Signal Processing over Triangle Meshes

Dinesh Shikhare

National Centre for Software Technology*

Abstract

Triangle mesh models have become established as the most general and flexible scheme to represent complex surface models in the areas of architectural design, mechanical CAD, heritage sculptures, etc. for various applications such as interactive visualization, FEM analysis, progressive transmission for collaborative exchange of models in networked environments. With the arrival of powerful range image scanners, acquisition of very detailed and accurate data representing these models has become common. These data are also represented as triangle meshes. In the recent years a lot research has taken place in the use of surface meshes.

In this report we present our study of the state-of-the-art in development of signal processing techniques over the domain of unstructured triangle meshes. These techniques include fairing of surfaces, filtering of models for denoising, smoothing, detail suppression, feature enhancement, spectral decomposition. We also study the applications that have driven these developments: compression of 3D models, watermarking of models, multi-resolution editing, interactive geometric modeling and progressive transmission.

1 BACKGROUND

Signal processing toolkits on spatial data entities like discrete 1D signals, 2D signals (images), 3D signals (volumetric data) are well understood and utilized extensively. Various mathematical techniques like Fourier transform, Wavelets, and other separable transforms [2] have been developed to treat data like vectors, images, volumetric data as signals. Many signal processing transformations are applied on these data like denoising, low-pass, high-pass and band-pass filtering, convolution with various transformation kernels, and so on. Signal processing transforms have also been recently used for applications like compression of images [2], watermarking of digital content like images, audio signal, digital video and progressive transmission scheme for these media elements.

The common feature among all these data is that they have a regular structure. One-dimensional digital signals are sampled at regular intervals and are represented in a regular fashion in the sense that i-th data element has a neighbour at $(i-1)$-th position and a neighbour at $(i+1)$-th position (except the boundary elements). This regular structure lets us easily think of the data in terms of a weighted sum of some basis functions. For example, a one-dimensional digital signal can be decomposed into harmonics of sinusoidal basis functions. It could also be decomposed and represented as a "coefficient – basis function" summation using the Haar basis. In general, the decomposition may be represented as

$$f(x) = \sum a_i B_i$$

where, $f(x)$ is the original function, a_i are the coefficients for the corresponding basis functions B_i.

Most of the mathematical tools of these kind easily extend from their one-dimensional statement to higher-dimensional formulations. For example, discrete Fourier transformation extends to higher dimensional data models; Haar wavelet transforms are easily extended for use on higher dimensional data like 2D image arrays, 3D volumetric data. Again, the structure among the elements of these data is uniform in terms of their neighbourhood. This structure allows an easy parameterization of the data along the directions of the inherent indexes the data elements have.

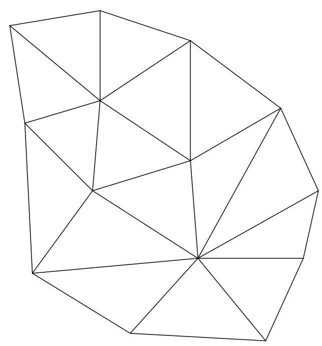

Figure 1: Triangle mesh – note the different degree of connectivity for each vertex

Triangle meshes representing surfaces of arbitrary topologies have now become very popular in many applications in 3D graphics and other related fields. These meshes lack a regular structure that is seen in some of the media elements discussed above. A *triangle mesh* is denoted as a pair (P, K), P is a set of N points positions $P = \{p_i \in R^3 | 1 \leq i \leq N\}$, in Euclidian space, $p_i = (x_i, y_i, z_i)$, and K is an *abstract simplicial complex* which contains all the

*Graphics & CAD Division, NATIONAL CENTRE FOR SOFTWARE TECHNOLOGY, Gulmohar Cross Road 9, Juhu, Mumbai 400049, India. {dinesh}@ncst.ernet.in

topological (*aka* connectivity) information. The complex K is a set of subsets of $\{1, ..., N\}$. These subsets are called simplices and come in three types: vertices $v = \{i\} \in V$, edges $e = \{i, j\} \in E$ and faces $f = \{i, j, k\} \in F$, so that $K = V \cup E \cup F$. Two vertices are *neighbours* if $\{i, j\} \in E$. The number of neighbouring vertices for the i-th vertex is denoted as its *degree*. The neighbourhood of a i-th vertex is denoted as $i^\star$. The mesh in Figure 1 shows a sample mesh having 14 vertices, 28 edges and 16 triangles. Note that this mesh has vertices with varying degrees.

