Complexity of Geometric Models

Dinesh Shikhare

National Centre for Software Technology, Gulmohar Cross Rd. 9, Juhu, Mumbai 400049, India.

{dinesh|mudur}@ncst.ernet.in

Abstract

In information theory, the term complexity has been used as measure of information content in a given piece of data. Based on such a measure, information encoding schemes have been evaluated for their efficiency. For measuring the complexity of a textual data, probabilistic measures have been used. Such measures do not seem relevant for the data representing geometric information. The recent interest in the area of compact encoding of geometric data motivates this study of defining complexity measures in the context of geometric models and compression of geometric information.

The specific issues we study are:

- What constitues information in geometric models?
- How to model and encode geometric information for compact encoding?
- What can be the measures of complexity for different kinds of geometric models?

1 Introduction

3D Geometric models are being extensively used in many applications ranging from engineering design, analysis, visualization, architectural modeling, virtual mock-ups, games, entertainment and so on. With the increasing capabilities of the computing environments, visualization hardware, modern interactive modeling tools [10] and semi-automatic 3D data acquisition system [3], large and complex 3D models are becoming commonplace. Large models pose basic problems of efficient storage, transmission, rendering, analysis, etc. Many applications using 3D models require time-bound response. For example, interactive visualization systems require certain number of frames to be rendered per second to be usable – irrespective of the size of the model. Complex models defined in great detail cannot be used as such due to the limited bandwidth of storage, transmission, processing. To deal with this mismatch between the complexity of the 3D models and the capability of the resources, three areas of research have emerged eminently: (a) simplification of geometric models (see a survey in [7]), (b) geometry compression [6, 11, 16, 19, 20] and (c) progressive transmission and disclosure [2, 14, 17, 18].

The simplification and compression techniques identify *redundancy* in the representation of the geometric models and arrive at a new representation that compactly encodes the *non-redundant* information in the model. The key point here is that some *features* in the models represent the significant information that are of geometric and visual importance for the consumption of the data. The simplification techniques suppress the less significant features and thereby also reduce the data that was needed to describe those features. The loss-less geometry compression algorithms reduce the repeated integer references to the vertices in the connectivity. The lossy geometry compression techniques, however, strip off the geometric details by either reducing the number of triangles/polygons or number of bits per the coordinates of vertices. The progressive transmission schemes identify and transmit first the information that representes the significant features of the model, followed by the other information the constitutes the other details.

The motivation of this study is to determine some measure of complexity of a geometric model (in particular for a 3D polygonal mesh models) for answering the following questions:

1. What constitutes information in geometric models?
2. How to model and encode geometric information for compact encoding?
3. What can be the measures of complexity for different kinds of geometric models?
4. What can be optimal compression achievable for a give geometric model?
5. Is it possible to arrive at an information theoretic framework for estimating information content and complexity of a given model?

Complexity measures are needed for practical requirements of processing large geometric models. For processing needs such as rendering, computation of global illumination, computational fluid dynamics, etc. the time required to complete the computations is largely dependent on the complexity of the given model. It is very helpful to know the complexity of the model to estimate the time required to complete the computational job. On the other hand is also useful to reduce the complexity of the model to some measure that can be handled by the computational task in a fixed amount of time.

The rest of the report is organized as follows. In the next section we characterize various kinds of models and describe the features that characterize them. We also survey the previous work in derivation of shape complexity measures. We conclude by highlighting the limitations of the work done so far in the development of complexity measure.

2 Characterization of Models

2.1 Some Preliminary Definitions

Polygon mesh models are typically defined in terms of (a) *geometry* – the coordinate values of vertices of the meshes making up the model, (b) *connectivity* – the relationship among the vertices which defines the polygonal faces of the mesh (also called as *topology* of the mesh), and (c) *attributes* - such as color, normal and texture coordinates at the vertices.

Other information such as texture images and material properties are also commonly present as a part of the models, usually associated with meshes or groups of meshes.

A *polygonal mesh model* O consists of a set M of polygon meshes and associated non-geometric properties. A *polygon mesh* $m \in M$ consists of a set V of vertices, a set E of edges and a set P of polygons. Each vertex corresponds to a point position from the set $X = \{x_i \in R^3\}$. An edge e is represented as a pair (v_1, v_2) of references into the list of vertices. A polygon p is represented as a sequence $(v_1, v_2, ..., v_k)$ of references into the list of vertices. A *triangle mesh* is a special case having all triangular polygons. Generally in the actual representation, edges are implied and not explicitly stored.

A mesh having each edge shared by at most two polygons and at least one polygon forms a *manifold*. A mesh having an edge shared by more than two polygons represents a *non-manifold*. A mesh representing a manifold surface is *closed* if all edges are shared by 2 polygons, otherwise it is *open*.

