

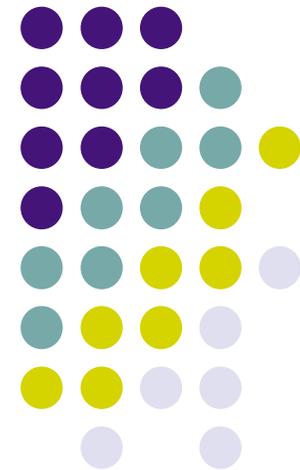
Compression and Progressive Visualization of Geometric Models



Frutuoso Silva¹ and Pranjul Yadav²

¹ IT/DCSE, University of Beira Interior,
Portugal

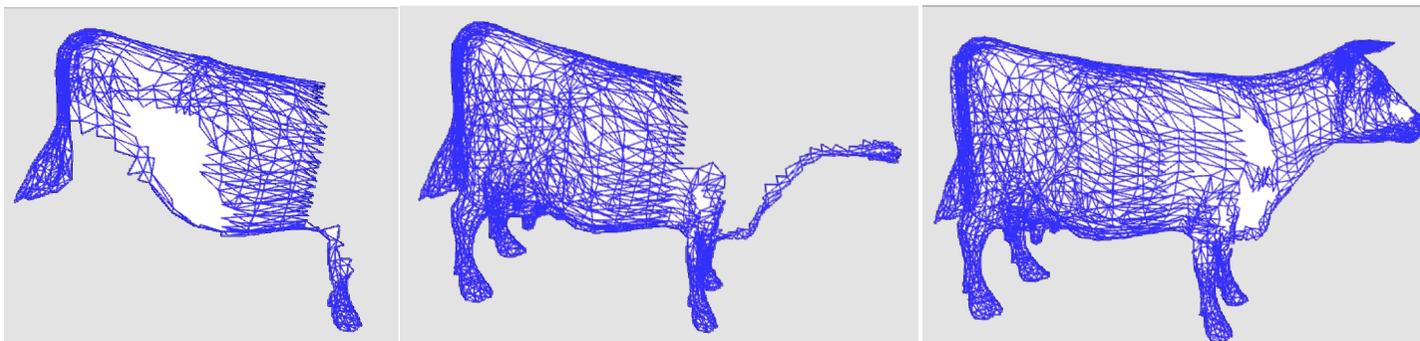
² Indian Institute of Technology, Guwahati,
India

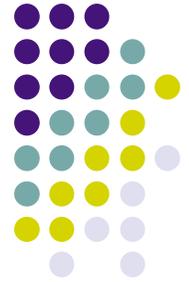




Outline

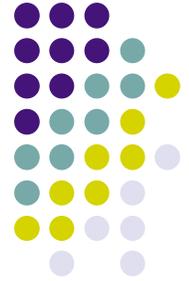
- Introduction
- Related work
- Compression and decompression algorithms
- Experimental results
- Conclusions
- Future work





Introduction

- The growth of graphical applications over the Internet
 - Rapid transmission of 3D models
 - Compression/decompression algorithms
 - Interactive visualization of models
 - The models are approximated by triangular meshes
 - The graphical hardware is optimized to support triangular polygons (e.g. triangle strips)



Related work

- Compression schemes
 - That encodes the geometry and connectivity separately [Tau98, Tou98, Ros99]
 - But these schemes are not adequate for progressive visualization
- Triangle stripification algorithms
 - SGI algorithm [Ake90]
 - STRIPE algorithm [Eva96]
 - FTSG algorithm [Xia99]



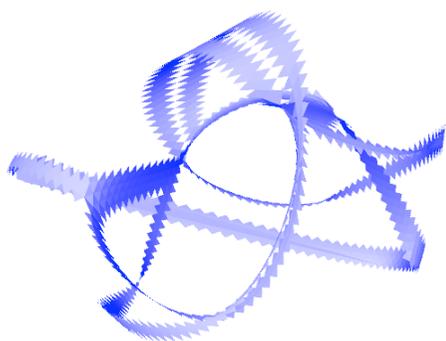
Related work (cont.)

- Bajaj *et al.* presented a compression algorithm that encodes the geometry and connectivity separately [Baj99]
- But, it uses four primitives to encode the mesh
 - Contours, branching points, triangle fans and triangle strips.
- Besides, the algorithm creates a representation of the model composed of multiple levels of detail. This levels of detail can be progressively transmitted and displayed.