Given this definition of neighbourhood between any two data elements, we know that general triangle meshes will have varying structure across different vertices. The earlier developments in the signal processing techniques cannot be easily extended to these data sets. Last few years' research in the handling of triangle mesh models has seen development of a new class of techniques for carrying out signal processing transforms which are intuitively identical to those applied on the "structured" models, but the detailed formulations vary substantially.

In this report, we present our study of the various approaches researchers have taken to treat unstructured triangle meshes as signals for carrying out global and local operations on these models. Since the field is still young it may not be easy to create a clear taxonomy of techniques, so we just present the contributions of various researchers and discuss the relationship among these efforts in the end.

2 FAIRING OF SURFACE DESIGN

Surfaces obtained from volumetric medical data by iso-surface generation algorithms [12] or constructed by integration of multiple range images [19] usually have a large number of triangles. During the acquision and integration of these surfaces, some parts become noisy either due to errors in registration or due to noise in the acquisition process. Fairing these surfaces automatically and efficiently has attracted some attention recently.

Fairing a surface model refers to removal of geometric discontinuities in the given triangle mesh model. The signal processing approach views this problem in terms of signal smoothing. The space of signals – functions defined on certain domain – is decomposed into orthogonal subspaces associated with different frequencies, with the low frequency components of the signal regarded as the subjacent data and the high frequency content as noise to be removed.

Taubin [16] has presented an efficient signal processing scheme for fairing surface designs. His approach is based on the classical result:

> *Fourier transform can be seen as the decomposition of the signal into a linear combination of the eigenvectors of the Laplacian operator.*

Karni and Gotsman [7] give a detailed formulation of this decomposition. Think of triangle mesh as a graph consisting of N vertices. The adjacency matrix A of this graph is populated by placing $A_{ij} = 1$ if $\{i, j\} \in E$ and 0 otherwise. The Laplacian operator associated with A is $L = I - DA$, where D is a diagonal matrix such that $D_{ii} = 1/d_i$, where d_i is the degree of i-th vertex. The eigenvectors of L form an orthogonal basis of R^n. The associated eigenvalues may be considered frequencies[1] and the three projections of each of the coordinate vectors of a 3D mesh geometry vector on the basis functions are the *spectrum* of the geometry. Note that there is a separate spectrum for each of x, y and z components of the geometry and they behave differently, depending on the geometric properties (e.g. curvature) of the mesh.

The essential observation is that geometries that are smooth relative to the mesh topology should yield spectra dominated by the low frequency components. By "smooth relative to the mesh topology" we mean that the local geometry, as defined by the topological neighbourhoods of the mesh, is such that the coordinates of the vertex are very close to the average coordinates of the vertex's neighbours. Hence, the Laplacian operator, when applied to the mesh geometry, will yield very small values.

It is interesting to note that the basis used for the decomposition of the triangle mesh signal is independent of the geometric realization of the surface in terms of coordinate value of the vertices. The basis is entirely derived by using the connectivity information of the mesh. Using this formulation as the basic tool for filtering of surface geometry has some limitations. The biggest limitation is the time complexity of computation of the eigenvectors, which runs into $O(n^3)$ steps, which means that computing the basis for meshes containing even 1000 vertices is unthinkable. Hence this scheme of signal processing is usable only for small meshes. Application of such a method for large meshes must be done by first partitioning the mesh into smaller pieces and then reconstructing the model after the application of the filtering process to the small pieces.

In order to avoid this computational limitation, Taubin [16] proposed an efficient scheme that uses repeated application of a spatial domain operator on the mesh to obtain fair surface. He uses the *first order* neighbourhood structure for the formulation of the operator.

