions with the Laplacian operator leads to an efficient algorithm, described in section 7, to compute the resampling dual vertex positions. Even though this algorithm, which can be implemented as a minor modification of the Laplacian smoothing algorithm, converges very fast, we exploit the relation to Laplacian smoothing even further, and define approximate algorithms that run in a fraction of the time in a pre-determined number of operations.

In the classical problem of uniform sampling rate conversion in signal processing [2, 16], under conditions determined by Shannon’s sampling theorem, when the sampling rate is reduced (fewer faces than vertices in the primal mesh) the frequency content of the signal determines whether loss of information (due to aliasing) occurs or not, and when the sampling rate is increased (more faces than vertices in the primal mesh), no loss of information occurs because the resampled signal is of low frequency. The situation here is more complex, due to lack of regularity, but in section 8 we establish the conditions under which loss of information occurs and is prevented, and study the asymptotic behavior of iterative dual mesh resampling, in fact defining the space of low frequency signals, i.e., the largest linear subspace of signals that can be resampled with no loss of information. For example, figures 1-(e), 1-(f), and 1-(g), shows the result of applying the first, second, and fourth powers of our new resampling dual mesh operator to the primal mesh of figure 1-(a). Note that the second and fourth powers (and any even power) recover the primal mesh because the conditions for lossless dual mesh resampling are satisfied.

In addition to Taubin’s low-pass filter algorithms [13, 15], a number of enhancements have been introduced in recent years to Laplacian smoothing to try to overcome some of its limitations, such as prevention of tangential drift [7, 3], implicit fairing for aggressive smoothing [3], the variational approach for interpolatory fairing [11], and the explicit incorporation of normals in the smoothing process for better control in shape design [18]. The algorithms introduced in this paper have potential applications in these areas. We do not explore these applications here, but we in section 10 we define a new interpolatory recursive subdivision scheme based on the primal-dual mesh operator, and we study the relation with variational fairing in section 12. Finally, we present our conclusions and plans for future work in section 13.

2 Meshes and Signals

The connectivity of a polygonal mesh $M$ is defined by the incidence relationships existing among its $V$ vertices, $E$ edges, and $F$ faces. We also use the symbols $V$, $E$, and $F$ to denote the sets of vertices, edges, and faces of $M$. A boundary edge of a polygonal mesh has
and arrange these one-dimensional signals as column vectors are linear and can be performed on each vertex coordinate independently. Vertex positions are three-dimensional vertex signals, and dual vertices, edges, and faces of a mesh, i.e., on the different connectivity components. The concepts of orientation and orientability play no role in this paper, and will be ignored as well. In our meshes every edge is regular, and the subgraph of the dual graph defined by all the faces incident to each mesh vertex form a closed loop, or cycle of faces. Figure 2 illustrates these concepts. The connectivity of the dual mesh of \( M \) is defined by the primal faces as dual vertices, and these dual graph loops as dual faces. Since each primal edge connects two vertices and has two incident faces, we identify primal and dual edges, and refer to them as just edges. Figures 1-(a) and 1-(b) show a mesh, and its dual.

We consider vertex, edge, and face signals defined on the vertices, edges, and faces of a mesh, i.e., on the different connectivity elements. These signals define vector spaces. For example, primal vertex positions are three-dimensional vertex signals, and dual vertex positions are three-dimensional face signals (vertex signals on the dual mesh). The role of the edge signals will become evident in subsequent sections. Since all the computations in this paper are linear and can be performed on each vertex coordinate independently, it is sufficient to consider one-dimensional signals. We arrange these one-dimensional signals as column vectors \( X_V, X_E, \) and \( X_F \), of dimension \( V, E, \) and \( F \), respectively. The element of \( X_V \) corresponding to a vertex \( v \) is denoted \( x_v \), the element \( X_E \) corresponding to an edge \( e \) is denoted \( x_e \), and the element of \( X_F \) corresponding to a face \( f \) is denoted \( x_f \).