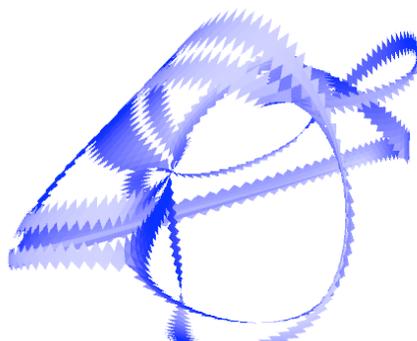


Compression algorithm

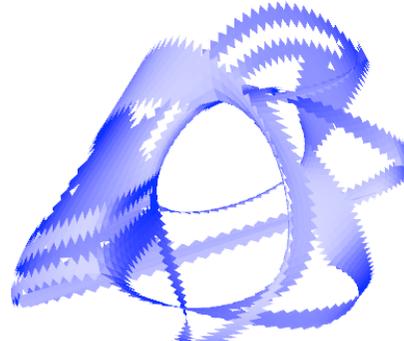
- Compress algorithm encodes the geometry and connectivity in a interwoven fashion.
- Main idea is converting a model into strips, which are auto-sufficient for visualization purposes.



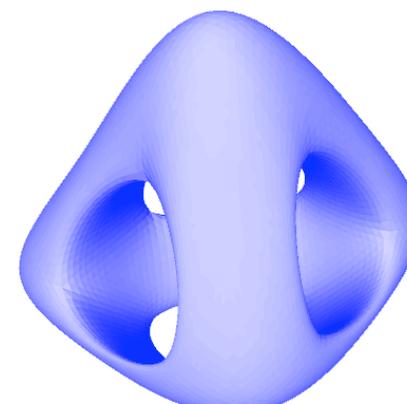
1



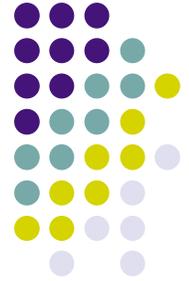
3



12

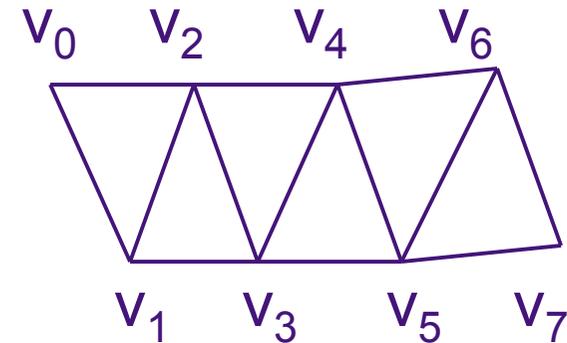


Genus3



Compression algorithm (cont.)

- The strips can be defined by:
 - Identifier of the vertex
 - e.g., $V_0 V_1 \dots V_6 V_7$
 - Coordinates of the vertex
 - e.g., $V_0 V_1X V_1Y V_1Z \dots V_6 V_7$
 - Linking agents
 - $/, \#, !, @, A, B, C, <$
 - e.g., $<0 <1 <2 <3 <4 <5 <6 <7 /$

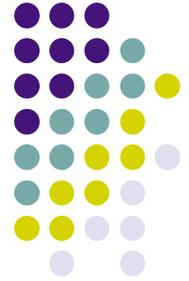


Compression algorithm (cont.)



- To distinguish the several situations we use linking agents
 - / - strip terminator
 - # - mark an auxiliary triangle
 - Change in the order of the last three vertices
 - ! - the shared edge will be defined by v_1 and v_3
 - @ - the shared edge will be defined by v_3 and v_1
 - Continue the strip based on an vertex
 - A - using v_1
 - B - using v_2
 - C - using v_3
 - < - identifier of a vertex