A discrete surface signal is a function $x = x_1, ..., x_n{}^t$ defined on the vertices of a polyhedral surface. The Laplacian of the signal is a weighted average over the neighbourhoods:

$$\triangle x_i = \sum_{j \in i^\star} w_{ij}(x_i - x_j)$$

where the weights w_{ij} are positive numbers that add up to one. The weights can be chosen in many different ways taking into consideration the neighbouring structures. One particularly simple choice is to set w_{ij} equal to the inverse of the

[1] The caveat here is that treating the eigenvalues as the frequencies is only a notional concept. Since the underlying mesh is unstructured, the idea of some frequency being associated with some discrete feature is very hard to establish. However for the application at hand, this question is not very bothersome, although it is interesting to investigate.

number of neighbours $1/|i^\star|$ of vertex v_i, for each element j of $i^\star$. A more general way of choosing weights for a surface with the first order neighbourhood structure is using a positive function $\phi(v_i, v_j) = \phi(v_j, v_i)$ defined on the edges of the surface

$$w_{ij} = \frac{\phi(v_i, v_j)}{\sum_{h \in i^\star} \phi(v_i, v_j)}$$

For example, the function can be the surface are of the two faces that share the edge, or some power of the length of the edge $\phi(v_i, v_j) = \|v_i - v_j\|^\alpha$. The exponent $\alpha = -1$ works in order to position each vertex in the centroid of its neighbourhood.

If $W = (w_{ij})$ is the matrix of weights, with $w_{ij} = 0$ when j is not a neighbour of i, matrix L can now be defined as $L = I - W$. Since the decomposition into eigen vectors is infeasible, an approximate projection is applied for achieving low-pass filtering. This is formulated as the multiplication of function $f(L)$ of matrix L by the original signal

$$x' = f(L)x,$$

and this process can be iterated multiple times

$$x^k = f(L)^k x$$

The function of one variable $f(l)$ is the transfer function of the filter. In the case of Gaussian smoothing the transfer function is $f(l) = 1 - \lambda l$. To define a low-pass filter, we ned to find a polynomial such that $f(l_i)^k \approx 1$ for low frequencies and $f(l_i)^k \approx 0$ for high frequencies in the region of interest, say $l \in [0, 2]$. Taubin chose

$$f(l) = (1 - \lambda l)(1 - \mu l)$$

where, $\lambda > 0$, and μ is a new negative scale factor such that $\mu < -\lambda$. That is, after performing the Gaussian smoothing step with the positive scale factor λ for all vertices – the shrinking step –, perform another similar step

$$x'_i = x_i + \mu \triangle x_i$$

for all the vertices, but with negative scale factor μ – the unshrinking step. The value of the band-pass frequency l_{BP}, such that $f(l_{BP}) = 1$. The value of l_{BP} is

$$l_{BP} = \frac{1}{\lambda} + \frac{1}{\mu} > 0.$$

This algorithm is linear in both time and space, simple to implement and produces smoothing without shrinkage. The λ value must be chosen so as to minimize k, the number of iterations of the filter. Taubin [16] claims that the algorithm produces acceptable faring within 50 iterations. Most of the literature on fair surface design combines subdivision schemes with smoothing formulations [10, 4]. Subdivision of the mesh is done to create more points to capture the smooth local geometry features. Fairing procedure and subdivision steps alternate until a surface satisfying the desired properties is obtained.

Guskov et al [4] present a toolbox of signal processing utilities on triangle mesh models. In addition to Taubin's work, they illustrate enhancement of features, generation of texture coordinates for the filtered and refined models (using subdivision). They call the Laplacian filter a *relaxation operator*, R

$$Rp_i = \sum_{j \in i^\star} w_{ij} p_j.$$

The weights w_{ij} are functions of not only the degree of connectivity in the locality but also the geometric positions of the vertices. This relaxation operator is used to achieve smoothing.

Enhancement of features is achieved to emphasize certain frequency ranges. The enhancement of points is achieved by

$$Ep_i = p_i + \xi(p_i - R^k p_i)$$

where $\xi > 1$. This technique enhances the features that are suppressed by low pass filtering of the model.

3 COMPRESSION

Many compression schemes for traditional media, such as images, employ spectral methods to achieve impressive lossy compression ratios, for example, the popular JPEG method [13] which relies on the discrete cosine transform. These involve expressing the data as a linear combination of a set of orthogonal basis functions, each function characterized by a "frequency." The underlying assumption is that a relatively good approaximation may be obtained by using only a small number of low-frequency basis functions. The coefficients of these selected basis functions may then be compactly encoded by using a variety predictive encoders, lossy quantizers and symbol encoders like Huffman compression or arithmetic encoder scheme.