3 The Barycenter Dual Mesh

The quad-edge data structure [6] can be used to efficiently represent and traverse a mesh, and in particular to construct the connectivity of the dual mesh. The faces of the dual mesh can be reconstructed by cycling around each vertex of the primal mesh using the information stored in the quad-edge data structure. In the dual mesh construction the dual vertex signal corresponding to a face \( f = (v_1, \ldots, v_n) \) with \( n \) corners is computed as the average of the primal vertex signals corresponding to the corners of the face

\[
x_f = \frac{1}{n} \sum_{i=1}^{n} x_{v_i}.
\]

We can also write this assignment in vector form as

\[
X_F = W_{VF} X_V,
\]

where \( W_{VF} \) is the vertex-face incident matrix \( I_{VF} \) normalized so that the sum of each row is equal to one. If this construction is repeated on the dual mesh, we obtain a mesh with the same connectivity as the primal mesh, but with vertex positions

\[
x'_F = W_{VF} W_{FP} X_V,
\]

where the matrix \( W_{VF} \) is the face-vertex incident matrix \( I_{VF} \) normalized so that the sum of each row is equal to one. The matrix \( W_{VF} W_{FP} \) is not symmetric, but is composed of non-negative elements, and its rows add up to one. It defines a second order smoothing operator closely related to Laplacian smoothing [13].

If we look at the set of vertex signals such that \( X'_V = X_V \), i.e., the invariant subspace of \( W_{VF} W_{FP} \) associated with the eigenvalue 1, we generally end up with a subspace spanned by the constant vector \( X_V = (1, \ldots, 1)^T \). Our approach, described in the next three sections, is to construct new matrices \( W_{VF} \) and \( W_{FP} \), as functions of the connectivity, to replace the matrix \( W_{VF} \) in the construction of dual vertex signal values, in such a way that the dimension of the invariant subspace of \( W_{VF} W_{FP} \) associated with the eigenvalue 1 is maximized.

4 Platonic Solids

Figure 3 shows the five Platonic solids: the tetrahedron, the cube, the octahedron, the icosahedron, and the dodecahedron. All of them are circumscribed by a sphere, say of unit radius. In terms of connectivity, the tetrahedron is dual of itself, and both the cube and the octahedron, and the icosahedron and the dodecahedron, are dual of each other. Because of the symmetries, if we construct the dual mesh of each of these meshes as described in section 3, with the dual vertex positions at the barycenters of the primal faces, we end up with the corresponding dual platonic solids, but circumscribed by spheres of smaller radii. This can be solved by adjusting the scale, moving the face positions away from the center of the primal mesh along the corresponding radial directions until the dual vertex positions are circumscribed by the unit sphere. This procedure solves the problem for the Platonic solids, but it does not work for other more general meshes. However, the construction has the following property [17], that can be observed in figure 4 for the case of the icosahedron and the dodecahedron: for each edge \( e = \{v_1, v_2, f_1, f_2\} \) connecting two vertices and two faces, the segments joining the corresponding vertex positions and face positions intersect at their midpoints, i.e.,

\[
\frac{1}{2}(x_{v_1} + x_{v_2}) = \frac{1}{2}(x_{f_1} + x_{f_2}).
\]

This means that the construction of the dual vertex positions of the Platonic solids can be described as the minimization of the following energy function

\[
\phi(X_V, X_F) = \sum_{e \in E} (\|x_{v_1} + x_{v_2} - x_{f_1} - x_{f_2}\|^2)
\]

with respect to \( X_F \) with \( X_V \) fixed, the sum taken over all the edges of the mesh. The value attained at the minimum is zero. Note that the vertex positions of the dual mesh are obtained by minimizing the same energy function with respect to \( X_V \) with \( X_F \) fixed. In addition, this energy function is defined for every manifold polygonal mesh without boundary.
the solution of the linear system be written for the matrix