Decompression algorithm

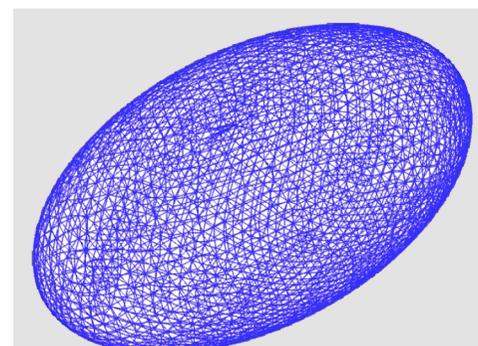
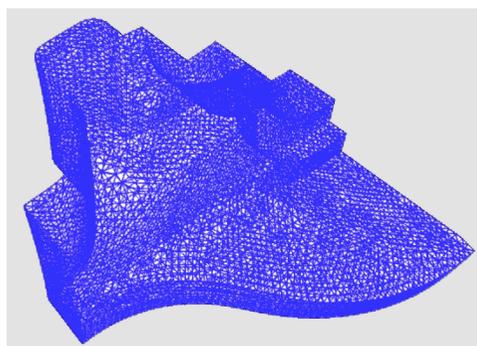
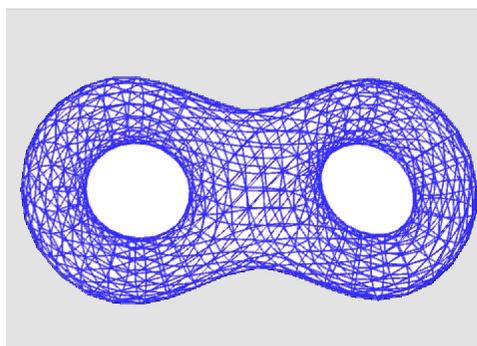
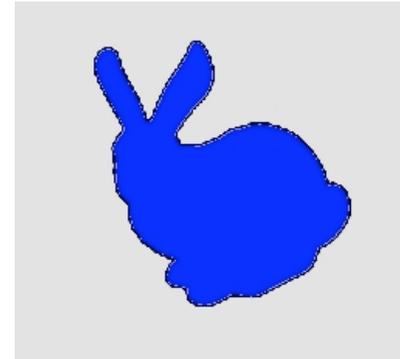
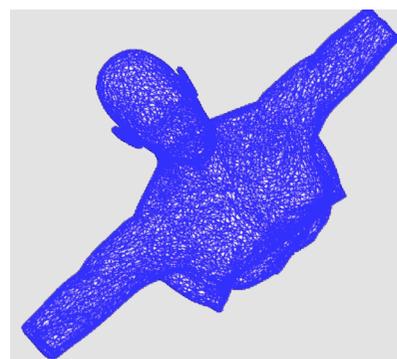
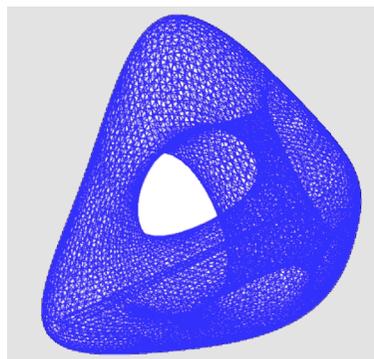
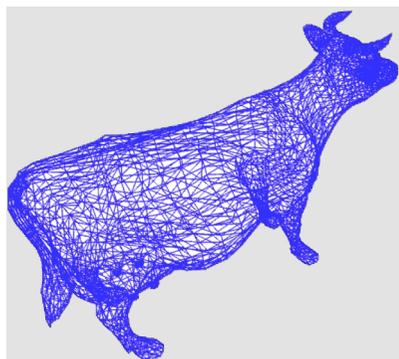


- Read the compression file and reverse the compression algorithm
 - This is a faster algorithm because we have only to know what means the linking agents
 - There are only two possibilities:
 - Read a linking agent
 - Read a vertex (x,y,z)



Experimental results

- Test models





Experimental results (cont.)

- Compression results

Model	Original File	Compressed File	Ratio
Cow	183 KB	106 KB	0.42
Bunny	2507 KB	1413 KB	0.43
Eight	47 KB	29 KB	0.38
Ellipsoid	212 KB	118 KB	0.44
Fandisk	493 KB	311 KB	0.37
Genus3	438 KB	253 KB	0.42
Torso	469 KB	268 KB	0.43



Experimental results (cont.)

- Number of vertices in strips

Model	# Triangles	Vert. in Strips	Vert. per Δ
Cow	5804	6637	1.14
Bunny	69473	79920	1.15
Eight	1535	1699	1.11
Ellipsoid	6923	7793	1.12
Fandisk	12946	15042	1.16
Genus3	13311	15117	1.13
Torso	14451	16389	1.13



Experimental results (cont.)

- Number of vertices in strips
 - Our algorithm achieved good results comparing with the results presented by Vanecek and Kolingerová [Van07]
 - Cow model
 - Our algorithm achieved 6637 vertices in strips
 - Other algorithms achieved results between 7000 and 8000
 - Bunny model
 - Our algorithm achieved 79920 vertices in strips
 - Other algorithms achieved results between 82000 and 102000 vertices



Experimental results (cont.)

