

Parallel 3D Split and Merge Segmentation with Oriented Boundary Graph

F. Baldacci (baldacci@labri.fr),
Pascal Desbarats (desbarats@labri.fr)

UMR5800, Laboratoire Bordelais de Recherche en Informatique, Université
Bordeaux1, France

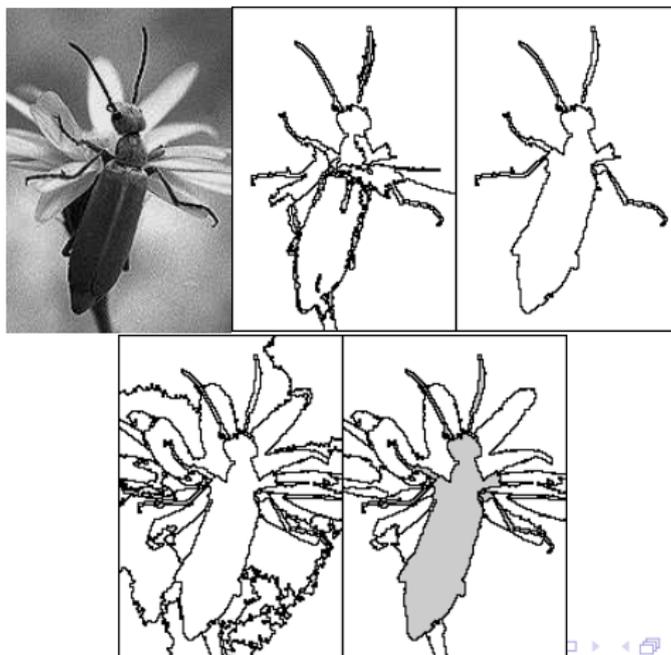
February 4-8, 2008



Motivations and goal

Motivations :

- 3D split and merge segmentation



Motivations and goal

Motivations :

- 3D split and merge segmentation
- Using a structuring model in order to efficiently compute the segmentation
- Existing models are neither not sufficient nor too complicated and segmentation is time and space consuming.

Our goal :

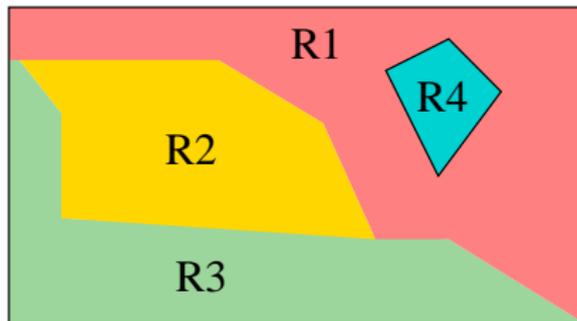
- Defining a new efficient structuring model
- Demonstrating that it is parallelizable



Outline

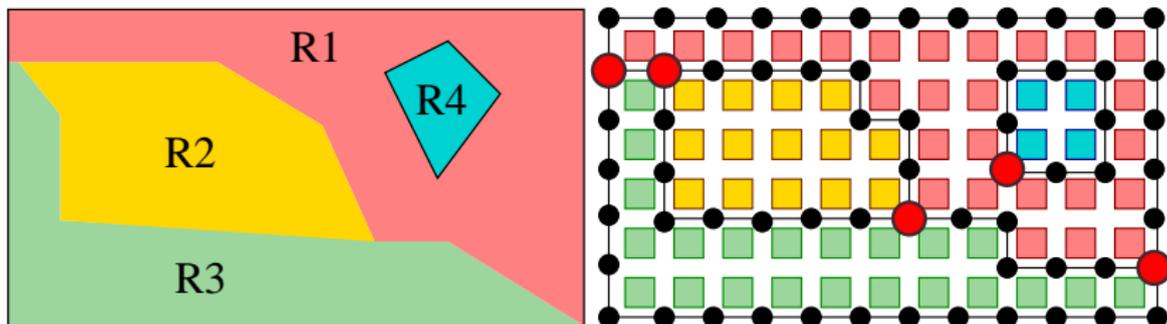
- 1 Introduction
- 2 Model presentation
 - Existing models
 - Oriented Boundary Graph model
- 3 Parallelization of the OBG
 - Description
 - Test
 - Results and execution time
- 4 Conclusion and future work

3D Split and Merge Segmentation:



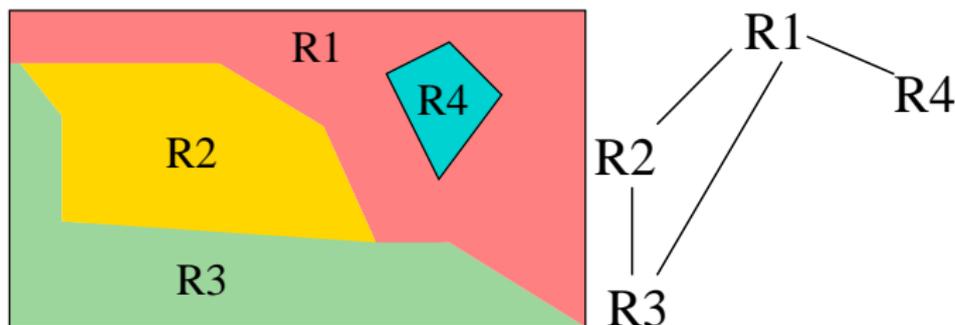
3D Split and Merge Segmentation:

- Needs the encoding of intervoxel elements



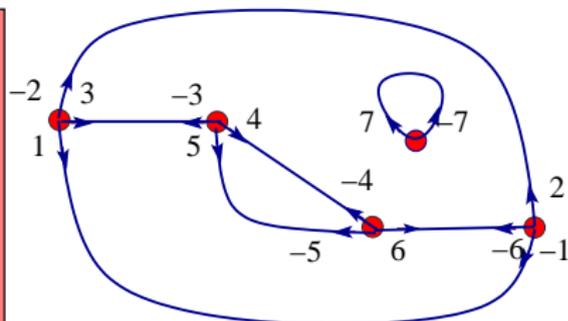
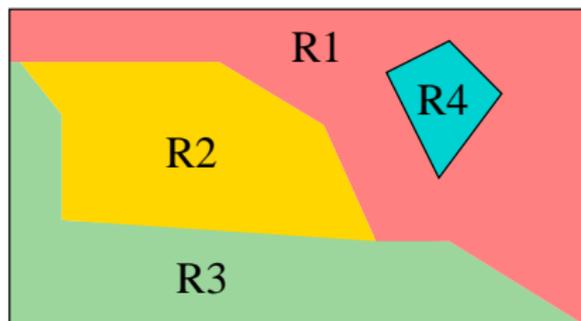
3D Split and Merge Segmentation:

- Needs the encoding of intervoxel elements
- Needs a structuring model encoding topology to be efficient
 - RAG



3D Split and Merge Segmentation:

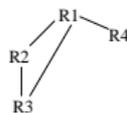
- Needs the encoding of intervoxel elements
- Needs a structuring model encoding topology to be efficient
 - RAG
 - Topological Map



3D Split and Merge Segmentation:

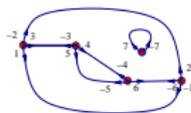
- Needs the encoding of intervoxel elements
- Needs a structuring model encoding topology to be efficient
 - RAG
 - Topological Map
 - Oriented Boundary Graph
- Have to be computed in parallel to have reasonable time complexity

- RAG (Region Adjacency Graph)



- Easy to implement
- Not sufficient: doesn't encode multiple adjacency or inclusion relation

- Topological Map



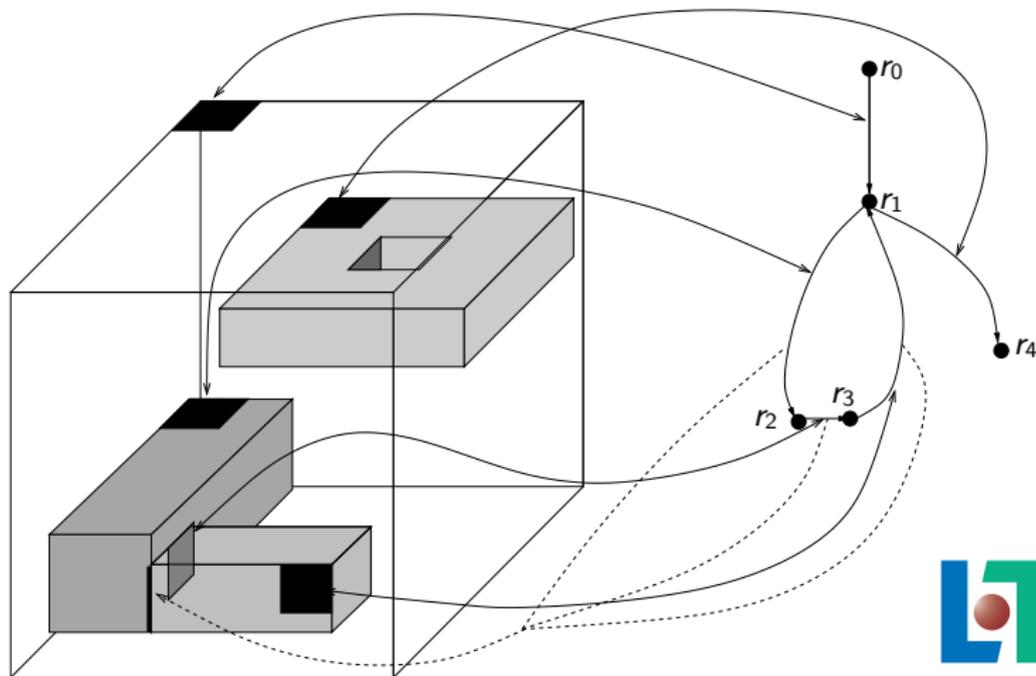
- Encode the whole topology including the Euler characteristic of regions
- Needs extra treatments on surfaces obtain from the segmentation description