Karni and Gotsman [7] use this basic idea for the compression of triangle mesh geometry. They use the formulation described in the previous section for decomposition of mesh geometry signal into its spectral components. Only a few low-frequency coefficients are selected for achieving a high compression ratio. The connectivity of the mesh is then compressed using best of the earlier compression schemes [18, 3, 1, 17, 15].

For large meshes, computation of eigenvectors is very expensive. Hence they partition the mesh into smaller submeshes, each of which is then treated separately. This, of course, results in degradation of coding quality and can be seen after reconstruction in the form of "edge-effects" along the sub-mesh boundaries, but has the advantage that local properties of the mesh are captured better. In order to minimize the damage, the partitions should be well balanced, that is, each sub-mesh should contain approximately the same number of vertices, and also, the number of edges straddling the different sub-meshes, the *edge-cut* be minimized. Optimal solution to this problem is NP-complete. Karni and Gotsman use an algorithm called MeTiS [8].

The encoding of the coefficients of the spectra are uniformly quantized to finite pricision. This, of course, introduces further loss in the encoding of the model. The resulting set of integers is then entropy coded using Huffman coder [13]. Quantization errors can be minimized by using sophisticated quantization schemes like median-cut quantization.

Khodakovsky et. al. [9] introduce a wavelet based technique for progressive compression of geometry. This algorithm requires the source mesh to have a regular structure. They have defined wavelet basis over manifold surfaces for multi-resolution analysis. The multi-resolution analysis of the the surface naturally provides levels of detail progressively. The data stored to represent the model begins with a coarse model and subsequent data stores the detail coefficients to refine the model. Compression is achieved since the detail coefficients are incrementally stored and are much smaller in magnitude. These small values can be encoded efficiently using very small number of bits. Also, many higher order detail coefficients can be ignored due to their negligible values. The limitation of this approach is that it only works for manifold surfaces and has a strong requirement of a regular connectivity in the mesh.

Both these approaches work effectively on meshes that model smooth surface geometry since a large amount of geometric information is captured by the low frequency components of the spectral analysis and in multi-resolution analysis. These methods do not work sufficiently well on engineering models containing sharp edges and folds. The sharp edges correspond to the high frequecies in the decomposed signal, forcing the coding of large number of coefficients.

Since both these schemes are lossy, the issue of controlling lossiness and also that defining lossiness is very important for use in precision applications.

4 FOURIER TRANSFORM BASED TECHNIQUES FOR SUPPRESSION OF DETAILS

In triangle mesh models, the geometric features that represent details like small notches, protuberances, slots, sockets, faired corners require a lot of triangular elements for accurate representation. Interactive visualization and FEM analysis applications do not require these detailed features in models because: (a) the visual or numerical results obtained by having those features in the model are not significant different from those obtained by simplifying the models, (b) the computational load increases significantly in having to process the additional large number of triangle present in the model to represent the features. Hence a lot of research has taken place to simplify the model to reduce the triangle count [5]. The criteria for selection of vertices and triangles to be removed determines the usability of a particular simplification algorithm for the given application.

Lee and Lee [11] have presented a Fourier transform based scheme for suppression of geometric details in polygon mesh models that represent solids. Their scheme is based on low-pass filtering in the frequency domain. The detail suppression procedure consists of the following steps:

1. Construct a bounding volume for the triangle mesh model.
2. Discretize this bounding volume into $X \times Y \times Z$ number of elements in x, y and z directions. These volumetric elements (voxels) are then labelled with colour values as follows:
   (a) if the voxel lies inside the solid object, label it with a large positive value, k,
   (b) if the voxel lies on the boundary of the solid model, label it with a positive value l such that $l < k$, and
   (c) if the voxel lies outside the solid, label it with a negative value.

