# View dependent decompression for web-based gigantic triangle meshes visualisation

A. CECCHIN
P. DU
M. PASTOR
A. AGOUZOUL

ARSKAN

## Abstract

We introduce a framework extending an existing progressive compression-decompression algorithm for 3D triangular meshes. First, a mesh is partitioned. Each resulting part is compressed, then joined with one of its neighbours. These steps are repeated following a binary tree of operations, until a single compressed mesh remains. Decompressing the mesh involves progressively performing those steps in reverse, per node, and locally, by selecting the branch of the tree to explore. This method creates a compact and lossless representation of the model that allows its progressive and local reconstruction. Previously unprocessable meshes can now be visualised on the web and mobile devices using this technique.

## Related work

- Triangle mesh compression with mesh traversal
- Out-of-core compression and visualisation
- View-dependent refinement

## Problem

With existing progressive lossless mesh traversal compression/decompression algorithm:

**Benefits**
- Progressive transfer and reconstruction
- Visualisation suitable for low performance devices

**Drawbacks**
- Unable to process gigantic models
- Reconstruction is independent from the view


Figure 2: Progressive decompression with a single view point on Marigny Aqueduct, provided by Air Marine

## Preprocessing

### Out-of-core partitioning

- Vertices clustered in 3D grid
- Grid is cut to satisfy a maximum vertex count per submesh
- Each submesh can then be processed in-core

### In-core parallel compression

Bottom up creation of binary tree by **recursive operations**:

1. **Progressive compression**
- capped by a vertex limit
- borders are locked

2. **Stitching two by two**
- shared border are unlocked

**1. & 2. until a single mesh remains**

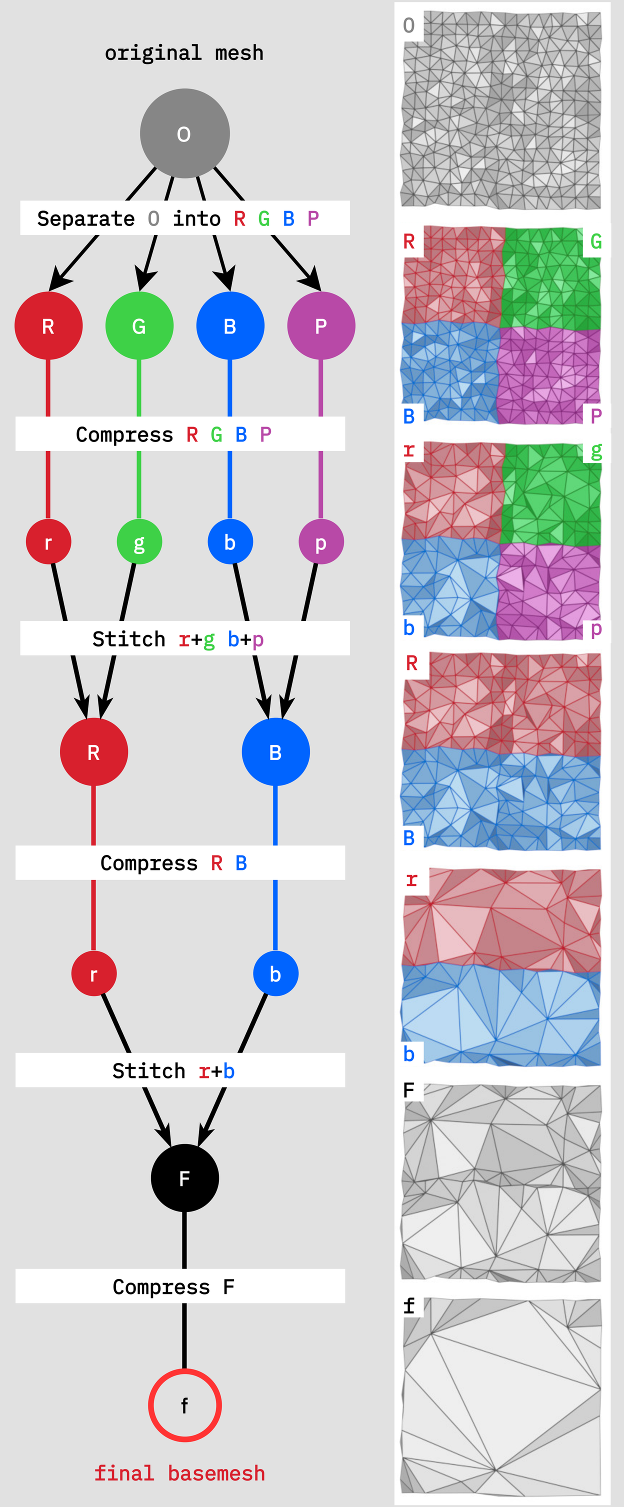
basemesh = remaining mesh

Progressive compression
=
Iterative valence driven decimation conquests

Simultaneous mesh simplification and connectivity encoding

Reversing operations to prepare for runtime

| Node | → | Node |
|---|---|---|
| Compression + Stitch | | Separation + Decompression |




