Report

Eigenmeshes

Author(s):
Gross, Markus H.; Hubeli, Andreas

Publication Date:
2000

Permanent Link:
https://doi.org/10.3929/ethz-a-006653911 →

Eidgenössische Technische Hochschule Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Institute of Scientific Computing

Computer Graphics Group

CS Technical Report #338, March 30, 2000

Eigenmeshes

Markus Gross, Andreas Hubeli

Computer Science Department
ETH Zurich, Switzerland
e-mail: {grossm, hubeli}@inf.ethz.ch
http://graphics.ethz.ch

Eigenmeshes

M.Gross A. Hubeli
Department of Computer Science
ETH Zurich, Switzerland

1 Introduction

Discretization of manifolds is of fundamental importance in computer graphics and geometric modeling. However, as opposed to classical processing of multidimensional signals, there has not been any *sampling theory* for manifolds introduced so far. As a consequence, most discretization methods in graphics either make use of ad hoc, heuristic approaches [4] or rely on theorems from discrete geometry or topology [1]. In many applications, mesh reduction methods are applied subsequently to reduce the complexity of the underlying triangulation. A prominent example is the progressive mesh of [7] that computes a sequence of progressively refineable meshes by successive application of an edge collapse operator. This method, in combination with appropriate error metrics [3], provides a powerful representation for triangle meshes. While mesh reduction can be considered as a *subsampling* of a discretized manifold, there is no rigorous mathematical theory to determine the *sampling rate* and the *sample positions*.

A significant step towards a *signal processing approach* to discrete manifolds has been *mesh fairing*, as pioneered by [9]. This approach generalizes the notion of *"frequency"* to meshes of arbitrary connectivity by taking the eigenfunctions of the discretized Laplacian. Hence, mesh smoothing can be accomplished by attenuation of the eigenvalues associated with the *"high frequencies"* of the mesh. Mesh fairing essentially relates to a diffusion of the perturbations over the mesh surface and various numerical schemes had been proposed, such as the implicit quadrature of [2]. A key ingredient of a fairing method is the discretized Laplacian approximating surface curvature, various approximations of which had been advocated [9], [8], [5]. Although the signal processing approach to mesh processing can provide visually pleasing results including low-, high-, and bandpass-behavior, the computational filtering schemes do not allow to discard isolated eigenvectors or ranges of eigenvectors with sharp boundaries. Explicit reduction of the dimensionality of the eigenspace of the Laplacian, however, is a very desirable feature for a wide range of analysis.

In this paper, we present first concepts towards a *signal processing based sampling framework* for manifolds. Although a rigorous treatment of eigenfunction expansions would require functional analysis, we take a pure algebraic approach to mesh fairing and reduction. By extending some of the work of [9], we introduce the notion of *eigenmeshes*. To this end, we expand the mesh as a linear combination of the eigenvectors (eigenfunctions) of the discretized Laplacian. Removing individual eigenvectors from the representation projects the mesh onto a lower dimensional subspace, spanned by the remaining eigenvectors. Hence, any mesh reduction method can be considered as a projection operator, computing the new mesh vertex positions (samples) in this lower dimensional space. This notion allows us to construct projection operators that minimize the error imposed by the lower dimensional approximation. The key ingredients of the eigenmesh framework are a fairing operator, a projection operator and a notion of error and detail.

In this report we briefly analyze and discuss the usefulness of eigenanalysis for the construction of subsampling and remeshing operators. Our analysis is confined to the computation of vertex positions. Topological issues, such as the determination of local mesh connectivity, is left to future research.

2 Fairing of One-Dimensional Curves

In order to better understand the usefulness of eigenanalysis for surface representations we start with a discrete, one-dimensional curve shown in Figure 1.

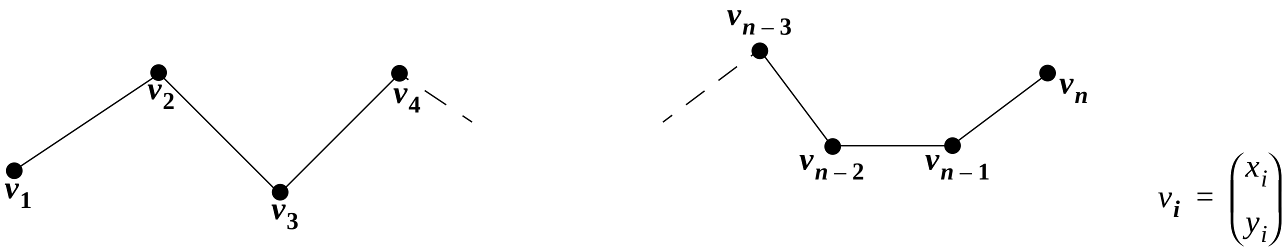

FIGURE 1. One-dimensional discrete curve represented by line segments

The curvature at each vertex $\boldsymbol{v_i}$ can be approximated by a discrete estimate of the Laplacian computed, for instance, using second order finite differences:

$$\Delta \boldsymbol{v_i} = \frac{1}{2}(\boldsymbol{v_{i-1}} - \boldsymbol{v_i}) + \frac{1}{2}(\boldsymbol{v_{i+1}} - \boldsymbol{v_i}) \tag{1}$$

In matrix form we write the Laplacian $\Delta \boldsymbol{x} = \{\Delta \boldsymbol{v_1}, \ldots, \Delta \boldsymbol{v_n}\}$ as a linear combination of the vertex vector $\boldsymbol{x} = \{\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}\}$ introducing the operator K:

$$\Delta \boldsymbol{x} = -K \cdot \boldsymbol{x} \tag{2}$$

K denotes a matrix of size $n \times n$ defined by

$$K = \frac{1}{2}\begin{bmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & \ldots & \ldots & \ldots & \\ & & -1 & 2 & -1 \\ -1 & & & -1 & 2 \end{bmatrix} \tag{3}$$

It is straightforward to compute the eigenvalues $\lambda_1, \ldots, \lambda_n$ and eigenvectors $\boldsymbol{u_1}, \ldots, \boldsymbol{u_n}$ of K. Since K, as defined in Eq 3, is symmetric the eigenvectors form an *orthonormal system* and we can expand $\boldsymbol{x}$ as the following linear combination:

$$x = \sum_{i=1}^{n} \vec{\xi}_i \cdot u_i \tag{4}$$

The coordinates $\vec{\xi}_i$ of the curve in eigenspace can be computed as the projection of $\boldsymbol{x}$ onto the $\boldsymbol{u_i}$ using the scalar product operator $\langle \bullet , \bullet \rangle$:

$$\vec{\xi}_i = \langle \boldsymbol{x}, \boldsymbol{u_i} \rangle \tag{5}$$

Since $\boldsymbol{x}$ is a 2xn-matrix representing the x and y-components of each vertex of the line segment, the coordinates $\vec{\xi}_i$ are vector-valued of type (ξ_{ix},ξ_{iy}) and define a *direction*. Hence, the products $\vec{\xi}_i \boldsymbol{u}_i$ can be interpreted as *uniform direction fields*, summing up to the line segment's vertex positions according to Eq 4. We call these direction fields the ***eigenmeshes*** of $\boldsymbol{x}$. *

Let us assume that the eigenvalues λ_i are sorted in ascending order, that is $\lambda_1 < \lambda_2 < \ldots < \lambda_n$. The discrete curve *"signal"* represented by the vector $\boldsymbol{x}$ can be smoothed by removing the m eigenvectors $\boldsymbol{u_{n-m+1}}, \ldots, \boldsymbol{u_n}$ from the linear combination described by Eq 4. Note that the removal of the m largest eigenvectors corresponds to the well-known *ideal* low-pass filtering of the regular setting.

The eigenvector and the eigenvalue representation diagonalizes K which is of Toeplitz type for our example:

$$K = U \cdot \lambda \cdot U^T \tag{6}$$

where

$$U = [\boldsymbol{u_1}, \ldots, \boldsymbol{u_n}] \tag{7}$$

and

$$\lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ldots & \\ 0 & & \lambda_n \end{bmatrix}. \tag{8}$$

3 Fairing of Arbitrary Meshes

The fairing process presented in Section 2 can be generalized to irregular two-dimensional discrete *"signals"*, such as triangular meshes.

* The geometric interpretation of an eigenmesh is not that obvious. The coordinate pair (ξ_{ix},ξ_{iy}) defines a direction for each eigenvector along which each mesh vertex is displaced. The magnitude of the displacement of vertex $\boldsymbol{v_k}$ is determined by the eigenvector coordinate $\boldsymbol{u_{ik}}$. Hence, an eigenmesh is a scalar field defined over the mesh and a displacement vector.