In a mesh model O, we call two polygons as *neighbouring polygons* if they share an edge. There exists a *path* between polygons p_i and p_j if there is a sequence of neighbouring polygons $p_i, p_1, p_2, ..., p_j$. A subset O_c of the mesh model O is called a *connected component* if there exists a path between any two polygons in O_c. Note that a given mesh model may have multiple connected components. A mesh can be trivially decomposed into its connected components using simple labelling algorithm.

When deriving a compact encoding for 3D models, there are three processes that work on a polygonal mesh model O and are in some sense related: compression, simplification and progressive disclosure.

Compression is the process of deriving a new encoding for a given polygonal mesh model O such that the number of bits needed in the new encoding is much smaller than the number of bits needed for the uncompressed representation. If $U(O)$ and $C(O)$ denote the number of bits for the uncompressed and the compressed versions respectively, then the *compression ratio* is defined as

$$CR = \frac{U(O) - C(O)}{U(O)}.$$

Compression is said to be lossy if the decompression process cannot give back the original polygon mesh model exactly.

Simplification is the process of deriving a new representation S, often referred to as an imposter, for a given polygonal mesh model O by which S becomes simpler to deal with normally for rendering purposes. This is usually in the form of lesser number of meshes or polygons or vertices and the elimination of fine detail in geometry and associated information, etc. Clearly simplification is a kind of lossy compression. A comprehensive review of simplification techniques may be found in [7].

Progressive disclosure is the process of deriving a new representation of a given polygonal mesh model O, which enables one to transmit a coarse representation of the model first and subsequently transmit the details to refine it. Compression strategies have been used in the progressive representation of models to also make the entire transmission data compact. Many groups have developed compression schemes incorporating progressive disclosure capabilities [2, 14, 17, 18]. The schemes that combine compression and progressive disclosure always have a trade-off between compression and the additional data needed for progressive representation of the models.

2.2 Classification of 3D Models

While a number of mathematical representations such as quadrics, bi-polynomials, NURBS, sweeps, Boolean operations etc. may be used during the initial creation of the 3D models, the final evaluated and stored representation is usually in the form of polygonal meshes, most commonly, triangle meshes. Many compression techniques have therefore focused on such triangle mesh representations. Quite naturally, specific compression schemes are best suited to specific classes of such complex 3D models with well distinguishable shape characteristics.

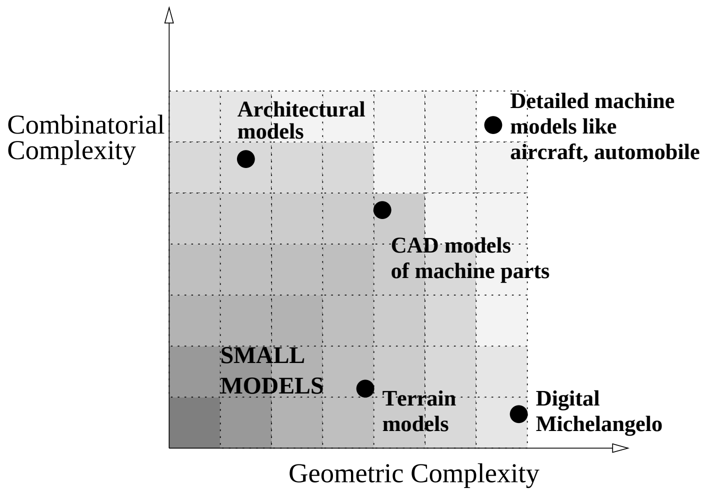

Figure 1: Classification of 3D models: Geometric complexity versus Combinatorial complexity

There have been a few attempts to classify 3D models. These are normally based on some formulation for measuring the complexity of the shapes and their representations. An early attempt is by Forrest [4] in which he classifies geometric models along three axes: geometric complexity (lines, planes, curves, surfaces, etc.), combinatorial complexity (number of components, edges, faces, etc.) and dimensional complexity (2D, 2.5D, 3D). More recent attempts have tried to relate polygonal representations to *shape factors* [21, 1, 5, 8, 12]. The shape factor addresses the complexity measure along the geometric complexity dimension. Some of the large 3D geometric models presently being worked upon in various computer graphics research laboratories around the world are shown in Figure 1 positioned in Forrest's model classification space.

Unlike digital models of natural shapes such as terrains, anatomies, sculptures and so on, such engineering models invariably have a number of repeated components, e.g., windows, pillars, fixtures, etc. in different orientations and positions. Typically such models consist of a large number of small polygonal meshes, each consisting of up to a few hundred triangles. In Figure 1, these models would be typically positioned in the upper right triangular region.