And due to symmetry, the minimizer of

Equation 2 can be written in matrix form for any mesh as follows

The Resampling Dual Mesh

Equation 2 can be written in matrix form for any mesh as follows

where \( I_{EV} \) and \( I_{EF} \) are the edge-vertex and edge-face incidence matrices without normalization. In general, these are full-rank matrices. Since equation 3 is quadratic, the minimizer of \( \phi(X_V, \star) \) is the solution of the linear system

which is obtained by differentiating \( \phi \) with respect to \( X_F \), or equivalently

with

And due to symmetry, the minimizer of \( \phi(\star, X_F) \) can be written as

with

The matrix in equations 5 can also be written as

where

is the pseudo-inverse of the matrix \( I_{EV} \). A similar expression can be written for the matrix \( W_{FEV} \) of equation 6.

6 Relation to Laplacian Smoothing

In this section we establish the relation between dual resampling formula \( X'_V = W_{VEF} X_F \) and Laplacian smoothing. In section 7 we use this formulation to define a simple and efficient algorithm to evaluate the resampling dual vertex signals as a minor modification of the Laplacian smoothing algorithm.

In its simplest form, the primal Laplacian operator is defined for a vector of vertex positions \( X_V \) as

where \( v^* \) is the set of vertices \( \star_{v^*} \) connected to vertex \( v \) by an edge, the weight \( w_{v,v^*} \) is equal to \( 1/|v^*| \), and \( |v^*| \) is the number of elements in the set \( v^* \). If we organize the weights as a matrix \( W_V \), we can write the Laplacian operator in matrix form as follows

where \( K_V = I_V \) has eigenvalues in \([0, 2]\), and \( I_V \) is the identity matrix in the space of vertex signals \([13]\). A similar expression
can be written for the dual Laplacian operator $\Delta_F X_F$. Pseudocode implementations of the algorithms to evaluate the primal and dual laplacian operators are described in figure 3.

Note that the diagonal element of the matrix $E_V E_V$ corresponding to a vertex $v$ is equal to the number $|\psi|$ of vertices connected to $v$ through an edge, and if we organize these numbers as a diagonal matrix $D_V$, we have

$$E_V E_V D_V = D_V (I_V + W_V) = 2D_V (I_V - \lambda K_V),$$

with $\lambda = 0.5$, and also

$$E_V E V D_V = 2D_V W_V X,$$

where $W_V$ is the matrix introduced in section 3 ($E_V$ normalized so that the sum of each row is equal to one). This allows us to rewrite the equation (dual of 4) used to compute the double dual vertex positions as a function of the face positions, as follows

$$(I_V - \lambda K_V) W_V X = X_F, \quad (7)$$

With a similar derivation, we can rewrite equation 4, used to compute the face positions as a function of the primal vertex positions, as follows

$$(I_F - \lambda K_F) W_F X = W_F X, \quad (8)$$

where $K_F$ is the matrix of the Laplacian operator defined on the dual mesh.

Note that in equation 8, the face positions are computed by applying implicit smoothing [3] to the barycenters of the faces with negative time step $dt = -0.5$. This process is not a smoother, but actually enhances high frequencies. The behavior of this process is closely related to Taubin's $\lambda \mu$ non-shrinking smoothing algorithm [13], where a true low pass-filter is constructed by two steps of Laplacian smoothing with positive (high frequency attenuating) and negative (high frequency enhancing) scaling factors. Here the computation of face barycenters has a high frequency attenuating effect, and the implicit smoother with negative time step has a high frequency enhancing effect.

$$W_F X = (I_F - \lambda K_F)^{-1} W_F X, \quad (9)$$

The final result, as in Taubin’s $\lambda \mu$ algorithm, is a low-pass filter effect without shrinkage, while the data is transferred from the primal to the dual mesh.