The vertices $\boldsymbol{v_i} = (v_{ix}, v_{iy}, v_{iz})^T$ of the two-dimensional mesh are stored in a vector $\boldsymbol{x} = \{\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}\}$ of size $3 \times n$ meaning that $\boldsymbol{x}$ stands for the *geometric or positional information* of the mesh. The discrete surface curvature at any mesh vertex is again estimated using the Laplacian

$$\Delta \boldsymbol{x} = -K\boldsymbol{x} \tag{9}$$

$K \in R^{n \times n}$ is a matrix operator used to approximate the Laplacian. Note that the discretization of the Laplacian is a non-trivial problem for irregular meshes and we will ignore it for now. Possible approaches are described in [8], [5] or [2]. For further analysis the only requirement imposed on K is that

$$\sum_{j=1}^{n} w_{i,j} = 1 \quad \Rightarrow \quad \sum_{j=1}^{n} k_{i,j} = 0 \tag{10}$$

where $w_{i,j}$ is defined as a weighted sum of type

$$\Delta \boldsymbol{x_i} = \sum_{\substack{j=1 \\ j \neq i}}^{n} w_{i,j} \cdot (\boldsymbol{x_j} - \boldsymbol{x_i}) \tag{11}$$

Here, we assume *linearity* of K with respect to the vertices and *independence* of their x,y and z-components, hence $K_x = K_y = K_z$. Although the Laplacian by its very nature is a linear operator, more sophisticated curvature operators usually lead to nonlinear expressions.

The very strength of this expansion is that the eigenvalues and eigenvectors of the matrix K provide us with generalized frequencies for irregular settings, even though a Fourier transform does not exist in the classical sense. Again, expressing the signal $\boldsymbol{x}$ as a linear combination of eigenvectors, such as shown in Eq 4, the mesh smoothing can be performed by removing the eigenvectors that correspond to high frequencies. An example of a smoothed mesh is given in Figure 2.

It is important to understand the difference between the fairing framework obtained using eigenanalysis and the fairing operators constructed in [9], [8] or [5]. For regularly sampled surfaces in the functional setting, that is, for surfaces defined over the x,y-plane, there is a one-to-one correspondence between the fairing computed by an *ideal low-pass filter* in Fourier space and the fairing obtained by the method developed in this section. Both methods compute orthogonal projections of a mesh onto the subspace of “smooth” meshes. However, most of the operators available in the literature use *iterative* smoothers to approximate the projection. Taubin, for example, employed an iterative Gaussian smoothing. It has been shown that iterative methods correspond to the classical, *recursive filtering* algorithms of the regular setting [5]. Furthermore, it is well known that iterative solvers only attenuate eigenvectors that correspond to high frequencies, as opposed to the above method which entirely removes them and guarantees a reduction of the dimensionality of the eigenspace. For the time being, we ignore the problem of numerically stable computation of the eigenvector decomposition for very large *K*.

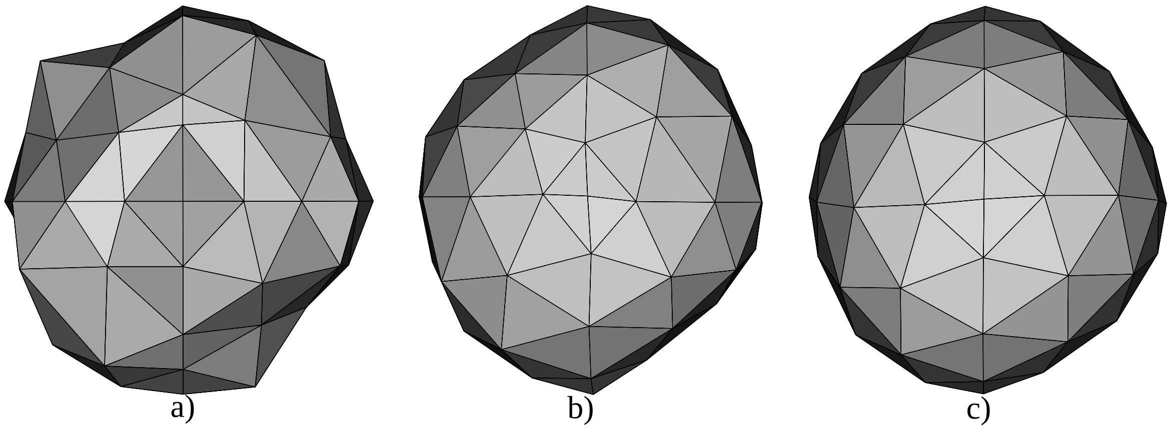

FIGURE 2. Smoothing (low pass filtering) of a triangular mesh
a) Noisy input mesh
b) Resulting mesh after removing 60% of the largest eigenvectors
c) Resulting mesh after removing 80% of the largest eigenvectors

Note that we changed our concept of *dimensionality* which from now refers to the number of mesh vertices.