2.3 Features in Models

Characterization of 3D models is also possible in the context of various features. In a general setting it is hard to give any useful definition of features. Some earlier attempts are:

- "A feature is a region of interest on the surface of a part." — Pratt, 1985 [15].
- "Features are defined as geometric and topological patterns of interest in a part model and which represent high level entities useful in part analysis." – Henderson, 1990 [9].

The main difficulty here is that, in trying to be general enough to cover all possibilities, these definitions fail to pin things down sufficiently to give a clear picture. To make these definitions more concrete, we give a classification of features:

- functional feature: for example, a pivot,

- design feature: a rotating pin supported by two raised lugs,
- manufacturing feature: a turned cylinder, a milled slot with an in-line reamed hole through both walls, etc.
- application specific feature: these could be any combination of topological, geometric, metric, colour and texture attributes or non-visual features.

It is important to note that for a given 3D model, different sets of features may be extracted. The features identified for a model in the design feature-space could be much different from those recognised in the manufacturing feature space. For different applications, the model under consideration may be described in different feature spaces.

In a given domain of application, such as mechanical CAD modeling, it is possible to describe a given model in terms of specific set of parameterised features. A description of this kind is always more compact than a purely geometric description of a given model. But we must note that such features have a lot of implied knowledge that needs to be incoportated into the software and algorithms that process the representation.

In order to make use of features in processing geometric models, some core common aspects exist. These apply irrespective of the domain of problems and the particular problems at hand. The relevance of individual aspects may vary. These are:

1. *Identification of features:* A particular feature of interest in a given 3D model must be specified in a way such that it can be algorithmically identified in the 3D data. This definition can be a mix of topological properties, geometric metrics, colour and texture attributes and so on. Some higher-level features may be defined in terms of simpler low-level features. Such hierarchies of features are common in literature.
2. *Recognition of features:* Recognition of a feature in the given 3D model is an essential computational part feature-based processing of geometric models. For detecting an individual model, typically specialised filters need to be implemented.
3. *Suppressing features:* By suppressing a feature we mean removing a local instance of the feature while minimally disturbing the data around the feature.
4. *Reconstruction of features:* This is an inverse of “suppressing features.” By reconstruction we mean restoration of a previously suppressed feature in the data, preferably, without any loss of information of the original data.
5. *Encoding of features:* A machine representation of features, preferably a compact one, must be developed.
6. *Feature space conversion:* Given a model described in one feature space, often there is a requirement to convert the description to another feature space.

The techniques for simplification of geometric models identify specific features, recognize and suppress them to obtain a *less complex* approaximation of the model. Different applications require different sets of features for their processing needs. Lossless geometry compression schemes too encode specific local features compactly. These features are topological (connectivity-based) as well as geometry based. The lossy techniques suppress features and also reduce the data by reducing the number of bits used to represent coordinate values for the vertices in the model.

3 Optimal Storage Allocation

In the setting of lossy compression, let O be a 3D model whose use requires it to be in a triangle mesh representation A. The mesh A may be compressed by simplifying the mesh to reduce the number of vertices V, by quantizing the vertex coordinates to B bits and by using some lossless compression techniques. Kind and Rossignac [12] have carried out extensive studies in developing a set of metrics for estimating the shape complexity K of a given mesh model. Their research contributions are summerised in this section.

If the end application has a fixed requirements of size and complexity of the model to be handled, then it is important to address the following questions:

1. Given a bound on the compressed file size F, which values of V and B minimize the error between the original model and the compressed model?

2. Given a bound on the geometric error E, which values of B and V minimize the total number of bits for storage?

3. How does the relationship between B, V, F and E affect depend on the shape complexity K of the model?

Note that for a given set of values B and V, there can be large number of different approximations A' of the original mesh model A. The specific approximation would be determined by the algorithm that gives the simplified version.

King and Rossignac formulate the notion of shape complexity K based on error E introduced in the model as a result of lossy operators that affect B and V. Based on their framework, simplification of the original model can be carried out while the error remains bounded. The number of bits needed to store the simplified model is then a measure of the inherent complexity of the model.

To measure the error between the original model and the current approximation in the process of simplification is formulated using various criteria in [12, 8]. These are based on geometric deviation from the original model. They demonstrate the effect of curvature and other shape characteristics like sharp edges, corners on the number of triangles.

The limitations of these measures are: (a) they are primarily based on local measurements and fail to consider the global distribution of error, (b) these measures attempt to compute the obtamal storage requirements in isolation from the specific algorithms that carry out lossy compression, (c) the simplifying assumption that the given model can be considered as a piecewise composition of spherical surfaces limits the domain of models they can handle in estimation of complexity measure.