7. Algorithm

To compute the dual vertex signals $X_F = W_EV X_V$ as a function of the primal vertex signals we solve the linear system of equation 8 using a simple iterative method, which, as we will see in this section, is a minor modification of the Laplacian smoothing algorithm.

Iterative methods are used to solve systems of linear equations such as

$$AY = Z,$$

where the non-singular square matrix $A$ is large and sparse, and $Y$ and $Z$ are vectors of the same dimension [5]. Several popular iterative solvers, such as Jacobi and Gauss-Seidel, are based on the following general structure. By decomposing the matrix $A$ as the sum of two square matrices $A = B + C$, such that $B$ is easy to invert and the spectral radius of the matrix $H = B^{-1}C$ is less than one, the problem is reduced to the solution of the equivalent system

$$(I - H) Y = Y_0,$$

where $Y_0$ is the matrix of the Laplacian operator defined on the dual mesh.

The final result, as in Taubin’s $\lambda \mu$ algorithm, is a low-pass filter effect without shrinkage, while the data is transferred from the primal to the dual mesh.

PrimalDualSmoothing ($X_V$, $n$, $\lambda$, steps)

for $s = 0, \ldots, \text{steps} - 1$

$X_{F0} = W_F X_V$;

for $j = 0, \ldots, n - 1$

$dX_F = \text{DualLaplacian} (X_{Fj})$;

$X_{Fj+1} = X_{F0} + \lambda dX_F$;

end;

$X_{V0} = W_F X_{F0}$;

for $j = 0, \ldots, n - 1$

$dX_V = \text{PrimalLaplacian} (X_{Vj})$;

$X_{Vj+1} = X_{V0} + \lambda dX_V$;

end;

$X_V = X_{Vn}$;

return $X_V$;

Figure 7: Primal-dual smoothing algorithm. Pseudocode for the primal and dual Laplacian operators is described in figure 3.

with $Y_0 = B^{-1} Z$. The following simple algorithm

$$Y_n = Y_0 + H Y_{n-1} \quad (10)$$

defines a sequence of estimates $\{Y_n : n = 1, 2, \ldots\}$ that converges to the solution of the original system of equations 9, because the series

$$\sum_{j=0}^{\infty} \theta^j = (I - \theta)^{-1}$$

converges absolutely and uniformly for $|\theta| < 1$, and

$$Y_n = \sum_{j=0}^{n} H^j Y_0.$$

The rate of convergence is determined by the spectral radius $\rho$ of $H$: if $\|H Y\| < \rho \|Y\|$ for all $Y$, and $0 \leq \rho < 1$, then

$$\|Y - Y_n\| \leq \sum_{j=n+1}^{\infty} \|H^j Y_0\| \leq \sum_{j=n+1}^{\infty} \rho^j \|Y_0\| = \frac{\rho^{n+1}}{1 - \rho} \|Y_0\|.$$  

For example, if $\rho \leq 1 / 2$, the relative error is less than $0.1\%$ after ten iterations, and the estimates have about six correct digits after twenty iterations.

To solve equation 8 we set $H = \lambda K_F$, $Y = X_F$, and $Y_0 = W_F X_V$. Although the spectral radius $H$ is bound above by 1 (because the eigenvalues are in the interval $[0, 1]$), in typical meshes this upper bound is closer to $1/2$, and we observe in practice convergence to an error of less than $0.1\%$ after ten iterations.

Note that, if we replace $Y_0$ by $Y_{n-1}$ in the iteration rule described in equation 10 we obtain

$$Y_n = Y_{n-1} + H Y_{n-1},$$

or equivalently

$$Y_n = (I + H)^n Y_0,$$

which corresponds to $n$ steps of the Laplacian smoothing with parameter $\lambda = 1/2$. The main difference is that in Laplacian smoothing the number of iterations is specified in advance, while in our new algorithm it depends on an error criterion. As an alternative, we can use the new algorithm with both an error tolerance and a maximum number of iterations, and stop as soon as either stopping criterion is satisfied. In our experience, a maximum number of iterations of 20 and error tolerance of 0.001 produces excellent results.
The result of applying primal-dual smoothing steps with parameters $\lambda = 0.6307$ and $\mu = -0.6732$. The result of applying primal-dual smoothing steps with parameters $\lambda = 0.5$; (D) 6 steps with $n = 1$, (E) 3 steps with $n = 2$, and (F) 1 step with $n = 6$. The computational cost is about the same in all cases.