Model	Number of strips	Average length of strips
Cow	231	28.71
Bunny	2645	30.21
Eight	42	40.45
Ellipsoid	246	31.67
Fandisk	544	27.65
Genus3	469	32.23
Torso	617	26.56



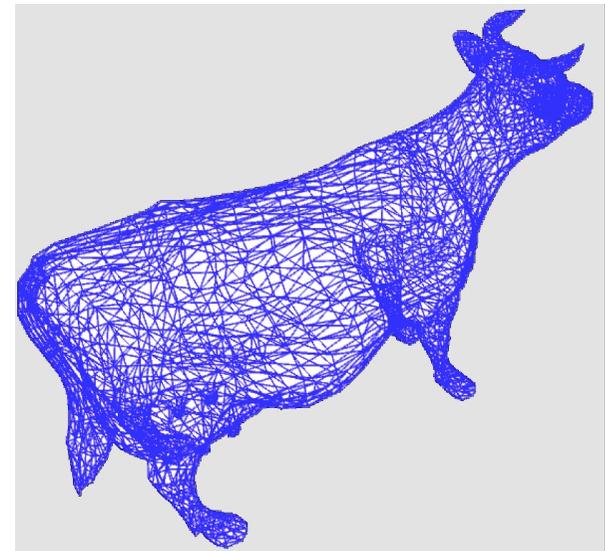
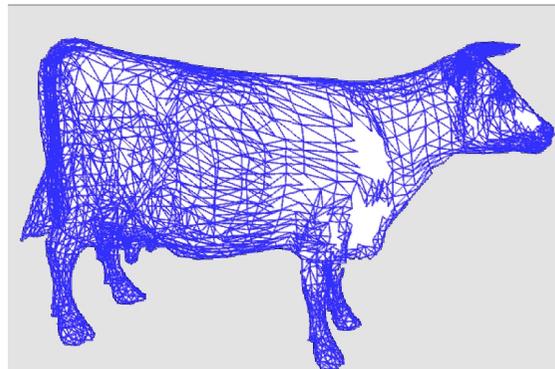
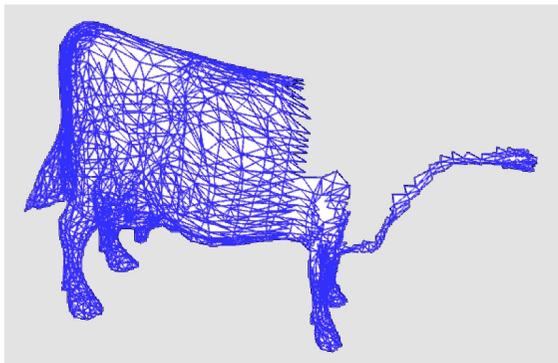
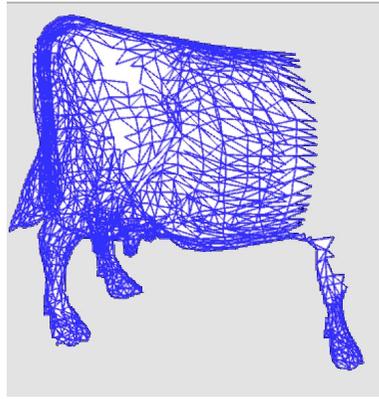
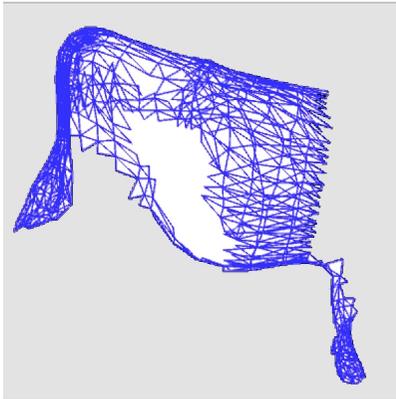
Experimental results (cont.)

- Our compression algorithm produces more strips than the other stripification algorithms
- However, the number of vertices per triangle in strips is smaller
- This means that other stripification algorithms produce fewer strips but with more number of vertices per triangle
- Therefore our algorithm is better to compression and visualization purposes



Experimental results (cont.)

