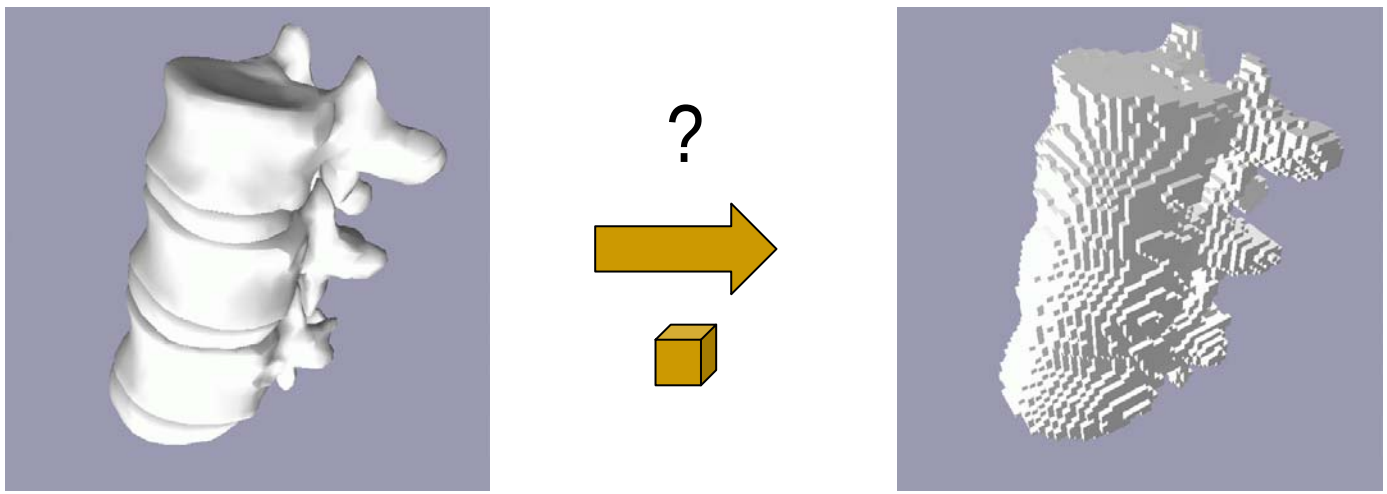
A Low Cost Antialiased Space Filled Voxelization Of Polygonal Objects

S. Thon, G. Gesquière, R. Raffin
LSIS Laboratory
Marseilles, France

Previous works

- Objectives



  - Transform a polygonal object into a set of voxel

Previous works

- Objectives
  - space filled voxelization
  - take into account holes or tunnels (image)
  - Decrease aliasing problems

Previous works

- Objectives …
  - Fast computation (although not in real time),
  - Low memory usage.
  - Using this method on a personal computer without specific graphics hardware.

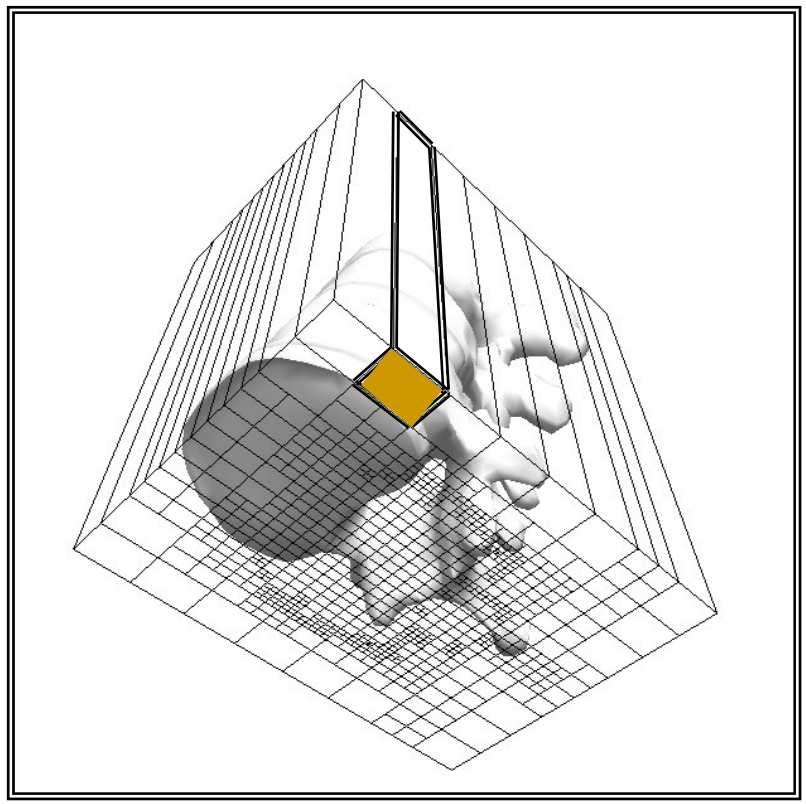
⇒ None of the existing methods matches all of our criteria