4 The Subsampling Problem

In the previous sections we explained the process of surface fairing using eigenanalysis. This method *"bandlimits"* the mesh "signal" by projection onto a lower dimensional subspace. Bandimitation is an essential prerequisite for subsampling, reducing the number of vertices from n to m (usually $m = n-1$ or $m = n/2$), Formally, we search a projection operator $D \in R^{n \times m}$ that both reduces the number of mesh vertices and computes the set of new vertex positions $\boldsymbol{x'}$:

$$\boldsymbol{x} = \sum_{i=1}^{n} \xi_i \cdot \boldsymbol{u_i} \quad \Rightarrow \quad \boldsymbol{x}' = D \cdot \boldsymbol{x} = \sum_{i=1}^{n} \xi_i \cdot D \cdot \boldsymbol{u_i} \tag{12}$$

$$U' = D \cdot U \quad \Rightarrow \quad K' = D \cdot U \cdot \lambda \cdot U^T \cdot D^T \tag{13}$$

Using the linear combination of eigenvectors $\boldsymbol{u_i}, 1 \le i \le n$ in Eq 4, $\boldsymbol{x}$ can be represented in an n-dimensional space defined by the orthonormal bases $\boldsymbol{e_i}, 1 \le i \le n$ and corresponding to the columns of the identity matrix I. Figure 3 depicts an example for $n = 3$.

The upper relations hold for the *x*,*y* and *z*-components of the mesh vertices, respectively. Recall that $\boldsymbol{x}$ encodes the geometry of the mesh; the eigenvectors $\boldsymbol{u_i}, 1 \le i \le n$ and the eigenvalues $\lambda_i, 1 \le i \le n$ carry more information, since they can be used to reconstruct K. K, however, bears information about local topology and connectivity depending on the approximation model used for the discretized

Laplacian. Conversely, the connectivity graph represented by K does not necessarily have to correspond to a valid triangulation.

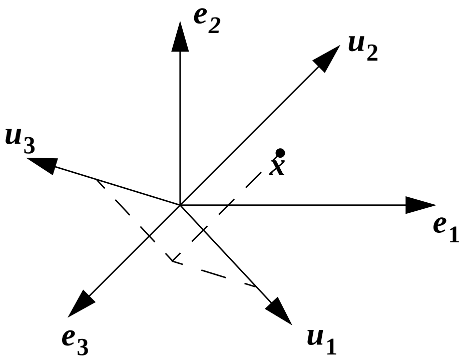

FIGURE 3. Mesh *x* represented as a linear combination of eigenvectors

For now, we will only consider the geometric part of the problem, that is we will analyze properties of the projection operator D. In order to better understand the problem, we start with a presentation of two trivial projection operators that reduce the dimension of the vector of mesh vertices $\boldsymbol{x} \in R^n$ by one, i.e. $\boldsymbol{x}' \in R^{n-1}$:

- A straightforward projection removes the eigenvector $\boldsymbol{u}_j$ corresponding to the highest frequency and projects the remaining eigenvectors onto the space spanned by $\{\boldsymbol{e}_i | 1 \le i \le n, i \ne j\}$. This requires a correspondence between every eigenvector $\boldsymbol{u}_i, 1 \le i \le n$ and every vector $\boldsymbol{e}_i, 1 \le i \le n$ that defines the coordinate system. Figure 4 illustrates the approach.