## Transfer

### Data structure

| | Downloaded once | Downloaded per node |
|---|---|---|
| for the decompression | • Basemesh | • Seed edge of mesh traversal<br>• Encoded traversal attributes |
| for the framework | • Hierarchy structure<br>• Node bounding volumes<br>• Node quality criteria | • Separation attributes |

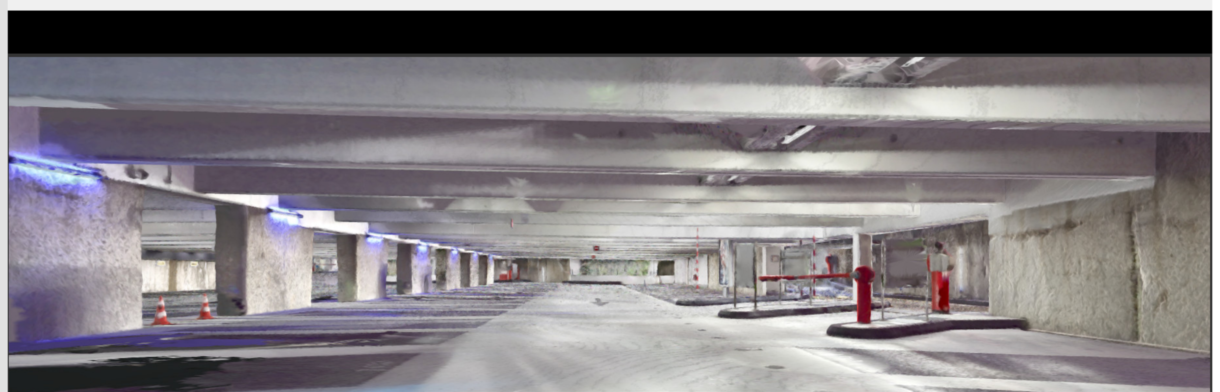


Figure 1: Underground Parking, provided by QPark

### Model presentation

Gigantic triangle mesh > 200M Vertices

| | Vertex count | Triangle count |
|---|---|---|
| Avignon Cathedral | 986,396,965 | 1,973,010,686 |
| Marigny Bridge | 499,722,772 | 999,342,292 |
| Underground Parking | 312,709,538 | 625,443,996 |

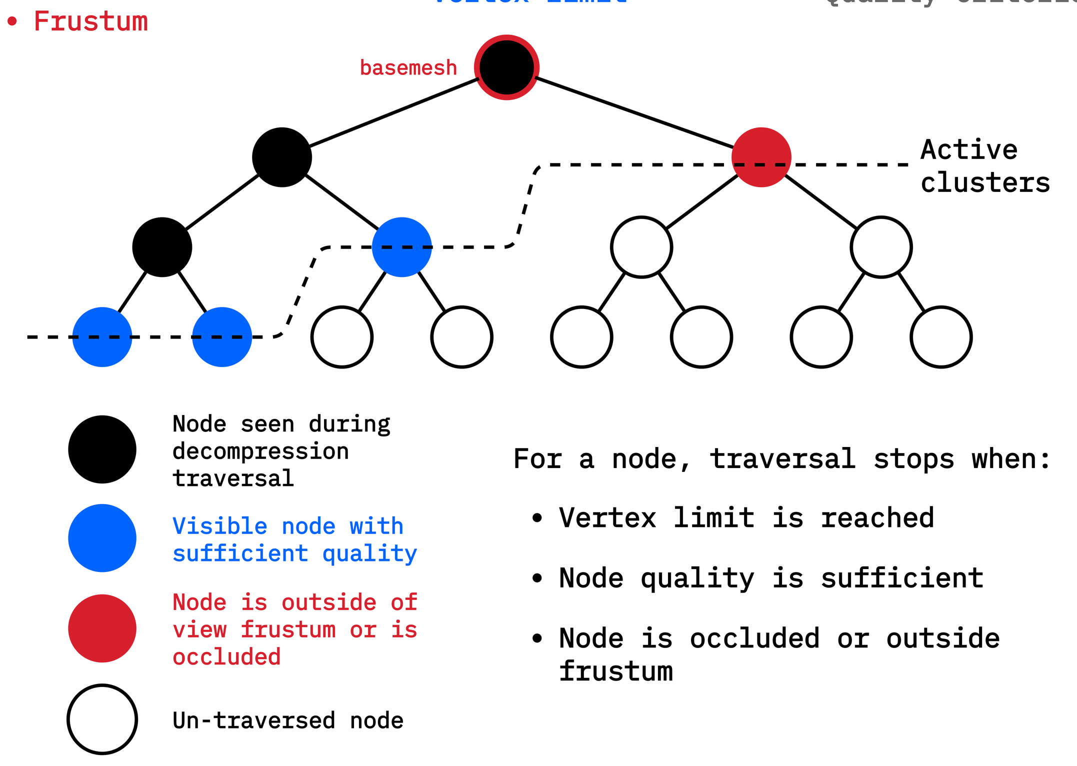
**Table 1:** Triangle meshes statistics

## Runtime

### Optimal view determination

reversed hierarchy traversal per frame

| View dependent | Device dependent | Mesh dependent |
|---|---|---|
| • Camera Transform<br>• Frustum | • Vertex limit | • Quality criterion |