Our method

- based on an optimized raycasting through the faces of the polygonal object
- Two steps:
  - a space partitioning of the object faces
  - voxelization by raycasting trough the space partitioned faces.

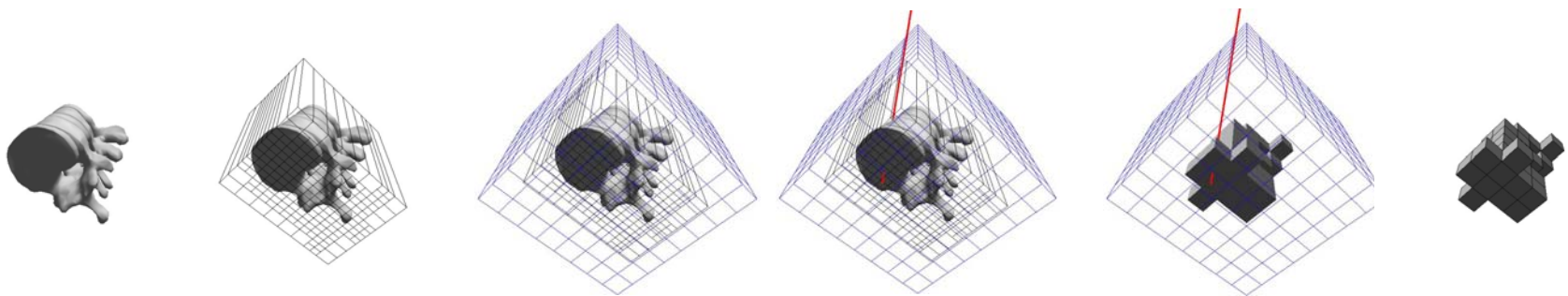
Space partitioning

1. Computation of the object bounding box
2. Subdivision of the bounding box as a quadtree of boxes
3. Each leaf of the quadtree contains a list of faces



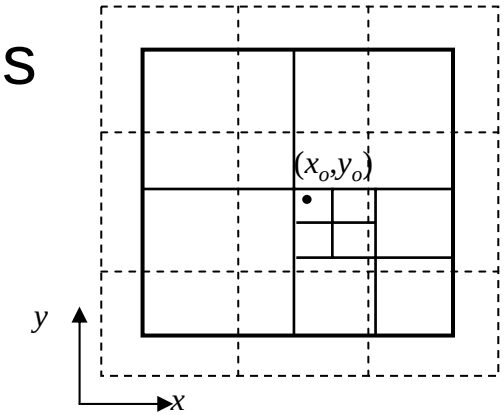
Voxelization

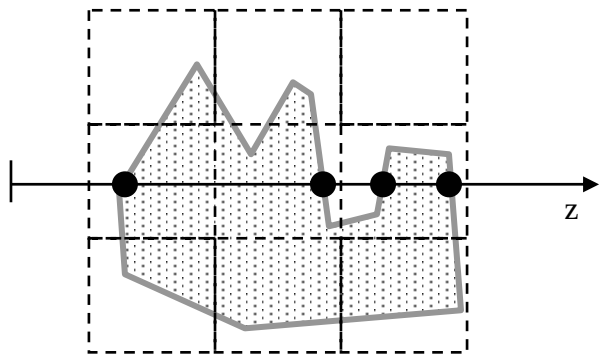
- Steps for the voxelization
  - Computation of a grid of voxels around the object
  - Raycasting
  - Inside/ outside determination