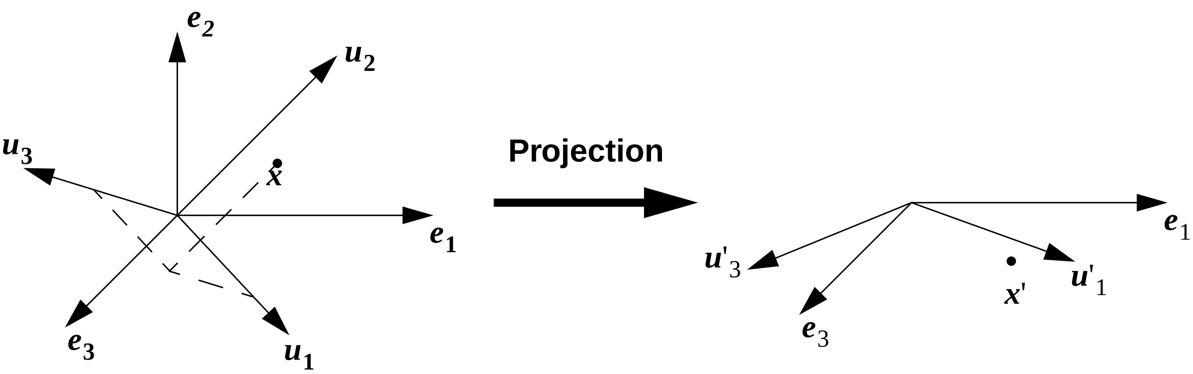

FIGURE 4. A simple projection operator removing the eigenvector of highest frequency.

The main disadvantage of this approach is that after the projection the eigenvectors loose orthogonality (remember that the matrix K defined in Eq 3 is symmetric). Furthermore the new set of eigenvectors is no longer *valid* in the sense that it represents an eigenvector of a meaningful new matrix K'.

- A slightly different approach consists of first rotating the eigenvector $\boldsymbol{u}_j$ being removed into the corresponding vector $\boldsymbol{e}_j$, and then projecting the system onto the subspace defined by $\{\boldsymbol{e}_i | 1 \le i \le n, i \ne j\}$. This second strategy is shown for $n = 3$ in figure 5.