   This gives a volumetric data representing the solid object, $h(x, y, z)$.
3. Obtain a Fourier transform of the volumetric data:
$$H(u,v,w) = \int\int\int h(x,y,z) e^{-j2\pi(ux+vy+wz)} dx dy dz$$
4. Apply a low-pass filter $G(u, v, w)$ to this the transformed volumetric data to obtain:
$$H'(u,v,w) = G(u,v,w) \circ H(u,v,w)$$
   This will eliminate the high-frequency components in the signal corresponding to the small and detailed features in the model.
5. Apply inverse Fourier transform to obtain low-pass filtered model in its volumetric representation.
$$h'(u,v,w) = \int\int\int H'(u,v,w) e^{-j2\pi(ux+vy+wz)} du dv dw$$
6. Reconstruct the vertices, edges and faces of the polygonal model to obtain the simplified polygon mesh. This step of the algorithm is the hardest to implement. The geometry is recovered as follows:
   (a) After the transform and low-pass filtering there may be some gaps in the boundary in the volumetric data. Fill the gaps in the boundary colour to ensure that the data represents a solid model.
   (b) Project the vertices of the original model onto the filtered volumetric model to obtain the new positions of the vertices. There may be cases when the vertices can not be projected because the boundary representing some small feature has completely vanished. In such cases there is no need to project these vertices – the small feature has been completely removed.

(c) Construct edges, faces using the connectivity in the original model.

(d) Decimate the those vertices that correspond to linear edges and planar collection of faces.

Lee and Lee have demonstrated this technique of simplification on solids that are formed by graph functions of the type $z = f(x, y)$. This method is necessarily interactive in its use. The selection of the cut-off frequency of the low-pass filter plays a crucial role in the kind of detail that is retained in the resulting model.

5 WATERMARKING

Watermarking provides a robust method for copyright protection of digital media by embedding in the data information identifying the owner. The bulk of research on digital watermarking has focused on media such as images, video and text. Robust watermarks must be able to survive a variety of "attacks", including resizing, cropping and filtering. For resilience to such attacks, recent watermarking schemes employ a *spread-spectrum* approach – they transform the document to the frequency domain and perturb the coefficients of the perceptually most significant basis functions.

Emil Praun et al [14] extend this spread spectrum approach to work for watermarking of arbitrary triangle meshes. They transform the triangle mesh geometry by decomposing it into the basis they defined using the *Progressive Meshes* scheme [6]. A randomly chosen watermark $w = (w_1, ..., w_m)$ is inserted by scaling the m largest coefficients by small perturbations $(1 + \alpha w + i)$. Given a suspect document, an extracted watermark $w^\star$ is computed as the difference on the same set of frequency coefficients between the suspect data and the original watermarked data. The watermark is declared to be present based on the statistical correlation of $w^\star$ and w. The robustness of this scheme derives from hiding the watermark in many different frequencies with the most energy.

They use the multi-resolution scheme of [6] to decompose the the mesh geometry into basis functions $\Phi = (\phi^1, ..., \phi^m)$. These are selected by identifying the m refinement operations that that cause the greatest geometric change to the model. For each of these m refinements, we define a scalar basis function over its corresponding neighbourhood in the original mesh. To embed the watermark in the model, the vertex corresponding to the centre of each of the selected neighbourhoods is then perturbed by a product of vector direction of the vertex computed within the neighbourhood and the watermark element taken from the vector w.

To extract the embedded watermark from possibly "attacked" model, the suspect mesh must be registered by finding the similarity transform (consisting of uniform scaling, rotation and translation) to coincide with the original model. The registered geometry is then resampled in order to produce a mesh with the same connectivity as the original. By taking the difference between the 3D coordinates of the vertices of this resampled mesh and those of the original, a vector of 3D residuals is accumulated. The watermark is then extracted by recovering the positions of those vertices that were perturbed in the multi-resolution representation of the mesh.

While the process of embedding of the watermark is automatic, detection of watermark cannot always be non-interactive. The interaction is needed to specify an initial registration between the original mesh and the suspect mesh. The registration is then refined iteratively before further steps are performed.

6 DISCUSSION AND SUMMARY

We have presented our study of how the basic signal processing tools can be extended to unstructured triangle meshes. So far, various formulations have been developed for decomposition of the mesh geometry into component signals for further operations like fairing, feature enhancement, feature detection, suppression based on global analysis of the models. In addition to transforming the triangle mesh models into frequency domain, there have been efforts to create spatial domain filters to apply local operators for achieving the same effects. These tools have been then used in applications such as compression of geometry and watermarking.

