Volume xx (200y), Number z, pp. 1–2

Polygonal Model Compression with Graph Symmetries

Andrej Jočić[1]

[1]University of Ljubljana, Faculty of Computer and Information Science, Slovenia

Abstract

TODO the abstract (see http://www.acm.org/about/class/class/2012)

CCS Concepts

•***Computing methodologies*** → ***Mesh models;*** *Graphics file formats;* •***Theory of computation*** → *Data compression;*

1. Introduction

Polygonal models are widely used for representing 3D objects in computer graphics. With increased rendering system capabilities and the desire for greater precision, the size of these models has been increasing. To cut down transmission and static storage costs, specialized compression schemes have been developed. In previously established [MLDH15] terms, we will be focusing on lossless single-rate global compression of static polygonal models. This means that we will be considering fixed (non-evolving) data, which will be decompressed it all at once (without progressive transmission at different levels of detail) and in its entirety (not focusing on specific regions of the model as requested by the user).

The information we usually store is the model's geometry (positions of vertices), connectivity / topology (incidence relations among elements), and optionally also attributes (normals, colours, etc.). Typically the connectivity information takes up the most space (around twice as much as the goemetry information in a triangle mesh homeomorphic to a sphere [Ros99]), so it is the most important to compress. Many compression methods focus exclusively on manifold triangle meshes (or another constrained structure) to get the best possible results. In this paper we will consider the case of arbitrary polygonal models. Since the connectivity relations can be represented as an undirected graph, we can reduce the problem to that of graph compression. Specifically, we will be exploiting symmetries in the graph, which can often occur in the graphical domain.

1.1. Related Work

this an that paper hasdealt with symmetry-based mesh compression, but to our knowledge it has not been done with automorphisms.

2. Methods

Čibej and Mihelič [ČM21] have developed a method general graph compression based on automorphisms. They define *symmetry-compressible* graphs G as ones that can be more succinctly represented with a "residual" graph G^π and an automorhpism (connectivity-preserving vertex permutation) π of G. Formally, $G \in \mathcal{SC}$ if $\exists \pi \in Aut(G)$ such that $|G^\pi| + |\pi| < |G|$, where $|\cdot|$ denotes a measure of representation size (e.g. number of edges).

The main computational issue is finding a (compressive) automorphism. The paper presents two approaches: graphlet search and bipartite completion. As Theorem TODO shows, we can just search for SC subgraphs.

2.1. Bipartite Completion

Though a star graph is a common occurence in the graphical domain, it is unfortunately not symmetry-compressible (measuring size by number of edges) (??) do we only need NSC?

2.2. Graphlet Search

This one is a lot slower, but may be more appropriate for our case (if we're not compressing on the fly during transmission).

2.3. Compressed file format

Most commonly used file formats for storing 3D models have the following structure: a header with metadata, then the vertex data, and finally the connectivity data. Connectivity data is usually stored as a list of faces, where each face is a list of vertex indices. This means the graph we will be compressing is the 1-skeleton of the model. Alternatively we could compress the face adjacency graph, but this would just inflate such an input file.

When decompressing, we will need to compute the facial walks of the skeleton in order to produce the original face list.

3. Results

4. Discussion

The 1-skeletons of manifold meshes (which the vast majority of real-world models are) are planar graphs. So the face adjacency graph, which is the dual of the 1-skeleton, is also planar. This means these graphs are very sparse with $m = O(n)$, where n is the number of vertices and m is the number of edges. This easily explains the poor performance of the bipartite completion method, since many edges have to be added to make a complete bipartite subgraph.

4.1. Conclusions

Please direct any questions to the production editor in charge of these proceedings.

References

[ČM21] ČIBEJ U., MIHELIČ J.: Graph automorphisms for compression. *Open Computer Science 11*, 1 (2021), 51–59. 1

[MLDH15] MAGLO A., LAVOUÉ G., DUPONT F., HUDELOT C.: 3d mesh compression: Survey, comparisons, and emerging trends. *ACM Comput. Surv. 47*, 3 (feb 2015). URL: https://doi.org/10.1145/2693443, doi:10.1145/2693443. 1

[Ros99] ROSSIGNAC J.: Edgebreaker: Connectivity compression for triangle meshes. *IEEE transactions on visualization and computer graphics 5*, 1 (1999), 47–61. 1