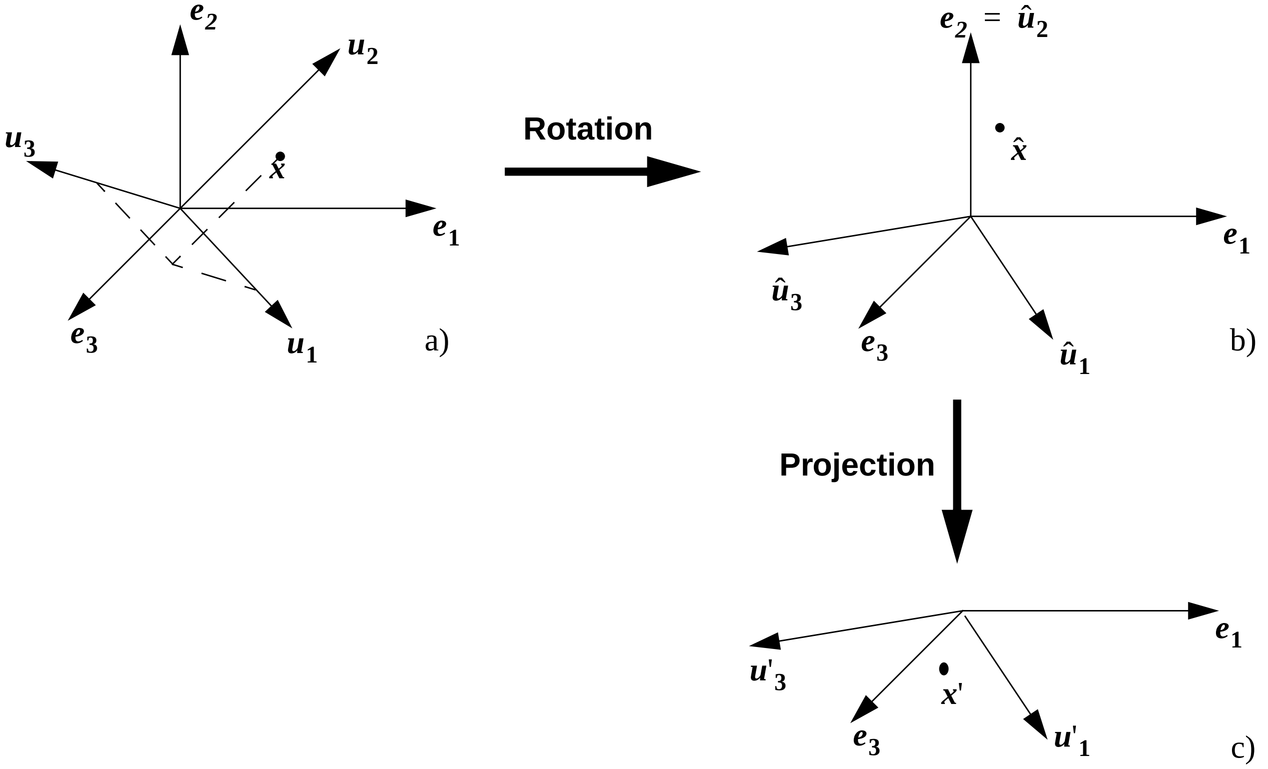

FIGURE 5. Projection after *n*-dimensional rotation:
a) Initial representation of *x* in *n*-space
b) n-dimensional rotation of u_2 to e_2
c) Projection onto the space defined by $\{e_1, e_2\}$

One of the problems of this approach is that an n-dimensional rotation is a non-trivial operation, since, in n dimensions, we have $(n \cdot (n-1))/2$ degrees of freedom. In our implementation we used the rolling ball algorithm developed in [6].

This strategy preserves orthogonality, that is, if the eigenvectors $\boldsymbol{u}_i$, $1 \le i \le n$ are orthogonal before projection the subset of eigenvectors remaining after the projection will also be orthogonal. This is due to the fact that all the projected eigenvectors are parallel to the subspace they are being projected to.

However, there are some important disadvantages that make this strategy useless:

- $\boldsymbol{x}$ is moved to $\hat{x}$ during the n-dimensional rotation, and, as a consequence, its subsequent projection is not orthogonal.
- The projected eigenvectors do not correspond to the eigenvectors of a meaningful matrix K'.

Apparently, none of the described simple projection methods can be used to develop an efficient subsampling strategy. Specifically, the orthogonal projection of the second example does not minimize the error introduced by the subsampling operator.

5 Analysis of Existing Subsampling Operators

In this section we analyze a well-known mesh reduction operator: the *edge collapse* operator first introduced in [7] as part of the progressive mesh algorithm.

As already explained the geometric information of the mesh can be stored in a vector $x \in R^n$, where n corresponds to the number of mesh vertices. In this n-dimensional space a collapse of two vertices v_i and v_j sharing one edge can be described as a *non-orthogonal projection*: one of the vertices, v_j for example, disappears, while the other, v_i, is moved so as to minimize some error norm. This operation is shown graphically in Figure 6:

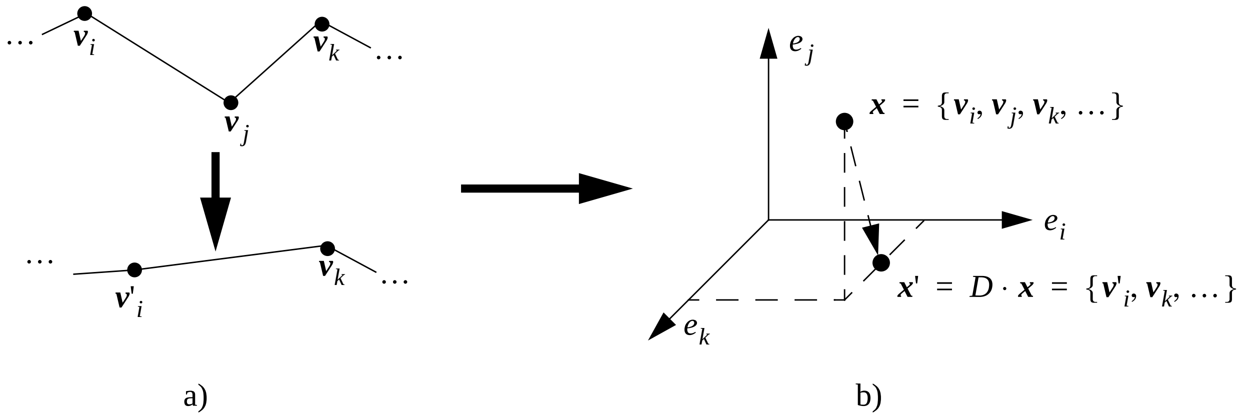

FIGURE 6. The edge collapse operator:
a) Standard edge collapse of a one-dimensional signal
b) Results of the edge collapse operation in *n*-space

In Figure 6 the value of $\boldsymbol{v}'_i$ is computed as a linear combination of the vertices of mesh $\boldsymbol{x}$ and the vertex $\boldsymbol{v}_j$ vanishes which corresponds to a dimensionality reduction by one. Formally, the progressive mesh operator D projects $\boldsymbol{x}$ onto a subspace spanned by the remaining unit vectors. Note that this projection is not necessarily orthogonal.