Voxelization- Raycasting

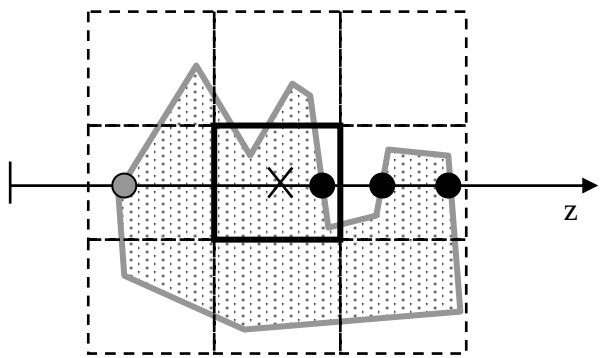
- 3D grid composed of *n* x *m* x *p* cells
- Only *n* x *m* rays are cast
- For each ray
  - Detect the intersected leaf in the quadtree
  - Compute intersections between ray and faces contained in this list leaf
  - Fill a sorted list with z-intersection between the ray and faces.



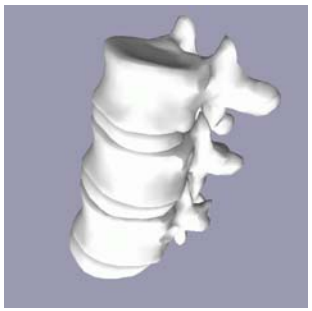


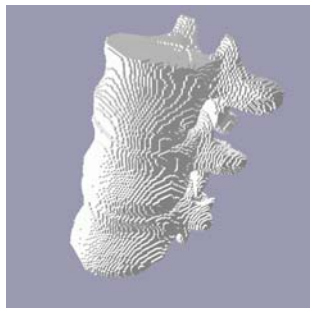
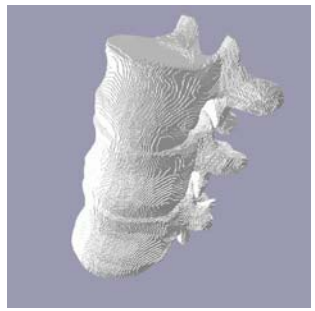
Voxelization- Inside/ outside determination

- For each voxel of the 3D Grid
  - Compute the z-coordinate of the center
  - Count the number of values in the sorted list that are greater than this z-coordinate
  - If this number is even
    - then the voxel is outside the object (0)
  - else
    - the voxel is inside the object (1)



Voxelization- Results



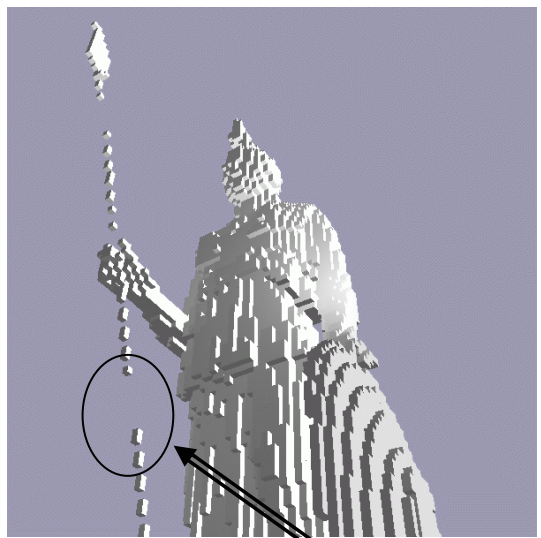



Resolution	64^3	128^3	256^3
Time (in sec.)	0,17	0,30	1

* Voxelization of polygonal object made of **10444** faces
Results obtained on an **Intel XEON 2.66 GHz with 512 Mo**.