The further research in this area is needed in the following directions:

1. The current formulations tend be in the form of transformations and filters that get applied globally on all elements of the triangle mesh. The user of these tools does not have the control on their application in a desired locality. To achieve localised application of signal processing operators, one needs a mechanism of specifying the locality of interest in terms of geometric or topological features and also a provision in the formulation to apply the operation only in the specified region.

2. Most of the work has concentrated on triangle meshes. However, in practice, many meshes are available as quadrilateral meshes, meshes consisting of a mix of different types of polygons. Also many practical models have non-manifold topology. Handling these special models will require extension of the formulation.

3. Efficiency and scalability is another issue in handling very large models that are being acquired now a days. Spectral analysis of large models can be a very time consuming task. For such model, different types of algorithms may be needed. One direction could be to do a multi-resolution analysis of the model and apply the spectral analysis to a low-resolution model and use the result to refine the analysis for the increasing complete model. Multi-grid techniques used in FEM and

CFD communities could be adapted here for handling the very large models.

References

[1] Mike Chow. Optimized Geometry Compression for Real-time Rendering. In *IEEE Visualization'97*, pages 347–354, 1997.

[2] Rafel Gonzalez and Richard Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1992.

[3] Stefan Gumhold and Wolfgang Strasser. Real-time Compression of Triangle Mesh Connectivity. In *SIGGRAPH 98*, pages 133–140, 1998.

[4] I. Guskov, W. Sweldens, and P. Schroeder. Multiresolution Signal Processing for Meshes. In *SIGGRAPH 99*, pages 325–334, 1999.

[5] Paul Heckbert and Michael Garland. Optimal Triangulation and Quadric-based Surface Simplification. *Journal of Computational Geometry: Theory and Applications*, November 1999.

[6] Hugues Hoppe. Progressive meshes. In *SIGGRAPH 96*, pages 99–108, 1996.

[7] Z. Karni and C. Gotsman. Spectral Compression of Mesh Geometry. In *SIGGRAPH 2000*, pages 279–286, 2000.

[8] G. Karypis and V. Kumar. MeTiS: a software package for partitioning unstructured graphs, partitioning meshes and computing fill-reducing orderings of sparse matrices. (version 4.0). University of Minnesota, Department of Computer Science, http://www-users.cs.umn.edu/k̃arypis/metis/metis.html, 1998.

[9] Andrei Khodakovsky, Peter Schroeder, and Win Sweldens. Progressive Geometry Compression. In *SIGGRAPH 2000*, pages 271–278, 2000.

[10] Leif Kobbelt, Swen Campagna, Jens Vorsatz, and Hans-Peter Seidel. Interactive Multi-resolution Modeling on Arbitrary Meshes. In *SIGGRAPH 98*, pages 105–114, 1998.

[11] Yong-Gu Lee and Kunwoo Lee. Geometric Detail Suppression by the Fourier Transform. *Computer-Aided Design*, 30(9):677–693, 1998.

[12] W. Lorenson and H. Cline. Marching cubes: A High Resolution 3D Surface Constrction Algorithm. In *SIGGRAPH 87*, pages 163–169, 1987.

[13] Mark Nelson. *Data Compression Handbook*. M & T Publishing, Inc., 1991.

[14] Emil Praun, Hugues Hoppe, and Adam Finkelstein. Robust Mesh Watermarking. In *SIGGRAPH 99*, pages 49–56, 2000.

[15] Jarek Rossignac. Edgebreaker: Connectivity Compression for Triangle Meshes. *IEEE Transactions on Visualization and Computer Graphics*, 5(1):47–61, January-March 1998.

[16] Gabriel Taubin. A Signal Processing Approach to Fair Surface Design. In *SIGGRAPH 95*, pages 351–358, 1995.

[17] Gabriel Taubin and Jarek Rossignac. Geometry Compression through Topological Surgery. *ACM Transactions on Graphics*, 17(2):84–115, April 1998.

[18] Costa Touma and Craig Gotsman. Triangle Mesh Compression. In *Proceeding of Graphics Interface 98*, June 1998.

[19] Greg Turk and Mark Levoy. Zippered Polygon Meshes from Range Images. In *SIGGRAPH 94*, pages 311–318, 1994.