If we use the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$ as a norm to measure the error introduced by the edge collapse, we see that the best collapse would remove the vertex $\boldsymbol{v}_l$ that is closest to zero. Furthermore in the case of straightforward vertex removal where the vertex $\boldsymbol{v}_l$ is collapsed into another vertex, the projection would be orthogonal.

However, [2] for instance, showed that straightforward collapses don't provide locally optimal representations. This result is somewhat disappointing, but can be explained easily: $\boldsymbol{x}$ contains only the geometric information of the mesh, not the topological information; this implies that the distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$ does not correspond to the "difference" between the meshes represented by $\boldsymbol{x}$ and $\boldsymbol{x}'$.

Let's extend the framework presented in Figure 6, such as shown in Figure 7. In this setting the error introduced by a collapse operation is not computed as the difference between $\boldsymbol{x}$ and $\boldsymbol{x}'$, but between $\boldsymbol{x}$ and $\boldsymbol{x}_e$, where the vector $\boldsymbol{x}_e$ is computed from the vector $\boldsymbol{x}'$ using the upsampling (interpolation, subdivision) operator U. Hence, the optimal edge collapse will minimize $\|\boldsymbol{x} - \boldsymbol{x}_e\|$ for a given U. The interpolation operator U re-introduces the vertex removed by the subsampling operator D based on the following prerequisites:

- topology of the mesh
- geometry of the mesh
- error norm

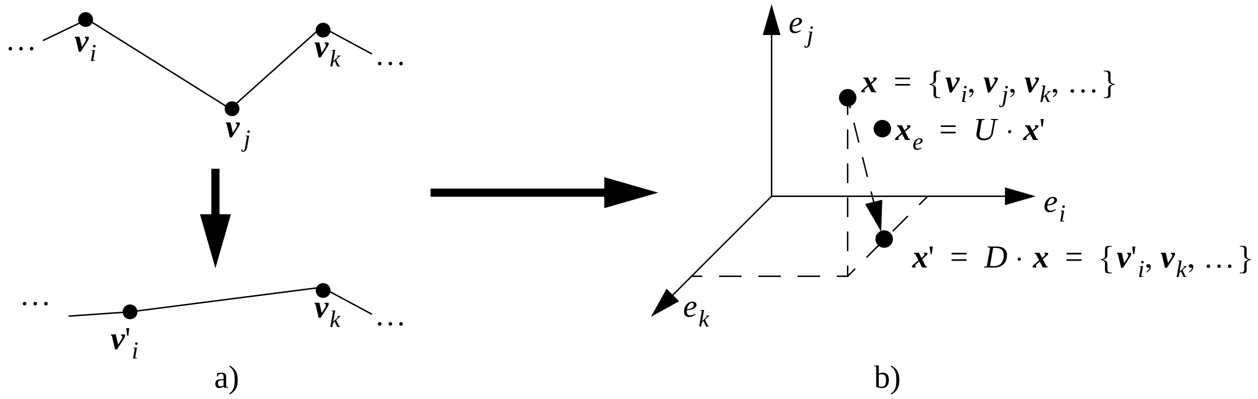

FIGURE 7. Edge collapse in n-space using a standard error norm
a) Standard edge-collapse for a one-dimensional signal
b) Result of the edge collapse operation in *n*-space

Finally, we merge the results presented in Section 3 with the results from this section. Our idea is to construct operators that compute a *combined subsampling and fairing of the mesh.* This concept is presented graphically in Figure 8:

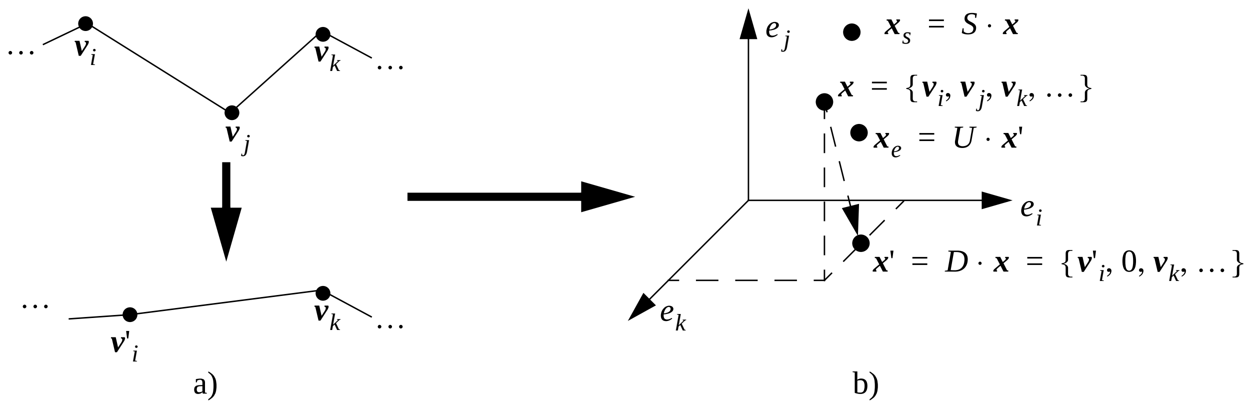

FIGURE 8. Edge collapse in n-space using a standard error norm
a) Standard edge-collapse for a one-dimensional signal
b) Result of the edge collapse operation in *n*-space

In this setting, the geometry vector $\boldsymbol{x}$ is projected to $\boldsymbol{x}'$, and, as before, the upsampling operator U used to compute the vector $\boldsymbol{x_e}$ storing the positions of the vertices $\{\boldsymbol{v}_l | 1 \le l \le n\}$ on the re-interpolated mesh. The difference in this setting is how we compute the error.

A disadvantage of a brute force minimization of $\|\boldsymbol{x} - \boldsymbol{x_e}\|$ is that error norm will only try to maintain the shape of the mesh, regardless of local artifacts. A different approach consists of imposing further constraints: We compute the edge collapse that introduces a small error, i.e. $\|x - x_e\|$ is "as small as possible" and smooth the surface, i.e. $\|x - x_s\| > \|x_e - x_s\|$ whenever possible. x_s is a smoothed version of the mesh computed by some predefined fairing operator *S*. We might combine the two error terms as follows:

$$min\{\alpha\|x - x_e\| + \beta\|x_e - x_s\|\} \quad (14)$$

The first term guarantees that important features are not removed from the mesh during the fairing, while the second term allows to obtain smoothed and subsampled versions of the mesh. The determination of *D* and *U* is left to future research.

6 References

[1] N. Amenta, M. Bern, and M. Kamvysselis. "A new voronoi-based surface reconstruction algorithm." In M. Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 415–422. ACM SIGGRAPH, Addison Wesley, July 1998. ISBN 0-89791-999-8.

[2] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. "Implicit fairing of irregular meshes using diffusion and curvature flow." In *SIGGRAPH '99 Proceedings*, Computer Graphics Proceedings, Annual Conference Series. ACM SIGGRAPH, ACM Press, Aug. 1999.

[3] M. Garland and P. S. Heckbert. "Surface simplification using quadric error metrics." In T. Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 209–216. ACM SIGGRAPH, Addison Wesley, Aug. 1997. ISBN 0-89791-896-7.

[4] J. P. Grossman and W. Dally. "Point sample rendering." pages 181–192. Rendering Techinques '98, Springer Verlag, 1998. Held in Vienna, Austria, July.

[5] I. Guskov, W. Sweldens, and P. Schröder. "Multiresolution signal processing for meshes." In *SIGGRAPH '99 Proceedings*, Computer Graphics Proceedings, Annual Conference Series. ACM SIGGRAPH, ACM Press, Aug. 1999.

[6] A. J. Hanson. "Geometry for N-dimensional graphics." In P. Heckbert, editor, *Graphics Gems IV*, pages 149–170. Academic Press, Boston, 1994.

[7] H. Hoppe. "Progressive meshes." In H. Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 99–108. ACM SIGGRAPH, Addison Wesley, Aug. 1996. held in New Orleans, Louisiana, 04-09 August 1996.

[8] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. "Interactive multi-resolution modeling on arbitrary meshes." In M. F. Cohen, editor, *Computer graphics: proceedings: SIGGRAPH 98 Conference proceedings, July 19–24, 1998*, Computer Graphics -proceedings- 1998, pages 105–114, New York, NY 10036, USA and Reading, MA, USA, 1998. ACM Press and Addison Wesley.

[9] G. Taubin. "A signal processing approach to fair surface design." In R. Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 351–358. ACM SIGGRAPH, Addison Wesley, Aug. 1995. held in Los Angeles, California, 06-11 August 1995.