Voxelization- Problems

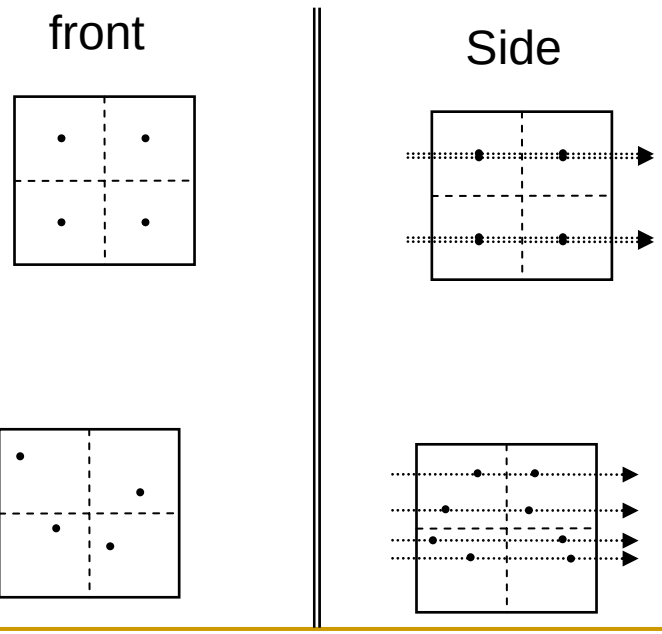
- Voxelization of a polygonal object is a 3D sampling process
  - ⇒ aliasing problems (missing details, disconnected parts)



- Solution : oversampling

Antialiasing

- Instead of casting only one ray through a row of voxels, we cast several rays.
- Two solutions for oversampling
  - Uniform
  - Stochastic