For a node, traversal stops when:
- Vertex limit is reached
- Node quality is sufficient
- Node is occluded or outside frustum

### Progressive decompression

download and decompression on separate thread

- Reversed compression steps, top down from basemesh
- Start by downloading the global information, then for each selected node :

if parent is done separating : Download → Decompress progressively → Separate

Intermediary meshes are created during the progressive decompression and are re-used during visualisation.
For every frame, the closest decompressed nodes to the optimal view are rendered using WebGL.

## Preprocessing time

| | (H:MM:SS) |
|---|---|
| Avignon Cathedral | 1:37:24 |
| Marigny Aqueduct | 0:32:35 |
| Underground Parking | 0:22:54 |

Time includes:
- Out-of-core partitioning
- In-core recursive compression-merging
- computed on Intel® Core™ i7-10750H CPU @ 2.60GHz × 12

**Table 2:** Preprocessing time for the three test models

## Compression rate

| | Original file size (GB) | Compressed file size (GB) | Compression rate (bit/vertex) |
|---|---|---|---|
| Avignon Cathedral | 45 | 3.1 | 27 |
| Marigny Bridge | 24 | 1.8 | 30 |
| Underground Parking | 15 | 1.0 | 27 |

**Table 3:** File size comparison and compression rate
Compressed file size is the maximum size of data transferred to view the original file details

## Decompression speed

(s/1M vertices)

| | PC | high-end smartphone | low-end smartphone |
|---|---|---|---|
| Avignon Cathedral | 3.4 | 7.5 | 17.5 |
| Marigny Aqueduct | 3.0 | 7.3 | 16.1 |
| Underground Parking | 3.5 | 8.3 | 18.0 |

- **PC**: Intel® Core™ i7-10750H CPU @ 2.60GHz × 12
- **high-end smartphone**: OnePlus nord: Qualcomm SM7250 Snapdragon 765G 5G (7 nm)
- **low-end smartphone**: Samsung Galaxy S6: Exynos 7420 Octa (14 nm)

**Table 4:** Decompression speed per million vertices on three devices with WebAssembly

Figure 3: View dependent details on Avignon Cathedral, provided by Art Graphique & Patrimoine
(5M vertices per view point, visible partitions top-left)

## Conclusion

We have introduced a framework for the interactive visualisation of gigantic triangular meshes. Our representation is compact, transferring only a simplified basemesh and the necessary steps to refine it hierarchically. No intermediate mesh is downloaded during the visualisation: the information of a node is downloaded only once. This allows a lightweight transfer with no redundancy of even gigantic models. Thanks to the view dependent hierarchy traversal, only targeted nodes are downloaded and decompressed. The visualisation is maintained at a constant framerate of 60 frames per second thanks to our device dependant criteria, independently from the model. The progressive decompression provides a high granularity of levels of details to adapt as best as possible to the view.

EG'22 TheEvent
ARSKAN

Acknowledgements
AGP Art Graphique & Patrimoine
AIR MARINE
bpifrance
QPARK
Investir l'avenir

### Bibliography

[MLDH15] MAGLO A., LAVOUÉ G., DUPONT F., HUDELOT C.: 3dmesh compression: Survey, comparisons, and emerging trends. ACMComput. Surv. 47, 3 (feb 2015).
[LCD13] LAVOUÉ G., CHEVALIER L., DUPONT F.: Streaming compressed 3d data on the web using javascript and webgl. pp. 19–27.
[AD01] ALLIEZ P., DESBRUN M.: Progressive compression for lossless transmission of triangle meshes. InProceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques(New York,NY, USA, 2001), SIGGRAPH '01, Association for Computing Machinery, p. 195–202.
[HLK01] HO J., LEE K.-C., KRIEGMAN D.: Compressing large polygonal models.Visualization '01(10 2001).
[YSGM04] YOON S.-E., SALOMON B., GAYLE R., MANOCHA D.: Quick-VDR: Interactive View-dependent Rendering of Massive Models. IEEE Visualization(2004), 131–138.
[PD15] PONCHIO F., DELLEPIANE M.: Fast decompression for web-based view-dependent 3d rendering. pp. 199–207.
[PCDS18] POTENZIANI M., CALLIERI M., DELLEPIANE M., SCOPIGNO R.: Publishing and consuming 3d content on the web: A survey. Foundations and Trends® in Computer Graphics and Vision 10 (12 2018), 244–333.