8 Analysis of Dual Mesh Resampling

In this section, and based on simple concepts from Linear Algebra, we establish necessary and sufficient conditions under which no loss of information occurs when primal vertex signals are remapped, and describe the general behavior of the dual resampling process.

The matrix $I_{EV}$ defines a linear mapping from the space of vertex signals into the space of edge signals $X_E = I_{EV}X_V$. Since normally meshes have more edges than vertices and the matrix $I_{EV}$ is full-rank, the image of this mapping is a subspace $S_V$ of dimensions $V$ in the space of edge signals. Let $T_V$ be the orthogonal complement of $S_V$ in the space of edge signals, i.e., $S_V \oplus T_V$ is the full space of edge signals. Every edge signal can be decomposed in a unique way as a sum of two edge signals; a first one $P_VX_E$ belonging to $S_V$ and a second one $(I_E - P_V)X_E$ belonging to $T_V$.

$$X_E = P_VX_E + (I_E - P_V)X_E,$$

Figure 8: Primal-Dual smoothing vs. Laplacian smoothing. (A) a mesh. The result of applying (B) 12 Laplacian smoothing steps with parameter $\lambda = 0.6307$, and (C) 12 steps of Taubin’s smoothing algorithm with parameters $\lambda = 0.6307$ and $\mu = -0.6732$. The result of applying primal-dual smoothing steps with parameters $\lambda = 0.5$: (D) 6 steps with $n = 1$, (E) 3 steps with $n = 2$, and (F) 1 step with $n = 6$. The computational cost is about the same in all cases.

where $P_V = I_{EV}I_{EV}^T$, and $I_E$ is the identity in the space of edge signals. The pseudo-inverse $I_{EV}^T$ of the matrix $I_{EV}$ defines a linear mapping from the space of edge signals into the space of vertex signals $X_V = I_{EV}^T X_E$ that recovers the vertex signal part of any edge signal

$$X_V = I_{EV}^T I_{EV} X_V,$$

because $I_{EV}^T I_{EV} = I_V$ (with $I_V$ the identity of vertex signals). The matrix $P_V$ is a projector ($P_V^2 = P_V$) in the space of edge signals which has $S_V$ as its invariant subspace associated with the eigenvalue 1 and $T_V$ as its invariant subspace associated with the eigenvalue 0. The matrix $I_{EF}$ also defines a linear mapping from the space of face signals into the space of edges signals $X_E = I_{EF}X_F$, orthogonal subspaces $S_F$ and $T_F$ of edge signals, and a projector $P_F = I_{EF}I_{EF}^T$.

The intersection $S = S_V \cap S_F$ of the subspaces $S_V$ and $S_F$
plays a key role in determining whether signal loss occurs or not in the dual mesh resampling process. We call a vertex signal \( X_V \) dual resamplable if \( P X_V = X_V \), i.e., if the dual resampling process produces no loss of information. These signals correspond to edge signals \( X_E = I_{EV} X_V \) that belong to \( S \). To prove this statement, let \( P = W_{VEF} W_{FEV} \) be the matrix corresponding to the square of the dual mesh resampling process, and let \( X_V \) be a vertex signal. If the corresponding edge signal \( X_E = I_{EV} X_V \) belongs to \( S \), then there is a face signal \( X_F \) so that \( X_E = I_{EF} X_F \). It follows that

\[
PX_V = I_{EV}^{1} I_{EF} P I_{EF} I_{EV} X_V \\
= I_{EV}^{1} I_{EF} P I_{EF} X_F \\
= I_{EV}^{1} I_{EF} X_F \\
= I_{EV} I_{EF} X_V \\
= X_V
\]

