Geometry Compression for ASCII Scenes

Martin Isenburg*
University of North Carolina at Chapel Hill

1 Compression and VRML

The most popular way of distributing 3D content on the Web is in form of a textual representation of the scene such as VRML and its variants. The advantage of such a description is that it is very author friendly in the sense of being meaningful to the human reader. A scene represented in a textual format can be viewed, understood, and modified with any text editor (see Figure 1). Most importantly, anyone can do this, even without knowledge about the specific software package that generated the 3D content.

One disadvantage of an ASCII format is that scene files can become too large for efficient Web transmission when the scene contains many and/or detailed polygon meshes. Although the scene files are usually compressed with gzip compression, such general purpose coders do not come close to the compression rates that can be achieved with a dedicated geometry coder.

For image, audio, and video data an on-going standardization process has produced binary compression formats such as JPEG, GIF, and MPEG that are widely accepted. Software to read, create, save, and modify these formats is plentiful and easy to use. Efforts to develop a similar standard for compressed polygonal data have so far been without success.

Geometry compression for VRML has been an important item on the wish-list of the Web3D Consortium since 1996. It has been widely assumed that a binary format would be required to allow compressed geometry. This lead to the formation of the Compressed Binary Format workgroup, which (a) created a binary format for all VRML nodes and (b) proposed new *compressed* versions of five data-heavy nodes that would only exist in binary. Despite excellent compression results, in the end the proposal was rejected. Some felt it was the sentiment against an unreadable binary format from the author-side and the reluctance to support two VRML formats from the browser company-side that influenced the decision.

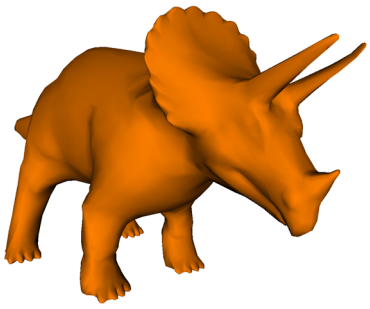
2 Eliminating the Binary Requirement

We have recently proposed a mesh compression technique that does not require a binary file format. In [Isenburg and Snoeyink 2002] we show how to code textured polygon meshes as a sequence of ASCII symbols that compresses well with standard gzip compression. One benefit of this approach is that it eliminates the binary requirement. In order to add compressed geometry to VRML (or now X3D) one no longer has to wait for a binary standard. The other benefit is that complete conformance between the ASCII version and the (eventual) binary version of a VRML scene would be possible—including the compressed nodes.

However, there was a significant gap between the compression rates of binary state-of-the-art geometry coders and the proposed ASCII coder. The reason for this was that binary coders use entropy coding to squeeze the produced symbol stream into as few bits as possible. Arithmetic coding, for example, outperforms gzip coding because it gives optimal compression in respect to the information entropy of the symbol sequence [Moffat et al. 1998].

We have combined the advantages of arithmetic and of non-binary coding by letting the arithmetic coder produce an ASCII string of zeros and ones instead of a binary bit-stream. Subsequent gzip compression aggressively compresses the resulting string of zeros and ones, since it only contains two different symbols.

*isenburg@cs.unc.edu http://www.cs.unc.edu/~isenburg/ac



```
Shape {
  appearance Appearance {
    material Material {
      diffuseColor 1 .5 0
    }
  }
  geometry IndexedFaceSet {
    creaseAngle .9
    coord Coordinate {
      point [ 3.660927 0.002173 ...  0.250595 -0.008488 ]
    }
    coordIndex [ 2805 2809 2814 2820 -1 ...  2606 2607 -1 ]
  }
}

Shape {
    ...
  geometry CompressedIndexedFaceSet {
    creaseAngle .9
    code [ 0 0 0 0 1 0 0 0 1 1 1 1 ...  0 0 0 1 1 1 1 0 0 0 1 ]
    bounding_box [ -10.2998 -3.6917 -2.9128 7.4163 4.0637 2.9442 ]
    bits 12
  }
}
```

Figure 1: A simple VRML scene containing the triceratops mesh (top) and its ASCII coded version (bottom). The compressed scene can still be viewed and edited in any text editor to, for example, change the material or the crease angle. However, the size of the gzipped VRML file is now only 8,392 instead of 56,307 bytes.

We have implemented an arithmetic coder that codes to and from ASCII and combined it with compression techniques for polygon mesh connectivity [Isenburg 2002] and geometry [Isenburg and Alliez 2002]. In my talk I will demonstrate that this approach is able to achieve bit-rates that are close to those of a benchmark coder.

scene	vertices	polygons	binary	plain	coded	ratio
triceratops	2832	2834	7,871	56,307	8,392	1:6.7
sandal	2636	2953	7,416	51,980	8,809	1:5.9
shark	2560	2562	6,196	49,198	5,578	1:8.8
tommygun	4171	3980	11,283	66,183	11,209	1:5.9

The table above lists compression rates for simple VRML scenes each containing a different model (like those in Figure 1). We compare the size of the gzipped file containing the *plain* versus the *coded* VRML scene and report the achieved compression ratio (see our web page for an interactive demo). For comparison we also give the rates achieved by a *binary* benchmark coder [Touma and Gotsman 1998]. Quantization of vertex positions was for both coders uniformly to 12 bits of precision.

References

ISENBURG, M. AND ALLIEZ, P. Compressing polygon mesh geometry with parallelogram prediction. *submitted for publication*, 2002.

ISENBURG, M. AND SNOEYINK, J. Compressing polygon meshes as compressable ASCII. In *Proceedings of Web3D Symposium'02* (Best Paper), pages 1–10, 2002.

ISENBURG, M. Compressing polygon mesh connectivity with degree duality prediction. In *Graphics Interface'02 Conference Proceedings*, pages 161–170, 2002.

MOFFAT, A., NEAL, R. M., AND WITTEN, I. H. Arithmetic coding revisited. In *ACM Transactions on Information Systems 16*, 3, pages 256–294, 1998.

TOUMA, C. AND GOTSMAN, C. Triangle mesh compression. In *Graphics Interface'98 Conference Proceedings*, pages 26–34, 1998.