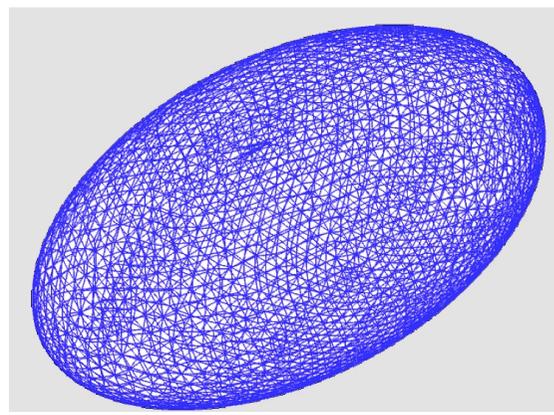
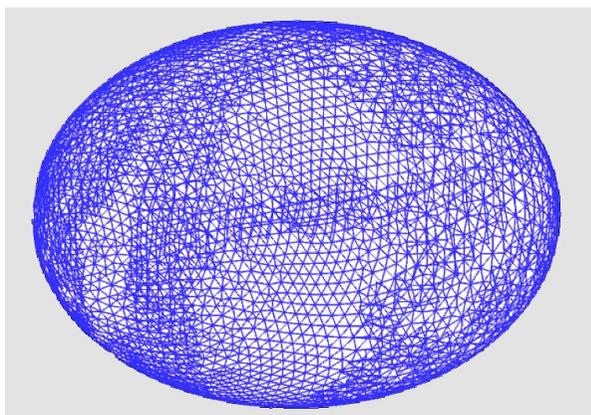
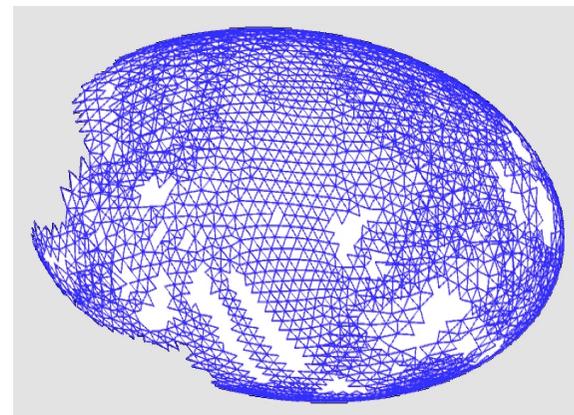
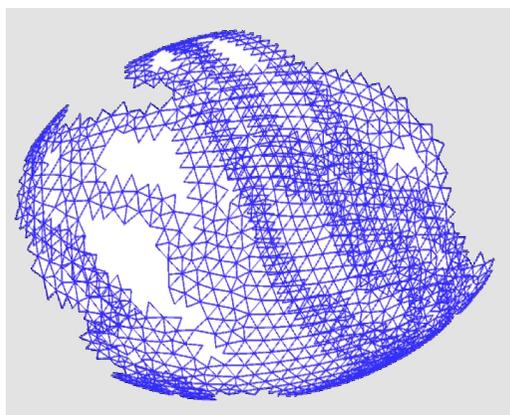
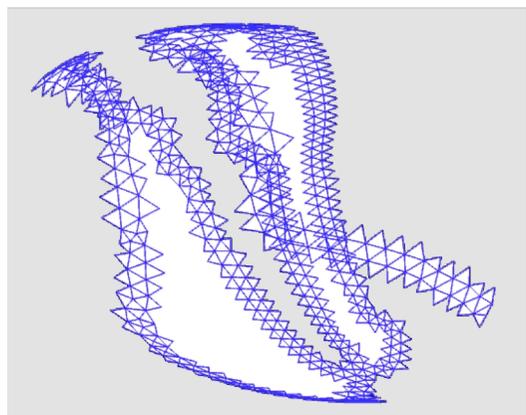
- Progressive visualization
 - Each strip is self-sufficient in terms of visualization

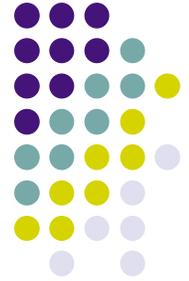




Experimental results (cont.)

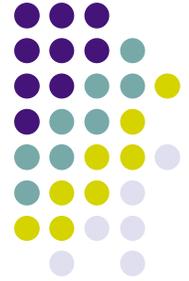
- Progressive visualization





Conclusions

- Compression algorithm was developed for compression and visualization purposes.
 - Thus, it produces normally more strips than other algorithms
 - But with less number of vertices per triangle.
 - Besides, it allows a progressive visualization of models during its transmission over the network.
- Our algorithm compress a file based on a combination of vertices and linking agents,
 - This combination helps us to achieve a compression factor above 40 percent over ASCII encoded formats.



Future work

- Improving the stripification algorithm
 - The main idea is to reduce the number of strips created by our algorithm
 - But maintaining the number of vertices per triangle

Questions?

Frutuoso Silva

<http://www.di.ubi.pt/~fsilva/>

fsilva@di.ubi.pt

Pranjul Yadav

pranjul@iitg.ernet.in



Thank you for your attention