Note that since the subspaces \( S_V \) and \( S_F \) are spanned by the columns of the matrices \( I_{EV} \) and \( I_{EF} \), the dimension of \( S \) is equal to the rank of the matrix \( I_{EF}^{1} I_{EV} \). We have three particular cases:

1) the dimension of \( S \) is \( V \) \( (S_V \) is a subspace of \( S_F ) \); 2) the dimension of \( S \) is \( F \) \( (S_F \) is a subspace of \( S_V ) \); and 3) the dimension of \( S \) is strictly less than the minimum of \( V \) and \( F \). In the first case (sampling rate increase), no loss of information occurs for any vertex signal, i.e., \( P = I \). In the second case (sampling rate decrease) the process is idempotent, i.e., \( P^2 = P \). This is so because since \( S_F \subseteq S_V \) we have \( P F_P F = P_F \), and so

\[
P^2 = I_{EV}^{1} P F_P F_P I_{EV} X_V \\
= I_{EV}^{1} P F_P I_{EV} X_V \\
= P .
\]

Neither one of these first two cases are very common. Most typically we encounter the third case, in which iterative dual mesh resampling produces a sequence of vertex signals that quickly converges to a resamplable one:

**Proposition 1** (Lossy resampling) For any vertex signal \( X_V \) the sequence \( P^n X_V \) converges to a dual resamplable signal.

**Proof 1:** Let us define the following sequence of vertex signals

\[
\begin{align*}
X_V^0 &= X_V \\
X_V^n &= P X_V^{n-1} & n > 0
\end{align*}
\]

Clearly, if the sequence converges, the limit vector \( X_V^\infty \) satisfies the desired property \( P X_V^\infty = X_V^\infty \). To show convergence, it is sufficient to prove that the sequence \( I_{EV} X_V^\infty \) converges. Our argument is based on an eigenvalue analysis. Note that

\[
I_{EV} X_V^\infty = I_{EV} P^n X_V = (P_F P_F^n) I_{EV} X_V ,
\]

and since \( P_F \) is a projector, and \( P_F I_{EV} X_V = I_{EV} X_V \), we have

\[
(P_F P_F^n) I_{EV} X_V = (P_F P_F^n) I_{EV} X_V .
\]

Now, the matrix \( P_F P_F P_F \) is symmetric and non-negative definite. Since \( P_F \) and \( P_F P \) are projectors, the eigenvalues of \( P_F P_F P_F \) are between zero and one, with eigenvalue \( 1 \) corresponding to eigenvectors in \( S \), and eigenvalues strictly less than \( 1 \) corresponding to eigenvectors orthogonal to \( S \). Let \( \lambda \) be the largest of the eigenvalues less than \( 1 \), which is equal to the cosine of the angle between the subspaces \( S_V \cap S \) and \( S_F \). Since the projection of \( I_{EV} X_V^\infty \) onto \( S \) remains constant, the projection onto the orthogonal subspace to \( S \) converges to zero at least as fast as \( \lambda^n \).

Figure 6 illustrates this last case. Numerical algorithms to determine the rank of the matrix \( I_{EF} I_{EV} \) can be based on the QR decomposition for small meshes, and the SVD algorithm for large meshes [5]. Computing the smallest singular value would be sufficient to know whether we are in cases 1 or 2, or 3. Further work is needed to relate local combinatorial relations between vertices, faces, and edges to the rank of the matrix \( I_{EF} I_{EV} \).