Oriented Boundary Graph:

- Encode all essential topological information needed for basic segmentation algorithm:
 - Multiple Region adjacency
 - Surface Adjacency
 - Inclusion (implicitly encoded)
 - Link with Intervoxel matrix

Oriented Boundary Graph:

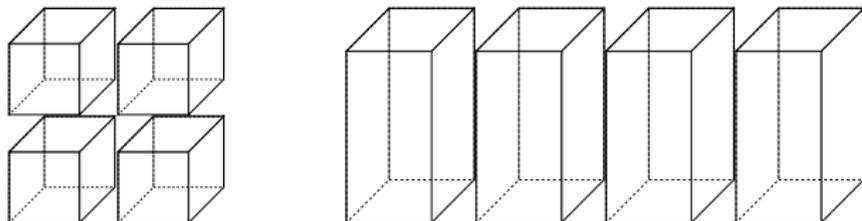
- Encode all essential topological information needed for basic segmentation algorithm:



Oriented Boundary Graph:

- Encode all essential topological information needed for basic segmentation algorithm:
 - Multiple Region adjacency
 - Surface Adjacency
 - Inclusion (implicitly encoded)
 - Link with Intervoxel matrix
- Doesn't need extra processing on surface defined by the segmentation description

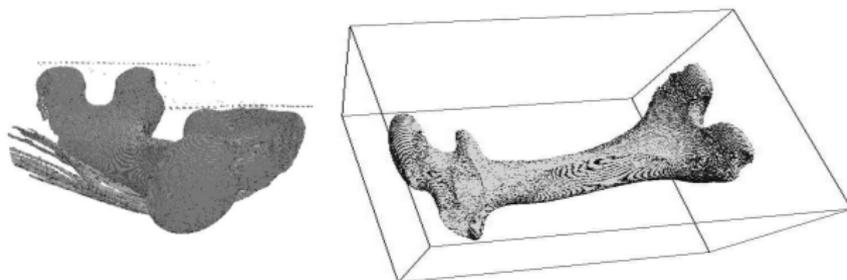
- Image cutting in many blocks

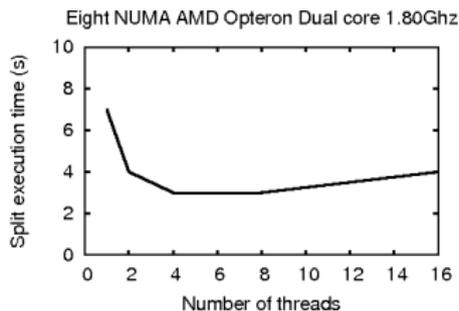
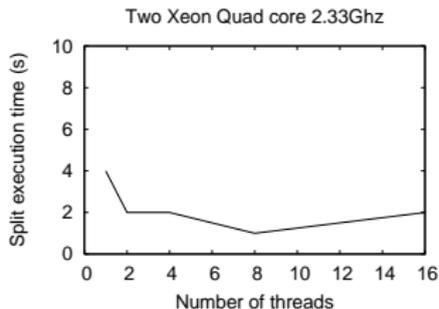


- Construction of the OBG by labelling the voxels(labels can be locally determined) with a parallel execution on each block
- Gluing the blocks (labelling the borders)

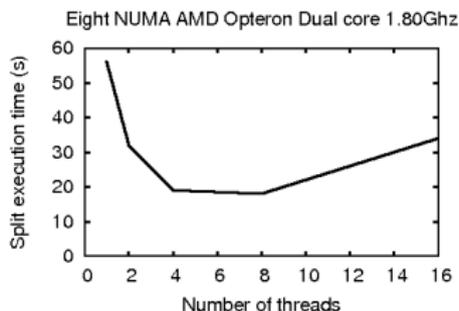
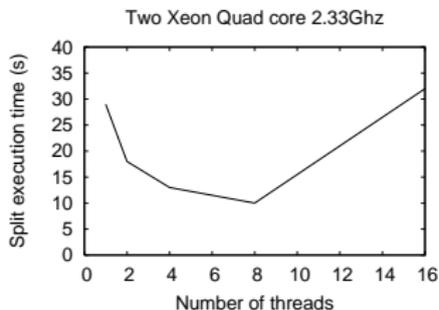
- Construction on 2 examples
- Use of a non optimal cutting

	Femur	Brain
size	512x512x475	256x256x256
Regions	456396	73453
Surfaces	1638726	291337





Execution time on the brain image.



Execution time on the femur image.

Conclusion:

- Construction time divided by 4
- Limit to the number of blocks due to:
 - naive cutting
 - glued labelling more complex than standard labelling

Future works:

- Implementation of an optimal cutting
- Parallelization of all the treatments: merge operation and surface traversing
- Study of the GPU use to create the labelling (and thus the construction of the model)