4 Feature Based Estimation of Complexity

Consider a scenario where a given geometric model can be completely described in term of application specific features of the model. These features could be design features, machining features, features for machine vision, and so on. Let us construct an alphabet F of parameterised features that are necessary for the description of the model O. The symbols in the model can then be entropy coded [13] to achieve compression.

The model O described as a sequence of features $f \in F$ can be compressed as follows. Let the model be described as

$$O = (f_0, f_1, f_2, ..., f_n)$$

where all the features are derived from the set F. The random occurrence of feature f_i may happen with probability $P(f_i)$, and such an event is said to contain information units

$$I(f_i) = log\frac{1}{P(f_i)} = -logP(f_i).$$

The quantity $I(f_i)$ would be called the self information of the parameterised feature f_i in the model O. The sum of such self information units of occurrences of the features in the sequence describing the model would be the shape complexity measure for the model

$$K(O, F) = \sum_i I(f_i)$$

where F is the set of features identified in the given model for the application at hand. The encoding may be carried out using some generic compression scheme such as Huffman coding or Arithmetic coding.

The set of features being modeled and recognised in such a complexity estimation framework could also be very generic and non-application specific. The feature patterns could be discovered from scratch. Discovery of feature patterns in geometric models is a part of our current work in progress. This work is described in the accompanying technical report for this semester.

5 Conclusions and Future Directions

In this report we have summarised the contemporary research in the area of complexity estimation for geometric models. Complexity measures are needed to estimate time needed for carrying out any analysis, rendering, transmission and storage of the models. Such measures are also required for the evaluation of model simplification and compression algorithms.

The current work in this direction is in very priminary stages and for now, makes a lot of simiplifying assumptions. The future directions for this work are: (a) feature-based measures for estimation of shape complexity, (b) discovery of generic self-information of the given model from the self-information of the individual features, (c) development of more sophisticated error measures between the original model and simplified models.

References

[1] N. Amenta, M. Bern, and M. Kamvysselis. A New Voronoi-based Surface Reconstruction. In *SIGGRAPH 98*, pages 415–421, 1998.

[2] D. Cohen-Or, D. Levin, and O. Remez. Progressive Compression of Arbitrary Triangle Meshes. In *IEEE Visualization 99*, pages 67–72, 1999.

[3] B. Curless and M. Levoy. A Volumentric Method for Building Complex Models from Range Images. In *SIGGRAPH 96*, pages 303–312, August 1996.

[4] A. R. Forrest. Computational geometry - achievements and problems. In R.E. Barnhill and R.F. Riesenfeld, editors, *Computer Aided Geometric Design*, pages 17–44. Academic Press, 1974.

[5] M. Garland. *Quadric-based Polygonal Surface Simplification*. PhD thesis, Carnegie Mellon University, 1998.

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

[7] P. Heckbert and M. Garland. Survey of polygonal simplification algorithms. In *Multi-resolution Surface Modeling Course*. ACM SIGGRAPH Course Notes, 1997.

[8] P. Heckbert and M. Garland. Optimal Triangulation and Quadric-based Surface Simplification. *Computational Geometry*, 14:49–65, 1999.

[9] M. R. Henderson, S. N. Chuang, P. Ganu, and P. Gavankar. Graph-based feature extraction. Technical report, Arizona State University, 1990.

[10] Discreet (Autodesk Inc.). 3dstudio MAX R4. http://www.discreet.com, 2000.

[11] M. Isenbueg and J. Snoeyink. Face Fixer: Compressing Polygon Meshes with Properties. In *SIGGRAPH 2000*, pages 263–270, 2000.

[12] D. King and J. Rossignac. Optimal Bit Allocation in Compressed 3D Models. *Computational Geometry*, 14:91–118, 1999.

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

[14] R. Pajarola and J. Rossignac. Compressed Progressive Meshes. Technical Report GIT-GVU-99-05, GVU Center, Georgia Tech., Atlanta, USA, 1999.

[15] M. J. Pratt and P. R. Wilson. Requirements for support of form features in a solid modeling system. Technical Report R-85-ASPP-01, CAM-I, 1985.

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

[17] G. Taubin, A. Gueziec, W. Horn, and F. Lazarus. Progressive Forest Split Compression. In *SIGGRAPH 98*, pages 123–132, 1998.

[18] G. Taubin, W. Horn, and P. Borrel. Compression and Transmission of Multi-resolution Clustered Meshes. Technical Report RC21398-(96630)-2FEB1999, IBM T.J. Watson Research Centre, New York, USA, 1999.

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

[20] C. Touma and C. Gotsman. Triangle Mesh Compression. In *Proceeding of Graphics Interface 98*, June 1998.

[21] G. Turk. Re-tiling Polygonal Surfaces. In *SIGGRAPH 92*, pages 55–64, 1992.