### 9 Primal-Dual Smoothing

If we use the new algorithm to compute the dual vertex positions by specifying just a maximum number of iterations, and then apply the same algorithm to recompute primal vertex positions as a function of the dual vertex positions, we obtain a new family of non-shrinking smoothing operators for the primal vertex signals \( X_V^\prime = P_n X_V \) described by the following steps:

1) \( X_V^\prime = (I_F + \cdots + \lambda^n K_F^n) W_{EV} X_V \)
2) \( X_V^\prime = (I_V + \cdots + \lambda^n K_V^n) W_{FEV} X_F \)

and illustrated in pseudocode in Figure 7.

Since for large \( n \) we have \( X_V^\prime \approx P X_V \), with \( P = W_{VEF} W_{FEV} \) which satisfies \( P^2 \approx P \), as \( n \) increases, these operators produce less smoothing. We also have the freedom of playing with the parameter \( \lambda \). Figure 8 shows some results compared to Laplacian and Taubin’s smoothing algorithms.

Note that to implement the primal-dual smoothing algorithm we do not need to construct the connectivity of the dual mesh explicitly. The two steps are based on recursively evaluating products of the Laplacian matrices \( K_V \) and \( K_F \) by vectors of dimensions \( V \) and \( F \), and by accumulating partial results in temporary arrays of the same dimensions. But both matrix vector products can be accumulated by traversing the same list of mesh edges.

### 10 The Primal-Dual Mesh

As noted by Kobbelt [10], the operator that transforms the connectivity of a mesh into its Catmull-Clark connectivity [1] has a square root. The result of applying this square root operator to the connectivity of a mesh has the vertices and faces of the original mesh as vertices, the edges of the original face as quadrilateral faces, and the vertex-face incident pairs as edges. The quad-edge data structure [6] can be used to operate on the primal-dual mesh. Figure 9
Figure 11: Non-shrinking Doo-Sabin subdivision operator is the composition of the square of the primal-dual operator followed by the resampling dual operator. (A) coarse mesh, (B) coarse mesh after three shrinking Doo-Sabin refinement steps, (C) coarse mesh after three non-shrinking Doo-Sabin refinement steps, (D) superposition of (A) and (C).

Figure 12: Surface designed by combining resampling-dual (RD) and primal-dual (PD) mesh operators. (A) coarse mesh, the result of applying (B) $PD \circ RD \circ PD$, (C) $PD \circ RD \circ PD$, and (D) $PD \circ RD \circ PD$ to the coarse mesh.

resampling dual. An example is shown in figure 11. This Doo-Sabin resampling operator is also an example of a resampling process to a different mesh with the same topology. More general cases will be studied in a subsequent paper.

Another application for other combinations of these two operators is as a design tool in an interactive modeling environment. One example of a surface designed in this way is shown in figure 12.

12 Relation to Variational Fairing

In this section we discuss the close relation existing between our primal-dual operator and the discrete fairing approach, which shows that the surfaces produced by recursive primal-dual subdivision are smooth in the variational sense. Further work is required to understand the local asymptotic behavior of primal-dual subdivision.

First of all, we modify the energy function $\phi(X_V, X_F)$ of equation 3 by introducing a symmetric positive definite $E \times E$ matrix $H$ as follows

$$\phi_H(X_V, X_F) = X_E^T H X_E , \quad (11)$$

where $X_E$ is the edge signal vector

$$X_E = I_{EV} X_V - I_{EF} X_F .$$

The qualitative behavior of the primal-dual subdivision process for different values of the diagonally dominating $H$ is very similar.

The Laplacian operator on the primal-dual mesh can be written in block matrix form as follows

$$\Delta \begin{pmatrix} X_V \\ X_F \end{pmatrix} = \begin{pmatrix} I_V & -W_{VF} \\ -W_{VF}^T & I_F \end{pmatrix} \begin{pmatrix} X_V \\ X_F \end{pmatrix}$$

where $I_V$ and $I_F$ are the identity matrices in the spaces of vertex and faces signals, respectively, and the matrices $W_{VF}$ and $W_{FV}$

11 Non-Shrinking Doo-Sabin

Since the Doo-Sabin connectivity of a mesh is the dual of the Catmull-Clark connectivity, and this is the square of the primal-dual connectivity, we can combine the resampling dual and primal-dual mesh operators to produce a non-shrinking version of the Doo-Sabin [4] subdivision scheme: apply primal-dual twice followed by

illustrates the construction, and how to recover the connectivity of the original mesh and its dual from the resulting mesh connectivity.