Antialiasing

- Two ways for the oversampling process
  - Binary values
  - Grey level

Antialiasing- Results with binary values

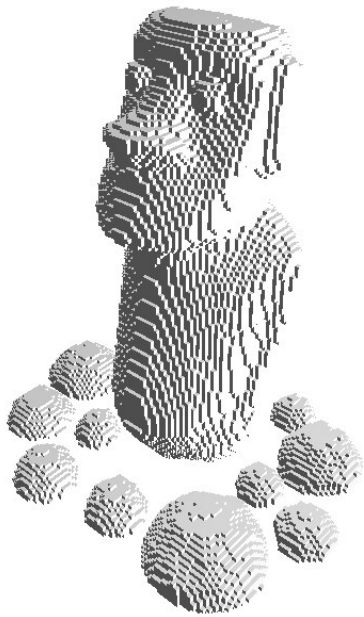
Polygonal model

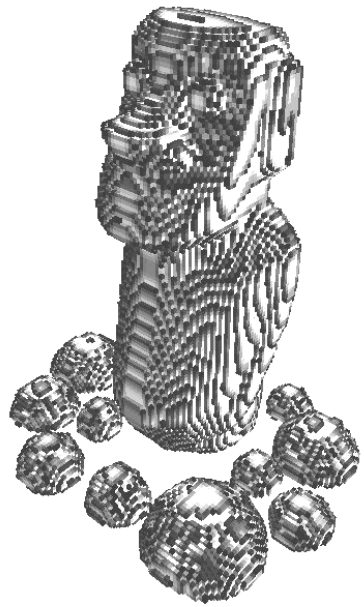
voxelization without antialiasing

uniform 2x2x2 oversampling

uniform 4x4x4 oversampling

Antialiasing- Results with grey values

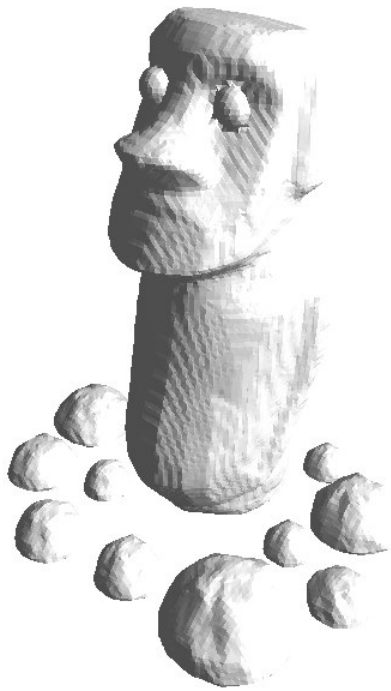




Antialiasing- Results with grey values

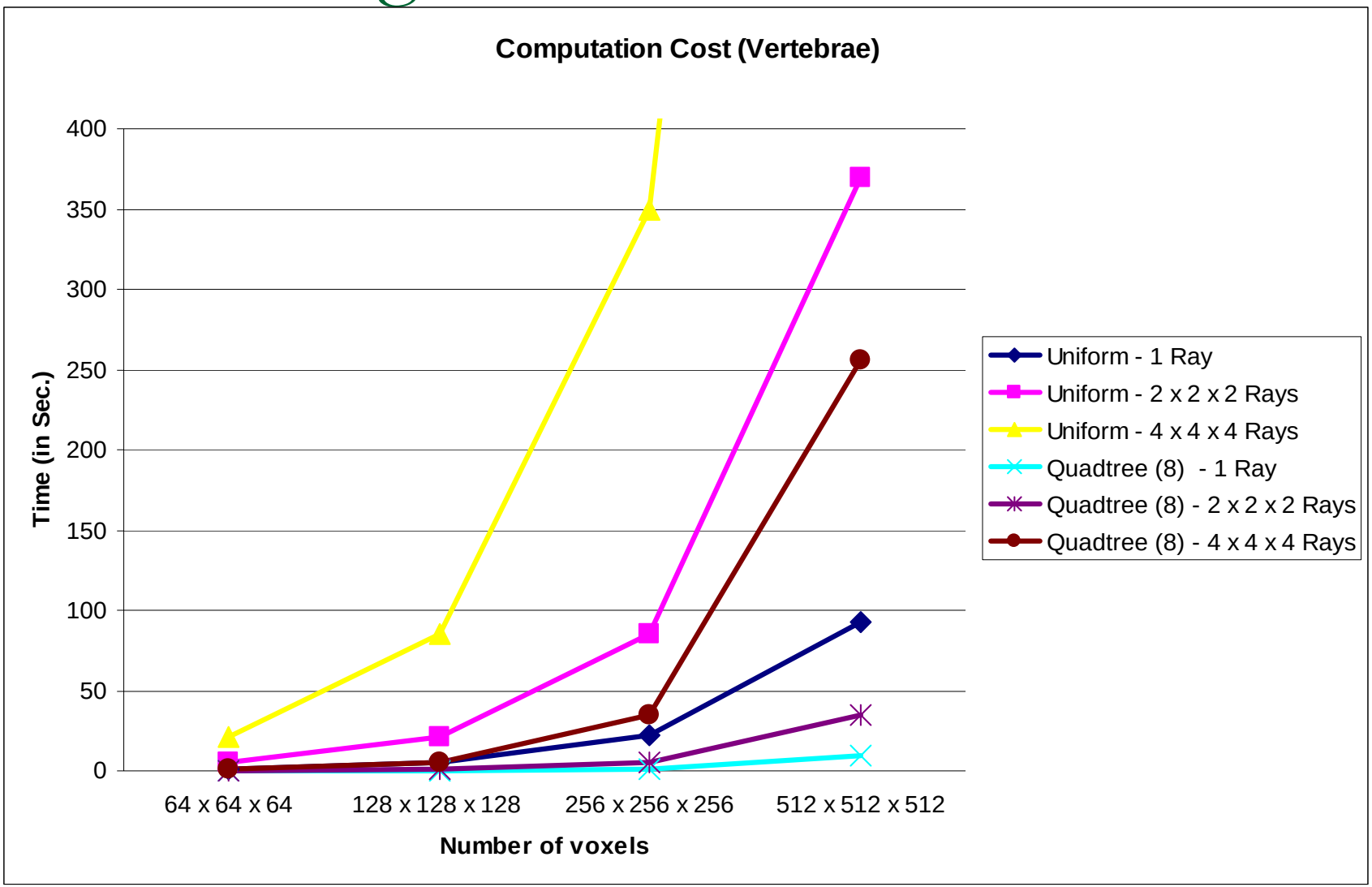
- Using the grey value for the marching cube

Marching Cubes with binary values



Marching Cubes with grey values

Antialiasing- Results



Conclusion

- voxelization method for polygonal objects based on an optimized raycasting.
- allows to fill the inner space of the object with voxels.
- Aliasing problems inherent to the sampling process of voxelization are tackled.
- Not real-time but fast enough to provide results within few seconds for polygonal objects made of several thousands of faces for large voxelizations such as 512^3.

Future Works

- As our method is based on a raycasting, it can be easily extended to the voxelization of other object (implicit surfaces, nurbs or analytic objects)
- Take into account polygonal objects that are not correctly closed, with missing faces.
- Compute adaptively the voxels in an octree.

A Low Cost Antialiased Space Filled Voxelization Of Polygonal Objects

S. Thon, G. Gesquière, R. Raffin
LSIS Laboratory
Marseilles, France