If we add to this connectivity refinement operator our algorithm to compute the resampling dual vertex positions, we obtain an interpolatory refinement mesh operator. Because of the symmetric role that primal and dual vertices play in this construction, we prefer to call it the primal-dual mesh operator. Note that this operator has two inverses that can be used to recover either the original mesh, or the resampling dual mesh. Also, since the scheme is interpolatory, and the original vertices are a subset of the vertices of the resulting mesh, there is no loss of information.

The primal-dual mesh operator defines a linear operator that maps vertex signals on the primal mesh to vertex signals on the primal-dual mesh

$$X_V \mapsto \begin{pmatrix} X_V \\ X_F \end{pmatrix} .$$

Since the matrix that defines this linear operator is full-rank, the image is a subspace of dimension $V$. And this is true even if the primal-dual mesh operator is applied iteratively several times to refine the mesh more and more. We will see in section 12 that these meshes are smooth in the variational sense. Figure 10 shows an example of applying the primal-dual mesh operator recursively several times to a coarse mesh as a mesh design tool.
are the face-vertex and vertex-face incident matrices normalized so that each row adds up to one introduced in section 3.

In the Discrete Fairing approach [11], the smoothness of a mesh is increased by minimizing an energy function such as the square of the Laplacian

\[ \Delta \left( \frac{X_V}{X_F} \right) \]

with some vertices fixed, or other linear constraints. In our case we would minimize this expression with the primal vertex positions \(X_V\) fixed to obtain the dual vertex positions \(X_F\), or with the dual vertex positions fixed to obtain the primal vertex positions.

If we replace the matrices \(W_{V,F}\) and \(W_{F,V}\) by the new matrices \(W_{V,EF}\) and \(W_{F,VE}\) defined in section 5, we obtain a new resampling Laplacian operator \(\Delta_T\) which behaves in a very similar way

\[ \Delta_T \left( \frac{X_V}{X_F} \right) = \begin{pmatrix} I_V & -W_{VEF} \\ -W_{FVE} & I_F \end{pmatrix} \begin{pmatrix} X_V \\ X_F \end{pmatrix}, \]

But if we expand the square of the resampling Laplacian operator we obtain

\[ \left| X_V - W_{VEF} X_F \right|^2 + \left| X_F - W_{FVE} X_V \right|^2, \]

or equivalently

\[ \left| \Delta_T \left( \frac{X_V}{X_F} \right) \right|^2 = \left| I_{EV} X_E \right|^2 + \left| I_{EF} X_E \right|^2 = \phi_H(X_V, X_F), \]

with

\[ H = (I_{EV})^T I_{EF} + (I_{EF})^T I_{EV}. \]

13 Conclusions and Future Work

In this paper we described a solution to the problem of shrinkage in the construction of the dual mesh, introduced efficient algorithms to solve the problem, and shown some applications. Through a signal processing resampling point of view, we established necessary and sufficient conditions under which no loss of information occurs, and analyzed the asymptotic behavior of iterative dual mesh resampling.

We regard the results introduced in this paper as a first step toward a general theory for general mesh resampling, and complementary to existing approaches to remeshing [12], recursive subdivision [10, 20, 19], and 3D geometry compression [9, 8, 14]. We plan to explore these applications in subsequent papers.

In this paper we restricted our meshes to oriented manifold meshes without boundary. We also plan to extend the formulation and algorithms to meshes with boundary and non-singular edges. In this extended formulation we will have explicit parameters (boundary conditions), such as normals, associated with boundary and singular edges, that could be used very effectively in an interactive free-form shape design environment.

References


